Communication-efficient Distributed Sparse Linear Discriminant Analysis

Lu Tian
University of Virginia

Quanquan Gu
University of Virginia

Abstract

We propose a communication-efficient distributed estimation method for sparse linear discriminant analysis (LDA) in the high dimensional regime. Our method distributes the data of size N into m machines, and estimates a local sparse LDA estimator on each machine using the data subset of size N/m. After the distributed estimation, our method aggregates the debiased local estimators from m machines, and sparsifies the aggregated estimator. We show that the aggregated estimator attains the same statistical rate as the centralized estimation method, as long as the number of machines m is chosen appropriately. Moreover, we prove that our method can attain the model selection consistency under a milder condition than the centralized method. Experiments on both synthetic and real datasets corroborate our theory.

1 INTRODUCTION

High dimensionality is a frequently confronted problem in many applications of machine learning. It increases time and space requirements for processing the data. Moreover, many machine learning methods tend to over-fit and become less interpretable in the presence of many irrelevant or redundant features. A common way to address this problem is the dimensionality reduction. Principal Component Analysis (PCA) (Jolliffe, 2002) is probably the most widely used dimensionality reduction method. However, it is an unsupervised dimensionality reduction method and does not consider the labels of the data. In order to take the label information into account, supervised dimensionality reduction methods are favored. Linear Discriminant Analysis (LDA) (Anderson, 1968), which is initially proposed as a classification method, is an important supervised dimensionality reduction method. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two d-dimensional random vectors following two normal distributions, $\boldsymbol{X} \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}^*)$ and $\boldsymbol{Y} \sim N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}^*)$, which share the same covariance matrix $\boldsymbol{\Sigma}^*$ but with different mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. For a new observation $\boldsymbol{Z}$ that is drawn with equal prior probability from the two normal distributions, the Fisher's linear discriminant rule is

$$\psi(\boldsymbol{Z}) = \mathbb{1}((\boldsymbol{Z} - \boldsymbol{\mu})^\top \boldsymbol{\Theta}^* \boldsymbol{\mu}_d > 0), \tag{1.1}$$

where $\boldsymbol{\mu} = (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$, $\boldsymbol{\mu}_d = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, $\boldsymbol{\Theta}^* = \boldsymbol{\Sigma}^{*-1}$ is the precision matrix (a.k.a., the inverse covariance matrix), and $\mathbb{1}(\cdot)$ is the indicator function. It is well known that the Fisher's linear discriminant rule minimizes the misclassification rate and it is Bayesian optimal. In practice, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}^*$ are unknown, and we need to estimate $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}^*$ from observations. More specifically, let $\{\boldsymbol{X}_i : 1 \leq i \leq n_1\}$ and $\{\boldsymbol{Y}_i : 1 \leq i \leq n_2\}$ be independently and identically distributed random samples from $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}^*)$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}^*)$ respectively. The classical estimations of $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and $\boldsymbol{\Theta}^*$ in the classical regime are the sample means $\widehat{\boldsymbol{\mu}}_1 = n_1^{-1} \sum_{i=1}^{n_1} \boldsymbol{X}_i$ and $\widehat{\boldsymbol{\mu}}_2 = n_2^{-1} \sum_{i=1}^{n_2} \boldsymbol{Y}_i$, and $\widehat{\boldsymbol{\Theta}} = \widehat{\boldsymbol{\Sigma}}^{-1}$, where $\widehat{\boldsymbol{\Sigma}} = n^{-1}\big[\sum_{i=1}^{n_1}(\boldsymbol{X}_i - \widehat{\boldsymbol{\mu}}_1)(\boldsymbol{X}_i - \widehat{\boldsymbol{\mu}}_1)^\top + \sum_{i=1}^{n_2}(\boldsymbol{Y}_i - \widehat{\boldsymbol{\mu}}_2)(\boldsymbol{Y}_i - \widehat{\boldsymbol{\mu}}_2)^\top\big]$ is the pooled sample covariance matrix with $n = n_1 + n_2$. Plugging these estimators into (1.1) gives rise to the empirical version of $\psi(\boldsymbol{Z})$, i.e., $\widehat{\psi}(\boldsymbol{Z})$. Theoretical properties of $\widehat{\psi}(\boldsymbol{Z})$ have been well studied when d is fixed, e.g., see Anderson (1968). However, in the high-dimensional regime where d increases as n, the pooled sample covariance matrix procedure is not well-conditioned and the plug-in estimator is not reliable. For example, Bickel and Levina (2004) showed that it is asymptotically equivalent to random guess when the dimensionality increases at some rate comparable to the number of samples. To overcome this curse of dimensionality, it is natural to impose some structural assumptions on the parameters of the discriminant rule in (1.1). For example, Cai and Liu (2011) made the assumption that $\boldsymbol{\beta}^* = \boldsymbol{\Theta}^* \boldsymbol{\mu}_d$ is a sparse vector and proposed the fol-

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

lowing estimator:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\boldsymbol{\beta}\|_1,$$
$$\text{subject to} \quad \|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta} - (\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_2)\|_\infty \le \lambda, \tag{1.2}$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^d |\beta_j|$ is the ℓ_1 norm, and $\|\cdot\|_\infty$ is the element-wise max norm, $\widehat{\boldsymbol{\Sigma}}$, $\widehat{\boldsymbol{\mu}}_1$ and $\widehat{\boldsymbol{\mu}}_2$ are defined as above and $\lambda > 0$ is a tuning parameter. In our study, we will focus on the above sparse LDA estimator, because it is comparable to or even better than many other sparse LDA estimators (Shao et al., 2011; Mai et al., 2012; Fan et al., 2012).

On the other hand, with the increase in the volume of data used for machine learning, and the availability of distributed computing resources, distributed statistical estimation (Mcdonald et al., 2009; Balcan et al., 2012; Zhang et al., 2012, 2013; Rosenblatt and Nadler, 2014; Lee et al., 2015; Battey et al., 2015) and distributed optimization (Zinkevich et al., 2010; Boyd et al., 2011; Dekel et al., 2012) have received increasing attention. The main bottleneck in distributed computing is usually the communication between machines, so the overarching goal of the algorithm design in distributed setting is to reduce the communication costs, while trying to achieve comparable performance as centralized algorithms. The problem becomes even more challenging when high dimensionality meets huge data size.

