

Distributed and Secure Sparse Machine Learning

A Thesis

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

Master of Science

by

Lu Tian

August 2018

APPROVAL SHEET

This Thesis
is submitted in partial fulfillment of the requirements
for the degree of
Master of Science

Author Signature: Shu Tian

This Thesis has been read and approved by the examining committee:

Advisor: Quanquan Gu

Committee Member: David Evans

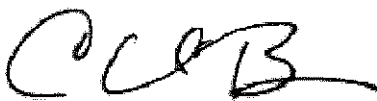
Committee Member: Farzad Farnoud

Committee Member: _____

Committee Member: _____

Committee Member: _____

Accepted for the School of Engineering and Applied Science:



Craig H. Benson, School of Engineering and Applied Science

August 2018

Abstract

With the growth of the volume of data used for machine learning, and the availability of distributed computing resources, distributed machine learning has received increasing attention from machine learning researchers and practitioners. In the meantime, the extensive usage of private data in machine learning makes the privacy issue a major concern of participants of collaborative machine learning. This thesis proposes a communication-efficient algorithm for distributed sparse linear discriminant analysis (LDA), where each local machine gets a debiased estimator from local data, the central machine aggregates the debiased estimators and outputs a final sparsified estimator. At the core of the algorithm is a debiasing step by which the bias caused by regularizer is compensated. It is proved that, with much less communication cost, the aggregated estimator attains the same statistical rate with the centralized estimator, as long as the number of machines is chosen appropriately. Based on the distributed sparse LDA algorithm, we propose a secure multi-party sparse learning method, in which a secure multi-party computation (MPC) protocol was employed to aggregate the local models. The protocol ensures that the local model owned by each party would not be revealed to other parties while the correct aggregated model can still be obtained. Experiments on both synthetic and real world datasets corroborate the performance of the distributed sparse LDA algorithm and the efficiency of the secure multi-party sparse learning method.

Acknowledgements

I would first like to thank my advisor Prof. Quanquan Gu. Working with him is a productive and enjoyable experience. Under his guidance, I got a solid understanding on the foundation of machine learning, including statistical learning theory, optimization, statistics, etc. I would also like to thank Prof. David Evans. He guided me to the field of secure computing, a field that I am always interested in but did not have a chance to study in depth before. I would also like to acknowledge Prof. Farzad Farnoud. His advice on the thesis is very valuable.

I would also like to thank my colleagues in the Department of Computer Science, especially Bargav Jayaraman and Pan Xu, who are also my close collaborators. Without their tremendous support, this thesis cannot be completed. Last but not least, I would like to thank my colleagues in the Statistical Machine Learning Lab. The daily discussion in various topics is very helpful to my study and life.

Author

Lu Tian

To my parents...

Contents

1	Introduction	2
2	Distributed Sparse Machine Learning	5
2.1	Introduction	5
2.2	Related Work	8
2.3	Distributed Sparse Linear Discriminant Analysis	10
2.4	Main Theory	13
2.5	Experiments	16
2.5.1	Synthetic Data Experiments	17
2.5.2	Real Date Experiments	19
2.6	Conclusions and Future Work	20
3	Secure Multi-Party Machine Learning	21
3.1	Introduction	21
3.2	Related Work	22
3.3	Garbled Circuit	22
3.3.1	Oblivious Transfer	23
3.3.2	Building Garbled Circuit by Oblivious Transfer	23
3.4	The Proposed Method	25
3.5	Experiments	25
3.6	Conclusion	27
4	Conclusion and Future Work	28
A	Proof of Theorems	36
A.1	Proof of Theorems, Corollaries and Propositions in Chapter 2	36
A.1.1	Proof of Proposition 2.4.5	37
A.1.2	Proof of Theorem 2.4.6	38

A.1.3	Proof of Corollary 2.4.8	39
A.1.4	Proof of Theorem 2.4.10	40
A.1.5	Proof of Corollary 2.4.11	41
A.2	Proof of Lemmas in Appendix A.1	41
A.2.1	Proof of Lemma A.1.1	43
A.2.2	Proof of Lemma A.1.2	46
A.3	Proof of Lemmas in Appendix A.2	47
A.3.1	Proof of Lemma A.2.1	48
A.3.2	Proof of Lemma A.2.2	48
A.3.3	Proof of Lemma A.2.3	49
A.3.4	Proof of Lemma A.2.4	50
A.3.5	Proof of Lemma A.2.5	51
A.4	Proof of Auxilliary Lemmas in Appendix A.3	53
A.4.1	Proof of Lemma A.3.1	53
A.4.2	Proof of Lemma A.3.2	53
A.5	Auxiliary Definitions, Lemmas and Theorems	54

Chapter 1

Introduction

With the explosive growth of data generated everyday, it has been very common to store data in different machines. In the meantime, the development of computer network makes it possible to process the data in a distributed fashion. Researchers have been studying distributed data processing algorithms for decades. The advantage of a distributed algorithm includes i) it can fully make use of the distributed computing resources; ii) there is no need to transfer data between machines, which can potentially be very expensive in communication.

Machine learning, which is originated in 1960s, has become one of the most active research topics in computer science. In this style of research, an agent learns a concept from large amount of instances, rather than user-defined rules. For example, a machine learning agent trying to learn a classifier for cat images get many cat images (and potentially non-cat images) as input and “trains” itself according to some pre-defined algorithm. Now machine learning algorithms have been applied ubiquitously in our daily life. For example, they have been used in the back-end of e-commercial websites, hospitals, financial institutions, etc. They are used to predict users’ preference on shopping [32], predict the patient’s readmission rate [16], detect fraud transactions [46], etc.

When distributed data encounters machine learning, many problem emerges. Typically, the performance of a machine learning algorithm improves as the input data size increases. Therefore, machine learning practitioners are willing to make use of data stored on distributed machines as fully as possible. This leads to the research of designing distributed machine learning algorithms [4, 27, 28, 40, 44, 48, 54]. The main challenges in distributed machine learning algorithm design include:

- In the situation that the communication between machines are limited, can we

achieve comparable performance to centralized machine learning algorithms?

- Can we develop algorithms without complicated inter-process communication, such as synchronization primitives?

Both of the two challenges are highly related with efficiency of the algorithms. Now the bandwidth between machines is still much less than that within a single machine. Therefore, communication between machines is still the main bottleneck restricting the speed of distributed algorithms. Moreover, too much synchronization steps causes many sleep and wake, as well as context switching between processes, which is very time consuming. In this thesis, we mainly focus on the first challenge, while the second one is also a hot topic in the machine learning community [28, 40].

High dimensionality is a major challenge in many machine learning applications. It not only increases the computation cost but also causes the issue of overfitting, i.e., the output of the learning algorithm performs very well on the training data but badly on other data that is not “seen” by the learning agent before. Moreover, in the scenario of distributed machine learning, high dimensionality causes additional problems. People usually employ regularizers to suppress overfitting. However the regularizer also causes bias to the machine learning algorithm. In distributed machine learning, the models learned by each machines are ususally aggregated by averaging. However, the bias cannot be eliminated by simple averaging. Some prior work [15, 22] proposed the debiased estimator for specific tasks such as linear regression, graphical model, etc. In this line of work, the bias contained in a model was estimated and subtracted, and an unbiased model remains. The original aim of debiased estimator designing lies in the field of statistics, such as confidence interval estimation and hypothesis testing. Very fortunately, it can also be used in distributed machine learning as a tool for improving the accuracy of model aggregation [27]. In this thesis we follow this line of research. We propose to improve the performance of distributed sparse linear discriminant analysis (LDA) [1] by the debiased estimator designed for sparse LDA. We will provide theoretical guarantees and empirical evidence of the performance of the proposed approach.

More and more private data are generated everyday. With the development of cloud storage and social network site, many private data are actually stored on Internet rather than private devices. Moreover, user preference or browsing/purchasing history on these websites are usually recorded in order to ease the users’ further using. With the deployment of machine learning algorithms on various web services,

users' concern on the privacy of their data increases. Generally, machine learning algorithms get users' own data as input and generate a model (classifier/predictor) as output, while these algorithms are often public. Therefore, given the knowledge of the algorithm used, a malicious party may infer user's data from the output. In order to protect user privacy to the maximum extent, we should design our machine learning algorithm very carefully to defend the data against adversaries.

There are mainly two lines of research in private machine learning. The first is to leverage cryptography protocols such that each party's input and intermediate results are not known by other parties while the final output of the algorithm is still accessible by all parties [30, 51, 55]. A representative protocol that can fulfill this requirement is the Garbled Circuit proposed by Yao [57]. In the other line, researchers try to inject the noise into the algorithm such that the output is random enough such that the adversary cannot infer the data accurately by the output. This line of research includes Chaudhuri and Monteleoni [10], Kifer, Smith, and Thakurta [24], and Talwar, Thakurta, and Zhang [47]. There are also studies combining the two lines, such as Pathak, Rane, and Raj [38].

In this thesis, we will follow the first line of research and propose a multi-party private sparse learning method which can be shown to be quite efficient in high-dimensional case. The method is based on secure multi-party computation (MPC) and the debiased estimator proposed in the last work. It is worth noting that although this work follows the first line, it has the potential to be extended such that the advantages of both lines are taken.

The thesis is organized as follows. Chapter 2 demonstrates the motivation, details and theoretical guarantees of the new distributed learning algorithm. In Chapter 3, we will combine it with Garbled Circuit, and show the effectiveness of the new secure algorithm. Chapter 4 concludes the thesis and proposes a couple of potential future work. The results described in Chapter 2 has been published as Tian and Gu [48]. Part of the content of Chapter 3 has been published as Tian, Jayaraman et al. [49].

Chapter 2

Distributed Sparse Machine Learning

2.1 Introduction

High dimensionality is a main challenge in machine learning applications. It usually leads to high time and space requirements for processing the data. What is more, overfitting is another problem that machine learning methods would meet in the presence of high dimensionality. A common way to address the problems caused by high dimensionality is the dimensionality reduction. Principal Component Analysis (PCA) [23] is probably the most well known and widely used dimensionality reduction method. However, PCA is basically an unsupervised dimensionality reduction method. It only considers the distributions of features of instances, without taking into account the labels of the data.

In order to leverage the label information into dimensionality reduction, supervised dimensionality reduction methods are favored. Linear Discriminant Analysis (LDA) [1], which is initially proposed as a classification method, is an important supervised dimensionality reduction method. The motivation of LDA is as follows. Let there be two classes, and the data of each class are drawn independently from a multivariate normal distribution. Moreover, we assume that the two normal distributions share the same covariance matrix but with different mean vectors. Formally, we denote the distribution of data in Class 1 as $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}^*)$ and Class 2 as $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}^*)$. For a new observation \mathbf{Z} that is drawn with equal prior probability from the two normal

distributions, the Fisher's linear discriminant rule takes the form

$$\psi(\mathbf{Z}) = \mathbb{1}((\mathbf{Z} - \boldsymbol{\mu})^\top \boldsymbol{\Theta}^* \boldsymbol{\mu}_d > 0), \quad (2.1.1)$$

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. However, 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. Typically, they can be estimated by sample means and sample covariance matrix. More specifically, let $\{\mathbf{X}_i : 1 \leq i \leq n_1\}$ and $\{\mathbf{Y}_i : 1 \leq i \leq n_2\}$ be independently and identically distributed random samples from $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}^*)$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}^*)$ respectively. The estimations of $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\Theta}^*$ in the classical regime are $\hat{\boldsymbol{\mu}}_1 = n_1^{-1} \sum_{i=1}^{n_1} \mathbf{X}_i$, $\hat{\boldsymbol{\mu}}_2 = n_2^{-1} \sum_{i=1}^{n_2} \mathbf{Y}_i$, and $\hat{\boldsymbol{\Theta}} = \hat{\boldsymbol{\Sigma}}^{-1}$, where $\hat{\boldsymbol{\Sigma}} = n^{-1} [\sum_{i=1}^{n_1} (\mathbf{X}_i - \hat{\boldsymbol{\mu}}_1)(\mathbf{X}_i - \hat{\boldsymbol{\mu}}_1)^\top + \sum_{i=1}^{n_2} (\mathbf{Y}_i - \hat{\boldsymbol{\mu}}_2)(\mathbf{Y}_i - \hat{\boldsymbol{\mu}}_2)^\top]$ is the pooled sample covariance matrix with $n = n_1 + n_2$. Plugging these estimators into (2.1.1) gives rise to the empirical version of $\psi(\mathbf{Z})$, i.e., $\hat{\psi}(\mathbf{Z})$. Theoretical properties of $\hat{\psi}(\mathbf{Z})$ have been well studied when d is fixed, e.g., see Anderson [1]. However, in the high-dimensional regime where d increases with the same order of n , the pooled sample covariance matrix procedure is not well-conditioned and the plug-in estimator is not reliable. For example, Bickel and Levina [5] 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 (2.1.1). For example, Cai and Liu [8] made the assumption that the weight vector of the classifier in (2.1.1), i.e., $\boldsymbol{\beta}^* = \boldsymbol{\Theta}^* \boldsymbol{\mu}_d$ is a sparse vector. They proposed the following estimator:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\boldsymbol{\beta}\|_1 \quad \text{subject to} \quad \|\hat{\boldsymbol{\Sigma}}\boldsymbol{\beta} - (\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)\|_\infty \leq \lambda, \quad (2.1.2)$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^d |\beta_j|$ is the ℓ_1 norm, and $\|\cdot\|_\infty$ is the element-wise max norm, $\hat{\boldsymbol{\Sigma}}$, $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\mu}}_2$ 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 [14, 33, 43].

On the other hand, with the increase in the volume of data used for machine learning, and the availability of distributed computing resources, distributed statisti-

cal estimation [3, 4, 27, 34, 41, 59, 60] and distributed optimization [7, 12, 61] 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, $s = \|\beta^*\|_0$ and $s' = \max_{1 \leq j \leq d} \|\theta_j^*\|_0$ are the number of nonzero elements in β^* and θ_j^* respectively, with θ_j^* being the j -th column of Θ^* . 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 [8], 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 [14, 25]. However, the model selection consistency established in Kolar and Liu [25] 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 In this chapter, we use lowercase letters x, y, \dots to denote scalars, bold lowercase letters $\mathbf{x}, \mathbf{y}, \dots$ for vectors, and bold uppercase letters $\mathbf{X}, \mathbf{Y}, \dots$ for matrices. We denote random vectors by \mathbf{X}, \mathbf{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

$\mathbf{x} = [x_1, \dots, x_d]^\top \in \mathbb{R}^d$ be a d -dimensional vector. For $0 < q < \infty$, we define the ℓ_0 , ℓ_q and ℓ_∞ vector norms as $\|\mathbf{x}\|_0 = \sum_{i=1}^d \mathbf{1}(x_i \neq 0)$, $\|\mathbf{x}\|_q = (\sum_{i=1}^d |x_i|^q)^{1/q}$, $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq d} |x_i|$, where $\mathbf{1}(\cdot)$ 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:

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

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 $|f_n| \leq C|g_n|$ 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.

