EXPLOITING SPARSITY FOR MACHINE LEARNING IN BIG DATA

BY

RONGDA ZHU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Chengxiang Zhai, Chair
Professor Jiawei Han
Assistant Professor Jian Peng
Doctor Yu Deng, IBM Research

Abstract

The rapid development of modern information technology has significantly facilitated the generation, collection, transmission and storage of all kinds of data. With the so-called "big data" generated in an unprecedented rate, we are facing significant challenges in learning knowledge from it. Traditional machine learning algorithms often suffer from the unmatched volume and complexity of such big data, however, sparsity has been recently studied to tackle this challenge. With reasonable assumptions and effective utilization of sparsity, we can learn models that are simpler, more efficient and robust to noise.

The goal of this dissertation is studying and exploiting sparsity to design learning algorithms to effectively and efficiently solve various challenging and significant real-world machine learning tasks. I will integrate and introduce my work from three different perspectives: sample complexity, computational complexity, and noise reduction. Intuitively, these three aspects correspond to models that require less data to learn, are more computationally efficient, and still perform well when the data is noisy. Specifically, this thesis is integrated from the three aspects as follows:

First, I focus on the sample complexity of machine learning algorithms for an important machine learning task, compressed sensing. I propose a novel algorithm based on nonconvex sparsity-inducing penalty, which is the first work that utilizes such penalty. I also prove that our algorithm improves the best previous sample complexity significantly by extensive theoretical derivation and numerical experiments.

Second, from the perspective of computational complexity, I study the expectation-maximization (EM) algorithms in high dimensional scenarios. In contrast to the conventional regime, the maximization step (M-step) in high dimensional scenario can be very computationally expensive or even not well defined. To address this challenge, I propose an efficient algorithm based on novel semi-stochastic gradient descent with variance reduction, which naturally incorporates the sparsity in model parameters, greatly economizes the computational cost at each iteration and enjoys faster convergence rates simultaneously. We believe the proposed unique semi-stochastic variance-reduced gradient is of general interest of nonconvex optimization of bivariate structure.

Third, I look into the noise reduction problem and target on an important text mining task, event detection.

To overcome the noise in the text data which hampers the detection of real events, I design an efficient algorithm based on sparsity-inducing fused lasso framework. Experiment results on various datasets show that our algorithm effectively smooths out noises and captures the real event, outperforming several state-of-the-art methods consistently in noisy setting.

To sum up, this thesis focuses on the critical issues of machine learning in big data from the perspective of sparsity in the data and model. Our proposed methods clearly show that utilizing sparsity is of great importance for various significant machine learning tasks.

To my family.

Acknowledgements

First of all, I would like to express my most sincere gratitude to my advisor Professor Chengxiang Zhai for his support and care during my PhD study. Professor Zhai is both a great researcher and a helpful advisor. He provides me with effective and insightful guidance when I run into difficulties, and also grants me great independence and freedom on my research at the same time. His passion for research always moves me deeply and encourages me to explore the unknown. I am very blessed to have him as my advisor. Without his help and guidance, this thesis would not have been possible.

I want to thank Professor Jian Peng, for his help and guidance in my PhD research work on sparsity in event detection. His knowledge, acumen and dedication inspire me and I have really learned a great deal from him. I am also grateful to the other two great researchers in my PhD committee, Professor Jiawei Han and Doctor Yu Deng, for their constructive comments and advices for this thesis. Doctor Deng was my intern manager at IBM research. She gave me great support and contributed a lot to my PhD research.

I would also like to thank Professor Quanquan Gu from University of Virginia. He led me into machine learning research systematically and sparked my genuine interest in sparsity. He also helped me a lot with the technical details of my research work.

I owe special thanks to my all friends and labmates in the group including Aston Zhang, Yinan Zhang, Hongwei Wang, Hongning Wang, Mingjie Qian, Yanen Li, Jingjing Wang, Xiaolong Wang, Jialu Liu, Sheng Wang, Shan Jiang, Xueqing Liu, Sean Massung, Chase Geigle, Jason Cho, Ismini Lourentzou, Yang Liu, Yiren Wang and every DAIS group member. Thanks for your help and encouragement!

Finally, I would like to thank my families for their wholehearted love and care, which support me to overcome all the difficulties. This thesis is dedicated to them.

Table of Contents

Chapter 1

Introduction and Motivation

Datasets in this era grow at a rapid pace across various fields of engineering and science. For example, the prosperity of online social media has led to overwhelming amount of text data; wireless sensor networks are gathering physical and environmental data such as temperature, sound and voltage; biomedical data are accumulating on computers and servers and facilitating the research of genomics and proteomics. Due to the enormous scales and complexity of these datasets, how we can efficiently learn simple and useful knowledge from them emerges as a critical challenge in this so-called "big data" era.

While such unprecedented massive amounts of data provide us with huge opportunities, conventional machine learning algorithms also show their limitations. Generally speaking, the framework of a learning algorithm is that it takes a certain **training dataset** as input and learns a desired objective through a designed **computation process**. Correspondingly, there are three important aspects for evaluation of a learning algorithm, in terms of the quantity and quality of the training dataset, and the complexity of the computation process:

- **Sample Complexity.** Sample complexity measures the number of samples a machine learning algorithm needs, so that the function returned by the algorithm is within an arbitrary small error of the best possible function, with probability arbitrary close to 1. In other words, a better learning algorithm in terms of sample complexity should need fewer examples to achieve a certain error bound with high probabilities.

- **Noise Reduction.** The input data can always be disturbed by irregular fluctuations and perturbances which is often referred to as noises. Noise reduction steps such as Gaussian smoothing and wavelet smoothing are often adopted to overcome such undesirable noises and facilitate the learning process. A robust learning algorithm should be able to address the noise challenge and still achieve desirable performance in the noisy settings.

- **Computational Complexity.** Computational complexity concerns the computational resources a learning algorithm needs to learn the desired target from the input dataset. In big data scenario, the

computational complexity has particularly been a rising challenge due to the scale and complexity of the data. Traditional methods may face prohibitive cost.

In order to better meet our needs in high dimensional and big data scenario, it's crucial that we propose learning algorithms improved from all three aspects above, i.e., algorithms requiring less examples for training, more robust to noise and more computationally efficient for learning.

To tackle such significant challenges in big data scenario, **sparsity** has been widely studied and used as a workhorse. Despite the ubiquitous high dimensional and complex data, many real-world signals and processes are concurrently sparse. For example, in speech recognition and image processing, the signals are often sparse in frequency domain or under some other appropriate basis; in biomedical research, only a few genes out of a huge number are of interest to a certain hereditary feature; in online social media, there are vast amount of short text snippets with sparsity in vocabulary. In some scenarios, we also want our learned models to be sparse, for lower computational cost and better interpretability. By exploring intrinsic sparsity of the data or applying reasonable sparsity assumptions, this thesis aims at learning compact, efficient and robust models that best fit the scale and dimensionality of this big data era.

This thesis attempts to exploit the sparsity in the data and model, and address the aforementioned three challenges. Specifically, we target at three important tasks in machine learning and text mining, i.e., one-bit compressed sensing, high dimensional expectation-maximization (EM) and event detection, and improve previous best results in the aspects of sample complexity, computational complexity and noise reduction by developing novel algorithms incorporating sparsity.

1.1 Lower Sample Complexity for Robust One-bit Compressed Sensing

The first component of this thesis is an efficient and robust algorithm for one-bit compressed sensing which improves the sample complexity significantly. Compressed sensing is the technique to recover a sparse signal using a few linear measurements. As we know, Nyquist rate is usually required for measurements to exactly recover the unknown signal [1]. However, when the signal is sparse, i.e., only a few entries are nonzero, we can restore the unknown signal with much fewer measurements by sophisticated measurement matrices and recovery algorithms. While conventional compressed sensing uses real-valued measurements, **one-bit** compressed sensing utilizes only one-bit, i.e., the sign of the measurements. Therefore, one-bit compressed sensing is often more robust to noise and non-linearity.

Sample complexity is one of the most important evaluation metrics for the problem of one-bit compressed

sensing, which is used to denote the number of measurements needed for an algorithm to obtain an estimator of the signal with error bounded by constant ϵ. For example, the sample complexity of [2], a convex estimator by linear programming, is $O(s \log^2 d/\epsilon^5)$. Such sample complexity means that when the signal dimension is d and at most s entries are nonzero, the algorithm needs $O(s \log^2 d/\epsilon^5)$ one-bit measurements to find an estimator with the estimation error bounded by ϵ. For the one-bit compressed sensing problem, it is crucial that we improve the sample complexity of algorithms to accommodate the scale and complexity of the data in high dimensional and big data scenarios. My proposed algorithm based on nonconvex penalty functions improves the sample complexity of the recovery of strong signals significantly from previous best results $O(s \log d/\epsilon^2)$ to $O(s/\epsilon^2)$, which is especially important for the high dimensional regime.

1.2 Accelerated Stochastic Gradient Expectation-Maximization Algorithm

The second contribution of this thesis focuses on the computational complexity. We propose an accelerated EM algorithm based on stochastic gradient. EM algorithm is widely used as a popular algorithm for the estimation of latent variable models. However, in high dimensional cases, the maximization step (M-step) can be time consuming or even not well defined. Therefore, the more general gradient EM algorithms, where the M-step is based on gradient ascent, have been attracting increasing research attention. However, these algorithms can still be computationally prohibitive in big data scenarios, since they need to compute the full gradient in each iteration.

To address this great challenge of computational complexity, we propose a novel algorithm based on a unique semi-stochastic gradient, where we only need to compute the gradient over a mini-batch each time. Our work is also the first method that brings variance reduction into the EM algorithm to overcome the intrinsic variance of stochastic gradient. The specially designed semi-stochastic structure and variance reduction distinguish our work from all existing methods. Our algorithm is proved to reduce the computational and concurrently outperform the state-of-the-art methods in terms of estimation error. Specifically, we show that with an appropriate initialization, our estimator achieves a linear convergence with the statistical rate of convergence matching the best previous result up to a logarithmic factor.

1.3 Noise Reduction in Event Detection

The third part of this thesis is noise reduction in event detection from text data. With the overwhelming text information, event detection has emerged as an important task that can significantly helps us understand the large-scale text data, such as scientific literature and social media. However, this detection process is often hampered by the heavy noise in the data. Therefore, noise reduction is always of great necessity for more accurate event detection in both retrospective and online settings.