To address the challenge of both high dimensionality and huge data size, in this paper, we propose a distributed sparse linear discriminant analysis method. In the proposed algorithm, each "worker" machine generates a local estimator for the sparse LDA and sends it to the "master" machine, where all local estimators are averaged to form an aggregated estimator. At the core of our algorithm is an unbiased estimator for the sparse linear discriminant analysis. It is worth noting that our proposed algorithm requires only one round of communication between the worker nodes and the master node. That is, each worker machine only needs to send a vector to the master node. Thus, our algorithm is very communication-efficient. We prove the estimation error bounds for the proposed algorithm in terms of different norms. More specifically, we show that the proposed distributed algorithm attains $O(\sqrt{s\log d/N} + \max(s,s')m\sqrt{s}\log d/N)$ estimation error bound in terms of ℓ_2 norm, where N is the total sample size, m is the number of machines, d is the dimensionality, $s = \|\boldsymbol{\beta}^*\|_0$ and $s' = \max_{1\le j\le d}\|\boldsymbol{\theta}_j^*\|_0$ are the number of nonzero elements in $\boldsymbol{\beta}^*$ and $\boldsymbol{\theta}_j^*$ respectively, with $\boldsymbol{\theta}_j^*$ being the j-th column of $\boldsymbol{\Theta}^*$. From the estimation error bound, we address an important question that how to choose m such that the information loss due to the data parallelism is negligible. In particular, if the machine number m satisfies $m \lesssim \sqrt{N/\log d}/\max(s,s')$, our distributed algorithm attains the same statistical rate as the centralized estimator (Cai and Liu, 2011), which is $O(\sqrt{s\log d/N})$ in terms of ℓ_2-norm. Furthermore, we show that given $\min_j |\beta_j^*| \gtrsim \sqrt{\log d/N}$, our estimator achieves the model selection consistency, which matches the minimax lower bound for support recovery in sparse LDA (Fan et al., 2012; Kolar and Liu, 2015). However, the model selection consistency established in Kolar and Liu (2015) relies on the irrepresentable condition, which is very stringent. In sharp contrast, the model selection consistency of our algorithm does not need this condition.

Notation We summarize here the notations to be used throughout the paper. We use lowercase letters $x, y, \ldots$ to denote scalars, bold lowercase letters $\mathbf{x}, \mathbf{y}, \ldots$ for vectors, and bold uppercase letters $\mathbf{X}, \mathbf{Y}, \ldots$ for matrices. We denote random vectors by $\boldsymbol{X}, \boldsymbol{Y}$. We denote $\mathbf{e}_j$ as the column vector whose j-th entry is one and others are zeros. Let $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{d\times d}$ be a $d\times d$ matrix and $\mathbf{x} = [x_1, \ldots, x_d]^\top \in \mathbb{R}^d$ be a d-dimensional vector. For $0 < q < \infty$, we define the ℓ_0, ℓ_q and ℓ_∞ vector norms as $\|\mathbf{x}\|_0 = \sum_{i=1}^d \mathbb{1}(x_i \neq 0), \|\mathbf{x}\|_q = (\sum_{i=1}^d |x_i|^q)^{1/q}, \|\mathbf{x}\|_\infty = \max_{1\le i\le d}|x_i|$, where $\mathbb{1}(\cdot)$ represents the indicator function. For any real number C and symmetric matrix $\mathbf{A}$, $\mathbf{A} \succ C$ means that the minimum eigenvalue of $\mathbf{A}$ is larger than C. Specifically, $\mathbf{A} \succ 0$ means that $\mathbf{A}$ is a positive definite matrix. We use the following notation for the matrix ℓ_∞, ℓ_1, $\ell_{\infty,\infty}$ and $\ell_{1,1}$ norms:

$$\|\mathbf{A}\|_\infty = \max_{1\le j\le d}\sum_{k=1}^d |A_{jk}|, \qquad \|\mathbf{A}\|_1 = \|\mathbf{A}^\top\|_\infty,$$
$$\|\mathbf{A}\|_{\infty,\infty} = \max_{1\le i,j\le d}|A_{ij}|, \qquad \|\mathbf{A}\|_{1,1} = \sum_{1\le i,j\le d}|A_{ij}|.$$

For a vector $\mathbf{x}$ and an index set S, $\mathbf{x}_S$ denotes the vector such that $[\mathbf{x}_S]_j = x_j$ if $j \in S$, and $[\mathbf{x}_S]_j = 0$ otherwise. For sequences f_n, g_n, we write $f_n = O(g_n)$ if $|f_n| \le C|g_n|$ for some $C > 0$ independent of n and all $n > D$, where D is a positive integer. We also make use of the notation $f_n \lesssim g_n$ ($f_n \gtrsim g_n$) if f_n is less than (greater than) g_n up to a constant. In this paper, C, c, C', C_1 etc. denote various absolute constants, not necessarily the same at each occurrence.

2 RELATED WORK

In this section, we briefly review the related work on sparse linear discriminant analysis (LDA) and distributed estimation.

LDA has been widely studied in the high dimensional regime where the number of features d can increase as the sample size n (Shao et al., 2011; Cai and Liu,

2011; Mai et al., 2012; Fan et al., 2012). One important problem in the high dimensional regime is that the estimation of $\boldsymbol{\Theta}^*$ will be unstable because the sample covariance matrix $\widehat{\boldsymbol{\Sigma}}$ is often singular. To address this problem, a common assumption is that both $\boldsymbol{\mu}_d$ and $\boldsymbol{\Sigma}^*$ are sparse. Under this assumption, Shao et al. (2011) proposed to use a thresholding procedure to estimate $\boldsymbol{\mu}_d$ and $\boldsymbol{\Sigma}^*$ respectively, followed by the standard procedure to estimate $\psi(\boldsymbol{Z})$. Cai and Liu (2011) assumed that $\boldsymbol{\beta}^* = \boldsymbol{\Theta}^*\boldsymbol{\mu}_d$ is sparse and estimated it directly. While sparse LDA has been investigated extensively, it is not clear how to extend it to the distributed setting, where the data are distributed on multiple machines.