2.2 Related Work

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

LDA has been widely studied in the high dimensional regime where the number of features d can increase as the sample size n [8, 14, 33, 43]. One important problem in the high dimensional regime is that the estimation of Θ^* will be unstable because the sample covariance matrix $\hat{\Sigma}$ is often singular. To address this problem, a common assumption is that both μ_d and Σ^* are sparse. Under this assumption, Shao et al. [43] proposed to use a thresholding procedure to estimate μ_d and Σ^* respectively, followed by the standard procedure to estimate $\psi(\mathbf{Z})$. Cai and Liu [8] assumed that $\beta^* = \Theta^* \mu_d$ 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 Balcan et al. [3], McDonald et al. [34], Zhang, Duchi, and Wainwright [59], Zhang, Wainwright, and Duchi [60], and Zinkevich et al. [61]. 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 any more [41]. 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 [50] is biased due to the ℓ_1 -norm penalty. Since averaging only reduces variances, not the bias, the performance of averaged estimator would not be better than the local estimator due to the aggregation of bias when averaging. To address this problem, Lee et al. [27] proposed distributed sparse regression methods, which exploits the debiased estimators proposed in Geer et al. [15] and Javanmard and Montanari [22] for distributed sparse regression. Similar distributed regression methods are proposed by Battey et al. [4] 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. [29] 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. [2] extended the kernel PCA to the distributed setting and proposed a communication-efficient distributed kernel PCA algorithm. Valcarcel Macua, Belanovic, and Zazo [52] 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.

2.3 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 there be m machines, where the l -th machine stores n_{1l} amount of data with label 1, and n_{2l} data with label 2. Let $\mathbf{X}^{(l)} \in \mathbb{R}^{n_{1l} \times d}$, $l \in \{1, 2, \dots, m\}$ 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 $\mathbf{Y}^{(l)} \in \mathbb{R}^{n_{2l} \times d}$, $l \in \{1, 2, \dots, m\}$ 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 $n_{11} = n_{12} = \dots = n_{1m} = n_1$ and $n_{21} = n_{22} = \dots = n_{2m} = n_2$. 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 [8] to directly estimate $\boldsymbol{\beta}^*$ in Algorithm 1.

Algorithm 1 Distributed Sparse Linear Discriminant Analysis

Require: $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(m)}, \mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(m)}$

Ensure: $\bar{\boldsymbol{\beta}}$, the aggregated sparse discriminant vector

Workers:

Each worker computes $\hat{\boldsymbol{\Sigma}}^{(l)}$ and $\hat{\boldsymbol{\mu}}_1^{(l)}, \hat{\boldsymbol{\mu}}_2^{(l)}$

Each worker computes a local sparse LDA estimator $\hat{\boldsymbol{\beta}}^{(l)}$ by (2.3.1)

Each worker computes a debiased estimator $\tilde{\boldsymbol{\beta}}^{(l)}$ by (2.3.4)

Each worker sends $\tilde{\boldsymbol{\beta}}^{(l)}$ to the master machine

Master:

while waiting for $\tilde{\boldsymbol{\beta}}^{(l)}$ sent from all workers **do**

if received $\tilde{\boldsymbol{\beta}}^{(l)}$ from all workers **then**

 Compute the aggregated sparse estimator $\bar{\boldsymbol{\beta}}$ by (2.3.5)

end if

end while

In detail, for the l -th machine, we denote by $\mathbf{X}_i^{(l)}$ and $\mathbf{Y}_i^{(l)}$ the i -th row of $\mathbf{X}^{(l)}$ and $\mathbf{Y}^{(l)}$ respectively. On each machine, we can use the sparse LDA estimator in (2.1.2) to obtain a local estimator as the following:

$$\hat{\boldsymbol{\beta}}^{(l)} = \underset{\boldsymbol{\beta} \in \mathbb{R}^d}{\operatorname{argmin}} \|\boldsymbol{\beta}\|_1 \quad \text{subject to} \quad \left\| \hat{\boldsymbol{\Sigma}}^{(l)} \boldsymbol{\beta} - \hat{\boldsymbol{\mu}}_d^{(l)} \right\|_{\infty} \leq \lambda, \quad (2.3.1)$$

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

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

which is the total intra-class sample covariance matrix of the l -th machine. The choice of λ will be discussed in Section 2.4.

The estimator in (2.3.1) is biased due to the shrinkage property of the estimator. Since averaging only reduce the variance, rather than the bias, if we naively average $\hat{\boldsymbol{\beta}}^{(l)}$'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. [27] and Battey et al. [4]. However, Lee et al. [27] focused on the Lasso estimator, and the debiasing method proposed in Battey et al. [4] 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 [9] is used to estimate the precision matrix:

$$\hat{\boldsymbol{\Theta}}^{(l)} = \operatorname{argmin} \|\boldsymbol{\Theta}\|_{1,1} \quad \text{subject to} \quad \|\boldsymbol{\Theta}^\top \hat{\boldsymbol{\Sigma}}^{(l)} - \mathbf{I}\|_{\infty, \infty} \leq \lambda', \quad (2.3.2)$$

where λ' is a tuning parameter, and its choice will be clear from Section 2.4. It is worth noting that (2.3.2) can be decomposed into d independent optimization problems, where each one takes the form

$$\hat{\boldsymbol{\theta}}_j^{(l)} = \operatorname{argmin} \|\boldsymbol{\theta}\|_1 \quad \text{subject to} \quad \|\hat{\boldsymbol{\Sigma}}^{(l)} \boldsymbol{\theta} - \mathbf{e}_j\|_\infty \leq \lambda', \quad (2.3.3)$$

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

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

$$\tilde{\boldsymbol{\beta}}^{(l)} = \hat{\boldsymbol{\beta}}^{(l)} - \hat{\boldsymbol{\Theta}}^{(l)\top} \left(\hat{\boldsymbol{\Sigma}}^{(l)} \hat{\boldsymbol{\beta}}^{(l)} - \hat{\boldsymbol{\mu}}_d^{(l)} \right). \quad (2.3.4)$$

Note that the second term in the right hand side of (2.3.4) can be seen as the estimation of the bias introduced by the penalized estimator in (2.3.2). We subtract the estimation of the bias from $\hat{\boldsymbol{\beta}}^{(l)}$ and obtain an unbiased estimator $\tilde{\boldsymbol{\beta}}^{(l)}$.

Finally, the workers send back the unbiased local estimators in (2.3.4) 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

$$\bar{\beta} = \text{HT}\left(\frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)}, t\right), \quad (2.3.5)$$

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

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

Here $t > 0$ is a pre-specified threshold. The setting of t will be discussed in Section 2.4.

The proposed distributed algorithm has a low communication cost. In detail, compared with the naive distributed algorithm in which $\hat{\Sigma}^{(l)}$'s and $\hat{\mu}_d^{(l)}$'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 $\hat{\beta}^{(l)}$, the main computation overhead lies on computing $\hat{\Sigma}^{(l)}$, whose time complexity is $O(Nd^2/m)$. For the CLIME estimator, using the FastCLIME method [37], 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.

2.4 Main Theory

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

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

Assumption 2.4.1 *There exists a constant $K \geq 1$, such that the maximum and minimal eigenvalues of Σ^* can be bounded as follows:*

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

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

Assumption 2.4.2 Θ^* *belongs to the following set:*

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

Assumption 2.4.2 is a common assumption made in the literature of sparse precision matrix estimation [9]. 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 2.4.2 immediately implies that $\|\theta_j^*\|_1 \leq \|\Theta^*\|_1 \leq M$ for all $j \in \{1, 2, \dots, d\}$.

In most literatures on high dimensional sparse estimation [6, 35], 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 the theory.

Definition 2.4.3 *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 2.4.4 For each $l \in \{1, 2, \dots, m\}$, $\hat{\Sigma}^{(l)}$ satisfies the restricted eigenvalue condition with respect to the parameters $(\max\{s, s'\}, 1, \lambda_{\min}(\Sigma^*)/16)$.

The following proposition shows that Condition 2.4.4 is satisfied with high probability when the sample size n is sufficiently large.

Proposition 2.4.5 If $n > \max\{s, s'\}r^{-1}C_1K^3 \log d$, Condition 2.4.4 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 $\bar{\beta}$.

Theorem 2.4.6 Under Assumptions 2.4.1, 2.4.2 and Condition 2.4.4, if $\lambda = C_1K^2\sqrt{\log d/(rn)}\|\beta^*\|_1$, $\lambda' = C_2K^2M\sqrt{\log d/n}$ for some C_1 and C_2 , and t is chosen as

$$t = C'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + C''\max(s, s')M\frac{m \log d}{N}\|\beta^*\|_1, \quad (2.4.1)$$

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

$$\|\bar{\beta} - \beta^*\|_\infty \leq C'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + C''\max(s, s')M\frac{m \log d}{N}\|\beta^*\|_1. \quad (2.4.2)$$

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

$$\|\bar{\beta} - \beta^*\|_2 \leq \sqrt{s}C'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + \sqrt{s}C''\max(s, s')M\frac{m \log d}{N}\|\beta^*\|_1, \quad (2.4.3)$$

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

$$\|\bar{\beta} - \beta^*\|_1 \leq sC'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + sC''\max(s, s')M\frac{m \log d}{N}\|\beta^*\|_1. \quad (2.4.4)$$

The proof of Theorem 2.4.6 is in Appendix A.1. It is worth noting that in the linear discriminant analysis, only the direction of $\bar{\beta}$ affects the discrimination, while the norm of $\bar{\beta}$ does not matter. Therefore, the relative error, i.e., the ratio of the norm of $\bar{\beta} - \beta^*$ to the norm of β^* , can better characterize the accuracy of the estimator.

Remark 2.4.7 *The centralized estimator can be regarded as a special case of the biased estimator (2.3.1) where $m = 1$ and $n = N$. Hence by Lemma A.2.4 the error bound of the centralized estimator can be obtained: with probability at least $1 - 6/d$ we have*

$$\|\hat{\beta}^{\text{centralized}} - \beta^*\|_1 \leq sCK^2 \sqrt{\frac{\log d}{N}} \|\beta^*\|_1,$$

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 2.4.8 *Under the same assumptions with Theorem 2.4.6, if the number of machines m is chosen to be*

$$m \lesssim \frac{1}{\max(s, s')} \sqrt{\frac{N}{\log d}}, \quad (2.4.5)$$

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

$$\begin{aligned} \|\bar{\beta} - \beta^*\|_\infty &\leq CM \sqrt{\frac{\log d}{N}} \|\beta^*\|_1, & \|\bar{\beta} - \beta^*\|_2 &\leq \sqrt{s} CM \sqrt{\frac{\log d}{N}} \|\beta^*\|_1, \\ \|\bar{\beta} - \beta^*\|_1 &\leq sCM \sqrt{\frac{\log d}{N}} \|\beta^*\|_1, \end{aligned}$$

where C is a constant.

Remark 2.4.9 *Generally speaking, the distributed estimation may cause information loss and lead to a worse estimation error bound. However, Corollary 2.4.8 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 2.4.6 ensures that the estimated parameter vector correctly excludes all non-informative variables and includes all useful variables provided that

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

where C' and C'' are the same as in Theorem 2.4.6. 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 2.4.10 *Under the same assumptions with Theorem 2.4.6, if*

$$\beta_{\min} > C' M \sqrt{\frac{\log d}{N}} \|\beta^*\|_1 + C'' \max(s, s') M \frac{m \log d}{N} \|\beta^*\|_1, \quad (2.4.6)$$

where C' and C'' are those appeared in Theorem 2.4.6, we have with probability higher than $1 - 18m/d - 4/d$ that $\text{sign}(\bar{\beta}_j) = \text{sign}(\beta_j^*)$ for any $j \in \{1, 2, \dots, d\}$.

Similar to Corollary 2.4.8, we have the following conclusion:

Corollary 2.4.11 *Under the same assumptions with Theorem 2.4.10, if the following two condition holds:*

$$m \lesssim \frac{1}{\max(s, s')} \sqrt{\frac{N}{\log d}}, \quad \beta_{\min} > CM \sqrt{\frac{\log d}{N}} \|\beta^*\|_1 \quad (2.4.7)$$

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 $j \in \{1, 2, \dots, d\}$.

Remark 2.4.12 *In Cai and Liu [8], the authors did not provide theoretical guarantee on the support recovery. Mai, Zou, and Yuan [33] showed that the condition on β_{\min} needed for model selection consistency is $\beta_{\min} \gtrsim s \sqrt{\log(sd)/N}$. The condition for the ROAD estimator proposed in Fan, Feng, and Tong [14] to satisfy the model selection consistency is $\beta_{\min} \gtrsim \sqrt{\log d/N}$ [25], which is proved to be minimax optimal. It is obvious that our condition implied by Corollary 2.4.11 matches the minimax lower bound in Kolar and Liu [25] and is better than Mai, Zou, and Yuan [33]. However, for the ROAD estimator, a very stringent irrerepresentable condition is required for the model selection consistency to hold. For our algorithm, the irrerepresentable condition is not required.*

2.5 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

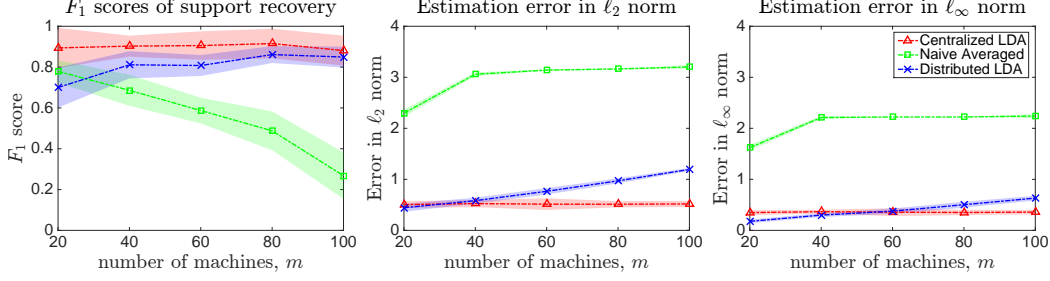


Figure 2.1: The F_1 score and estimation error (in ℓ_2 and ℓ_∞ norms) of the proposed estimator versus the centralized estimator and the naive averaged estimator when the total sample size N is fixed as 10000.