I propose a novel event detection based on the undiscovered temporal divergence of topic distributions to tackle this challenge. I find that enforcing sparsity in this divergence greatly helps with the noise issue. Sparsity-inducing longitudinal regularization is applied to such divergence to effectively combat the noise and capture the real events. Our proposed algorithm can be smoothly adapted to both retrospective and online settings, and is also scalable to work on social media like Twitter.

Organization: The rest of this thesis is organized as follows. In Chapter 2, I discuss the representative related work. I present my work on one-bit compressed sensing in Chapter 3, high dimensional EM in Chapter 4 and event detection from text data in Chapter 5 . Finally, Chapter 6 concludes the thesis and discusses potential future work.

Notation: Let $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{d\times d}$ be a matrix and $\mathbf{v} = (v_1, \ldots, v_d)^\top \in \mathbb{R}^d$ be a vector. We define the ℓ_q-norm ($q \geq 1$) of $\mathbf{v}$ as $\|\mathbf{v}\|_q = \left(\sum_{j=1}^d |v_j|^q\right)^{1/q}$. Specifically, $\|\mathbf{v}\|_0$ denotes the number of nonzero entries of $\mathbf{v}$, $\|\mathbf{v}\|_2 = \sqrt{\sum_{j=1}^d v_j^2}$ and $\|\mathbf{v}\|_\infty = \max_j |v_j|$. For $q \geq 1$, we define $\|\mathbf{A}\|_q$ as the operator norm of $\mathbf{A}$. Specifically, $\|\mathbf{A}\|_2$ is the spectral norm. We let $\|\mathbf{A}\|_{\infty,\infty} = \max_{i,j} |A_{ij}|$. For an integer $d > 1$, we define $[d] = \{1, \ldots, d\}$. For an index set $\mathcal{I} \in [d]$ and vector $\mathbf{v} \in \mathbb{R}^d$, we use $\mathbf{v}_\mathcal{I} \in \mathbb{R}^d$ to denote the vector where $[\mathbf{v}_\mathcal{I}]_j = v_j$ if $j \in \mathcal{I}$, and $[\mathbf{v}_\mathcal{I}]_j = 0$ otherwise. We use $\mathrm{supp}(\mathbf{v})$ to denote the index set of its nonzero entries, and $\mathrm{supp}(\mathbf{v}, s)$ to denote the index set of top s largest $|v_j|$'s. $C, C', C_1, C_2, \ldots$ are used to denote some absolute constants. The values of these constants may be different from case to case. Let $\|X\|_{\psi_q}$ ($q \geq 1$) be the Orlicz norm of random variable X. $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ are used to denote the largest and smallest eigenvalues of matrix $\mathbf{A}$. We use $\mathcal{B}(r; \boldsymbol{\beta})$ to denote the ball centered at $\boldsymbol{\beta}$ with radius r.

Chapter 2

Related Work

In this chapter, we discuss the related work in details. Specifically, we will review the existing literature by the machine learning tasks we focus on separately: one-bit compressed sensing, high dimensional EM algorithm and event detection.

2.1 One-bit Compressed Sensing

One-bit compressed sensing was first introduced in [3] where the authors minimized the ℓ_1 norm of a unit vector which is consistent with the measurements, and further shown effective recovering sparse signals from nonlinearly distorted measurements [4]. Suppose $\mathbf{x}^*$ is the unknown signal vector, and $\{\mathbf{u}_i\}_{i=1}^n$ is a set of measurement vectors. The sign of real-valued measurement is observed as follows:

$$y_i = \text{sign}(\langle \mathbf{u}_i, \mathbf{x}^* \rangle), i = 1, 2, \ldots, n$$

where y_i is the binary one-bit measurement we use.

In general, there are two major tasks in one-bit compressed sensing: (1) approximate signal vector recovery [5, 6, 7], which aims at finding an estimator $\widehat{\mathbf{x}}$ with an estimation error $\|\widehat{\mathbf{x}} - \mathbf{x}^*\|_2$ small enough; and (2) support recovery, which finds the support, i.e., positions of the nonzero entries [5, 8, 9].

For the first task, approximate signal vector recovery, it is worth noting that since only the sign of the real-valued measurements are used, we cannot recover the magnitude of the signal, i.e., we always assume that the signal $\mathbf{x}^*$ is a unit vector with $\|\mathbf{x}^*\|_2 = 1$, which further makes this problem nonconvex. A convex formulation is proposed in [2], where ℓ_1 norm is put on the measurement vectors instead of signal vectors. The sample complexity of this work is $O(s \log^2 d/\epsilon^5)$, where ϵ is the guaranteed estimation error bound. [10] also proposed a popular convex approach by maximizing the dot product of the one-bit measurements and the real-valued measurements. Their sample complexity is $O(s \log d/\epsilon^4)$.

The best previous results in terms of sample complexity for approximate vector recovery is achieved in [7]. The authors proposed an efficient algorithm with close-form solution based on ℓ_1 regularization. Their

sample complexity is $O(s \log d/\epsilon^2)$.

For the second task, support recovery, we have the current best sample complexity of $O(s \log d)$ in [9]. However, their result depends on specially designed measurement matrices based on the signals, thus not universal. A universal method for support recovery is proposed in [5], which is based on two combinatorial structures: union free families of sets and expanders. For their method, all signals can be recovered using a single measurement matrix. The sample complexity is $O(s^2 \log d)$.

Gaussian measurements are used in the majority of the cases for its generality, and recently one-bit compressed sensing is also extended to non-Gaussian measurements [11]. The authors use sub-Gaussian measurements to recover both exactly and approximately sparse signals that are not extremely sparse. Other extensions have also been studied. For example, in [12] the sparse signals to be recovered can be with unknown and time-variant sparsity levels and the measurements are noisy. [13] studied one-bit compressed sensing on piece-wise smoothing signals.

Most of the previous studies only focus on one of the two major tasks. In contrast, my proposed algorithm [14] is proved to improve the best previous sample complexity significantly and achieve exact support recovery at the same time. At the core of my algorithm is nonconvex sparsity-inducing penalty function, which has been studied and utilized in various fields of statistics [15, 16, 17, 18]. My work is the first ever study to introduce such penalty functions into the problem of one-bit compressed sensing.

2.2 High Dimensional EM Algorithm

EM algorithm and its variants [19, 20] are widely used for the estimation of latent variable models and studied for a long time [21, 22, 23, 24]. There has been a long history of convergence analysis for EM algorithms [20, 25], however, only until recent research efforts [26, 27, 28] do we have rigorous understanding on the statistical convergence guarantees of EM algorithms.

The first study of definite statistical rate of convergence was introduced in [26], where the authors showed that with a suitable initialization, their algorithm can always converge to a reasonable local optima at a linear rate. Nonetheless, their work is only for low dimensional regime. The conventional EM algorithm as well as its gradient variants were extended to the high dimensional setting in [27], where the number of parameters of the latent variable is comparable to or even larger than the number of data points. According to their study, EM algorithms in the high dimensional regime must be carefully regularized by sparsity-type assumptions. Specifically, they applied a truncation step (T-step) after the M-step at each iteration. Another relevant study was introduced in [28], where the authors used a regularized estimator in M-step. It

is worth noting that all of the methods mentioned above are deterministic requiring the computation of full gradient at each iteration.

In order to avoid the prohibitive computational complexity in large-scale optimization [29, 30], stochastic gradient methods are always a popular workaround. For such methods, we only need to compute a partial gradient based on a stochastic mini-batch of data. However, the inherent variance is another challenge which hampers the convergence rate of stochastic methods [31, 32]. Accordingly, variance reduction techniques are studied to overcome this challenge. One of the most popular methods is the stochastic variance-reduced gradient (SVRG) [33], which has been widely utilized for a lot of optimization problems [34, 35, 36], and even for nonconvex problems [37, 38] for variance reduction.

Nonetheless, all the previous studies only tackle the univariate scenario, i.e, the optimization depends on only one variable. In EM algorithm, the structure is bivariate, and whether variance reduction can be applied to such structure is still remained to be seen. To the best of our knowledge, our work [39] is the first algorithm that incorporates variance reduction into EM algorithms in the high dimensional regime.

It is worth noting that reasonable initialization is a necessary condition for the convergence and statistical guarantees of high dimensional EM algorithms. Without a proper initialization, the estimator can be far away from the true model parameter and statistical properties of the objective function may not apply. Therefore, it is possible the estimation error accumulates instead of converges along the iterations. For different latent variable models such as Gaussian mixture model and mixture or linear regression, there are various spectral methods [40, 41] that helps with the initialization.

2.3 Event Detection with Noise Reduction

In a collection of documents, events are significant and novel stories. The discovery of such significant stories that have not been aforementioned, known as event detection, is often of great importance in understanding the data. For example, event detection on scientific literature can greatly help new researchers understand how the research interests evolve over time [42]; event detection on Twitter has been a popular approach to discover the bursty or trending topics and public interests [43], and even faster earthquake detection has been proposed using such methods [44].

Existing studies on event detection can be generally classified into two categories:

- *document-pivot* methods.
- *feature-pivot* methods.

Document-pivot methods focus on clustering the documents and analyze these clusters of documents to find features for events. A representative method is used in the UMASS system [45] exploiting *term frequency–inverse document frequency* (TF-IDF) weight vectors to represent the document features, and identifies a new document as an event if it is different enough from all existing clusters. Otherwise, it is assigned to the closest cluster and the cluster center is updated. This method has achieved best performances in several topic detection and tracking (TDT) competitions. To make the UMASS system efficient enough for working on social media scale like Twitter, [46] improved the scalability by locality-sensitive hashing (LSH).

Feature-pivot methods aims at detecting the statistical patterns of the corpus and get event features from these patterns, which can be term frequency, term cooccurrences and distributions.

For example, in [47], the frequency of each term is modeled by a binomial distribution. The bursty features are detected as a set of words when the parameters of these distributions change. They use a set of cooccuring bursty features to feature a detected events. This idea is further extended to an event hierarchy construction in [48], where the documents are clustered based on their bursty features into a hierarchical event structure. In [49], *Discrete Fourier Transform* (DFT) is applied to extract the bursty features from term frequency. Since frequency-domain techniques are involved, this method naturally distinguishes periodic and aperiodic events well.