With the growth of the size of available datasets, distributed algorithms become more and more attractive in the machine learning and optimization communities. In general, distributed algorithm can be categorized into two families: (1) data parallelism, which distributes the data across different parallel computing nodes/machines; and (2) task parallelism, which distributes tasks performed by threads across different parallel computing nodes. In this paper, we focus on data parallelism. The most important problem in data parallelism is how to minimize the communication cost among different machines. A commonly used approach in distributed statistical estimation is averaging: each "worker" machine generates a local estimator and sends it to the "master" machine where all local estimators are averaged to form an aggregated estimator. This type of approach has been first studied by Mcdonald et al. (2009); Zinkevich et al. (2010); Zhang et al. (2012, 2013); Balcan et al. (2012). Nevertheless, the above distributed statistical estimation methods are in the classical regime. In the high dimensional regime, averaging is not an effective way for aggregation (Rosenblatt and Nadler, 2014). Moreover, many estimators in the high dimensional regime are based on the penalized estimation, which introduces some bias to the estimator. For example, the Lasso estimator (Tibshirani, 1996) is biased due to the ℓ_1-norm penalty. Since averaging only reduces variances, not the bias, the performance of averaged estimator is no better than the local estimator due to the aggregation of bias when averaging. To address this problem, Lee et al. (2015) proposed distributed sparse regression methods, which exploits the debiased estimators proposed in Javanmard and Montanari (2014); Van de Geer et al. (2014) for distributed sparse regression. Similar distributed regression methods are proposed by Battey et al. (2015) for both distributed statistical estimation and hypothesis testing. However, all the above studies on distributed statistical estimation are focused on regression. It is not easy to extend them to distributed dimensionality reduction.

In fact, the problem of distributed dimensionality reduction is still relatively under-studied. Liang et al. (2014) proposed a distributed approximate PCA algorithm, which speeds up the computation and needs low communication cost but with a low accuracy loss. Balcan et al. (2015) extended the kernel PCA to the distributed setting and proposed a communication-efficient distributed kernel PCA algorithm. Valcarcel Macua et al. (2011) developed a distributed algorithm for linear discriminant analysis on a single-hop network. Nevertheless, all these algorithms are in the classical regime, and cannot be applied to sparse LDA in the high dimensional regime.

3 DISTRIBUTED SPARSE LINEAR DISCRIMINANT ANALYSIS

In this section, we present a distributed linear discriminant analysis algorithm. The problem setup of distributed sparse linear discriminant analysis is as follows: Let $\mathbf{X}^{(l)} \in \mathbb{R}^{n_{1l}\times d}, l \in \{1, 2, \ldots, m\}$ be the data matrix of the first class stored on the l-th machine, each row of which is sampled i.i.d. from the multivariate normal distribution $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}^*)$. Similarly, let $\mathbf{Y}^{(l)} \in \mathbb{R}^{n_{2l}\times d}, l \in \{1, 2, \ldots, m\}$ be the data matrix of the second class stored on the l-th machine, where each row is sampled i.i.d. from the multivariate normal distribution $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}^*)$. Without loss of generality, we assume $n_{11} = n_{12} = \ldots = n_{1m} = n_1$ and $n_{21} = n_{22} = \ldots = n_{2m} = n_2$. Let $n = n_1 + n_2$, which is the total number of data stored in a single machine. We also assume $n_1 \leq n_2$ and $n_1 = rn$, where $r \leq 1/2$ is a constant. We propose a distributed sparse LDA algorithm based on Cai and Liu (2011) to directly estimate $\boldsymbol{\beta}^*$ in Algorithm 1.

Algorithm 1 Distributed Sparse Linear Discriminant Analysis

Require: $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)}, \mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(m)}$
Ensure: $\bar{\boldsymbol{\beta}}$, the aggregated sparse discriminant vector
Workers:
Each worker computes $\widehat{\boldsymbol{\Sigma}}^{(l)}$ and $\widehat{\boldsymbol{\mu}}_1^{(l)}, \widehat{\boldsymbol{\mu}}_2^{(l)}$
Each worker computes a local sparse LDA estimator $\widehat{\boldsymbol{\beta}}^{(l)}$ by (3.1)
Each worker computes a debiased estimator $\widetilde{\boldsymbol{\beta}}^{(l)}$ by (3.4)
Each worker sends $\widetilde{\boldsymbol{\beta}}^{(l)}$ to the master machine
Master:
while waiting for $\widetilde{\boldsymbol{\beta}}^{(l)}$ sent from all workers **do**
  if received $\widetilde{\boldsymbol{\beta}}^{(l)}$ from all workers **then**
    Compute the aggregated sparse estimator $\bar{\boldsymbol{\beta}}$ by (3.5)
  end if
end while