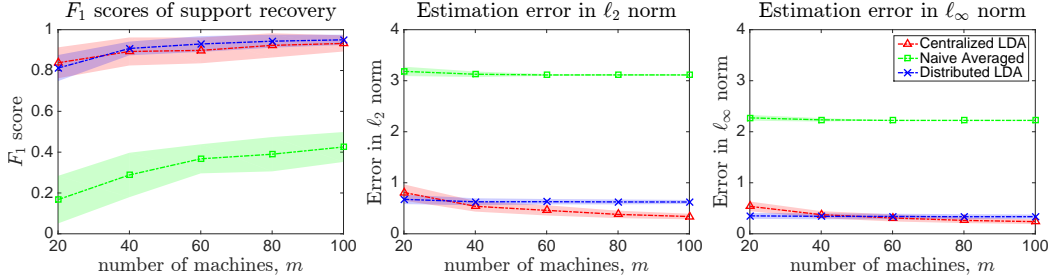


Figure 2.2: The F_1 score and estimation error (in ℓ_2 and ℓ_∞ norms) of the proposed estimator versus the centralized estimator and the naive averaged estimator when the sample size on each machine n is set to 200.

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 [8]. In the naively averaged SLDA estimator, we apply Cai and Liu [8] 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 $\hat{\beta}_n = (\sum_{l=1}^m \hat{\beta}^{(l)})/m$.

2.5.1 Synthetic Data Experiments

The synthetic data are generated by setting Σ^* and μ_1, μ_2 as follows: $\Sigma^* \in \mathbb{R}^{d \times d}$ with $d = 200$, and $\Sigma_{jk}^* = 0.8^{|j-k|}$ for all $j, k \in \{1, \dots, d\}$. Additionally, we choose $\mu_1, \mu_2 \in \mathbb{R}^d$ as $\mu_1 = \mathbf{0}$ and $\mu_2 = (1, 1, \dots, 1, 0, 0, \dots, 0)^\top$, where the number of 1's is 10. It is easy to get that β^* is a sparse vector with 11 nonzero entries. We set $r = 0.5$, which means that 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 com-

Table 2.1: 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

parison: 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 $\text{precision} = |\text{supp}(\bar{\beta}) \cap \text{supp}(\beta^*)| / |\text{supp}(\bar{\beta})|$ and $\text{recall} = |\text{supp}(\bar{\beta}) \cap \text{supp}(\beta^*)| / |\text{supp}(\beta^*)|$, where $\bar{\beta}$ is some estimator. The symbol $|\cdot|$ here means 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 2.1 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 2.1, 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 2.4.6. That is, if m is too big, the dominating term in the estimation error bound will be the second term, which depends on m .

Table 2.2: Misclassification rates of different methods on the real dataset

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

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 2.2 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 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 2.4.6 suggests: in (2.4.2) and (2.4.3), 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 rather than zero.

The empirical computation time of distributed LDA and centralized LDA are summarized in Table 2.1. 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 2.1 we can see that the distributed algorithm has lower time cost than the centralized algorithm. Furthermore, Table 2.1 also demonstrates a near linear speedup with the number of machines, which is consistent with the time complexity analysis in Section 2.3.

2.5.2 Real Date Experiments

To verify the effectiveness of the proposed algorithm on real datasets, we use the Heart Disease dataset¹ 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

¹<https://archive.ics.uci.edu/ml/datasets/Heart+Disease>

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 along with their standard deviations are reported in Table 2.2. 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 LDA estimator. This verifies the effectiveness of our algorithm on real data.

2.6 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 corroborate our theory.

Chapter 3

Secure Multi-Party Machine Learning

3.1 Introduction

As stated in Chapter 1, privacy is one of the major concerns of participants in collaborative machine learning. Communication between machines is unavoidable in distributed machine learning. When sensitive data are stored in machines, communication may leak the privacy to other parties, even if the content of communication is just a processing result of the data, rather than the raw data themselves, since a malicious adversary may infer the private input data through the output of an algorithm. Privacy-preserving machine learning has been investigated by researchers for years [10, 11, 20, 21, 24, 45, 47]. More specifically, security problems on different threat models and different privacy notions have been investigated. Most of the work focuses on preserving the privacy of a single party, such as Chaudhuri and Monteleoni [10], Chaudhuri, Monteleoni, and Sarwate [11], Jain, Kothari, and Thakurta [20], Jain and Thakurta [21], Smith and Thakurta [45], and Talwar, Thakurta, and Zhang [47]. In the meantime, multi-party privacy preserving machine learning methods have also been studied by Hamm, Cao, and Belkin [17], Pathak, Rane, and Raj [38], and Shokri and Shmatikov [44]. An important component in multi-party privacy preserving machine learning is a protocol to securely aggregate models generated by different parties. This is typically fulfilled by Multi-Party Computation (MPC) protocol. A well-known implementation of MPC is Garbled Circuit [31, 56, 57], which will be discussed in detail in Section 3.3.

In this chapter we propose a secure multi-party sparse learning algorithm. This

algorithm leverages the debiased estimator presented in Chapter 2 to get an accurate aggregated model while employing MPC to keep the aggregating process being secure. The detail of the proposed method will be demonstrated in Section 3.4. The experiment to verify the effectiveness of the proposed method is presented in Section 3.5.

3.2 Related Work

Some prior arts [30, 51, 55] have proposed encrypting the data and using MPC throughout the machine learning algorithm. This approach provides strong privacy guarantee, and the output of the approach is exactly the same as non-secure approaches since no noise or approximation was involved. However, it is less effective due to the relatively high computational cost of MPC.

In Pathak, Rane, and Raj [38], each party first trains a local model, and then the local models are aggregated using MPC and published after adding some noise. Shokri and Shmatikov [44] proposed to add noise on the update of each iteration in training deep neural networks. These methods are very efficient because the local models are trained in a distributive way and can be aggregated asynchronously. However, the accuracy of the output model is decreased due to the injected noise in the algorithm. Hamm, Cao, and Belkin [17] proposed to learn a differential private classifier from local models with the help of unlabeled data. All these models are restricted to the classical low-dimensional regime. It is not clear how to extend them to high-dimensional setting.

3.3 Garbled Circuit

Garbled Circuit is a protocol that allowing multiple parties collaboratively perform computation without revealing the input data of each party to other parties. It is based on a more fundamental protocol called Oblivious Transfer (OT) [13]. Here we first briefly introduce oblivious transfer, then discuss how to build garbled circuit based on oblivious transfer. Part of the content in this section is based on Lindell and Pinkas [31].

3.3.1 Oblivious Transfer

Oblivious transfer is a building block for secure multi-party computation. Here we use a simple example to show the functionality of oblivious transfer. Let there be two parties, a sender and a receiver. The sender input a pair strings (s_0, s_1) and the receiver inputs a bit $\sigma \in \{0, 1\}$. By oblivious transfer, the receiver gets s_σ , i.e., the string corresponding to the receiver's own input. Moreover, both the sender and the receiver learn nothing from the transfer except what they are supposed to know. In other words, after the transfer, the sender does not know which string was got by the receiver, and the receiver does not know the content of the other string. If there is a protocol that meets the requirements above, it is called a 1-out-of-2 oblivious transfer.

Oblivious transfer can be implemented by cryptological approaches. For example, if we assume the semi-honest threat model, i.e., we assume that the parties honestly follow the protocol but are curious about other parties' secret, the 1-out-of-2 OT can be implemented as follows. The receiver randomly samples a public key P_σ whose decryption key it knows, and another public key $P_{1-\sigma}$ whose decryption key it does not know. The receiver sends the two keys to the sender, the sender encrypts the two strings by the two keys and send the encrypted strings back to the receiver. Since the receiver only knows one decryption key, it can only get one string among the two. In the threat model that parties can be malicious, the implementation would be a little bit difficult. The detailed implementation of oblivious transfer is out of the scope of this thesis.

3.3.2 Building Garbled Circuit by Oblivious Transfer

Let us take two-party secure computation as an example. Let $f(x, y)$ be the function that we wish to compute securely, where x is the input of party A and y is the input of party B . Without loss of generality, we assume that x and y are two strings of bits. Garbled Circuit is based on decomposing $f(x, y)$ into combination of logic gates, i.e., a circuit. Based on the circuit representation of $f(\cdot, \cdot)$, garbled circuit generally consists of the following three steps:

1. Party A hardwires its input into the circuit, i.e., generating the circuit computing $f(x, \cdot)$.
2. For every wire in the circuit (corresponding to every intermediate result in the

Table 3.1: Garbled truth table for an AND gate

Wire a	Garbled	Wire b	Garbled	Wire c	Garbled	Encrypted Entry
0	W_a^0	0	W_b^0	0	W_c^0	$E_{W_a^0}(E_{W_b^0}(W_c^0))$
0	W_a^0	1	W_b^1	0	W_c^0	$E_{W_a^0}(E_{W_b^1}(W_c^0))$
1	W_a^1	0	W_b^0	0	W_c^0	$E_{W_a^1}(E_{W_b^0}(W_c^0))$
1	W_a^1	1	W_b^1	1	W_c^1	$E_{W_a^1}(E_{W_b^1}(W_c^1))$

calculation), party A assign two random numbers for two states, one and zero. The random numbers are called garbled values.

3. For every gate, party A prepares the encrypted truth table according to the random numbers generated in the last step. Each entry of the table is the encrypted value of the garbled output using the garbled inputs as the encryption key. The details are shown below.
4. For all wires for party B 's input, use oblivious transfer to transfer the corresponding garbled values to party B .
5. Party A sends all garbled values of its input to Party B . Party B evaluate all gates by decoding the truth tables.

Preparing the garbled truth table for each gate is the core of the protocol. Taking AND gate as an example, let the input wires be as the a and b -th wire, and the output wire be the c -th. Furthermore, let the two random numbers for wire i be W_i^0, W_i^1 , corresponding to value 0 and 1. The real and garbled values, as the encrypted version of the truth table are shown in Table 3.1. Here $E_k(\cdot)$ is a symmetric encryption function with key k . The last column of Table 3.1 would be calculated by party A and sent to party B . If party B knows the garbled value of two input, say W_a^0 on wire a and W_b^1 on wire b , it can decrypt only one of the four entries and get W_c^0 . Here we assume that the decryption algorithm can tell party B that the decryption is correct or not. Here party B can get the garbled representation of the result but not the real result, because the correspondence between garbled value and real value is hold by party A . Party B can use the garbled result for further gate evaluation.

3.4 The Proposed Method

Based on the MPC protocol and debiased estimator presented in Chapter 2, we propose the following method for secure multi-party sparse machine learning. The data flow of the method is illustrated in Figure 3.1. In this framework, each party possesses some private data, and generates a debiased estimator by the algorithm presented in Chapter 2. Then the debiased estimators are aggregated using MPC module. It is worth noting that garbled circuit is designed for calculating functions defined on boolean values, and potentially it can be extended to any discrete values. However, the value appeared in machine learning is typically a floating point number. Although the addition of floating numbers can also be implemented by gates, it is usually too complicated and may cause a high computational cost for MPC. To simplify the calculation and ensure that all input numbers are represented by the same number of bits, we multiply each floating point number input by a big integer (say, 10^8) and ignore the fractional part. Since the MPC module in the method only executes the accumulation, we can get back to the original scale by dividing the output by the same big integer. Note that the truncation may cause some accuracy loss. Nevertheless, the experiment result reveals that the accuracy loss is negligible.

To simplify the procedure of MPC, we employ secret sharing [42] to convert the multi-party computation task to a 2-party computation task. Secret sharing is a method to distribute a secret among a group of parties such that the secret can be recovered only when a sufficient number of parties combine the pieces owned by them together. In other words, each party cannot recover the secret individually. In our case, a two-party secret sharing is enough. More specifically, we assume that there are two trustful parties (i.e., they do not collude), S_1 and S_2 . For any data owner P_i , it generates a random bit string $r^{(i)}$, sends it to S_2 , performs bitwise Exclusive OR (XOR) between $r^{(i)}$ and the debiased estimator $\tilde{\beta}^{(i)}$ and sends the result to S_1 . Then we only need to perform secure two-party computation between S_1 and S_2 . Due to the presence of the random bit strings, either S_1 or S_2 have no knowledge about the output of each data owner as long as they do not collude.

3.5 Experiments

It is worth noting that using MPC to perform calculation on floating point numbers requires the user to round the numbers to fixed number of decimals, therefore might

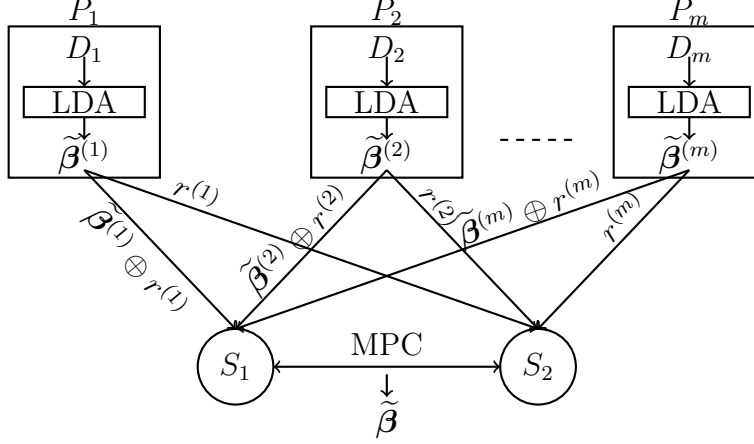


Figure 3.1: Illustration for Secure Multi-Party Sparse Linear Discriminant Analysis

be inaccurate. Moreover, the time consumption of the MPC protocol is also worth investigation. We perform experiments on both synthetic dataset and real-world dataset.

We use Obliv-C [58] to implement the MPC protocol in the proposed method. Obliv-C is a wrapper for C compilers that enables users to easily embed secure computation protocols inside C programs. Moreover, Obliv-C contains recent improvements on garbled circuits [18, 19, 26].

The experiment setting is basically the same as in Chapter 2. The difference is that we involve the secure multi-party LDA in comparison. We only consider the experiment setting that the dataset size in each machine is fixed and the number of machines varies among $\{20, 40, 60, 80, 100\}$. Here we focus on the misclassification rate of the model learned by different methods. We are also interested about the computational cost of MPC. The misclassification rates, as well as the number of gates evaluated in both the synthetic and real datasets experiments are shown in Table 3.2.