With the increasing popularity of user-generated data such as citizen journalism and social media, the "noise" in data is also emerging as a significant challenge. For example, meaningless "babbles" [46] are generated at a very high rate on Twitter. A straightfoward solution was proposed in [50], where the authors simply used the hash tag *#breakingnews* to pick the valuable news posts out of the noises. The wavelet signals generated from term frequency was first used in [51] to filter out the noises. Specifically, the authors determine all the signals with the auto-correlation lower than a threshold as trivial. They then cluster the terms based on cross-correlation of different wavelet signals. To group the similar Tweets which might be of short length, the weight of proper nouns is boosted in TF-IDF weighting. The temporal and geographical features of social media such as Twitter are also important for event detection. For example, [52] proposed an event detection framework based on time and location-based topics.

As we have introduced, *feature-pivot* methods [43, 47, 48, 51] study the distribution of terms and detect events by clustering these terms. By nature, these methods are closely related to topic models which aim to extract hidden topics from text data and also characterize these topics by word clusters. Static topic models such as probabilistic latent semantic analysis (PLSI) [53] and latent Dirichlet allocation (LDA) [54] have gained great success, and time is also incorporated [55, 56, 57] to discover evolving topics in text corpus. Specifically, an online variant of LDA with applications to event detection was proposed in [57]. The authors

learn the model in an incremental fashion, where the model from last iteration passes its parameters to the next iteration as priors. Then word distributions of models from two consecutive iterations are compared to see if there is enough difference indicating an event. This model is further extended to a dynamic vocabulary in [58].

Despite the increasing research attention on topic model-based methods on event detection, most of them detect the events by exploiting the divergence of *word distributions* of topics. The *topic distributions* of documents featuring the coverage of topics in the corpus, have not received research attention. Our work is the first study that looks into topic distributions for the problem of event detection.

2.4 Sparsity

Sparsity is utilized in a wide variety of machine learning problems [18, 59, 60], which helps us learn more compact and interpretable models with lower sample and computational complexity. As we have mentioned, sparsity is ubiquitous especially in high dimensional scenarios. Therefore, it is often reasonable to apply sparsity-inducing regularizers to enforce sparse structure in high-dimensional data or models. The most commonly used techniques include ℓ-0 regularization [61, 62, 63] and ℓ-1 regularization [64, 65, 66].

In this thesis, we incorporate sparsity to our machine learning algorithms from different perspectives. We look into different sparse-inducing regularization functions, and apply them to the output of our models. In our work, we focus sparsity in both the original data and resulting model.

More specifically, in our work on one-bit compressed sensing, sparsity is in the signal we want to recover; in our work on stochastic gradient EM algorithm, we want to enhance sparsity in the model parameters we learn; in the event detection problem, we also enforce sparsity in the parameter we learn to better encode the nature of real events. We can see that sparsity can really be exploited flexibly to match our needs in different machine learning tasks.

Chapter 3

Lower Sample Complexity for Robust One-bit Compressed Sensing

In this chapter, I will present my work on one-bit compressed sensing [14]. I propose an efficient algorithm with close-form solution, achieving a significantly improved sample complexity for vector recovery and exact support recovery simultaneously.

3.1 Background

We first briefly describe the general framework of one-bit compressed sensing. We let $\mathbf{x}^*$ is the unknown signal vector, and $\|\mathbf{x}^*\|_0 \leq s$. $\{\mathbf{u}_i\}_{i=1}^n$ is a set of measurement vectors and the one-bit measurements are the signs of real-valued measurements observed as follows:

$$y_i = \text{sign}(\langle \mathbf{u}_i, \mathbf{x}^* \rangle), i = 1, 2, \ldots, n.$$

Our goal is to recover $\mathbf{x}^*$ from $\{(y_i, \mathbf{u}_i)\}_{i=1}^n$. Note that in one-bit compressed sensing the norm of the signal does not affect the measurements, thus we let $\|\mathbf{x}^*\|_2 = 1$. We focus on the more realistic noisy setting, where y_i can be influenced by irrational perturbances. As described in [10], we assume y_i can be treated as independently drawn from a distribution with the following expectation

$$\mathrm{E}(y_i|\mathbf{u}_i) = \theta\big(\langle \mathbf{u}_i, \mathbf{x}^* \rangle\big), i = 1, 2, \ldots, n$$

where $\theta(z)$ is the function modeling the expectation with value domain $[-1, 1]$. We define

$$\mathrm{E}[\theta(g)g] =: \gamma > 0, \tag{3.1.1}$$

where $g \sim N(0, 1)$ is a standard Gaussian random variable, and γ measures the correlation between y_i and $\langle \mathbf{u}_i, \mathbf{x}^* \rangle$. When the noise is not significant, these two are well correlated, which means that γ will get a higher value. When y_i is equal to $\text{sign}(\langle \mathbf{u}_i, \mathbf{x}^* \rangle)$, there is no noise and γ will get the maximal value $\sqrt{2/\pi}$.

3.2 Nonconvex Penalty Functions

At the core of my framework is the nonconvex penalty functions. In this work, we also have these functions as decomposable

$$\mathcal{G}_{\lambda,b}(\mathbf{x}) = \sum_{i=1}^{d} g_{\lambda,b}(x_i),$$

where $\mathcal{G}_{\lambda,b}(\mathbf{x})$ is the decomposable function on the signal vector and $g_{\lambda,b}(x_i)$ is the component function on the entries. λ and b are regularization parameters shaping the function.

There are a variety of nonconvex penalties that are decomposable. Representatives include the smoothly clipped absolute deviation (SCAD) penalty [15] and minimax concave penalty (MCP) [16]. Specifically, MCP is given by

$$g_{\lambda,b}(t) = \begin{cases} \lambda|t| - \dfrac{t^2}{2b}, & \text{if } |t| \le b\lambda, \\ \dfrac{b\lambda^2}{2}, & \text{if } |t| > b\lambda, \end{cases} \tag{3.2.1}$$

where $b > 0, \lambda > 0$ are fixed regularization parameters. An important property of $g_{\lambda,b}(t)$ is that it can be written as the sum of a ℓ_1 penalty part and a concave part $h_{\lambda,b}(t) : g_{\lambda,b}(t) = \lambda|t| + h_{\lambda,b}(t)$.

Our work does not depend on specific form of $g_{\lambda,b}(t)$, such as MCP or SCAD. Generally, our work only depends on the following conditions on $g_{\lambda,b}(t)$ and $h_{\lambda,b}(t)$:

C1. $g'_{\lambda,b}(t) = 0$, for $|t| \ge \nu \ge 0$.

C2. $h'_{\lambda,b}(t)$ is monotone, and for $t' > t$, there is a constant $\zeta_- \ge 0$ such that

$$-\zeta_-(t' - t) \le h'_{\lambda,b}(t') - h'_{\lambda,b}(t).$$

C3. $h_{\lambda,b}(0) = h'_{\lambda,b}(0) = 0$.

C4. $|h'_{\lambda,b}(t)| \le \lambda$ for any t.

The above conditions hold for a wide variety of nonconvex penalty functions. For example, it can be proved that MCP and SCAD are valid choices. Specifically, $\nu = b\lambda$ and $\zeta_- = 1/b$ for MCP. I use MCP as the nonconvex penalty function in my algorithm, and g, $\mathcal{G}$ and h, $\mathcal{H}$ will be used to denote the component and sum functions of MCP in (3.2.1) and its concave part for the rest of this thesis.

3.3 One-bit Compressed Sensing with Nonconvex Penalty

We start with the framework of passive algorithm for one-bit compressed sensing [7], which is given by

$$\underset{\|\mathbf{x}\|_2\leq 1}{\operatorname{argmin}} -\frac{1}{n}\mathbf{x}^\top \mathbf{U}\mathbf{y} + \tau\|\mathbf{x}\|_1 \tag{3.3.1}$$

where $\mathbf{U}$ is the measurement matrix. Since the estimator should be reasonably consistent with the one-bit measurements, we need to maximize the dot product of $\mathbf{U}^\top\mathbf{x}$ and $\mathbf{y}$, which is the first part in (3.3.1). The second part is a ℓ_1 regularizer to enforce sparsity of the estimator.

Accordingly, our estimator $\widehat{\mathbf{x}}$ is any local optimal solution to the following optimization problem

$$\underset{\|\mathbf{x}\|_2\leq 1}{\operatorname{argmin}} -\frac{1}{n}\sum_{i=1}^{n} y_i\langle \mathbf{u}_i, \mathbf{x}\rangle + \mathcal{G}_{\lambda,b}(\mathbf{x}) + \frac{\tau}{2}\|\mathbf{x}\|_2^2, \tag{3.3.2}$$

where $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n \in \mathbb{R}^d$ are the rows of the known measurement matrix $\mathbf{U} \in \mathbb{R}^{n\times d}$, and $\mathcal{G}_{\lambda,b}(\cdot)$ is the nonconvex penalty function. I use ℓ_2 regularizer here. I will later show why the penalty function and regularizer are necessary.

I also propose a novel algorithm to efficiently compute the estimator as the local minima in (3.3.2). The basic idea here is divide and conquer. We denote $\mathbf{v} = \mathbf{U}^\top\mathbf{y}/n \in \mathbb{R}^d$ for simplicity.

To go over the details of the proposed algorithm, we start with the following lemma tackling the subproblem of the optimization in (3.3.2).

Lemma 3.3.1. The solution to the following optimization problem

$$\widehat{x} = \underset{x}{\operatorname{argmin}} \frac{1}{2}(x-y)^2 + g_{\lambda,b}(|x|)$$

is given by

- **if** $b > 1$

$$\widehat{x} = \begin{cases} \dfrac{S(y,\lambda)}{1-1/b}, & \text{if } |y| \leq b\lambda, \\ y, & \text{if } |y| > b\lambda, \end{cases} \tag{3.3.3}$$

- **if** $b \leq 1$

$$\widehat{x} = \begin{cases} 0, & \text{if } |y| \leq \sqrt{b}\lambda, \\ y, & \text{if } |y| > \sqrt{b}\lambda, \end{cases} \tag{3.3.4}$$

where $S(y,\lambda)$ is the soft-thresholding operator [67] defined for $\lambda \geq 0$ by

$$S(y,\lambda) = \begin{cases} y-\lambda, & \text{if } y > \lambda, \\ 0, & \text{if } |y| \leq \lambda, \\ y+\lambda, & \text{if } y < -\lambda. \end{cases}$$

Proof. For $b > 1$, please see [68]. For $b \leq 1$, please refer to Section 3.7. □

A similar version of Lemma 3.3.1 with $\tau > 0$ can be derived easily.