In detail, for the l-th machine, we denote by $\boldsymbol{X}_i^{(l)}$ and $\boldsymbol{Y}_i^{(l)}$ the i-th row of $\mathbf{X}^{(l)}$ and $\mathbf{Y}^{(l)}$ respectively. On each machine, we can use the sparse LDA estimator in (1.2) to obtain a local estimator as the following:

$$\widehat{\boldsymbol{\beta}}^{(l)} = \underset{\boldsymbol{\beta}\in\mathbb{R}^d}{\operatorname{argmin}} \|\boldsymbol{\beta}\|_1 \text{ subject to } \left\|\widehat{\boldsymbol{\Sigma}}^{(l)}\boldsymbol{\beta} - \widehat{\boldsymbol{\mu}}_d^{(l)}\right\|_\infty \leq \lambda, \tag{3.1}$$

where $\lambda > 0$ is a tuning parameter, $\widehat{\boldsymbol{\mu}}_d^{(l)} = \widehat{\boldsymbol{\mu}}_1^{(l)} - \widehat{\boldsymbol{\mu}}_2^{(l)}$ with sample means $\widehat{\boldsymbol{\mu}}_1^{(l)} = (\sum_{i=1}^{n_1} \boldsymbol{X}_i^{(l)})/n_1$ and $\widehat{\boldsymbol{\mu}}_2^{(l)} = (\sum_{i=1}^{n_2} \boldsymbol{Y}_i^{(l)})/n_2$ and

$$\widehat{\boldsymbol{\Sigma}}^{(l)} = \frac{1}{n}\Bigg[\sum_{i=1}^{n_1}(\boldsymbol{X}_i^{(l)} - \widehat{\boldsymbol{\mu}}_1^{(l)})(\boldsymbol{X}_i^{(l)} - \widehat{\boldsymbol{\mu}}_1^{(l)})^\top + \sum_{i=1}^{n_2}(\boldsymbol{Y}_i^{(l)} - \widehat{\boldsymbol{\mu}}_2^{(l)})(\boldsymbol{Y}_i^{(l)} - \widehat{\boldsymbol{\mu}}_2^{(l)})^\top\Bigg],$$

which is the total intra-class sample covariance matrix of the l-th machine. The choice of λ will be discussed in Section 4.

The estimator in (3.1) is biased due to the shrinkage property of the estimator. Since averaging only reduce the variance, rather than the bias, if we naively average $\widehat{\boldsymbol{\beta}}^{(l)}$'s, the error bound of the averaged estimator will remain in the same order as that of the local estimators. To address the bias, several debiasing techniques have been proposed, such as Lee et al. (2015) and Battey et al. (2015). However, Lee et al. (2015) focused on the Lasso estimator, and the debiasing method proposed in Battey et al. (2015) is only suitable for regularized estimators. In order to construct an unbiased estimator for the Dantzig-type estimator, we propose a new debiasing procedure as follows: First, the CLIME estimator (Cai et al., 2011) is used to estimate the precision matrix:

$$\widehat{\boldsymbol{\Theta}}^{(l)} = \operatorname{argmin} \|\boldsymbol{\Theta}\|_{1,1} \quad \text{subject to} \quad \|\boldsymbol{\Theta}^\top\widehat{\boldsymbol{\Sigma}}^{(l)} - \mathbf{I}\|_{\infty,\infty} \leq \lambda', \tag{3.2}$$

where λ' is a tuning parameter, and its choice will be clear from Section 4. It is worth noting that (3.2) can be decomposed into d independent optimization problems, where each one takes the form

$$\widehat{\boldsymbol{\theta}}_j^{(l)} = \operatorname{argmin} \|\boldsymbol{\theta}\|_1 \text{ subject to } \|\widehat{\boldsymbol{\Sigma}}^{(l)}\boldsymbol{\theta} - \mathbf{e}_j\|_\infty \leq \lambda', \tag{3.3}$$

for $j \in \{1, 2, \ldots, d\}$ and $\widehat{\boldsymbol{\theta}}_j^{(l)}$ corresponds to the j-th column of $\widehat{\boldsymbol{\Theta}}^{(l)}$. Therefore, they can be solved in parallel.

Second, based on $\widehat{\boldsymbol{\Theta}}^{(l)}$, we construct a debiased estimator $\widetilde{\boldsymbol{\beta}}^{(l)}$ in the following way:

$$\widetilde{\boldsymbol{\beta}}^{(l)} = \widehat{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\Theta}}^{(l)\top}\left(\widehat{\boldsymbol{\Sigma}}^{(l)}\widehat{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\mu}}_d^{(l)}\right). \tag{3.4}$$

Note that the second term in the right hand side of (3.4) can be seen as the estimation of the bias introduced by the penalized estimator in (3.2). We subtract the estimation of the bias from $\widehat{\boldsymbol{\beta}}^{(l)}$ and obtain an unbiased estimator $\widetilde{\boldsymbol{\beta}}^{(l)}$.

Finally, the workers send back the unbiased local estimators in (3.4) generated by different machines to the master node, and the master node averages all the debiased local estimators followed by hard thresholding in order to get a sparse estimator. More specifically, the sparse aggregated estimator is as follows

$$\bar{\boldsymbol{\beta}} = \mathrm{HT}\left(\frac{1}{m}\sum_{l=1}^{m}\widetilde{\boldsymbol{\beta}}^{(l)}, t\right), \tag{3.5}$$

where $\mathrm{HT}(\cdot)$ is the hard thresholding operator, which is defined as

$$[\mathrm{HT}(\boldsymbol{\beta}, t)]_j = \begin{cases} \beta_j, & \text{if } |\beta_j| > t, \\ 0, & \text{if } |\beta_j| \leq t. \end{cases}$$

Here $t > 0$ is a pre-specified threshold. The setting of t will be discussed in Section 4.

The proposed distributed algorithm has a low communication cost. In detail, compared with the naive distributed algorithm in which $\widehat{\boldsymbol{\Sigma}}^{(l)}$'s and $\widehat{\boldsymbol{\mu}}_d^{(l)}$'s are computed separately on each machine and then sent back to the master node, our algorithm only needs to send d-dimensional vectors rather than $d \times d$ matrices to the master node, which significantly reduces the communication cost. Moreover, we will prove later that, while keeping low communication cost, our algorithm can attain the same convergence rate as the centralized method if we choose the number of machines appropriately.