From Table 3.2 it can be seen that the approach introduced in Chapter 3 has nearly the same performance with the distributed LDA introduced in Chapter 2. This means that the round error is negligible in practice. In terms of the computational cost, we can see that the number of gates evaluated scales linearly with the number of parties, which is consistent with our expectation. Since the MPC gate evaluation speed is over 4M gates per second, the time to perform MPC is within one second. This indicates that our method can scale to larger datasets.

Dataset	m	Misclassification Rate			Number of gates
		Centralized LDA	Distributed LDA	Our Approach	for MPC
Synthetic	20	0.168 ± 0.002	0.182 ± 0.003	0.182 ± 0.003	1,056,800
Synthetic	40	0.167 ± 0.001	0.180 ± 0.002	0.180 ± 0.002	1,295,600
Synthetic	60	0.166 ± 0.001	0.179 ± 0.002	0.179 ± 0.002	1,559,800
Synthetic	80	0.166 ± 0.001	0.179 ± 0.001	0.179 ± 0.001	1,786,400
Synthetic	100	0.165 ± 0.001	0.179 ± 0.001	0.179 ± 0.001	2,062,600
Real	4	0.208 ± 0.012	0.220 ± 0.017	0.220 ± 0.017	94,200

Table 3.2: Experimental Results (20 repetitions on Synthetic data and 10 repetitions on Real data)

3.6 Conclusion

In this chapter we introduced several important concepts in secure multi-party computing and proposed a framework for secure multi-party sparse linear discriminant analysis. We conducted experiments to show that in this application, the MPC module is very efficient and does not cause any perceivable performance loss to the output of the machine learning algorithm.

Chapter 4

Conclusion and Future Work

In this thesis, a new communication-efficient distributed sparse LDA algorithm is proposed. Combining with secure multi-party computation protocols, we also propose a secure multi-party sparse LDA algorithm. We proved the theoretical guarantee of the proposed distributed algorithm. More specifically, we proved that, as long as the number of machines is less than a threshold, the distributed algorithm can attain the same statistical rate with centralized algorithms, where all data are stored in a single machine and easily accessible by the algorithm. For the secure multi-party sparse LDA algorithm, we conducted experiments on both synthetic and real-world datasets to show that the secure algorithm has no accuracy loss compared with the non-secure algorithm.

It is worth noting that the debiasing technique can not only be used in high-dimensional linear discriminant analysis, but also in other machine learning methods that employ a convex loss function and a regularizer to avoid overfitting, such as sparse linear regression [27], sparse Logistic regression, ridge regression, high dimensional graphical model [15, 54], etc. Therefore the proposed approaches presented in both chapters can be generalized to a wider range of machine learning methods.

In the future we will also explore other privacy notions in secure machine learning, such as differential privacy and its variants. We will also explore secure machine learning methods involving non-convex optimization, such as training deep neural networks, sparse learning by iterative hard-thresholding, etc.

Bibliography

- [1] T. T. W. Anderson. *An introduction to multivariate statistical analysis*. John Wiley & Sons, 1968.
- [2] M.-F. Balcan, Y. Liang, L. Song, D. Woodruff, and B. Xie. “Distributed Kernel Principal Component Analysis”. In: *arXiv preprint arXiv:1503.06858* (2015).
- [3] M.-F. Balcan, A. Blum, S. Fine, and Y. Mansour. “Distributed learning, communication complexity and privacy”. In: *arXiv preprint arXiv:1204.3514* (2012).
- [4] H. Battey, J. Fan, H. Liu, J. Lu, and Z. Zhu. “Distributed Estimation and Inference with Statistical Guarantees”. In: *arXiv preprint arXiv:1509.05457* (2015).
- [5] P. J. Bickel and E. Levina. “Some theory for Fisher’s linear discriminant function, ‘naive Bayes’, and some alternatives when there are many more variables than observations”. In: *Bernoulli* (2004), pp. 989–1010.
- [6] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. “Simultaneous analysis of Lasso and Dantzig selector”. In: *The Annals of Statistics* (2009), pp. 1705–1732.
- [7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122.
- [8] T. Cai and W. Liu. “A direct estimation approach to sparse linear discriminant analysis”. In: *Journal of the American Statistical Association* 106.496 (2011).
- [9] T. Cai, W. Liu, and X. Luo. “A constrained ℓ_1 minimization approach to sparse precision matrix estimation”. In: *Journal of the American Statistical Association* 106.494 (2011), pp. 594–607.
- [10] K. Chaudhuri and C. Monteleoni. “Privacy-preserving logistic regression”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 289–296.

- [11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. “Differentially Private Empirical Risk Minimization”. In: *Journal of Machine Learning Research* 12.Mar (2011), pp. 1069–1109.
- [12] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. “Optimal distributed online prediction using mini-batches”. In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 165–202.
- [13] S. Even, O. Goldreich, and A. Lempel. “A randomized protocol for signing contracts”. In: *Communications of the ACM* 28.6 (1985), pp. 637–647.
- [14] J. Fan, Y. Feng, and X. Tong. “A Road to Classification in High Dimensional Space: the Regularized Optimal Affine Discriminant”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74.4 (2012), pp. 745–771.
- [15] S. Van de Geer, P. Bühlmann, Y. Ritov, R. Dezeure, et al. “On Asymptotically Optimal Confidence Regions and Tests for High-Dimensional Models”. In: *The Annals of Statistics* 42.3 (2014), pp. 1166–1202.
- [16] M. Grzyb, A. Zhang, C. Good, K. Khalil, B. Guo, L. Tian, J. Valdez, and Q. Gu. “Multi-task cox proportional hazard model for predicting risk of unplanned hospital readmission”. In: *Systems and Information Engineering Design Symposium (SIEDS), 2017*. IEEE. 2017, pp. 265–270.
- [17] J. Hamm, P. Cao, and M. Belkin. “Learning Privately from Multiparty Data”. In: *arXiv preprint arXiv:1602.03552* (2016).
- [18] Y. Huang, D. Evans, J. Katz, and L. Malka. “Faster Secure Two-Party Computation Using Garbled Circuits”. In: *USENIX Security Symposium*. 2011.
- [19] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. “Extending Oblivious Transfers Efficiently”. In: *Annual International Cryptology Conference*. 2003.
- [20] P. Jain, P. Kothari, and A. Thakurta. “Differentially Private Online Learning”. In: *arXiv preprint arXiv:1109.0105* (2011).
- [21] P. Jain and A. Thakurta. “Differentially Private Learning with Kernels”. In: *International Conference on Machine Learning* 28 (2013), pp. 118–126.
- [22] A. Javanmard and A. Montanari. “Confidence Intervals and Hypothesis Testing for High-Dimensional Regression”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 2869–2909.
- [23] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

- [24] D. Kifer, A. Smith, and A. Thakurta. “Private Convex Empirical Risk Minimization and High-Dimensional Regression”. In: *Journal of Machine Learning Research* 1 (2012), p. 41.
- [25] M. Kolar and H. Liu. “Optimal feature selection in high-dimensional discriminant analysis”. In: *Information Theory, IEEE Transactions on* 61.2 (2015), pp. 1063–1083.
- [26] V. Kolesnikov and T. Schneider. “Improved Garbled Circuit: Free XOR Gates and Applications”. In: *International Colloquium on Automata, Languages, and Programming*. 2008.
- [27] J. D. Lee, Y. Sun, Q. Liu, and J. E. Taylor. “Communication-Efficient Sparse Regression: a One-Shot Approach”. In: *arXiv preprint arXiv:1503.04337* (2015).
- [28] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su. “Scaling Distributed Machine Learning with the Parameter Server.” In: *OSDI*. Vol. 14. 2014, pp. 583–598.
- [29] Y. Liang, M.-F. F. Balcan, V. Kanchanapally, and D. Woodruff. “Improved Distributed Principal Component Analysis”. In: *Advances in Neural Information Processing Systems*. 2014, pp. 3113–3121.
- [30] Y. Lindell and B. Pinkas. “Privacy Preserving Data Mining”. In: *Advances in Cryptology?-CRYPTO*. Springer. 2000, pp. 36–54.
- [31] Y. Lindell and B. Pinkas. “Secure Multiparty Computation for Privacy-Preserving Data Mining”. In: *Journal of Privacy and Confidentiality* 1.1 (2009), p. 5.
- [32] G. Linden, B. Smith, and J. York. “Amazon. com recommendations: Item-to-item collaborative filtering”. In: *IEEE Internet computing* 7.1 (2003), pp. 76–80.
- [33] Q. Mai, H. Zou, and M. Yuan. “A Direct Approach to Sparse Discriminant Analysis in Ultra-High Dimensions”. In: *Biometrika* (2012), asr066.
- [34] R. Mcdonald, M. Mohri, N. Silberman, D. Walker, and G. S. Mann. “Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models”. In: *NIPS*. 2009.
- [35] S. Negahban, B. Yu, M. J. Wainwright, and P. K. Ravikumar. “A unified framework for high-dimensional analysis of M -estimators with decomposable regularizers”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 1348–1356.

- [36] M. Neykov, Y. Ning, J. S. Liu, and H. Liu. “A Unified Theory of Confidence Regions and Testing for High Dimensional Estimating Equations”. In: *arXiv preprint arXiv:1510.08986* (2015).
- [37] H. Pang, H. Liu, and R. J. Vanderbei. “The fastclime package for linear programming and large-scale precision matrix estimation in R.” In: *Journal of Machine Learning Research* 15.1 (2014), pp. 489–493.
- [38] M. Pathak, S. Rane, and B. Raj. “Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers”. In: *NIPS*. 2010.
- [39] G. Raskutti, M. J. Wainwright, and B. Yu. “Restricted eigenvalue properties for correlated Gaussian designs”. In: *Journal of Machine Learning Research* 11.Aug (2010), pp. 2241–2259.
- [40] B. Recht, C. Re, S. Wright, and F. Niu. “Hogwild: A lock-free approach to parallelizing stochastic gradient descent”. In: *Advances in neural information processing systems*. 2011, pp. 693–701.
- [41] J. Rosenblatt and B. Nadler. “On the Optimality of Averaging in Distributed Statistical Learning”. In: *arXiv preprint arXiv:1407.2724* (2014).
- [42] A. Shamir. “How to share a secret”. In: *Communications of the ACM* 22.11 (1979), pp. 612–613.
- [43] J. Shao, Y. Wang, X. Deng, S. Wang, et al. “Sparse Linear Discriminant Analysis by Thresholding for High Dimensional Data”. In: *The Annals of Statistics* 39.2 (2011), pp. 1241–1265.
- [44] R. Shokri and V. Shmatikov. “Privacy-Preserving Deep Learning”. In: *ACM Conference on Computer and Communications Security*. 2015.
- [45] A. Smith and A. Thakurta. “Differentially Private Feature Selection via Stability Arguments, and the Robustness of the Lasso”. In: *Proceedings of Conference on Learning Theory*. 2013.
- [46] S. Stolfo, D. W. Fan, W. Lee, A. Prodromidis, and P Chan. “Credit card fraud detection using meta-learning: Issues and initial results”. In: *AAAI-97 Workshop on Fraud Detection and Risk Management*. 1997.
- [47] K. Talwar, A. Thakurta, and L. Zhang. “Nearly Optimal Private Lasso”. In: *NIPS*. 2015.

- [48] L. Tian and Q. Gu. “Communication-efficient Distributed Sparse Linear Discriminant Analysis”. In: *Artificial Intelligence and Statistics*. 2017, pp. 1178–1187.
- [49] L. Tian, B. Jayaraman, Q. Gu, and D. Evans. “Aggregating Private Sparse Learning Models Using Multi-Party Computation”. In:
- [50] R. Tibshirani. “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), pp. 267–288.
- [51] J. Vaidya, M. Kantarcioğlu, and C. Clifton. “Privacy-Preserving Naive Bayes Classification”. In: *The VLDB Journal* 17.4 (2008).
- [52] S Valcarcel Macua, P. Belanovic, and S. Zazo. “Distributed linear discriminant analysis”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE. 2011, pp. 3288–3291.
- [53] R. Vershynin. “Introduction to the non-asymptotic analysis of random matrices”. In: *arXiv preprint arXiv:1011.3027* (2010).
- [54] P. Xu, L. Tian, and Q. Gu. “Communication-efficient Distributed Estimation and Inference for Transelliptical Graphical Models”. In: *arXiv preprint arXiv:1612.09297* (2016).
- [55] Z. Yang, S. Zhong, and R. N. Wright. “Privacy-Preserving Classification of Customer Data without Loss of Accuracy”. In: *SIAM International Conference on Data Mining*. 2005.
- [56] A. C. Yao. “How to Generate and Exchange Secrets”. In: *Symposium on Foundations of Computer Science*. 1986.
- [57] A. C. Yao. “Protocols for Secure Computations”. In: *Symposium on Foundations of Computer Science*. 1982.
- [58] S. Zahur and D. Evans. *Obliv-C: A Language for Extensible Data-Oblivious Computation*. Cryptology ePrint Archive, Report 2015/1153. <http://eprint.iacr.org/2015/1153>. 2015.
- [59] Y. Zhang, J. C. Duchi, and M. J. Wainwright. “Divide and Conquer Kernel Ridge Regression: A Distributed Algorithm with Minimax Optimal Rates”. In: *arXiv preprint arXiv:1305.5029* (2013).
- [60] Y. Zhang, M. J. Wainwright, and J. C. Duchi. “Communication-Efficient Algorithms for Statistical Optimization”. In: *NIPS*. 2012.

- [61] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola. “Parallelized stochastic gradient descent”. In: *Advances in neural information processing systems*. 2010, pp. 2595–2603.

Appendix A

Proof of Theorems

A.1 Proof of Theorems, Corollaries and Propositions in Chapter 2

Before we prove the main results, we first lay out a key lemma, which is crucial to establish the main theory.

Lemma A.1.1 *Under Assumptions 2.4.1, 2.4.2 and Condition 2.4.4, if $\lambda = C_1 K^2 \sqrt{\log d / (rn)} \|\beta^*\|_1$, $\lambda' = C_2 K^2 M \sqrt{\log d / n}$, we have with probability at least $1 - 18m/d - 4/d$ that*

$$\left\| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)} - \beta^* \right\|_{\infty} \leq C' M \sqrt{\frac{\log d}{N}} \|\beta^*\|_1 + C'' \max(s, s') M \frac{m \log d}{N} \|\beta^*\|_1,$$

where C' and C'' are constants.