Lemma 3.3.2. The solution to the following optimization problem

$$\widehat{x} = \underset{x}{\operatorname{argmin}} \frac{1}{2}(x-y)^2 + g_{\lambda,b}(|x|) + \frac{\tau}{2}x^2$$

is given by

- **if** $b(1+\tau) > 1$

$$\widehat{x} = \begin{cases} \dfrac{S(y,\lambda)}{1+\tau-1/b}, & \text{if } |y| \leq b\lambda(1+\tau), \\ \dfrac{y}{1+\tau}, & \text{if } |y| > b\lambda(1+\tau). \end{cases} \tag{3.3.5}$$

- **if** $b(1+\tau) \leq 1$

$$\widehat{x} = \begin{cases} 0, & \text{if } |y| \leq \sqrt{b(1+\tau)}\lambda, \\ \dfrac{y}{1+\tau}, & \text{if } |y| > \sqrt{b(1+\tau)}\lambda. \end{cases} \tag{3.3.6}$$

Proof. Please see Section 3.7. □

From Lemma 3.3.3 and 3.3.4, we can see that the decomposed subproblems in (3.3.2) have close-form solutions.

Now we are in position to solve (3.3.2). For the sake of simplicity, we first consider the case where $\tau = 0$ to illustrate our method. The $\tau > 0$ case can be solved similarly, as we will show later.

We consider the Lagrange function $f(\mu)$ of (3.3.2) given by

$$\begin{aligned} f(\mu) &= \min_{\mathbf{x}} -\mathbf{x}^\top \mathbf{v} + \mathcal{G}_{\lambda,b}(\mathbf{x}) + \mu(\|\mathbf{x}\|_2^2 - 1) \\ &= \min_{\mathbf{x}} 2\mu \left(\frac{1}{2}\|\mathbf{x} - \frac{\mathbf{v}}{2\mu}\|_2^2 + \frac{\mathcal{G}_{\lambda,b}(\mathbf{x})}{2\mu} \right) - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu \\ &= 2\mu \left(\sum_i \min_{x_i} \frac{1}{2}\Big(x_i - \frac{v_i}{2\mu}\Big)^2 + g_{\lambda/(2\mu),2\mu b}(|x_i|) \right) - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu, \end{aligned} \tag{3.3.7}$$

where the last equation comes from the property of MCP.

We use μ^* to denote the optimal solution to the dual problem.

According to Lemma 3.3.1, we divide the problem into two cases: (1) $2\mu b \leq 1$ and (2) $2\mu b > 1$. For each subproblem in (3.3.7), we can just determine the value of μ by dividing the feasible region of v_i into intervals where the optimal value of μ can be determined. The outlines of our algorithms for these two cases are outlined in Algorithm 1 and 2 respectively.

I will only briefly introduce the algorithms in two cases, and the derivation and technical details of Algorithm 1 and 2 can be found in Section 3.7.

- $2\mu b \leq 1$: In this case, the solution to (3.3.7) comes from (3.3.4). Therefore, we need to compare the value of $|v_i/2\mu|$ and $\lambda\sqrt{b/2\mu}$ according to Lemma 3.3.1, which is equivalent to comparing μ and $v_i^2/2b\lambda^2$, to decide the value of each term in the summation in (3.3.7). After sorting $|v_i|$ and dividing the feasible region into intervals, we will compute $f(\mu)$ and find μ^* within each interval, which has a close form solution as in Line 5 to 11 of Algorithm 1 to get $f(\mu)$. Finally, among the optimal solutions in each interval, we find μ_1^* that maximizes $f(\mu)$.

- $2\mu b > 1$: In this case, the solution to (3.3.7) comes from (3.3.3). We do similar sorting and dividing operation, yet within each interval, we need to solve a simple optimization as in Line 8, Algorithm 2. Then we will find the final μ_2^* by comparing the values from each interval.

After finding the optimal values of μ from the above two cases, we compare the objective function values of outputs of Algorithm 1 and 2 to get the final μ^*:

$$\mu^* = \operatorname*{argmax}_{\mu \in \{\mu_1^*, \mu_2^*\}} f(\mu). \tag{3.3.8}$$

The optimal primal solution is further given by

$$\widehat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \frac{\mathbf{v}}{2\mu^*}\|_2^2 + \frac{\mathcal{G}_{\lambda,b}(\mathbf{x})}{2\mu^*}.$$

By Lemma 3.3.1, we would finally get our estimator as follows:

- ***if*** $2\mu^* b > 1$

$$\widehat{x}_i = \begin{cases} \dfrac{S(v_i, \lambda)}{2\mu^* - 1/b}, & \text{if } |v_i| \leq 2\mu^* \lambda b, \\ \dfrac{v_i}{2\mu^*}, & \text{if } |v_i| > 2\mu^* \lambda b. \end{cases}$$

- ***if*** $2\mu^* b \leq 1$

$$\widehat{x}_i = \begin{cases} 0, & \text{if } |v_i| \leq \sqrt{2\mu^* b}\lambda, \\ \dfrac{v_i}{2\mu^*}, & \text{if } |v_i| > \sqrt{2\mu^* b}\lambda. \end{cases}$$

For the case $\tau > 0$, we have a similar Lagrange function $f(\mu')$ with $\mu' = \mu + \tau/2$. The optimization of $f(\mu')$ is in a similar manner.

Algorithm 1 Find maximizer of $f(\mu)$ when $\mu \leq 1/2b$

Input: $\lambda, b, \mathbf{v}$
Output: μ_1^*
Initialize $f = f(1/2b), \mu_1^* = 1/2b$
$v_{(1)}, v_{(2)}, ..., v_{(d)} =$ **Sort**$(|v_1|, |v_2|, ..., |v_d|)$
$v_{(0)} = 0, v_{(d+1)} = \infty$
$l =$ **Find**$(v_{(l)} \leq 1/2b < v_{(l+1)})$
for i:=0 ... l **do**
  if $\sqrt{\sum_{j=i}^{n} v_{(j)}^2}/2 \in (v_{(i)}^2/2b\lambda^2, v_{(i+1)}^2/2b\lambda^2]$ **then**
    $\mu = \sqrt{\sum_{j=i}^{d} v_{(j)}^2}/2$
  else
    $\mu = v_{(i+1)}^2/2b\lambda^2$
  end if
  if $f(\mu) > f$ **and** $\mu < 1/2b$ **then**
    $f = f(\mu),\ \mu_1^* = \mu$
  end if
end for

3.4 Theoretical Results

We will prove that under a reasonable assumption on the elements of the true signal $\mathbf{x}^*$, our estimator will have oracle property, i.e., identical to the oracle estimator, with high probability. This indicates exact support recovery. We will also show the advantage of our method in terms of sample complexity.

Algorithm 2 Find maximizer of $f(\mu)$ when $\mu > 1/2b$

Input: $\lambda, b, \mathbf{v}$
Output: μ_2^*
Initialize $f = f(1/2b), \mu_2^* = 1/2b$
$v_{(1)}, v_{(2)}, ..., v_{(d)} = \mathbf{Sort}(|v_1|, |v_2|, ..., |v_d|)$
$v_{(0)} = 0, v_{(d+1)} = \infty$
$l = \mathbf{Find}(v_{(l)} \leq 1/2b < v_{(l+1)})$
for i:=l ... n **do**
  $S_1 = \sum_{j=i+1}^{n} v_{(j)}^2$
  $S_2 = \sum_{j=l}^{i} (|v_{(j)}| - \lambda)^2$
  $J(\mu) = \frac{S_1}{4\mu} + \frac{S_2}{2(2\mu - 1/b)} + \mu$
  if $\mu_i = \operatorname{argmin}_\mu J(\mu) \in (|v_{(i)}|/2b\lambda, |v_{(i+1)}|/2b\lambda]$ **then**
    $\mu = \mu_i$
  else
    $\mu = |v_{(i+1)}|/2b\lambda$
  end if
  if $f(\mu) > f$ **and** $\mu > 1/2b$ **then**
    $f = f(\mu),\ \mu_2^* = \mu$
  end if
end for

3.4.1 Oracle Property of Our Estimator

We will start with presenting the oracle property of the proposed estimator in (3.3.2). The definition of the oracle estimator $\widehat{\mathbf{x}}_O$ is given by

$$\widehat{\mathbf{x}}_O = \underset{\text{supp}(\mathbf{x}) \subset S, \|\mathbf{x}\|_2 \leq 1}{\operatorname{argmin}} \mathcal{L}_O(\mathbf{x}), \tag{3.4.1}$$

where $\mathcal{L}_O(\mathbf{x}) = -1/n \sum_{i=1}^{n} y_i \langle \mathbf{u}_i, \mathbf{x} \rangle$. We can see that for the oracle estimator, the support information is known as prior knowledge. The oracle property for an estimator is indicating this estimator is identical to the oracle estimator.

It is worth noting that support information is critical to the problem of one-bit compressed sensing. With the support information, the recovery problem will be much easier. Therefore, oracle property is often a strong criteria for estimators.

For the rest of this chapter, we use the following notations

$$\begin{aligned}
&\mathcal{H}_{\lambda,b}(\mathbf{x}) = \sum_{i=1}^{d} h_{\lambda,b}(x_i) = \mathcal{G}_{\lambda,b}(\mathbf{x}) - \lambda\|\mathbf{x}\|_1, \\
&\mathcal{L}(\mathbf{x}) = \mathcal{L}_O(\mathbf{x}) + \frac{\tau}{2}\|\mathbf{x}\|_2^2 = -\frac{1}{n}\mathbf{y}^\top \mathbf{U}\mathbf{x} + \frac{\tau}{2}\|\mathbf{x}\|_2^2, \\
&\widetilde{\mathcal{L}}_\lambda(\mathbf{x}) = \mathcal{L}(\mathbf{x}) + \mathcal{H}_{\lambda,b}(\mathbf{x}) = -\frac{1}{n}\mathbf{y}^\top \mathbf{U}\mathbf{x} + \frac{\tau}{2}\|\mathbf{x}\|_2^2 + \mathcal{H}_{\lambda,b}(\mathbf{x}).
\end{aligned} \tag{3.4.2}$$

We have the following important property for the oracle estimator.