The time complexity of our algorithm can be illustrated as follows: in order to obtain $\widehat{\boldsymbol{\beta}}^{(l)}$, the main computation overhead lies on computing $\widehat{\boldsymbol{\Sigma}}^{(l)}$, whose time complexity is $O(Nd^2/m)$. For the CLIME estimator, using the FastCLIME method (Pang et al., 2014), the time complexity is $O(d^2)$. Thus the total time complexity of the proposed algorithm per machine is $O(Nd^2/m)$. In contrast, for centralized estimator which collects the data from all local machines and performs the estimation, the time complexity is $O(Nd^2)$. Therefore, as the number of machine grows, a near linear speedup in the number of machines can be achieved for our distributed algorithm. Furthermore, as will be demonstrated in the main theory, in order to make the information loss caused by the data parallelism negligible, the appropriate choice of m can be as large as $O(\sqrt{N})$, which implies a time complexity of $O(d^2\sqrt{N})$ on each machine. This suggests that the proposed algorithm has a lower time complexity while attaining the same statistical rate as the centralized method.

4 MAIN THEORY

In this section, we establish the main theory for our distributed LDA algorithm. Before we present the main result of this paper, we first lay out some necessary assumptions.

We make the following assumptions on the covariance matrix and the precision matrix of the two normal distributions.

Assumption 4.1 *There exists a constant $K \geq 1$, such that the maximum and minimal eigenvalues of $\boldsymbol{\Sigma}^*$ can be bounded as follows:*

$$1/K \leq \lambda_{\min}(\boldsymbol{\Sigma}^*) \leq \lambda_{\max}(\boldsymbol{\Sigma}^*) \leq K.$$

Furthermore we assume that K does not increase as d goes to infinity.

Assumption 4.2 *$\boldsymbol{\Theta}^*$ belongs to the following set:*

$$\mathcal{U}(s', M) = \Big\{\boldsymbol{\Theta} : \boldsymbol{\Theta} \succ 0, \|\boldsymbol{\Theta}\|_1 \leq M, \max_{1\leq j\leq d} \sum_{k=1}^{d} \mathbb{1}(\Theta_{jk} \neq 0) \leq s'\Big\}.$$

Assumption 4.2 is a common assumption made in the literature of sparse precision matrix estimation (Cai et al., 2011). It implies that the data can be viewed as generated from a sparse Gaussian graphical model, where the maximum degree of the graph is no larger than s'. Note that Assumption 4.2 immediately implies that $\|\boldsymbol{\theta}_j^*\|_1 \leq \|\boldsymbol{\Theta}^*\|_1 \leq M$ for all $j \in \{1, 2, \ldots, d\}$.

In most literatures on high dimensional sparse estimation (Bickel et al., 2009; Negahban et al., 2009), it is assumed that the sample covariance matrix satisfies the restricted eigenvalue condition. Following is the definition of the restricted eigenvalue condition that we use in this paper.

Definition 4.3 *A matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ satisfies the restricted eigenvalue (RE) condition with parameters (s, α, γ) if and only if for any index set S with $|S| \leq s$, for any vector $\mathbf{v}$ in the cone*

$$\mathbb{C}(S, \alpha) = \{\mathbf{v} \in \mathbb{R}^d : \|\mathbf{v}_{S^c}\|_1 \leq \alpha\|\mathbf{v}_S\|_1\},$$

we have $\mathbf{v}^\top \mathbf{A}\mathbf{v} \geq \gamma\|\mathbf{v}\|_2^2$.

With this definition, the assumption made on the sample covariance matrices can be presented as follows.

Condition 4.4 *For each $l \in \{1, 2, \ldots, m\}$, $\widehat{\boldsymbol{\Sigma}}^{(l)}$ satisfies the restricted eigenvalue condition with respect to the parameters $(\max\{s, s'\}, 1, \lambda_{\min}(\boldsymbol{\Sigma}^*)/16)$.*

The following proposition shows that Condition 4.4 is satisfied with high probability when the sample size n is sufficiently large.

Proposition 4.5 *If $n > \max\{s, s'\}r^{-1}C_1K^3 \log d$, Condition 4.4 is satisfied with probability at least $1 - mC_2\exp(-C_3n) - 2m/d$, where C_1, C_2 and C_3 are absolute constants.*

Now we are ready to present the main theorem bounding the estimation error of $\bar{\boldsymbol{\beta}}$.

Theorem 4.6 *Under Assumptions 4.1, 4.2 and Condition 4.4, if $\lambda = C_1K^2\sqrt{\log d/(rn)}\|\boldsymbol{\beta}^*\|_1$, $\lambda' = C_2K^2M\sqrt{\log d/n}$ for some C_1 and C_2, and t is chosen as*

$$t = C'M\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1 + C''\max(s, s')M\frac{m\log d}{N}\|\boldsymbol{\beta}^*\|_1, \tag{4.1}$$

where C' and C'' are absolute constants, then the following inequality holds with probability at least $1 - 18m/d - 4/d$:

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_\infty \leq C'M\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1 + C''\max(s, s')M\frac{m\log d}{N}\|\boldsymbol{\beta}^*\|_1. \tag{4.2}$$

Moreover, with probability at least $1 - 18m/d - 4/d$ we have

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \leq \sqrt{s}C'M\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1 + \sqrt{s}C''\max(s, s')M\frac{m\log d}{N}\|\boldsymbol{\beta}^*\|_1, \tag{4.3}$$

and with probability at least $1 - 18m/d - 4/d$ we have

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1 \leq sC'M\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1 + sC''\max(s, s')M\frac{m\log d}{N}\|\boldsymbol{\beta}^*\|_1. \tag{4.4}$$

The proof of Theorem 4.6 is in Appendix A. It is worth noting that in the linear discriminant analysis, only the direction of $\bar{\boldsymbol{\beta}}$ affects the discrimination, while the norm of $\bar{\boldsymbol{\beta}}$ does not matter. Therefore, the relative error, i.e., the ratio of the norm of $\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$ to the norm of $\boldsymbol{\beta}^*$, can better characterize the accuracy of the estimator.

Remark 4.7 *The centralized estimator can be regarded as a special case of the biased estimator (3.1) where $m = 1$ and $n = N$. Hence by Lemma B.4 the error bound of the centralized estimator can be obtained: with probability at least $1 - 6/d$ we have*

$$\|\widehat{\boldsymbol{\beta}}^{\text{centralized}} - \boldsymbol{\beta}^*\|_1 \leq sCK\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1,$$

where C is a constant. Compared with our distributed estimator, it can be seen that the error bound of the centralized estimator is of the same order with the first term of our proposed estimator, which is in the order of $O(\sqrt{\log d/N})$. And the second term of the error bound of our estimator is in the order of $O(m \log d/N)$, reflecting the loss caused by the data distribution and one round of communication.