Lemma A.1.1 gives an upper bound on the ℓ_{∞} estimation error for the averaged debiased estimator.

Lemma A.1.2 *Under Assumption 2.4.1, for any $l \in \{1, 2, \dots, d\}$, if we define*

$$\tilde{\Sigma}^{(l)} = \frac{1}{n} \left[\sum_{i=1}^{n_1} (\mathbf{X}_i^{(l)} - \boldsymbol{\mu}_1)(\mathbf{X}_i^{(l)} - \boldsymbol{\mu}_1)^{\top} + \sum_{i=1}^{n_2} (\mathbf{Y}_i^{(l)} - \boldsymbol{\mu}_2)(\mathbf{Y}_i^{(l)} - \boldsymbol{\mu}_2)^{\top} \right], \quad (\text{A.1.1})$$

then with probability at least $1 - 2/d$ the following inequality holds:

$$\|\hat{\Sigma}^{(l)} - \tilde{\Sigma}^{(l)}\|_{\infty, \infty} \leq \frac{CK^2 \log d}{rn}.$$

Now we are ready to prove the main theorems and corollaries.

A.1.1 Proof of Proposition 2.4.5

Proof We denote $s^* = \max(s, s')$. For any index set S satisfying $|S| \leq s^*$, and for any vector \mathbf{u} in the cone $\{\mathbf{u} : \|\mathbf{u}_{S^c}\|_1 \leq \|\mathbf{u}_S\|_1\}$, we have

$$\mathbf{u}^\top \widehat{\Sigma}^{(l)} \mathbf{u} = \mathbf{u}^\top \widetilde{\Sigma}^{(l)} \mathbf{u} + \mathbf{u}^\top (\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}) \mathbf{u} \geq \mathbf{u}^\top \widetilde{\Sigma}^{(l)} \mathbf{u} - |\mathbf{u}^\top (\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}) \mathbf{u}|. \quad (\text{A.1.2})$$

Note that Σ^* satisfies the RE condition with parameter $(s^*, 1, \lambda_{\min}(\Sigma^*))$ since Σ^* is a positive definite matrix following Assumption 2.4.1. From the definition of $\widetilde{\Sigma}^{(l)}$ in (A.1.1) and Theorem A.5.7, it gives rise that there exist three universal constants C_1 , C_2 and C_3 , such that if n satisfies the following inequality:

$$n > \frac{C_1 \rho^2(\Sigma^*)}{\lambda_{\min}^2(\Sigma^*)} s^* \log d,$$

then with probability at least $1 - C_2 \exp(-C_3 n)$, $\widetilde{\Sigma}^{(l)}$ satisfies the RE condition with parameters $(s^*, 1, \lambda_{\min}(\Sigma^*)/8)$, i.e., $\mathbf{u}^\top \widetilde{\Sigma}^{(l)} \mathbf{u} \geq \lambda_{\min}(\Sigma^*)/8 \|\mathbf{u}\|_2^2$. From Assumption 2.4.1 we get that $\lambda_{\min}(\Sigma^*) \geq 1/K$ and $\rho^2(\Sigma^*) \leq \lambda_{\max}(\Sigma^*) \leq K$. Therefore we have

$$\frac{C_1 \rho^2(\Sigma^*)}{\lambda_{\min}^2(\Sigma^*)} s^* \log d \leq C_1 K^3 s^* \log d.$$

Next we give an bound on $|\mathbf{u}^\top (\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}) \mathbf{u}|$: we have

$$|\mathbf{u}^\top (\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}) \mathbf{u}| \leq \|\mathbf{u}\|_1 \cdot \|(\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}) \mathbf{u}\|_\infty \leq \|\mathbf{u}\|_1^2 \cdot \|\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}\|_{\infty, \infty}, \quad (\text{A.1.3})$$

where the first and second inequality follow from Hölder's inequality. Moreover, by $\|\mathbf{u}_{S^c}\|_1 \leq \|\mathbf{u}_S\|_1$ we have $\|\mathbf{u}\|_1 = \|\mathbf{u}_{S^c}\|_1 + \|\mathbf{u}_S\|_1 \leq 2\|\mathbf{u}_S\|_1$. Substituting it into (A.1.3) gives rise to

$$\begin{aligned} |\mathbf{u}^\top (\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}) \mathbf{u}| &\leq (2\|\mathbf{u}_S\|_1)^2 \|\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}\|_{\infty, \infty} \leq 4s^* \|\mathbf{u}_S\|_2^2 \cdot \|\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}\|_{\infty, \infty} \\ &\leq 4s^* \|\mathbf{u}\|_2^2 \cdot \|\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}\|_{\infty, \infty}, \end{aligned}$$

where the second inequality follows from Cauchy-Schwartz inequality. By (A.2.10) with probability at least $1 - 2/d$ we have

$$|\mathbf{u}^\top (\widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)}) \mathbf{u}| \leq \frac{4s^* C_4 K^2 \log d}{rn} \|\mathbf{u}\|_2^2.$$

Applying this bound on (A.1.2) gives that

$$\mathbf{u}^\top \widehat{\Sigma}^{(l)} \mathbf{u} \geq \left(\frac{\lambda_{\min}(\Sigma^*)}{8} - \frac{4s^* C_4 K^2 \log d}{rn} \right) \|\mathbf{u}\|_2^2$$

with probability at least $1 - C_1 \exp(-C_2 n) - 2/d$. If we set $n > s^* r^{-1} \max\{64C_4, C_1\} K^3 \log d$, it yields that $\mathbf{u}^\top \widehat{\Sigma}^{(l)} \mathbf{u} \geq \lambda_{\min}(\Sigma^*)/16$, i.e., $\widehat{\Sigma}^{(l)}$ satisfies the RE condition with parameters $(s^*, 1, \lambda_{\min}(\Sigma^*)/16)$. Applying union bound over $l \in \{1, 2, \dots, d\}$, the probability of the RE condition to be satisfied by all $\widehat{\Sigma}^{(l)}$'s is $1 - mC_1 \exp(-C_2 n) - 2m/d$. This completes the proof. \blacksquare

A.1.2 Proof of Theorem 2.4.6

Proof From the definition of $\text{HT}(\cdot, t)$ we have that $\forall \mathbf{u} \in \mathbb{R}^d, \forall j \in \{1, 2, \dots, d\}, |(\text{HT}(\mathbf{u}, t))_j - u_j| \leq t$. Hence $\|\text{HT}(\mathbf{u}, t) - \mathbf{u}\|_\infty \leq t$. By triangle inequality, we have

$$\begin{aligned} \|\bar{\beta} - \beta^*\|_\infty &\leq \left\| \bar{\beta} - \frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)} \right\|_\infty + \left\| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)} - \beta^* \right\|_\infty \\ &\leq t + \left\| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)} - \beta^* \right\|_\infty \\ &\leq 2t = 2C' M \sqrt{\frac{\log d}{N}} \|\beta^*\|_1 + 2C'' \max(s, s') M \frac{m \log d}{N} \|\beta^*\|_1, \end{aligned} \quad (\text{A.1.4})$$

where the third inequality follows from Lemma A.1.1. Now we consider the error bound of the ℓ_2 and ℓ_1 norm. If the event $\mathcal{E} := \{\|(\sum_{l=1}^m \tilde{\beta}^{(l)})/m - \beta^*\|_\infty \leq t\}$ happens, for any $j \in S^c$, we have

$$\left| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}_j^{(l)} \right| = \left| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}_j^{(l)} - \beta_j^* \right| \leq t,$$

where the first equality follows from the fact that $\beta_j^* = 0$. By the truncation rule, we have $\bar{\beta}_j = 0$. Hence $\text{supp}(\bar{\beta}) \subseteq S$. Therefore, under event \mathcal{E} , we have

$$\begin{aligned}\|\bar{\beta} - \beta^*\|_2 &= \|(\bar{\beta} - \beta^*)_S\|_2 \leq \sqrt{s} \|(\bar{\beta} - \beta^*)_S\|_\infty \\ &\leq \sqrt{s} \|\bar{\beta} - \beta^*\|_\infty \leq 2\sqrt{st} \\ &= 2\sqrt{s}C'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + 2\sqrt{s}C''\max(s, s')M\frac{m \log d}{N}\|\beta^*\|_1,\end{aligned}$$

where the second inequality follows from Cauchy-Schwartz inequality and the fourth inequality follows from (A.1.4). Similarly, we have

$$\begin{aligned}\|\bar{\beta} - \beta^*\|_1 &= \|(\bar{\beta} - \beta^*)_S\|_1 \leq s \|(\bar{\beta} - \beta^*)_S\|_\infty \leq 2st \\ &\leq 2sC'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + 2sC''\max(s, s')M\frac{m \log d}{N}\|\beta^*\|_1.\end{aligned}$$

This completes the proof. ■

A.1.3 Proof of Corollary 2.4.8

Proof Substituting (2.4.5) into (2.4.2), we obtain with probability at least $1 - 18m/d - 4/d$ that

$$\begin{aligned}\|\bar{\beta} - \beta^*\|_\infty &\leq 2C'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + 2C''\max(s, s')M\left(\frac{C'''}{\max(s, s')}\sqrt{\frac{N}{\log d}}\right)\frac{\log d}{N}\|\beta^*\|_1 \\ &= CM\sqrt{\frac{\log d}{N}}\|\beta^*\|_1.\end{aligned}$$

Substituting (2.4.5) into (2.4.3), we obtain with probability at least $1 - 18m/d - 4/d$ that

$$\begin{aligned}\|\bar{\beta} - \beta^*\|_2 &\leq 2\sqrt{s}C'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + 2\sqrt{s}C''\max(s, s')M\left(\frac{C'''}{\max(s, s')}\sqrt{\frac{N}{\log d}}\right)\frac{\log d}{N}\|\beta^*\|_1 \\ &= \sqrt{s}CM\sqrt{\frac{\log d}{N}}\|\beta^*\|_1,\end{aligned}$$

Substituting (2.4.5) into (2.4.4), we obtain with probability at least $1 - 18m/d - 4/d$ that

$$\begin{aligned}\|\bar{\beta} - \beta^*\|_1 &\leq 2sC'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + 2sC''\max(s, s')M\left(\frac{C'''}{\max(s, s')}\sqrt{\frac{N}{\log d}}\right)\frac{\log d}{N}\|\beta^*\|_1 \\ &= sCM\sqrt{\frac{\log d}{N}}\|\beta^*\|_1.\end{aligned}$$

This completes the proof. ■

A.1.4 Proof of Theorem 2.4.10

Proof We define t as in (2.4.1), and the event \mathcal{E} is defined as

$$\mathcal{E} := \left\{ \left\| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)} - \beta^* \right\|_{\infty} \leq t \right\}.$$

This event is the event that the conclusion of Lemma A.1.1 holds. From Lemma A.1.1 we know that \mathcal{E} happens with probability at least $1 - 18m/d - 4/d$. Under the condition that $\beta_{\min} > 2t$, we have that

1. if $\beta_j^* > 0$, which means that $\beta_j^* > 2t$, it holds that $\bar{\beta}_j \geq \beta_j^* - |\bar{\beta}_j - \beta_j^*| > 2t - 2t = 0$.
2. if $\beta_j^* < 0$, which means that $\beta_j^* < -2t$, it holds that $\bar{\beta}_j \leq \beta_j^* + |\bar{\beta}_j - \beta_j^*| < -2t + 2t = 0$.
3. if $\beta_j^* = 0$, because event \mathcal{E} happens we have $|(\sum_{l=1}^m \tilde{\beta}_j^{(l)})/m| \leq t$. By the definition of the hard thresholding operator $\text{HT}(\cdot, t)$, we have $\bar{\beta}_j = 0$.

Conclusively we have $\text{sign}(\beta_j^*) = \text{sign}(\bar{\beta}_j)$ for all $j \in \{1, 2, \dots, d\}$ if the event \mathcal{E} happens, which has a probability at least $1 - 18m/d - 4/d$. ■

A.1.5 Proof of Corollary 2.4.11

Proof Substituting (2.4.7) into (2.4.6), we obtain that the condition of β_{\min} in Theorem 2.4.10 can be rewritten as

$$\begin{aligned}\beta_{\min} &> 2C'M\sqrt{\frac{\log d}{N}}\|\beta^*\|_1 + 2C''\max(s, s')M\left(\frac{C_1}{\max(s, s')}\sqrt{\frac{N}{\log d}}\right)\frac{\log d}{N}\|\beta^*\|_1 \\ &= CM\sqrt{\frac{\log d}{N}}\|\beta^*\|_1.\end{aligned}$$

It is obvious that our assumption on β_{\min} satisfies this condition. Therefore by Theorem 2.4.10 we have with probability at least $1 - 18m/d - 4/d$ that $\text{sign}(\beta_j^*) = \text{sign}(\bar{\beta}_j)$ for all $j \in \{1, 2, \dots, d\}$. \blacksquare

A.2 Proof of Lemmas in Appendix A.1

First, we lay out some lemmas which are crucial to the proof of Lemma A.1.1.

Lemma A.2.1 *Let $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n \in \mathbb{R}^d$ be i.i.d. random vectors following normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. And the sample mean $\bar{\mathbf{X}} = (\sum_{i=1}^n \mathbf{X}_i)/n$. Then the difference between $\bar{\mathbf{X}}$ and $\boldsymbol{\mu}$ can be bounded by*

$$\|\bar{\mathbf{X}} - \boldsymbol{\mu}\|_{\infty} \leq CK_{\mathbf{X}}\sqrt{\frac{\log d}{n}}$$

with probability at least $1 - 1/d$, where C is an absolute constant, $K_{\mathbf{X}} = \|\mathbf{X}_1\|_{\psi_2}$.

Lemma A.2.2 *Let $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n$ be i.i.d. random vectors following multivariate normal distribution with zero mean and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$, then with probability at least $1 - 2/d$, the following inequality holds:*

$$\left\| \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^{\top} - \boldsymbol{\Sigma} \right\|_{\infty, \infty} \leq CK_{\mathbf{X}}^2 \sqrt{\frac{\log d}{n}},$$

where C is an absolute constant, and $K_{\mathbf{X}} = \|\mathbf{X}_1\|_{\psi_2}$.