Lemma 3.4.1. If $\tau \leq \|\mathbf{v}_S\|_2$ where $\mathbf{v} = -1/n \sum_{i=1}^n y_i \mathbf{u}_i$ and S is the support of $\mathbf{x}^*$. The following optimization problem

$$\widehat{\mathbf{x}} = \underset{\text{supp}(\mathbf{x}) \subset S, \|\mathbf{x}\|_2 \leq 1}{\text{argmin}} -\frac{1}{n} \sum_{i=1}^n y_i \langle \mathbf{u}_i, \mathbf{x} \rangle + \frac{\tau}{2} \|\mathbf{x}\|_2^2, \tag{3.4.3}$$

has the same solution as the oracle estimator in (3.4.1).

Proof of Lemma 3.4.1. We will first give the following lemma which features the close-form solution for the oracle estimator.

Lemma 3.4.2. The following optimization problem

$$\widehat{\mathbf{x}} = \underset{\text{supp}(\mathbf{x}) \subset S, \|\mathbf{x}\|_2 \leq 1}{\text{argmin}} -\frac{1}{n} \sum_{i=1}^n y_i \langle \mathbf{u}_i, \mathbf{x} \rangle, \tag{3.4.4}$$

has closed form solution, i.e.,

$$\widehat{x}_j = \begin{cases} v_j / \|\mathbf{v}_S\|_2, & \text{if } j \in S, \\ 0, & \text{otherwise.} \end{cases}$$

where $\mathbf{v} = 1/n \sum_{i=1}^n y_i \mathbf{u}_i = 1/n \cdot \mathbf{U}^\top \mathbf{y}$.

Proof. To solve the optimization problem in (3.4.4), it is sufficient to

$$\underset{\|\mathbf{z}\|_2 \leq 1}{\text{argmin}} -\langle \mathbf{v}_S, \mathbf{z} \rangle. \tag{3.4.5}$$

The Lagrange function of (3.4.5) is

$$L(\mathbf{z}, \alpha) = -\mathbf{v}_S^\top \mathbf{z} + \alpha(\|\mathbf{z}\|_2^2 - 1).$$

Taking the gradient of $L(\mathbf{z}, \alpha)$ with respect to $\mathbf{z}$ and setting it to zero, we obtain

$$-\mathbf{v}_S + 2\alpha \mathbf{z} = 0. \tag{3.4.6}$$

Therefore, we have $\mathbf{z} = 1/(2\alpha)\mathbf{v}_S$. Substituting it back into (3.4.5), we obtain the dual problem as follows

$$\underset{\alpha}{\text{argmin}} -\frac{1}{4\alpha} \|\mathbf{v}_S\|_2^2 - \alpha. \tag{3.4.7}$$

The optimal solution to the dual problem (3.4.7) is $\alpha^* = \|\mathbf{v}_S\|_2/2$. Substituting α^* back into (3.4.6), and solving for $\mathbf{z}$, we obtain that $\widehat{\mathbf{z}} = \mathbf{v}_S/\|\mathbf{v}_S\|_2$. This completes the proof. □

Now we are ready to prove Lemma 3.4.1. The optimization problem in (3.4.3) is equivalent to

$$\underset{\text{supp}(\mathbf{x})\subset S, \|\mathbf{x}\|_2\leq 1}{\text{argmin}} \frac{\tau}{2}\left\|\mathbf{x} - \frac{1}{\tau}\mathbf{v}\right\|_2^2 + \frac{1}{2\tau}\|\mathbf{v}\|_2^2. \tag{3.4.8}$$

It is sufficient to solve the following reduced problem restricted on S

$$\underset{\|\mathbf{z}\|_2\leq 1}{\text{argmin}} \frac{\tau}{2}\left\|\mathbf{z} - \frac{1}{\tau}\mathbf{v}_S\right\|_2^2.$$

If $1/\tau\|\mathbf{v}_S\|_2 < 1$, the optimal solution is $\widehat{\mathbf{z}} = 1/\tau\mathbf{v}_s$. If $1/\tau\|\mathbf{v}_S\|_2 \geq 1$, according to the proof of Lemma 3.4.2, the optimal solution is $\widehat{\mathbf{z}} = \mathbf{v}_S/\|\mathbf{v}_S\|_2$, and the corresponding optimal solution to (3.4.3) is

$$\widehat{x}_j = \begin{cases} v_j/\|\mathbf{v}_S\|_2, & \text{if } j \in S, \\ 0, & \text{otherwise.} \end{cases}$$

It is identical to the solution of the oracle estimator. Therefore, when $\tau \leq \|\mathbf{v}_S\|_2$, the oracle estimator is identical to the solution of (3.4.3). This completes the proof. □

We will now investigate the oracle property of our estimator in the following theorem:

Theorem 3.4.3 (**Oracle Property for Strong Signals**)**.** Assume that we have the nonconvex penalty $\mathcal{G}_\lambda(\mathbf{x}) = \sum_{i=1}^d g_{\lambda,b}(x_i)$ that satisfies conditions C1 and C2. If the true signal $\mathbf{x}^*$ satisfies the magnitude condition $\min_{j\in S}|x_j^*| \geq \nu + \|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2$, for our estimator $\widehat{\mathbf{x}}$ with regularization parameter $\lambda = C\sqrt{\log d/n} + |\gamma - \tau|$ and $\zeta_- < \tau \leq \|\mathbf{v}_S\|_2$ as in Lemma 3.4.1, there will be $\widehat{\mathbf{x}} = \widehat{\mathbf{x}}_O$.

Proof of Theorem 3.4.3. We start with the following two lemmas:

Lemma 3.4.4. For loss function $\mathcal{L}(\mathbf{x}')$ defined in (3.4.2), we have

$$\widetilde{\mathcal{L}}_\lambda(\mathbf{x}') \geq \widetilde{\mathcal{L}}_\lambda(\mathbf{x}) + \langle \nabla\widetilde{\mathcal{L}}_\lambda(\mathbf{x}), \mathbf{x}' - \mathbf{x}\rangle + \frac{\tau - \zeta_-}{2}\|\mathbf{x}' - \mathbf{x}\|_2^2.$$

Proof of Lemma 3.4.4. From condition C2, we have

$$-\zeta_-(\mathbf{x}_i' - \mathbf{x}_i)^2 \leq (h_{\lambda,b}'(\mathbf{x}') - h_{\lambda,b}'(\mathbf{x}))(\mathbf{x}_i' - \mathbf{x}_i),$$

which yields

$$\left\langle \nabla(-\mathcal{H}_{\lambda,b}(\mathbf{x}')) - \nabla(-\mathcal{H}_{\lambda,b}(\mathbf{x})), \mathbf{x}' - \mathbf{x} \right\rangle \leq \zeta_- \|\mathbf{x}' - \mathbf{x}\|_2^2,$$

which is equivalent to

$$-\mathcal{H}_{\lambda,b}(\mathbf{x}') \leq -\mathcal{H}_{\lambda,b}(\mathbf{x}) - \left\langle \nabla(-\mathcal{H}_{\lambda,b}(\mathbf{x})), \mathbf{x}' - \mathbf{x} \right\rangle + \frac{\zeta_-}{2}\|\mathbf{x}' - \mathbf{x}\|_2^2. \tag{3.4.9}$$

For $\mathcal{L}(\mathbf{x})$, it is strongly convex with modulus τ, we have

$$\mathcal{L}(\mathbf{x}') \geq \mathcal{L}(\mathbf{x}) + \left\langle \nabla(\mathcal{L}(\mathbf{x})), \mathbf{x}' - \mathbf{x} \right\rangle + \tau\|\mathbf{x}' - \mathbf{x}\|_2^2. \tag{3.4.10}$$

Subtracting (3.4.9) from (3.4.10), we obtain

$$\widetilde{\mathcal{L}}_\lambda(\mathbf{x}') \geq \widetilde{\mathcal{L}}_\lambda(\mathbf{x}) + \langle \nabla\widetilde{\mathcal{L}}_\lambda(\mathbf{x}), \mathbf{x}' - \mathbf{x} \rangle + \frac{\tau - \zeta_-}{2}\|\mathbf{x}' - \mathbf{x}\|_2^2. \tag{3.4.11}$$

□

Lemma 3.4.5. With a probability at least $1 - e/d$, we have

$$\left\| \frac{1}{n}\mathbf{U}^\top \mathbf{y} - \gamma \mathbf{x}^* \right\|_\infty \leq C\sqrt{\frac{\log d}{n}}.$$

Proof of Lemma 3.4.5. Please refer to [7]. □

We now prove Theorem 3.4.3. We let $\widehat{Z} \in \partial\|\widehat{\mathbf{x}}\|_1$, and $\widehat{\mathbf{x}}$ satisfies the optimality condition

$$\max_{\|\mathbf{x}'\|_2 \leq 1} \langle \widehat{\mathbf{x}} - \mathbf{x}', \nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}) + \lambda\widehat{Z} \rangle \leq 0.$$

Now we want to show that there exists some $\widehat{Z}_O \in \partial\|\widehat{\mathbf{x}}_O\|_1$ such that $\widehat{\mathbf{x}}_O$ also satisfies the same optimality condition, i.e.,

$$\max_{\|\mathbf{x}'\|_2 \leq 1} \langle \widehat{\mathbf{x}}_O - \mathbf{x}', \nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O \rangle \leq 0. \tag{3.4.12}$$