Corollary 4.8 *Under the same assumptions with Theorem 4.6, if the number of machines m is chosen to be*

$$m \lesssim \frac{1}{\max(s, s')} \sqrt{\frac{N}{\log d}}, \tag{4.5}$$

then with probability at least $1 - 18m/d - 4/d$ the following inequalities holds:

$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_\infty \le CM\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1,$$
$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \le \sqrt{s}CM\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1,$$
$$\|\bar{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1 \le sCM\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1,$$

where C is a constant.

Remark 4.9 *Generally speaking, the distributed estimation may cause information loss and lead to a worse estimation error bound. However, Corollary 4.8 suggests that if the number of machines m satisfies $m \lesssim \sqrt{N/\log d}/\max(s, s')$ when N, d, s and s' grow, the information loss is negligible and the distributed algorithm can attain the same rate of convergence as the centralized algorithm.*

In fact, the ℓ_∞ estimation error bound in Theorem 4.6 ensures that the estimated parameter vector correctly excludes all non-informative variables and includes all useful variables provided that

$$|\beta_j^*| > C'M\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1 + C''\max(s, s')M\frac{m\log d}{N}\|\boldsymbol{\beta}^*\|_1,$$

where C' and C'' are the same as in Theorem 4.6. Therefore, in order to achieve the model selection consistency, it is sufficient to assume that the minimum signal strength $\beta_{\min} := \min_{j\in S}|\beta_j^*|$ is not too small. More specifically, we have the following theorem:

Theorem 4.10 *Under the same assumptions with Theorem 4.6, if*

$$\beta_{\min} > C'M\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1 + C''\max(s, s')M\frac{m\log d}{N}\|\boldsymbol{\beta}^*\|_1, \tag{4.6}$$

where C' and C'' are those appeared in Theorem 4.6, we have with probability higher than $1 - 18m/d - 4/d$ that $\text{sign}(\bar{\beta}_j) = \text{sign}(\beta_j^*)$ for any $j \in \{1, 2, \ldots, d\}$.

Similar to Corollary 4.8, we have the following conclusion:

Corollary 4.11 *Under the same assumptions with Theorem 4.10, if the following two condition holds:*

$$m \lesssim \frac{1}{\max(s, s')}\sqrt{\frac{N}{\log d}}, \quad \beta_{\min} > CM\sqrt{\frac{\log d}{N}}\|\boldsymbol{\beta}^*\|_1 \tag{4.7}$$

for some C, then we have with probability at least $1 - 18m/d - 4/d$ that $\text{sign}(\bar{\beta}_j) = \text{sign}(\beta_j^*)$ for any $j \in \{1, 2, \ldots, d\}$.

Remark 4.12 *In Cai and Liu (2011), the authors did not provide theoretical guarantee on the support recovery. Mai et al. (2012) showed that the condition on $\beta_{\min}$ needed for model selection consistency is $\beta_{\min} \gtrsim s\sqrt{\log(sd)/N}$. The condition for the ROAD estimator proposed in Fan et al. (2012) to satisfy the model selection consistency is $\beta_{\min} \gtrsim \sqrt{\log d/N}$ (Kolar and Liu, 2015), which is proved to be minimax optimal. It is obvious that our condition implied by Corollary 4.11 matches the minimax lower bound in Kolar and Liu (2015) and is better than Mai et al. (2012). However, for the ROAD estimator, a very stringent irrepresentable condition is required for the model selection consistency to hold. For our algorithm, the irrepresentable condition is not required.*

5 EXPERIMENTS

In this section, we verify the performance of the distributed LDA algorithm using both synthetic data and real data. We compared it with the centralized sparse LDA estimator, and naively averaged sparse LDA estimator. In the centralized SLDA, all samples are collected in one machine and the model is estimated by Cai and Liu (2011). In the naively averaged SLDA estimator, we apply Cai and Liu (2011) to the data on each machine to obtain local estimators. The local estimators are directly averaged without debiasing. In other words, the naively averaged SLDA estimator can be written as $\widehat{\boldsymbol{\beta}}_{\text{n}} = (\sum_{l=1}^m \widehat{\boldsymbol{\beta}}^{(l)})/m$.

5.1 Synthetic Data Experiments

The synthetic data are generated by setting $\boldsymbol{\Sigma}^*$ and $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ as follows: $\boldsymbol{\Sigma}^* \in \mathbb{R}^{d\times d}$ with $d = 200$, and $\Sigma_{jk}^* = 0.8^{|j-k|}$ for all $j, k \in \{1, \ldots, d\}$. Additionally, we choose $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^d$ as $\boldsymbol{\mu}_1 = \mathbf{0}$ and $\boldsymbol{\mu}_2 = (1, 1, \ldots, 1, 0, 0, \ldots, 0)^\top$, where the number of 1's is 10. It is easy to get that $\boldsymbol{\beta}^*$ is a sparse vector with 11 nonzero entries. We set $r = 0.5$, which means that

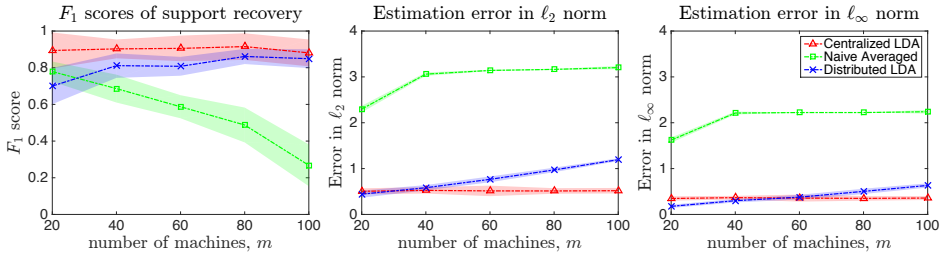

Figure 1: The F_1 score and estimation error (in ℓ_2 and ℓ_∞ norms) of the proposed estimator versus the centralized estimator and the naive averaged estimator when the total sample size N is fixed as 10000.