Lemma A.2.3 *Under Assumption 2.4.1, for any $l \in \{1, 2, \dots, m\}$, we have with probability at least $1 - 4/d$ that*

$$\left\| \widehat{\Sigma}^{(l)} - \Sigma^* \right\|_{\infty, \infty} \leq C_2 K^2 \sqrt{\frac{\log d}{n}}, \quad (\text{A.2.1})$$

where C_1 and C_2 are absolute constants.

Lemma A.2.4 *Under Assumption 2.4.1 and Condition 2.4.4, if we set λ as*

$$\lambda \geq C K^2 \sqrt{\frac{\log d}{rn}} \|\beta^*\|_1, \quad (\text{A.2.2})$$

then we have with probability at least $1 - 6/d$ that

$$\|\widehat{\beta}^{(l)} - \beta^*\|_1 \leq \frac{128 \lambda s}{\lambda_{\min}(\Sigma^*)}. \quad (\text{A.2.3})$$

Lemma A.2.5 *Under Assumption 2.4.1, 2.4.2 and Condition 2.4.4, if we set λ'*

$$\lambda' \geq C K^2 M \sqrt{\frac{\log d}{n}}, \quad (\text{A.2.4})$$

then for each l with probability at least $1 - 4/d$ we have

$$\|\widehat{\theta}_j^{(l)} - \theta^*\|_1 \leq \frac{128 \lambda' s'}{\lambda_{\min}(\Sigma^*)}$$

for all $j \in \{1, 2, \dots, d\}$.

Note that Lemma A.2.5 implies that for each machine,

$$\|\widehat{\Theta}^{(l)} - \Theta^*\|_1 \leq \frac{128 \lambda' s'}{\lambda_{\min}(\Sigma^*)}$$

holds with probability at least $1 - 4/d$.

A.2.1 Proof of Lemma A.1.1

Proof Substituting the definition of $\tilde{\beta}^{(l)}$, we have

$$\begin{aligned}
\left\| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)} - \beta^* \right\|_{\infty} &= \left\| \frac{1}{m} \sum_{l=1}^m \hat{\beta}^{(l)} - \beta^* - \hat{\Theta}^{(l)\top} (\hat{\Sigma}^{(l)} \hat{\beta}^{(l)} - \hat{\mu}_d^{(l)}) \right\|_{\infty} \\
&= \left\| \frac{1}{m} \sum_{l=1}^m (\mathbf{I} - \hat{\Theta}^{(l)\top} \hat{\Sigma}^{(l)}) (\hat{\beta}^{(l)} - \beta^*) - \hat{\Theta}^{(l)\top} \hat{\Sigma}^{(l)} \beta^* + \hat{\Theta}^{(l)\top} \hat{\mu}_d^{(l)} \right\|_{\infty} \\
&= \left\| \frac{1}{m} \sum_{l=1}^m (\mathbf{I} - \hat{\Theta}^{(l)\top} \hat{\Sigma}^{(l)}) (\hat{\beta}^{(l)} - \beta^*) + \hat{\Theta}^{(l)\top} (\hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^*) \right\|_{\infty}.
\end{aligned} \tag{A.2.5}$$

Using triangle inequality, we can split (A.2.5) into two terms:

$$\left\| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)} - \beta^* \right\|_{\infty} \leq \underbrace{\left\| \frac{1}{m} \sum_{l=1}^m (\mathbf{I} - \hat{\Theta}^{(l)\top} \hat{\Sigma}^{(l)}) (\hat{\beta}^{(l)} - \beta^*) \right\|_{\infty}}_{I_1} + \underbrace{\left\| \frac{1}{m} \sum_{l=1}^m \hat{\Theta}^{(l)\top} (\hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^*) \right\|_{\infty}}_{I_2}.$$

Firstly we give an upper bound of I_1 . By triangle inequality we have

$$\begin{aligned}
I_1 &\leq \frac{1}{m} \sum_{l=1}^m \|(\mathbf{I} - \hat{\Theta}^{(l)\top} \hat{\Sigma}^{(l)}) (\hat{\beta}^{(l)} - \beta^*)\|_{\infty} \\
&\leq \frac{1}{m} \sum_{l=1}^m \|\mathbf{I} - \hat{\Theta}^{(l)\top} \hat{\Sigma}^{(l)}\|_{\infty, \infty} \cdot \|\hat{\beta}^{(l)} - \beta^*\|_1.
\end{aligned}$$

By (2.3.2) and Lemma (A.2.4) we have with probability at least $1 - 6m/d$ that

$$I_1 \leq \frac{128\lambda'\lambda s}{\lambda_{\min}(\Sigma^*)} = \frac{128sC_1C_2K^4M \log d}{\lambda_{\min}(\Sigma^*)rn} \|\beta^*\|_1 \leq \frac{sC_3M \log d}{n} \|\beta^*\|_1,$$

where C_3 is a constant. Now we take a closer look at I_2 . This term can be further rewritten as

$$\begin{aligned}
I_2 &= \left\| \frac{1}{m} \sum_{l=1}^m \Theta^{*\top} (\hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^*) + (\hat{\Theta}^{(l)} - \hat{\Theta}^*)^{\top} (\hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^*) \right\|_{\infty} \\
&\leq \underbrace{\left\| \frac{1}{m} \sum_{l=1}^m \Theta^{*\top} (\hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^*) \right\|_{\infty}}_{I_3} + \underbrace{\left\| \frac{1}{m} \sum_{l=1}^m (\hat{\Theta}^{(l)} - \hat{\Theta}^*)^{\top} (\hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^*) \right\|_{\infty}}_{I_4}.
\end{aligned}$$

Firstly we bound I_4 . By triangle inequality we have

$$I_4 \leq \frac{1}{m} \sum_{l=1}^m \left\| (\hat{\Theta}^{(l)} - \hat{\Theta}^*)^\top (\hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^*) \right\|_\infty \leq \frac{1}{m} \sum_{l=1}^m \left\| (\hat{\Theta}^{(l)} - \hat{\Theta}^*)^\top \right\|_\infty \cdot \left\| \hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^* \right\|_\infty.$$

In the proof of Lemma A.2.4, we get the conclusion that $\left\| \hat{\mu}_d^{(l)} - \hat{\Sigma}^{(l)} \beta^* \right\|_\infty \leq \lambda$ with probability at least $1 - 6/d$. Moreover, using Lemma A.2.5 and union bound, we have with probability at least $1 - 10m/d$ that

$$I_4 \leq \frac{128\lambda\lambda's'}{\lambda_{\min}(\Sigma^*)} = \frac{128s'C_1C_2K^4M \log d}{\lambda_{\min}(\Sigma^*)rn} \|\beta^*\|_1 \leq \frac{s'C_4M \log d}{n} \|\beta^*\|_1,$$

where C_4 is a constant. Finally, we give a bound on I_3 . This term can be rewritten as follows:

$$\begin{aligned} I_3 &= \left\| \frac{1}{m} \sum_{l=1}^m \Theta^{*\top} (\hat{\mu}_d^{(l)} - \mu_d + \Sigma^* \beta^* - \hat{\Sigma}^{(l)} \beta^*) \right\|_\infty \\ &\leq \left\| \frac{1}{m} \sum_{l=1}^m \Theta^{*\top} (\hat{\mu}_d^{(l)} - \mu_d) \right\|_\infty + \left\| \frac{1}{m} \sum_{l=1}^m \Theta^{*\top} (\Sigma^* - \hat{\Sigma}^{(l)}) \beta^* \right\|_\infty \\ &\leq \underbrace{\left\| \frac{1}{m} \sum_{l=1}^m \Theta^{*\top} (\hat{\mu}_d^{(l)} - \mu_d) \right\|_\infty}_{I_5} + \underbrace{\left\| \frac{1}{m} \sum_{l=1}^m \Theta^{*\top} (\Sigma^* - \tilde{\Sigma}^{(l)}) \beta^* \right\|_\infty}_{I_6} + \underbrace{\frac{1}{m} \sum_{l=1}^m \left\| \Theta^{*\top} (\tilde{\Sigma}^{(l)} - \hat{\Sigma}^{(l)}) \beta^* \right\|_\infty}_{I_7}, \end{aligned}$$

where the first equality uses the fact that $\mu_d = \Sigma^* \beta^*$. Furthermore, I_7 can be bounded by

$$\begin{aligned} I_7 &\leq \frac{1}{m} \sum_{l=1}^m \left\| \Theta^{*\top} \right\|_\infty \cdot \left\| (\tilde{\Sigma}^{(l)} - \hat{\Sigma}^{(l)}) \beta^* \right\|_\infty \\ &\leq \frac{1}{m} \sum_{l=1}^m \left\| \Theta^* \right\|_1 \cdot \left\| \tilde{\Sigma}^{(l)} - \hat{\Sigma}^{(l)} \right\|_{\infty, \infty} \cdot \left\| \beta^* \right\|_1. \end{aligned}$$

By Lemma A.2.3, we have with probability at least $1 - 2m/d$ that

$$I_7 \leq \frac{1}{m} \sum_{l=1}^m M \frac{C_5 \log d}{n} = \frac{C_5 M \log d}{n},$$

where the first inequality follows from the fact that $\|\Theta^*\|_1 \leq M$. In terms of I_5 , we have

$$\begin{aligned} I_5 &= \left\| \Theta^{*\top} \left[\left(\frac{1}{m} \sum_{l=1}^m \hat{\mu}_1^{(l)} - \mu_1 \right) - \left(\frac{1}{m} \sum_{l=1}^m \hat{\mu}_2^{(l)} - \mu_2 \right) \right] \right\|_\infty \\ &\leq \|\Theta^{*\top}\|_\infty \cdot (\|\hat{\mu}_1 - \mu_1\|_\infty + \|\hat{\mu}_2 - \mu_2\|_\infty), \end{aligned}$$

where $\hat{\mu}_1 = (\sum_{l=1}^m \hat{\mu}_1^{(l)})/m$, $\hat{\mu}_2 = (\sum_{l=1}^m \hat{\mu}_2^{(l)})/m$. By Lemma A.2.1, we have with probability at least $1 - 1/d$ that

$$\|\hat{\mu}_1 - \mu_1\|_\infty \leq C_6 K \sqrt{\frac{\log d}{mn_1}}.$$

Similarly, we have with probability at least $1 - 1/d$ that

$$\|\hat{\mu}_2 - \mu_2\|_\infty \leq C_6 K \sqrt{\frac{\log d}{mn_2}}.$$

Therefore we have with probability at least $1 - 2/d$ that

$$I_5 \leq C_6 M K \left(\sqrt{\frac{\log d}{mn_1}} + \sqrt{\frac{\log d}{mn_2}} \right).$$

In terms of I_6 , we can apply similar procedure. Let us denote $\tilde{\Sigma} = (\sum_{l=1}^m \tilde{\Sigma}^{(l)})/m$. By Lemma A.2.2 we have with probability at least $1 - 2/d$ that

$$\|\tilde{\Sigma} - \Sigma^*\|_{\infty, \infty} \leq C_7 K^2 \sqrt{\frac{\log d}{N}}.$$

Therefore I_6 can be bounded by

$$\begin{aligned} I_6 &= \|\Theta^{*\top} (\Sigma^* - \tilde{\Sigma}) \beta^*\|_\infty \leq \|\Theta^{*\top}\|_\infty \cdot \|(\Sigma^* - \tilde{\Sigma}) \beta^*\|_\infty \leq M \cdot \|\Sigma^* - \tilde{\Sigma}\|_{\infty, \infty} \cdot \|\beta^*\|_1 \\ &\leq C_7 M K^2 \sqrt{\frac{\log d}{N}} \|\beta^*\|_1 \end{aligned}$$

with probability at least $1 - 2/d$. Combining the bound of I_1, I_4, I_5, I_6 and I_7 , we get that

$$\begin{aligned}
\left\| \frac{1}{m} \sum_{l=1}^m \tilde{\beta}^{(l)} - \beta^* \right\|_{\infty} &\leq \frac{sC_3M \log d}{n} \|\beta^*\|_1 + \frac{s'C_4M \log d}{n} \|\beta^*\|_1 + \frac{C_5MK^2 \log d}{n} + \\
&\quad C_6MK \left(\sqrt{\frac{\log d}{mn_1}} + \sqrt{\frac{\log d}{mn_2}} \right) + C_7MK^2 \sqrt{\frac{\log d}{N}} \|\beta^*\|_1 \\
&\leq C'M \sqrt{\frac{\log d}{N}} \|\beta^*\|_1 + C'' \max(s, s') M \frac{m \log d}{N} \|\beta^*\|_1
\end{aligned}$$

with probability at least $1 - 18m/d - 4/d$, where C' and C'' are constants. \blacksquare

A.2.2 Proof of Lemma A.1.2

Proof By the definition of $\hat{\Sigma}^{(l)}$ and $\tilde{\Sigma}^{(l)}$, we have

$$\begin{aligned}
\|\hat{\Sigma}^{(l)} - \tilde{\Sigma}^{(l)}\|_{\infty, \infty} &= \frac{1}{n} \left\| \sum_{i=1}^{n_1} \left((\mathbf{X}_i^{(l)} - \hat{\mu}_1^{(l)})(\mathbf{X}_i^{(l)} - \hat{\mu}_1^{(l)})^\top - (\mathbf{X}_i^{(l)} - \mu_1)(\mathbf{X}_i^{(l)} - \mu_1)^\top \right) \right. \\
&\quad \left. + \sum_{i=1}^{n_2} \left((\mathbf{Y}_i^{(l)} - \hat{\mu}_2^{(l)})(\mathbf{Y}_i^{(l)} - \hat{\mu}_2^{(l)})^\top - (\mathbf{Y}_i^{(l)} - \mu_2)(\mathbf{Y}_i^{(l)} - \mu_2)^\top \right) \right\|_{\infty, \infty}.
\end{aligned} \tag{A.2.6}$$

Note that

$$\begin{aligned}
&\sum_{i=1}^{n_1} \left((\mathbf{X}_i^{(l)} - \hat{\mu}_1^{(l)})(\mathbf{X}_i^{(l)} - \hat{\mu}_1^{(l)})^\top - (\mathbf{X}_i^{(l)} - \mu_1)(\mathbf{X}_i^{(l)} - \mu_1)^\top \right) \\
&= \sum_{i=1}^{n_1} \mathbf{X}_i^{(l)} \mathbf{X}_i^{(l)\top} - n_1 \hat{\mu}_1^{(l)} \hat{\mu}_1^{(l)\top} - n_1 \hat{\mu}_1^{(l)} \hat{\mu}_1^{(l)\top} + n_1 \hat{\mu}_1^{(l)} \hat{\mu}_1^{(l)\top} \\
&\quad - \sum_{i=1}^{n_1} \mathbf{X}_i^{(l)} \mathbf{X}_i^{(l)\top} + n_1 \hat{\mu}_1^{(l)} \mu_1^\top + n_1 \mu_1 \hat{\mu}_1^{(l)\top} - n_1 \mu_1 \mu_1^\top \\
&= -n_1 \hat{\mu}_1^{(l)} \hat{\mu}_1^{(l)\top} + n_1 \hat{\mu}_1^{(l)} \mu_1^\top + n_1 \mu_1 \hat{\mu}_1^{(l)\top} - n_1 \mu_1 \mu_1^\top \\
&= -n_1 (\hat{\mu}_1^{(l)} - \mu_1)(\hat{\mu}_1^{(l)} - \mu_1)^\top.
\end{aligned} \tag{A.2.7}$$