Since we have $\widetilde{\mathcal{L}}_\lambda(\mathbf{x}) = \mathcal{L}(\mathbf{x}) + \mathcal{H}_{\lambda,b}(\mathbf{x})$, therefore,

$$\langle \widehat{\mathbf{x}}_O - \mathbf{x}', \nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O\rangle = \underbrace{\sum_{i\in S}(\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i}_{(1)} + \underbrace{\sum_{i\in S^c}(\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i}_{(2)}, \tag{3.4.13}$$

where S is the support of true signal $\mathbf{x}^*$. For the term (1) in (3.4.13), we first know $\|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_\infty \le \|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2$ and by the assumption that $\min_{i\in S}|\mathbf{x}^*| \ge \nu + \|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2$, there is

$$\begin{aligned}\min_{i\in S}|(\widehat{\mathbf{x}}_O)_i| = \min_{i\in S}|(\widehat{\mathbf{x}}_O - \mathbf{x}^* + \mathbf{x}^*)_i| &\ge -\max_{i\in S}|(\widehat{\mathbf{x}}_O - \mathbf{x}^*)_i| + \min_{i\in S}|\mathbf{x}^*_i| \\ &\ge -\|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2 + \nu + \|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2 = \nu.\end{aligned}$$

Since $\mathcal{G}_{\lambda,b}(\mathbf{x}) = \mathcal{H}_{\lambda,b}(\mathbf{x}) + \lambda\|\mathbf{x}\|_1$, according to condition C1, $(\nabla\mathcal{H}_{\lambda,b}(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i = (\nabla\mathcal{G}_{\lambda,b}(\widehat{\mathbf{x}}_O))_i = g'_{\lambda,b}(\widehat{\mathbf{x}}_O)_i = 0$ for $i \in S$, therefore,

$$\sum_{i\in S}(\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i = \sum_{i\in S}(\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\mathcal{L}(\widehat{\mathbf{x}}_O))_i.$$

Note that by Lemma 3.4.1, $\widehat{\mathbf{x}}_O$ satisfies the optimality condition

$$\max_{\|\mathbf{x}'\|_2\le 1}\sum_{i\in S}(\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\mathcal{L}(\widehat{\mathbf{x}}_O))_i \le 0,$$

so we can get

$$\sum_{i\in S}(\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i \le 0. \tag{3.4.14}$$

For the term (2) in (3.4.13), we have for $i \in S^c$, by condition C3, $(\nabla\mathcal{H}_{\lambda,b}(\widehat{\mathbf{x}}_O))_i = h'_{\lambda,b}((\widehat{\mathbf{x}}_O)_i) = 0$, then we have

$$\sum_{i\in S^c}(\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i = \sum_{i\in S^c}(\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\mathcal{L}(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i.$$

Since $\mathcal{L}(\mathbf{x}) = -1/n \cdot \mathbf{y}^\top \mathbf{U}\mathbf{x} + \tau/2\|\mathbf{x}\|_2^2$, we know that $\nabla\mathcal{L}(\mathbf{x}) = -1/n\mathbf{U}^\top\mathbf{y} + \tau\mathbf{x}$. We further have

$$\begin{aligned}
\|\nabla\mathcal{L}(\mathbf{x}^*)\|_\infty &= \left\|\frac{1}{n}\mathbf{U}^\top\mathbf{y} - \tau\mathbf{x}^*\right\|_\infty \\
&= \left\|\frac{1}{n}\mathbf{U}^\top\mathbf{y} - \tau\mathbf{x}^* + \gamma\mathbf{x}^* - \gamma\mathbf{x}^*\right\|_\infty \\
&= \left\|\left(\frac{1}{n}\mathbf{U}^\top\mathbf{y} - \gamma\mathbf{x}^*\right) + (\gamma\mathbf{x}^* - \tau\mathbf{x}^*)\right\|_\infty
\end{aligned} \tag{3.4.15}$$

For the last term in (3.4.15), we have

$$\begin{aligned}
\left\|\left(\frac{1}{n}\mathbf{U}^\top\mathbf{y} - \gamma\mathbf{x}^*\right) + (\gamma\mathbf{x}^* - \tau\mathbf{x}^*)\right\|_\infty &\leq \left\|\frac{1}{n}\mathbf{U}^\top\mathbf{y} - \gamma\mathbf{x}^*\right\|_\infty + \|\gamma\mathbf{x}^* - \tau\mathbf{x}^*\|_\infty \\
&\leq \left\|\frac{1}{n}\mathbf{U}^\top\mathbf{y} - \gamma\mathbf{x}^*\right\|_\infty + |\gamma - \tau|\|x^*\|_2 \\
&\leq C\sqrt{\frac{\log d}{n}} + |\gamma - \tau|,
\end{aligned} \tag{3.4.16}$$

where the last inequality holds with probability of at least $1 - e/d$ according to Lemma 3.4.5.

Therefore, for $i \in S^c$, we have that

$$\begin{aligned}
|(\nabla\mathcal{L}(\widehat{\mathbf{x}}_O))_i| &= |(\nabla\mathcal{L}(\mathbf{x}^*))_i| \\
&\leq |\nabla\mathcal{L}(\mathbf{x}^*)|_\infty \\
&\leq C\sqrt{\frac{\log d}{n}} + |\gamma - \tau| = \lambda.
\end{aligned} \tag{3.4.17}$$

For $i \in S$, we have $(\widehat{\mathbf{x}}_O)_i = 0$, so $|\widehat{Z}_O| \leq 1$. We can just set $(\widehat{Z}_O)_i = -(\nabla\mathcal{L}(\widehat{\mathbf{x}}_O))_i/\lambda$ for $i \in S^c$, we will have $(\nabla\mathcal{L}(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i = 0$ and hence

$$\sum_{i \in S^c} (\widehat{\mathbf{x}}_O - \mathbf{x}')_i \cdot (\nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda\widehat{Z}_O)_i = 0. \tag{3.4.18}$$

Adding (3.4.18) and (3.4.19), and taking maximum over $\|\mathbf{x}'\|_2 \leq 1$, we obtain (3.4.12). We thus have proved that the same optimality condition holds for $\widehat{\mathbf{x}}_O$. Now we are going to prove $\widehat{\mathbf{x}}_O = \widehat{\mathbf{x}}$. In fact, by Lemma 3.4.4 we have

$$\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}) \geq \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \langle\nabla\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O), \widehat{\mathbf{x}} - \widehat{\mathbf{x}}_O\rangle + \frac{\tau - \zeta_-}{2}\|\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_O\|_2^2, \tag{3.4.19}$$

$$\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) \geq \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}) + \langle \nabla \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}), \widehat{\mathbf{x}}_O - \widehat{\mathbf{x}} \rangle + \frac{\tau - \zeta_-}{2} \|\widehat{\mathbf{x}}_O - \widehat{\mathbf{x}}\|_2^2. \tag{3.4.20}$$

By the convexity of ℓ_1 norm, we have

$$\lambda \|\widehat{\mathbf{x}}\|_1 \geq \lambda \|\widehat{\mathbf{x}}_O\|_1 + \lambda \langle \widehat{\mathbf{x}} - \widehat{\mathbf{x}}_O, \widehat{Z}_O \rangle, \tag{3.4.21}$$

$$\lambda \|\widehat{\mathbf{x}}_O\|_1 \geq \lambda \|\widehat{\mathbf{x}}\|_1 + \lambda \langle \widehat{\mathbf{x}}_O - \widehat{\mathbf{x}}, \widehat{Z} \rangle. \tag{3.4.22}$$

We add (3.4.19) to (3.4.22) and obtain

$$0 \geq \langle \nabla \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}) + \lambda \widehat{Z}, \widehat{\mathbf{x}}_O - \widehat{\mathbf{x}} \rangle + \langle \nabla \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}_O) + \lambda \widehat{Z}_O, \widehat{\mathbf{x}} - \widehat{\mathbf{x}}_O \rangle + (\tau - \zeta_-) \|\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_O\|_2^2.$$

The first two terms are non-negative by optimality conditions of $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{x}}_O$, and we have $\tau - \zeta_- \geq 0$, hence we have $\|\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_O\|_2^2 = 0$, which means $\widehat{\mathbf{x}} = \widehat{\mathbf{x}}_O$. □

Remark 3.4.6. Theorem 3.4.3 indicates that our estimator is identical to oracle estimator under a magnitude assumption, while requiring no oracle information a priori. This will lead to exact support recovery directly. As we have mentioned, the oracle property is often a very strong criterion for estimators. For example, even [7] has achieved the best previous results on sample complexity in the noisy setting, there is still no guarantee of oracle property for their estimator.

Now we are in a position to analyze the error bound of oracle estimator, which is also the error bound of our estimator for strong signals. We will also show that the magnitude assumption is actually a weak assumption.

3.4.2 Sample Complexity of Our Estimator for Strong Signals

We now analyze the error bound of out method. Note that the error bound ϵ can be easily transformed into sample complexity with fixed s and d. Therefore, the error bound analysis is equivalent to sample complexity analysis.

We start with the following lemma characterizing the distance between true signal and the measurements.

Lemma 3.4.7. With a probability at least $1 - 1/d$, we have

$$\left\| \frac{\mathbf{U}_S^\top \mathbf{y}}{n} - \gamma \mathbf{x}_S^* \right\|_2 \leq C \sqrt{\frac{s}{n}}, \tag{3.4.23}$$

where C is a universal constant and S is the support of $\mathbf{x}^*$.