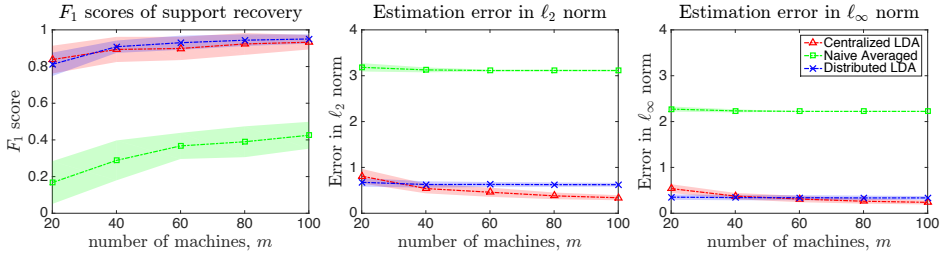

Figure 2: The F_1 score and estimation error (in ℓ_2 and ℓ_∞ norms) of the proposed estimator versus the centralized estimator and the naive averaged estimator when the sample size on each machine n is set to 200.

there are equal number of samples from the two normal distributions on each machine.

We use the following metrics to evaluate the performance of algorithms for comparison: the ℓ_2 and ℓ_∞ norms of parameter estimation error. Additionally, to measure the support recovery, F_1 score is used to measure the overlap of estimated supports and true supports. The definition of F_1 score is as follows

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})},$$

where $\text{precision} = |\operatorname{supp}(\bar{\boldsymbol{\beta}}) \cap \operatorname{supp}(\boldsymbol{\beta}^*)|/|\operatorname{supp}(\bar{\boldsymbol{\beta}})|$, $\text{recall} = |\operatorname{supp}(\bar{\boldsymbol{\beta}}) \cap \operatorname{supp}(\boldsymbol{\beta}^*)|/|\operatorname{supp}(\boldsymbol{\beta}^*)|$, where $\bar{\boldsymbol{\beta}}$ is some estimator. Here $|\cdot|$ is the cardinality of a set.

For the centralized estimator and the naively averaged estimator, there is one regularization parameter λ. By the theoretical result, a proper choice of λ should be in the order of $O(\sqrt{N^{-1}\log d})$ for centralized estimator, and $O(\sqrt{n^{-1}\log d})$ for naively averaged estimator. Therefore, we set $\lambda = C\sqrt{N^{-1}\log d}$ (or $C\sqrt{n^{-1}\log d}$) and tune C by grid search. For the proposed estimator, other than λ, there are two more parameters to be tuned: λ' and t. The theoretical result reveals that λ' should be in the order of $O(\sqrt{n^{-1}\log d})$. Thus, we simply set $\lambda' = \lambda$. The parameter t is tuned in a similar way as the tuning of λ. We report the best results for all methods for the sake of fairness.

To investigate the effect of number of machines m, we fix the total sample size $N = 10000$ and vary the number of machines. Figure 1 shows how the F_1 score and estimation error (in ℓ_2 and ℓ_∞ norm) of the proposed estimator change as the number of machine grows. The widths of the curves represent the standard deviations of metrics such as F_1 scores and ℓ_2, ℓ_∞ norms. The standard deviations are obtained after repeating the experiments 20 times.

From Figure 1, it can be seen that the proposed distributed LDA algorithm is comparable to the centralized LDA estimator in both support recovery and parameter estimation when m is small, while the naive averaged estimator is much worse. Moreover, we can see that the estimation error of distributed LDA will be larger than that of centralized LDA as m surpasses a certain threshold. This is consistent with the result of Theorem 4.6. That is, if m is too big, the dominating term in the estimation error bound will be the second term, which depends on m.

Next we focus on the effect of averaging. We increase the number of machines m linearly as the total sample size N, that is, the sample size on each machine n is fixed. More specifically, we choose $n = 200$. Figure 2 displays the F_1 score, estimation error of our estimator, naively averaged estimator and centralized estimator in terms of ℓ_2 and ℓ_∞ norms. We can see that the performance of distributed LDA is comparable to that of centralized LDA, while the performance of naively averaged estimator is much worse. We can

Table 1: The computation time of distributed LDA vs. centralized LDA ($m = 1$ indicates centralized algorithm).

m	1	20	40	60	80	100
time (in second)	863.4	48.37	33.65	21.87	15.46	10.38

Table 2: Result of Real Data Experiments: Misclassification rates of different methods

m	Centralized SLDA	Naive Averaged SLDA	Distributed SLDA
4	0.208 ± 0.012	0.329 ± 0.035	0.220 ± 0.017

also observe that as N grows linearly with respect to m (i.e., n is fixed), the estimation error of distributed LDA decreases slower than that of centralized LDA. This is consistent with what Theorem 4.6 suggests: in (4.2) and (4.3), if n is fixed and m is growing, the first term of the error bounds will decrease because it is of the order $O(1/\sqrt{N})$. However, the second term in the error bounds will not decrease because it depends on $m/N = 1/n$. Therefore, the total estimation error of our algorithm will converge to a positive constant.

The empirical computation time of distributed LDA and centralized LDA are summarized in Table 1. We set $d = 200, N = 10^6$ and vary m between 20 and 100. For distributed LDA algorithm, we only take into account the time used in one local machine, rather than the total CPU time consumed by all machines, because the local computations are carried out in parallel. The experiment platform is Linux operating system with 2.8GHz CPU. From Table 1 we can see that the distributed algorithm has lower time cost than the centralized algorithm. Furthermore, Table 1 also demonstrates a near linear speedup with the number of machines, which is consistent with the time complexity analysis in Section 3.