Similarly, we have

$$\sum_{i=1}^{n_2} \left((\mathbf{Y}_i^{(l)} - \hat{\boldsymbol{\mu}}_2^{(l)}) (\mathbf{Y}_i^{(l)} - \hat{\boldsymbol{\mu}}_2^{(l)})^\top - (\mathbf{Y}_i^{(l)} - \boldsymbol{\mu}_2) (\mathbf{Y}_i^{(l)} - \boldsymbol{\mu}_2)^\top \right) = -n_2 (\hat{\boldsymbol{\mu}}_2^{(l)} - \boldsymbol{\mu}_2) (\hat{\boldsymbol{\mu}}_2^{(l)} - \boldsymbol{\mu}_2)^\top. \quad (\text{A.2.8})$$

Substituting (A.2.7) and (A.2.8) into (A.2.6) and using the triangle inequality, we can obtain

$$\begin{aligned} \|\hat{\boldsymbol{\Sigma}}^{(l)} - \tilde{\boldsymbol{\Sigma}}^{(l)}\|_{\infty, \infty} &\leq \frac{n_1}{n} \|(\hat{\boldsymbol{\mu}}_1^{(l)} - \boldsymbol{\mu}_1)(\hat{\boldsymbol{\mu}}_1^{(l)} - \boldsymbol{\mu}_1)^\top\|_{\infty, \infty} + \frac{n_2}{n} \|(\hat{\boldsymbol{\mu}}_2^{(l)} - \boldsymbol{\mu}_2)(\hat{\boldsymbol{\mu}}_2^{(l)} - \boldsymbol{\mu}_2)^\top\|_{\infty, \infty} \\ &= \frac{n_1}{n} \|\hat{\boldsymbol{\mu}}_1^{(l)} - \boldsymbol{\mu}_1\|_\infty^2 + \frac{n_2}{n} \|\hat{\boldsymbol{\mu}}_2^{(l)} - \boldsymbol{\mu}_2\|_\infty^2. \end{aligned} \quad (\text{A.2.9})$$

By Lemma A.2.1, we know that $\|\hat{\boldsymbol{\mu}}_1^{(l)} - \boldsymbol{\mu}_1\|_\infty \leq C' K \sqrt{\log d / n_1}$ with probability at least $1 - 1/d$ and $\|\hat{\boldsymbol{\mu}}_2^{(l)} - \boldsymbol{\mu}_2\|_\infty \leq C' K \sqrt{\log d / n_2}$ with probability at least $1 - 1/d$. Submitting the two high probability bounds into (A.2.9) and using the union bound gives rise to

$$\|\hat{\boldsymbol{\Sigma}}^{(l)} - \tilde{\boldsymbol{\Sigma}}^{(l)}\|_{\infty, \infty} \leq \frac{C'^2 K^2 \log d}{\min(n_1, n_2)} \leq \frac{C'^2 K^2 \log d}{rn}. \quad (\text{A.2.10})$$

This inequality holds with probability at least $1 - 2/d$. Setting $C = C'^2$ completes the proof. \blacksquare

A.3 Proof of Lemmas in Appendix A.2

First of all, we present some lemmas which are crucial to the proof of lemmas in this section.

Lemma A.3.1 *For the l -th machine, if the underlying true parameter $\boldsymbol{\beta}^*$ lies in the feasible set of the optimization problem (2.3.1), then the biased estimator $\hat{\boldsymbol{\beta}}^{(l)}$ lies in the set $\{\boldsymbol{\beta} : \|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{S^c}\|_1 \leq \|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_S\|_1\}$, where $S = \text{supp}(\boldsymbol{\beta}^*)$.*

Lemma A.3.2 *For the l -th machine, if the underlying true parameter $\boldsymbol{\theta}_j^*$ lies in the feasible set of the optimization problem (2.3.3), then the optimal solution $\hat{\boldsymbol{\theta}}_j^{(l)}$ lies in the set $\{\boldsymbol{\theta} : \|(\boldsymbol{\theta} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}^c}\|_1 \leq \|(\boldsymbol{\theta} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1\}$, where $S_{\theta_j} = \text{supp}(\boldsymbol{\theta}_j^*)$.*

A.3.1 Proof of Lemma A.2.1

Proof For the j -th component, by Theorem A.5.4, we have

$$\mathbb{P}(|\bar{X}_j - \mu_j| > t) \leq \exp\left(-\frac{C_1 n t^2}{K_{\mathbf{X}}^2}\right)$$

for any $t > 0$, where $C_1 > 0$ is an absolute constant. Using the union bound, we have

$$\mathbb{P}(\|\bar{\mathbf{X}} - \boldsymbol{\mu}\|_{\infty} > t) \leq d \exp\left(-\frac{C_1 n t^2}{K_{\mathbf{X}}^2}\right).$$

Taking $t = K_{\mathbf{X}} \sqrt{2 \log d / C_1 n}$ gives that with probability at least $1 - 1/d$ the following inequality holds:

$$\|\bar{\mathbf{X}} - \boldsymbol{\mu}\|_{\infty} \leq K_{\mathbf{X}} \sqrt{\frac{2 \log d}{C_1 n}}.$$

Setting $C = \sqrt{2/C_1}$ yields the conclusion of this lemma. ■

A.3.2 Proof of Lemma A.2.2

Proof Denote the j -th entry of \mathbf{X}_i as x_{ij} . We have $(\mathbf{X}_i \mathbf{X}_i^{\top})_{jk} = x_{ij} x_{ik}$ following sub-Exponential distribution. Because $\mathbb{E}(x_{ij} x_{ik}) = \Sigma_{jk}$ for all i , we know that $x_{ij} x_{ik} - \Sigma_{jk}$ is a centered sub-Exponential random variable. The ψ_1 norm of $x_{ij} x_{ik}$ can be bounded using Lemma A.5.6 as $\|x_{ij} x_{ik}\|_{\psi_1} \leq C_1 \max\{\|x_{ij}\|_{\psi_2}^2, \|x_{ik}\|_{\psi_2}^2\} \leq C_1 K_{\mathbf{X}}^2$. By Theorem A.5.5, for any $t > 0$ we have

$$\mathbb{P}\left(\left|\frac{1}{n} \sum_{i=1}^n x_{ij} x_{ik} - \Sigma_{jk}\right| \geq t\right) \leq 2 \exp\left[-C_2 \min\left(\frac{t^2 n}{C_1^2 K_{\mathbf{X}}^4}, \frac{t n}{C_1 K_{\mathbf{X}}^2}\right)\right].$$

Using the union bound, we get the conclusion that

$$\mathbb{P}\left(\left\|\frac{1}{n} \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^{\top} - \boldsymbol{\Sigma}\right\|_{\infty, \infty} \geq t\right) \leq 2d^2 \exp\left[-C_2 \min\left(\frac{t^2 n}{C_1^2 K_{\mathbf{X}}^4}, \frac{t n}{C_1 K_{\mathbf{X}}^2}\right)\right].$$

Setting $2d^2 \exp[-C_2 t^2 n / (C_1^2 K_{\mathbf{X}}^4)] = \delta$, we get

$$t = C_1 K_{\mathbf{X}}^2 \sqrt{\frac{\log(2d^2/\delta)}{C_2 n}}.$$

Setting $\delta = 2/d$ and $C = \sqrt{3}C_1/\sqrt{C_2}$, we get the conclusion that with probability at least $1 - 2/d$, the following inequality holds:

$$\left\| \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^\top - \Sigma \right\|_{\infty, \infty} \leq C K_{\mathbf{X}}^2 \sqrt{\frac{\log d}{n}}.$$

■

A.3.3 Proof of Lemma A.2.3

Proof Following [36], using triangle inequality, we have

$$\left\| \widehat{\Sigma}^{(l)} - \Sigma^* \right\|_{\infty, \infty} \leq \left\| \widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)} \right\|_{\infty, \infty} + \left\| \widetilde{\Sigma}^{(l)} - \Sigma^* \right\|_{\infty, \infty}.$$

Now we bound the first term. In Lemma A.1.2 we have got that with probability at least $1 - 2/d$, the first term is bounded by

$$\left\| \widehat{\Sigma}^{(l)} - \widetilde{\Sigma}^{(l)} \right\|_{\infty, \infty} \leq \frac{C' K^2 \log d}{rn}.$$

For the second term, note that in each machine, $\mathbf{X}_i^{(l)} - \boldsymbol{\mu}_1$'s and $\mathbf{Y}_i^{(l)} - \boldsymbol{\mu}_2$'s are i.i.d. random vectors following normal distribution with zero mean and covariance matrix Σ^* . Hence by Lemma A.2.2 we have with probability at least $1 - 2/d$ that

$$\left\| \widetilde{\Sigma}^{(l)} - \Sigma^* \right\|_{\infty, \infty} \leq C'' K^2 \sqrt{\frac{\log d}{n}}.$$

Combining the two high probability bounds together, we have with probability at least $1 - 4/d$ that

$$\left\| \widehat{\Sigma}^{(l)} - \Sigma^* \right\|_{\infty, \infty} \leq C'' K^2 \sqrt{\frac{\log d}{n}} + \frac{C' K^2 \log d}{rn} \leq 2C'' K^2 \sqrt{\frac{\log d}{n}}.$$

Setting $C_1 = C'^4/C''^2$ and $C_2 = 2C''$ completes the proof. \blacksquare

A.3.4 Proof of Lemma A.2.4

Proof First we will show that with high probability the true parameter $\beta^* = \Theta^* \mu_d$ satisfies the constraint in (2.3.1), i.e., with high probability the inequality $\|\widehat{\Sigma} \beta^* - \widehat{\mu}_d^{(l)}\|_\infty < \lambda$ holds. To show this, we consider

$$\begin{aligned} \|\widehat{\Sigma}^{(l)} \beta^* - \widehat{\mu}_d^{(l)}\|_\infty &= \|\Sigma^* \beta^* - \Sigma^* \beta^* + \widehat{\Sigma}^{(l)} \beta^* - \mu_d + \mu_d - \widehat{\mu}_d^{(l)}\|_\infty \\ &\leq \|\Sigma^* \beta^* - \mu_d\|_\infty + \|(\widehat{\Sigma}^{(l)} - \Sigma^*) \beta^*\|_\infty + \|\mu_1 - \mu_2 - \widehat{\mu}_1^{(l)} + \widehat{\mu}_2^{(l)}\|_\infty \\ &\leq \|\Sigma^* \beta^* - \mu_d\|_\infty + \|\widehat{\Sigma}^{(l)} - \Sigma^*\|_{\infty, \infty} \cdot \|\beta^*\|_1 + \|\widehat{\mu}_1^{(l)} - \mu_1\|_\infty + \|\widehat{\mu}_2^{(l)} - \mu_2\|_\infty, \end{aligned} \quad (\text{A.3.1})$$

where the second inequality follows from triangle inequality and the definition of μ_d and $\widehat{\mu}_d^{(l)}$, and the third inequality follows from Hölder's inequality and triangle inequality. Note that by the definition of β^* we have $\Sigma^* \beta^* - \mu_d = \mathbf{0}$. For other terms, by Lemma A.2.3 we have with probability at least $1 - 4/d$ that $\|\widehat{\Sigma}^{(l)} - \Sigma^*\|_{\infty, \infty} \leq C_1 K^2 \sqrt{\log d/n}$. Additionally Lemma A.2.1 gives that $\|\widehat{\mu}_1^{(l)} - \mu_1\|_\infty \leq C_2 K \sqrt{\log d/n_1}$ and $\|\widehat{\mu}_2^{(l)} - \mu_2\|_\infty \leq C_2 K \sqrt{\log d/n_2}$. Substituting the three high probability bounds into (A.3.1), we have with probability at least $1 - 6/d$ that

$$\|\widehat{\Sigma}^{(l)} \beta^* - \widehat{\mu}_d^{(l)}\|_\infty \leq C_1 K^2 \sqrt{\frac{\log d}{n}} \|\beta^*\|_1 + C_2 K \sqrt{\frac{\log d}{\min(n_1, n_2)}} \leq C K^2 \sqrt{\frac{\log d}{rn}} \|\beta^*\|_1.$$

This means that if λ satisfies (A.2.2), β^* will lie in the feasible set of (2.3.1) with probability at least $1 - 6/d$. Applying Lemma A.3.1 gives $\|(\widehat{\beta}^{(l)} - \beta^*)_{S^c}\|_1 \leq \|(\widehat{\beta}^{(l)} - \beta^*)_S\|_1$. By Condition 2.4.4 we have

$$(\widehat{\beta}^{(l)} - \beta^*)^\top \widehat{\Sigma}^{(l)} (\widehat{\beta}^{(l)} - \beta^*) \geq \frac{\lambda_{\min}(\Sigma^*)}{16} \|\widehat{\beta}^{(l)} - \beta^*\|_2^2 \geq \frac{\lambda_{\min}(\Sigma^*)}{16} \|(\widehat{\beta}^{(l)} - \beta^*)_S\|_2^2. \quad (\text{A.3.2})$$