Proof of Lemma 3.4.7. We have $\mathbb{E}[\mathbf{u}_i y_i] = \gamma \mathbf{x}^*$ for $i = 1, \ldots, n$. Consider the j-th element of $1/n\mathbf{U}_S^\top \mathbf{y} - \gamma \mathbf{x}_S^*$, i.e.,

$$\left[\frac{1}{n}\mathbf{U}\mathbf{y} - \gamma \mathbf{x}^*\right]_j = \frac{1}{n}\sum_{i=1}^{n} u_{ij} y_i - \gamma x_j^*,$$

where $\mathbf{U} = [u_{ij}]$. Since $u_{ij} y_i$ is a sub-Gaussian random variable, according to Lemma 3.7.2, we have

$$\|u_{ij} y_i - \gamma x_j\|_{\psi_2} \leq 2\|u_{ij} y_i\|_{\psi_2}.$$

Since u_{ij} is sub-Gaussian random variable, we assume that $\|u_{ij}\|_{\psi_2} \leq C$ where $C > 0$ is an absolute constant. Since $y_i = \{-1, 1\}$, we have $\|u_{ij} y_i\|_{\psi_2} \leq C$. Thus, we have

$$\|u_{ij} y_i - \gamma x_j\|_{\psi_2} \leq C.$$

Let $\mathbf{a} = \mathbf{U}_S^\top \mathbf{y}/n - \gamma \mathbf{x}_S^*$. According to Lemma 3.7.4, for any $t > 0$, we have

$$\begin{aligned}
\mathbb{P}\Big(\sup_{\mathbf{u} \in \mathbb{S}^s} \big|\langle \mathbf{u}, \mathbf{a}\rangle\big| > t\Big) &\leq \mathbb{P}\Big(\sup_{\mathbf{u} \in N_\epsilon^s} \frac{1}{1-\epsilon}\big|\langle \mathbf{u}, \mathbf{a}\rangle\big| > t\Big) \\
&\leq \left(1 + \frac{2}{\epsilon}\right)^s \mathbb{P}\Big(\frac{1}{1-\epsilon}\big|\langle \mathbf{u}, \mathbf{a}\rangle\big| > t\Big). \qquad (3.4.24)
\end{aligned}$$

Setting $\epsilon = 1/2$ in the right hand side of (3.4.24), and invoking the Hoeffding's inequality in Lemma 3.7.3, we have

$$\begin{aligned}
\mathbb{P}\Big(\sup_{\mathbf{u} \in \mathbb{S}^s} \big|\langle \mathbf{u}, \mathbf{a}\rangle\big| > t\Big) &\leq 5^s \mathbb{P}(\big|\langle \mathbf{u}, \mathbf{a}\rangle\big| > 2t) \qquad (3.4.25) \\
&\leq 5^s e \cdot \exp\big(-Cnt^2\big) \\
&= e \cdot \exp\big(s \log 5 - Cnt^2\big). \qquad (3.4.26)
\end{aligned}$$

Setting $t = C\sqrt{s/n}$ in the right hand side of (3.4.25), we have with a probability at least $1 - \exp(1 - s)$ that

$$\left\| \frac{\mathbf{U}_S^\top \mathbf{y}}{n} - \gamma \mathbf{x}_S^* \right\|_2 \leq C\sqrt{\frac{s}{n}}.$$

This completes the proof. □

Now we are ready to present the main theorem on the sample complexity for strong signals of our proposed algorithm.

Theorem 3.4.8 (**Sample Complexity for Strong Signals**)**.** Under the same conditions of Theorem 3.4.3, we have with probability at least $1 - 1/d$ that

$$\|\widehat{\mathbf{x}} - \mathbf{x}^*\|_2 \leq \frac{C}{\gamma}\sqrt{\frac{s}{n}},$$

where C is a universal constant.

Proof. According to Lemma 3.4.1 and Theorem 3.4.3, our estimator and the oracle estimator have the same solution as (3.4.3). So it is sufficient to analyze (3.4.1). Since $\widehat{\mathbf{x}}_O$ is the optimal solution to (3.4.1), we have

$$0 \geq \left\langle \frac{\mathbf{U}_S^\top \mathbf{y}}{n}, (\widehat{\mathbf{x}}_O - \mathbf{x}^*)_S \right\rangle. \tag{3.4.27}$$

The right hand side of (3.4.27) can be further lower bounded by

$$\begin{aligned} 0 &\geq \left\langle \frac{\mathbf{U}_S^\top \mathbf{y}}{n}, (\widehat{\mathbf{x}}_O - \mathbf{x}^*)_S \right\rangle \\ &\geq \left\langle \frac{\mathbf{U}_S^\top \mathbf{y}}{n} - \gamma \mathbf{x}_S^*, (\widehat{\mathbf{x}}_O - \mathbf{x}^*)_S \right\rangle + \langle \gamma \mathbf{x}_S^*, (\widehat{\mathbf{x}}_O - \mathbf{x}^*)_S \rangle \\ &\geq -\left\| \frac{\mathbf{U}_S^\top \mathbf{y}}{n} - \gamma \mathbf{x}_S^* \right\|_2 \|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2 + \gamma\big(1 - \widehat{\mathbf{x}}_O^\top \mathbf{x}^*\big). \end{aligned} \tag{3.4.28}$$

Note that here we have used the fact that for $i \in S^c$,$(\widehat{\mathbf{x}}_O)_i = x_i^* = 0$. Since $\|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2^2 = 2\big(1 - \widehat{\mathbf{x}}_O^\top \mathbf{x}^*\big)$, invoking (3.4.28), we can obtain that

$$\|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2^2 \leq \frac{2}{\gamma}\left\| \frac{\mathbf{U}_S^\top \mathbf{y}}{n} - \gamma \mathbf{x}_S^* \right\|_2 \|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2,$$

which yields

$$\|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2 \leq \frac{2}{\gamma}\left\| \frac{\mathbf{U}_S^\top \mathbf{y}}{n} - \gamma \mathbf{x}_S^* \right\|_2 \leq \frac{C}{\gamma}\sqrt{\frac{s}{n}},$$

where the last inequality follows from Lemma 3.4.7 for some universal constant C. This completes the proof. □

Remark 3.4.9. From Theorem 3.4.8, we can see that the recovery error of our method for strong signals is just $O(\sqrt{s/n})$. We let $\epsilon = \sqrt{s/n}$ to get a sample complexity of $O(s/\epsilon^2)$. This is a significant improvement

from previous best result $O(s \log d/\epsilon^2)$.

Further more, we have $\|\widehat{\mathbf{x}}_O - \mathbf{x}^*\|_2 \leq C/\gamma\sqrt{s/n}$ with high probability. Therefore, we will only need

$$\min_{j \in S} |x_j^*| \geq \nu + C/\gamma\sqrt{s/n} \tag{3.4.29}$$

to get $\widehat{\mathbf{x}} = \widehat{\mathbf{x}}_O$ with probability at least $1 - 1/d$. This is a weak assumption, since one-bit measurements can be acquired at very high rates. When n is very large, the right-hand side of (3.4.29) will converge to a constant ν. Note that for the oracle estimator, the error bound is always of the order of $O(\sqrt{s/n})$, which does not depend on the magnitude assumption. We only need the magnitude assumption to make $\widehat{\mathbf{x}} = \widehat{\mathbf{x}}_O$, and thus enjoy the improved sample complexity.

3.4.3 Sample Complexity for General Signals

We now focus on the case of general signals, where the magnitude assumption does not hold necessarily. For the sake of simplicity, we focus on our estimator in (3.3.2) with $\tau = 0$, and for $\tau > 0$ it works in a similar way.

We start with the following lemma, which characterizes the curvature of the loss function in the ball $\|\mathbf{x}\|_2 \leq 1$.

Lemma 3.4.10. For any $\mathbf{x}$ where $\|\mathbf{x}\|_2 \leq 1$, we have

$$\frac{\langle \mathbb{E}[\mathbf{U}^\top \mathbf{y}], \mathbf{x}^* - \mathbf{x} \rangle}{\gamma} \geq \frac{1}{2}\|\mathbf{x}^* - \mathbf{x}\|_2^2.$$

Proof. We have

$$\begin{aligned}
\langle \mathbb{E}[\mathbf{U}^\top \mathbf{y}], \mathbf{x}^* - \mathbf{x} \rangle &= \langle \gamma \mathbf{x}^*, \mathbf{x}^* - \mathbf{x} \rangle \\
&= \langle \gamma \mathbf{x}^* - \gamma \mathbf{x} + \gamma \mathbf{x}, \mathbf{x}^* - \mathbf{x} \rangle \\
&= \gamma \|\mathbf{x}^* - \mathbf{x}\|_2^2 + \gamma \langle \mathbf{x}, \mathbf{x}^* - \mathbf{x} \rangle.
\end{aligned} \tag{3.4.30}$$

On the other hand, we have

$$\begin{aligned}
\langle \mathbf{x}, \mathbf{x}^* - \mathbf{x} \rangle &= \mathbf{x}^\top \mathbf{x}^* - \|\mathbf{x}\|_2^2 \geq \mathbf{x}^\top \mathbf{x}^* - \frac{1}{2} - \frac{1}{2}\|\mathbf{x}\|_2^2 \\
&= \mathbf{x}^\top \mathbf{x}^* - \frac{1}{2}\|\mathbf{x}^*\|_2^2 - \frac{1}{2}\|\mathbf{x}\|_2^2 \\
&= -\frac{1}{2}\|\mathbf{x} - \mathbf{x}^*\|_2^2.
\end{aligned} \tag{3.4.31}$$

Substituting (3.4.31) into (3.4.30), we obtain

$$\langle \mathbb{E}[\mathbf{U}^\top \mathbf{y}], \mathbf{x}^* - \mathbf{x} \rangle \geq \frac{\gamma}{2} \|\mathbf{x} - \mathbf{x}^*\|_2^2,$$

which completes the proof. □

Theorem 3.4.11 (**Sample Complexity for General Signals**). Suppose the nonconvex penalty $\mathcal{G}_{\lambda,b}(\mathbf{x}) = \sum_{i=1}^d g_{\lambda,b}(x_i)$ satisfies conditions C2, C3 and C4. For any local optimal solution $\widehat{\mathbf{x}}$ to (3.3.2) with $\tau = 0$, $\lambda = C\sqrt{\frac{\log d}{n}}$ and $\zeta_- < \frac{\gamma}{2}$, we have with probability at least $1 - 1/d$ that

$$\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2 \leq \underbrace{\frac{2C}{\gamma - 2\zeta_-}\sqrt{\frac{s_1}{n}}}_{S_1:|\mathbf{x}_i^*| \geq \nu} + \underbrace{\frac{6C\sqrt{s_2}}{\gamma - 2\zeta_-}\sqrt{\frac{\log d}{n}}}_{S_2:0<|\mathbf{x}_i^*|<\nu}, \tag{3.4.32}$$

where C is a universal constant.