5.2 Real Date Experiments

To verify the effectiveness of the proposed algorithm on real datasets, we use the Heart Disease dataset[1] to conduct the experiment. This dataset contains information of 920 heart disease patients across 4 hospitals. For each patient, there are 13 attributes associated, including gender, age, laboratory test results, etc. Every patient is labeled with the diagnosis result, i.e., whether he or she is diagnosed as heart disease. In the preprocessing step, we extend all categorical attributes into binary dummy variables. For the missing values in any numeric attributes in the dataset, we replace them with the average value of the attribute that it belongs to. After the preprocessing, we get 920 entries, each with 22 numerical attributes.

The dataset is naturally divided into 4 parts by the hospital where each patient is diagnosed. We consider each part as the local data stored in one machine. In every part, we randomly choose half of the data as the training set and the remaining half as the test set. To get a proper choice of parameters, as in the synthetic data experiment, we set $\lambda = C\sqrt{N^{-1}\log d}$ (or $C\sqrt{n^{-1}\log d}$), $\lambda' = \lambda$ and use 5-fold cross validation on the training set to tune C and t. After the training phase, we test the misclassification rate of classifiers obtained by different methods on the test set. The experiment is repeated 10 times (i.e., training and test set splitting) and the averaged misclassification rates with their standard deviations are reported in Table 2. It can be seen that the proposed method greatly decreases the misclassification rate compared with the naive averaged estimator, and achieves a comparable performance with the centralized estimator. This verifies the effectiveness of our algorithm on real data.

6 CONCLUSIONS AND FUTURE WORK

We proposed a communication efficient distributed algorithm for sparse linear discriminant analysis in the high dimensional regime. The key idea is constructing a local debiased estimator on each machine and averaging them over all machines. We addressed an important question that how to choose the number of machines such that the aggregated estimator will attain the same convergence rate as the centralized estimator. Experiments on both synthetic and real datasets support our theory. In the future, we will extend our algorithm and theory to multi-class sparse LDA.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This research was sponsored in part by the National Science Foundation IIS-1618948 and IIS-1652539. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

[1] https://archive.ics.uci.edu/ml/datasets/Heart+Disease

References

Anderson, T. T. W. (1968). *An introduction to multivariate statistical analysis*. John Wiley & Sons.

Balcan, M.-F., Blum, A., Fine, S. and Mansour, Y. (2012). Distributed learning, communication complexity and privacy. *arXiv preprint arXiv:1204.3514* .

Balcan, M.-F., Liang, Y., Song, L., Woodruff, D. and Xie, B. (2015). Distributed kernel principal component analysis. *arXiv preprint arXiv:1503.06858* .

Battey, H., Fan, J., Liu, H., Lu, J. and Zhu, Z. (2015). Distributed estimation and inference with statistical guarantees. *arXiv preprint arXiv:1509.05457* .

Bickel, P. J. and Levina, E. (2004). Some theory for fisher's linear discriminant function,'naive bayes', and some alternatives when there are many more variables than observations. *Bernoulli* 989–1010.

Bickel, P. J., Ritov, Y. and Tsybakov, A. B. (2009). Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics* 1705–1732.

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* **3** 1–122.

Cai, T. and Liu, W. (2011). A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association* **106**.

Cai, T., Liu, W. and Luo, X. (2011). A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association* **106** 594–607.

Dekel, O., Gilad-Bachrach, R., Shamir, O. and Xiao, L. (2012). Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research* **13** 165–202.

Fan, J., Feng, Y. and Tong, X. (2012). A road to classification in high dimensional space: the regularized optimal affine discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **74** 745–771.

Javanmard, A. and Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research* **15** 2869–2909.

Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.

Kolar, M. and Liu, H. (2015). Optimal feature selection in high-dimensional discriminant analysis. *Information Theory, IEEE Transactions on* **61** 1063–1083.

Lee, J. D., Sun, Y., Liu, Q. and Taylor, J. E. (2015). Communication-efficient sparse regression: a one-shot approach. *arXiv preprint arXiv:1503.04337* .

Liang, Y., Balcan, M.-F. F., Kanchanapally, V. and Woodruff, D. (2014). Improved distributed principal component analysis. In *Advances in Neural Information Processing Systems*.

Mai, Q., Zou, H. and Yuan, M. (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika* asr066.

Mcdonald, R., Mohri, M., Silberman, N., Walker, D. and Mann, G. S. (2009). Efficient large-scale distributed training of conditional maximum entropy models. In *Advances in Neural Information Processing Systems*.

Negahban, S., Yu, B., Wainwright, M. J. and Ravikumar, P. K. (2009). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*.

Neykov, M., Ning, Y., Liu, J. S. and Liu, H. (2015). A unified theory of confidence regions and testing for high dimensional estimating equations. *arXiv preprint arXiv:1510.08986* .

Pang, H., Liu, H. and Vanderbei, R. J. (2014). The fastclime package for linear programming and large-scale precision matrix estimation in r. *Journal of Machine Learning Research* **15** 489–493.

Raskutti, G., Wainwright, M. J. and Yu, B. (2010). Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research* **11** 2241–2259.

Rosenblatt, J. and Nadler, B. (2014). On the optimality of averaging in distributed statistical learning. *arXiv preprint arXiv:1407.2724* .

Shao, J., Wang, Y., Deng, X., Wang, S. et al. (2011). Sparse linear discriminant analysis by thresholding for high dimensional data. *The Annals of statistics* **39** 1241–1265.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Valcarcel Macua, S., Belanovic, P. and Zazo, S. (2011). Distributed linear discriminant analysis. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE.

Van de Geer, S., Bühlmann, P., Ritov, Y., Dezeure, R. et al. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics* **42** 1166–1202.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027* .

Zhang, Y., Duchi, J. C. and Wainwright, M. J. (2013). Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *arXiv preprint arXiv:1305.5029* .

Zhang, Y., Wainwright, M. J. and Duchi, J. C. (2012). Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*.

Zinkevich, M., Weimer, M., Li, L. and Smola, A. J. (2010). Parallelized stochastic gradient descent. In *Advances in neural information processing systems*.