Additionally, we have

$$\|\widehat{\Sigma}^{(l)} (\widehat{\beta}^{(l)} - \beta^*)\|_\infty \leq \|\widehat{\Sigma}^{(l)} \widehat{\beta}^{(l)} - \widehat{\mu}_d^{(l)}\|_\infty + \|\widehat{\Sigma}^{(l)} \beta^* - \widehat{\mu}_d^{(l)}\|_\infty \leq 2\lambda, \quad (\text{A.3.3})$$

where the second inequality follows from the fact that both $\hat{\beta}^{(l)}$ and β^* are feasible solutions of optimization problem (2.3.1). Therefore we have

$$\begin{aligned} (\hat{\beta}^{(l)} - \beta^*)^\top \hat{\Sigma}^{(l)} (\hat{\beta}^{(l)} - \beta^*) &\leq \|\hat{\Sigma}^{(l)} (\hat{\beta}^{(l)} - \beta^*)\|_\infty \cdot \|\hat{\beta}^{(l)} - \beta^*\|_1 \leq 2\lambda \|\hat{\beta}^{(l)} - \beta^*\|_1 \\ &\leq 4\lambda \|(\hat{\beta}^{(l)} - \beta^*)_S\|_1 \\ &\leq 4\lambda \sqrt{s} \|(\hat{\beta}^{(l)} - \beta^*)_S\|_2, \end{aligned} \quad (\text{A.3.4})$$

where the first inequality follows from Hölder's inequality, the second inequality follows from (A.3.3), the third follows from the fact that $\|(\hat{\beta}^{(l)} - \beta^*)_{S^c}\|_1 \leq \|(\hat{\beta}^{(l)} - \beta^*)_S\|_1$ and the last follows from Cauchy-Schwartz inequality. Combining (A.3.4) and (A.3.2) gives that

$$\|(\hat{\beta}^{(l)} - \beta^*)_S\|_2 \leq \frac{64\lambda\sqrt{s}}{\lambda_{\min}(\Sigma^*)}.$$

Based on this result we can provide the estimation error bound of $\hat{\beta}^{(l)}$ in terms of ℓ_1 norm:

$$\|\hat{\beta}^{(l)} - \beta^*\|_1 \leq 2\|(\hat{\beta}^{(l)} - \beta^*)_S\|_1 \leq 2\sqrt{s}\|(\hat{\beta}^{(l)} - \beta^*)_S\|_2 \leq \frac{128\lambda s}{\lambda_{\min}(\Sigma^*)}.$$

■

A.3.5 Proof of Lemma A.2.5

Proof First we will show that with high probability the true parameter θ_j^* satisfies the constraint in (2.3.3), i.e., with high probability the inequality $\|\hat{\Sigma}\theta_j^* - \mathbf{e}_j\|_\infty < \lambda'$ holds. To show this, we consider

$$\left\| \hat{\Sigma}^{(l)} \theta_j^* - \mathbf{e}_j \right\|_\infty = \left\| \hat{\Sigma}^{(l)} \theta_j^* - \Sigma^* \theta_j^* \right\|_\infty \leq \|\hat{\Sigma}^{(l)} - \Sigma^*\|_{\infty, \infty} \cdot \|\theta_j^*\|_1, \quad (\text{A.3.5})$$

where the first equality follows from the fact that $\Sigma^* \theta_j^* = \mathbf{e}_j$, and second inequality follows from Hölder's inequality. By Lemma A.2.3 we have with probability at least $1 - 4/d$ that $\|\hat{\Sigma}^{(l)} - \Sigma^*\|_{\infty, \infty} \leq CK^2 \sqrt{\log d/n}$. Assumption 2.4.2 indicates that $\|\theta_j^*\|_1 \leq M$ for all j . Substituting the two high probability bounds into (A.3.5), for

all $j \in \{1, 2, \dots, d\}$ we have with probability at least $1 - 4/d$ that

$$\|\widehat{\Sigma}^{(l)} \boldsymbol{\theta}_j^* - \mathbf{e}_j\|_\infty \leq CK^2 M \sqrt{\frac{\log d}{n}}.$$

This means that if λ' satisfies (A.2.4), $\boldsymbol{\theta}_j^*$ will lie in the feasible set of (2.3.1) with probability at least $1 - 4/d$. Applying Lemma A.3.2 gives $\|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}^c}\|_1 \leq \|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1$. By Condition 2.4.4 we have

$$(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)^\top \widehat{\Sigma}^{(l)} (\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*) \geq \frac{\lambda_{\min}(\Sigma^*)}{16} \|\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*\|_2^2 \geq \frac{\lambda_{\min}(\Sigma^*)}{16} \|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_2^2. \quad (\text{A.3.6})$$

Additionally, we have

$$\|\widehat{\Sigma}^{(l)} (\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)\|_\infty \leq \|\widehat{\Sigma}^{(l)} \widehat{\boldsymbol{\theta}}_j^{(l)} - \mathbf{e}_j\|_\infty + \|\widehat{\Sigma}^{(l)} \boldsymbol{\theta}_j^* - \mathbf{e}_j\|_\infty \leq 2\lambda', \quad (\text{A.3.7})$$

where the second inequality follows from the fact that both $\widehat{\boldsymbol{\theta}}_j^{(l)}$ and $\boldsymbol{\theta}_j^*$ are feasible solutions of optimization problem (2.3.3). Therefore we have

$$\begin{aligned} (\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)^\top \widehat{\Sigma}^{(l)} (\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*) &\leq \|\widehat{\Sigma}^{(l)} (\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)\|_\infty \cdot \|\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*\|_1 \leq 2\lambda' \|\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*\|_1 \\ &\leq 4\lambda' \|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1 \\ &\leq 4\lambda' \sqrt{s'} \|(\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_2, \end{aligned} \quad (\text{A.3.8})$$

where the first inequality follows from Hölder's inequality, the second inequality follows from (A.3.7), the third follows from the fact that $\|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}^c}\|_1 \leq \|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1$ and the last follows from Cauchy-Schwartz inequality. Combining (A.3.8) and (A.3.6) gives that

$$\|(\widehat{\boldsymbol{\theta}}_j - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_2 \leq \frac{64\lambda' \sqrt{s'}}{\lambda_{\min}(\Sigma^*)}.$$

Based on this result we can provide the estimation error bound of $\widehat{\boldsymbol{\theta}}_j^{(l)}$ in terms of ℓ_1 -norm:

$$\|\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*\|_1 \leq 2 \|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1 \leq 2\sqrt{s'} \|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_2 \leq \frac{128\lambda' s'}{\lambda_{\min}(\Sigma^*)}.$$

■

A.4 Proof of Auxilliary Lemmas in Appendix A.3

A.4.1 Proof of Lemma A.3.1

Proof In the optimization problem (2.3.1), under the condition that β^* is a feasible solution, the optimality of $\hat{\beta}^{(l)}$ yields

$$\|\beta_S^*\|_1 = \|\beta^*\|_1 \geq \|\hat{\beta}^{(l)}\|_1 = \|\hat{\beta}_S^{(l)}\|_1 + \|\hat{\beta}_{S^c}^{(l)}\|_1 = \|\hat{\beta}_S^{(l)}\|_1 + \|(\hat{\beta}^{(l)} - \beta^*)_{S^c}\|_1, \quad (\text{A.4.1})$$

where the last equality follows from the fact that $(\beta^*)_{S^c} = \mathbf{0}$. (A.4.1) immediately leads to

$$\|\beta_S^*\|_1 - \|\hat{\beta}_S^{(l)}\|_1 \geq \|(\hat{\beta}^{(l)} - \beta^*)_{S^c}\|_1.$$

Moreover, by triangle inequality, we have

$$\|(\hat{\beta}^{(l)} - \beta^*)_S\|_1 \geq \|\beta_S^*\|_1 - \|\hat{\beta}_S^{(l)}\|_1.$$

Combining the above two inequalities, we can obtain

$$\|(\hat{\beta}^{(l)} - \beta^*)_S\|_1 \geq \|(\hat{\beta}^{(l)} - \beta^*)_{S^c}\|_1.$$

This completes the proof. ■

A.4.2 Proof of Lemma A.3.2

Proof In the optimization problem (2.3.3), under the condition that θ_j^* is a feasible solution, the optimality of $\hat{\theta}_j^{(l)}$ yields

$$\|(\theta_j^*)_{S_{\theta_j}}\|_1 = \|\theta_j^*\|_1 \geq \|\hat{\theta}_j^{(l)}\|_1 = \|(\hat{\theta}_j^{(l)})_{S_{\theta_j}}\|_1 + \|(\hat{\theta}_j^{(l)})_{S_{\theta_j}^c}\|_1 = \|(\hat{\theta}_j^{(l)})_{S_{\theta_j}}\|_1 + \|(\hat{\theta}_j^{(l)} - \theta_j^*)_{S_{\theta_j}^c}\|_1, \quad (\text{A.4.2})$$

where the last equality follows from the fact that $(\boldsymbol{\theta}_j^*)_{S_{\theta_j}^c} = \mathbf{0}$. (A.4.2) immediately leads to

$$\|(\boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1 - \|(\widehat{\boldsymbol{\theta}}_j^{(l)})_{S_{\theta_j}}\|_1 \geq \|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}^c}\|_1. \quad (\text{A.4.3})$$

Moreover, by triangle inequality, we have

$$\|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1 \geq \|(\boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1 - \|(\widehat{\boldsymbol{\theta}}_j^{(l)})_{S_{\theta_j}}\|_1. \quad (\text{A.4.4})$$

Combining (A.4.3) and (A.4.4), we can obtain

$$\|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}}\|_1 \geq \|(\widehat{\boldsymbol{\theta}}_j^{(l)} - \boldsymbol{\theta}_j^*)_{S_{\theta_j}^c}\|_1.$$

This completes the proof. ■

A.5 Auxiliary Definitions, Lemmas and Theorems

We define sub-Exponential random variables and its corresponding ψ_1 norm as follows.

Definition A.5.1 (Definition 5.13 in [53]) *A random variable X is called sub-Exponential if there exists a constant $K > 0$ such that for all $p \geq 1$ the following inequality holds:*

$$(\mathbb{E}(|X|^p))^{1/p} \leq Kp. \quad (\text{A.5.1})$$

The ψ_1 norm of X , denoted as $\|X\|_{\psi_1}$, is the smallest K that makes (A.5.1) holds. In other words,

$$\|X\|_{\psi_1} = \sup_{p \geq 1} p^{-1} (\mathbb{E}(|X|^p))^{1/p}.$$

Similarly, sub-Gaussian random variables and the corresponding ψ_2 norm are defined as follows:

Definition A.5.2 (Definition 5.7 in [53]) *A random variable X is called sub-Gaussian if there exists a constant $K > 0$ such that for all $p \geq 1$ the following inequality holds:*

$$(\mathbb{E}(|X|^p))^{1/p} \leq K\sqrt{p}. \quad (\text{A.5.2})$$

The ψ_2 norm of X , denoted as $\|X\|_{\psi_2}$, is the smallest K that makes (A.5.2) holds. In other words,

$$\|X\|_{\psi_2} = \sup_{p \geq 1} p^{-1/2} (\mathbb{E}(|X|^p))^{1/p}.$$

We can generalize the concept of sub-Gaussian random variable to sub-Gaussian random vector.

Definition A.5.3 (Definition 5.22 in [53]) A random vector $\mathbf{X} \in \mathbb{R}^d$ is sub-Gaussian if for any vector $\mathbf{u} \in \mathbb{R}^d$ the inner product $\langle \mathbf{X}, \mathbf{u} \rangle$ is a sub-Gaussian random variable. And the corresponding ψ_2 norm of \mathbf{X} is defined as

$$\|\mathbf{X}\|_{\psi_2} = \sup_{\|\mathbf{u}\|_2=1} \|\langle \mathbf{X}, \mathbf{u} \rangle\|_{\psi_2}.$$

It is obvious that for any sub-Gaussian random vector $\mathbf{X} \in \mathbb{R}^d$, $\|\mathbf{X}\|_{\psi_2} \geq \max_{j=1}^d \|X_j\|_{\psi_2}$.

It is proved in [53] that a centered Gaussian random variable X with variance σ^2 is also a sub-Gaussian random variable with $\|X\|_{\psi_2} \leq C\sigma$ where C is an absolute constant. Therefore, we can easily show that a centered Gaussian random vector \mathbf{X} with covariance matrix Σ is also a sub-Gaussian random vector with $\|\mathbf{X}\|_{\psi_2} \leq C\lambda_{\max}(\Sigma)$.

The following theorem is the Hoeffding type inequality for sub-Gaussian random variables, which characterizes the tail bound for the weighted sum of independent sub-Gaussian random variables.

Theorem A.5.4 (Proposition 5.10 in [53]) Let X_1, X_2, \dots, X_n be independent centered sub-Gaussian random variables, and let $K = \max_i \|X_i\|_{\psi_2}$. Then for every $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n$ and for every $t > 0$, we have

$$\mathbb{P}\left(\left|\sum_{i=1}^n a_i X_i\right| > t\right) \leq \exp\left(-\frac{Ct^2}{K^2 \|\mathbf{a}\|_2^2}\right),$$

where $C > 0$ is an absolute constant.

Similarly, the following theorem is the Bernstein type inequality for sub-Exponential random variables, which provides the tail bound on the weighed sum of independent sub-Exponential random variables.

Theorem A.5.5 (Proposition 5.16 in [53]) *Let X_1, X_2, \dots, X_n be independent centered sub-Exponential random variables, and let $K = \max_i \|X_i\|_{\psi_1}$. Then for every $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n$ and for every $t > 0$, we have*

$$\mathbb{P}\left(\left|\sum_{i=1}^n a_i X_i\right| > t\right) \leq 2 \exp\left(-C \min\left\{\frac{t^2}{K^2 \|\mathbf{a}\|_2^2}, \frac{t}{K \|\mathbf{a}\|_\infty}\right\}\right),$$

where $C > 0$ is an absolute constant.

Lemma A.5.6 *For X_1 and X_2 being two sub-Gaussian random variables, $X_1 X_2$ is a sub-Exponential random variable with*

$$\|X_1 X_2\|_{\psi_1} \leq C \max\{\|X_1\|_{\psi_2}^2, \|X_2\|_{\psi_2}^2\},$$

where $C > 0$ is an absolute constant.

Lemma A.5.6 reveals that the product of two sub-Gaussian random variables is a sub-Exponential random variable and gives an upper bound on its ψ_1 norm.

Theorem A.5.7 (Corollary 1 in [39]) *For any design matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ with i.i.d. rows following Gaussian distribution $N(0, \Sigma)$, if Σ satisfies the RE condition with parameter (s, α, γ) , and the sample size n satisfies*

$$n > \frac{C'' \rho^2(\Sigma)(1 + \alpha)^2}{\gamma^2} s \log d,$$

then the sample covariance matrix $\hat{\Sigma} = \mathbf{X}^\top \mathbf{X} / n$ satisfies the restricted eigenvalue condition with parameter $(s, \alpha, \gamma/8)$ with probability at least $1 - C' \exp(-Cn)$, where C, C' and C'' are absolute constants and $\rho^2(\Sigma) = \max_{1 \leq j \leq d} \Sigma_{jj}$.