Proof. We denote the subgradient by $Z \in \partial\|\mathbf{x}\|_1$, $Z^* \in \partial\|\mathbf{x}^*\|_1$ and $\widehat{Z} \in \partial\|\widehat{\mathbf{x}}\|_1$. We know by the optimality condition of $\widehat{\mathbf{x}}$ that

$$\max_{\|\mathbf{x}'\| \leq 1} \langle \widehat{\mathbf{x}} - \mathbf{x}', \nabla \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}} + \lambda \widehat{Z}) \rangle \leq 0.$$

According to Lemma 3.4.4 with $\tau = 0$, we know that

$$\widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}) \geq \widetilde{\mathcal{L}}_\lambda(\mathbf{x}^*) + \langle \nabla \widetilde{\mathcal{L}}_\lambda(\mathbf{x}^*), \widehat{\mathbf{x}} - \mathbf{x}^* \rangle - \frac{\zeta_-}{2}\|\widehat{\mathbf{x}} - \mathbf{x}^*\|_2^2, \tag{3.4.33}$$

$$\widetilde{\mathcal{L}}_\lambda(\mathbf{x}^*) \geq \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}) + \langle \nabla \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}), \mathbf{x}^* - \widehat{\mathbf{x}} \rangle - \frac{\zeta_-}{2}\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2^2. \tag{3.4.34}$$

By the convexity of ℓ_1 norm, we have

$$\lambda\|\widehat{\mathbf{x}}\|_1 \geq \lambda\|\mathbf{x}^*\|_1 + \lambda\langle \widehat{\mathbf{x}} - \mathbf{x}^*, Z^* \rangle, \tag{3.4.35}$$

$$\lambda\|\mathbf{x}^*\|_1 \geq \lambda\|\widehat{\mathbf{x}}\|_1 + \lambda\langle \mathbf{x}^* - \widehat{\mathbf{x}}, \widehat{Z} \rangle. \tag{3.4.36}$$

Adding up (3.4.33) to (3.4.36) yields

$$0 \geq \langle \nabla \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}) + \lambda \widehat{Z}, \mathbf{x}^* - \widehat{\mathbf{x}} \rangle + \langle \nabla \widetilde{\mathcal{L}}_\lambda(\mathbf{x}^*) + \lambda Z^*, \widehat{\mathbf{x}} - \mathbf{x}^* \rangle - \zeta_-\|\widehat{\mathbf{x}} - \mathbf{x}^*\|_2^2.$$

By the optimality condition of $\widehat{\mathbf{x}}$, we know that

$$\langle \nabla \widetilde{\mathcal{L}}_\lambda(\widehat{\mathbf{x}}) + \lambda \widehat{Z}, \mathbf{x}^* - \widehat{\mathbf{x}} \rangle \geq 0.$$

Therefore,

$$-\zeta_-\|\widehat{\mathbf{x}}-\mathbf{x}^*\|_2^2 \le \langle \nabla\widetilde{\mathcal{L}}_\lambda(\mathbf{x}^*)+\lambda Z^*, \mathbf{x}^*-\widehat{\mathbf{x}}\rangle. \tag{3.4.37}$$

According to Lemma 3.4.10, we have

$$\frac{\gamma}{2}\|\mathbf{x}^*-\widehat{\mathbf{x}}\|_2^2 \le \langle \mathbb{E}[\mathbf{U}^\top\mathbf{y}], \mathbf{x}^*-\widehat{\mathbf{x}}\rangle = \langle \gamma\mathbf{x}^*, \mathbf{x}^*-\widehat{\mathbf{x}}\rangle. \tag{3.4.38}$$

Adding (3.4.37) and (3.4.38), we get

$$\begin{aligned}\left(\frac{\gamma}{2}-\zeta_-\right)\|\mathbf{x}^*-\widehat{\mathbf{x}}\|_2^2 &\le \left\langle \gamma\mathbf{x}^*+\nabla\widetilde{\mathcal{L}}_\lambda(\mathbf{x}^*)+\lambda Z^*, \mathbf{x}^*-\widehat{\mathbf{x}}\right\rangle \\ &= \langle \gamma\mathbf{x}^*+\nabla\mathcal{L}(\mathbf{x}^*)+\nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*)+\lambda Z^*, \mathbf{x}^*-\widehat{\mathbf{x}}\rangle \\ &\le \sum_{i=1}^d |(\gamma\mathbf{x}^*+\nabla\mathcal{L}(\mathbf{x}^*)+\nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*)+\lambda Z^*)_i|\cdot|(\mathbf{x}^*-\widehat{\mathbf{x}})_i|.\end{aligned} \tag{3.4.39}$$

In the following, we will decompose the summation in (3.4.39) into three parts: $i\in S^c$, $i\in S_1$ and $i\in S_2$, where $S_1=\{i\in S:|\mathbf{x}_i^*|\ge\nu\}$ and $S_2=\{i\in S:|\mathbf{x}_i^*|<\nu\}$.

For $i\in S^c$, by condition C3, we will have

$$(\nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*))_i = h'_{\lambda,b}(\mathbf{x}_i^*) = h'_{\lambda,b}(0) = 0,$$

and

$$(\mathcal{H}_{\lambda,b}(\mathbf{x}^*))_i = h_{\lambda,b}(\mathbf{x}_i^*) = h_{\lambda,b}(0) = 0.$$

We also have

$$\begin{aligned}\max_{i\in S^c}|(\gamma\mathbf{x}^*+\nabla\mathcal{L}(\mathbf{x}^*))_i| &= \max_{i\in S^c}\left|\left(\gamma\mathbf{x}^*-\frac{1}{n}\mathbf{U}^\top\mathbf{y}\right)_i\right| \\ &\le \left\|\gamma\mathbf{x}^*-\frac{1}{n}\mathbf{U}^\top\mathbf{y}\right\|_\infty \\ &\le C\sqrt{\frac{\log d}{n}} \\ &= \lambda,\end{aligned} \tag{3.4.40}$$

where the second inequality comes from Lemma 3.4.5. Hence, we have

$$\max_{i\in S^c}|(\gamma\mathbf{x}^*+\nabla\mathcal{L}(\mathbf{x}^*)+\nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*))_i| \le \lambda.$$

Since $Z^* \in \partial\|\mathbf{x}^*\|_1$, we will have $\lambda Z_i \in [-\lambda, \lambda]$. That is, we can always find a $Z^* \in \partial\|\mathbf{x}^*\|_1$ such that for any $i \in S^c$,

$$|(\gamma\mathbf{x}^* + \nabla\mathcal{L}(\mathbf{x}^*) + \nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*) + \lambda Z^*)_i| = 0.$$

Therefore, we will have

$$\sum_{i\in S^c} |(\gamma\mathbf{x}^* + \nabla\mathcal{L}(\mathbf{x}^*) + \nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*) + \lambda Z^*)_i| \cdot |(\mathbf{x}^* - \widehat{\mathbf{x}})_i| = 0. \tag{3.4.41}$$

For the second part in (3.4.39), we have $|\mathbf{x}_i^*| \geq \nu$ for $i \in S_1 \subset S$. By condition C1, we have

$$(\nabla\mathcal{H}_{\lambda,b_\lambda}(\mathbf{x}^*) + \lambda Z^*)_i = g'_{\lambda,b}(\mathbf{x}_i^*) = 0,$$

which means that

$$\begin{aligned}\sum_{i\in S_1} |(\gamma\mathbf{x}^* + \nabla\mathcal{L}(\mathbf{x}^*) + \nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*) + \lambda Z^*)_i| \cdot |(\mathbf{x}^* - \widehat{\mathbf{x}})_i| &= \sum_{i\in S_1} |(\gamma\mathbf{x}^* + \nabla\mathcal{L}(\mathbf{x}^*))_i| \cdot |(\mathbf{x}^* - \widehat{\mathbf{x}})_i| \\ &= \sum_{i\in S_1} \left|\left(\gamma\mathbf{x}^* - \frac{1}{n}\mathbf{U}^\top\mathbf{y}\right)_i\right| \cdot |(\mathbf{x}^* - \widehat{\mathbf{x}})_i| \\ &\leq \left\|\left(\gamma\mathbf{x}^* - \frac{1}{n}\mathbf{U}^\top\mathbf{y}\right)_{S_1}\right\|_2 \cdot \|(\mathbf{x}^* - \widehat{\mathbf{x}})_{S_1}\|_2, \end{aligned} \tag{3.4.42}$$

where the first inequality follows from Cauchy-Schwartz inequality. According to Lemma 3.4.7, we know that

$$\left\|\gamma\mathbf{x}^*_{S_1} - \frac{1}{n}\mathbf{U}^\top_{S_1}\mathbf{y}\right\|_2 \leq C\sqrt{\frac{s_1}{n}}. \tag{3.4.43}$$

Therefore, substituting (3.4.43) into (3.4.42), we obtain

$$\sum_{i\in S_1} |(\gamma\mathbf{x}^* + \nabla\mathcal{L}(\mathbf{x}^*) + \nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*) + \lambda Z^*)_i| \cdot |(\mathbf{x}^* - \widehat{\mathbf{x}})_i| \leq C\sqrt{\frac{s_1}{n}}\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2. \tag{3.4.44}$$

For the third part in (3.4.39), we have $|\mathbf{x}_i^*| < \nu$ for $i \in S_2 \subset S$. By Lemma 3.4.5, we have

$$\max_{i\in S_2}\left|\left(\gamma\mathbf{x}^* - \frac{1}{n}\mathbf{U}^\top\mathbf{y}\right)_i\right| \leq \left\|\gamma\mathbf{x}^* - \frac{1}{n}\mathbf{U}^\top\mathbf{y}\right\|_\infty \leq C\sqrt{\frac{\log d}{n}} = \lambda.$$

By regularity condition C4, we have

$$\max_{i\in S_2}|(\nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*))_i| = \max_{i\in S_2}|h'_{\lambda,b}(\mathbf{x}_i^*)| \le \lambda$$

We also have $|Z_i^*| \le 1$ since $Z^* \in \partial\|\mathbf{x}^*\|_1$, therefore, for any $i \in S_2$,

$$\begin{aligned}|(\gamma\mathbf{x}^* + \nabla\mathcal{L}(\mathbf{x}^*) + \nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*) + \lambda Z^*)_i| &\le |(\gamma\mathbf{x}^* + \nabla\mathcal{L}(\mathbf{x}^*))_i| + |(\mathcal{H}_{\lambda,b}(\mathbf{x}^*))_i| + |(\lambda Z^*)_i| \\ &\le \left\|\gamma\mathbf{x}^* - \frac{1}{n}\mathbf{U}^\top\mathbf{y}\right\|_\infty + |(\mathcal{H}_{\lambda,b}(\mathbf{x}^*))_i| + \lambda|Z_i^*| \\ &\le 3\lambda.\end{aligned} \tag{3.4.45}$$

This yields that

$$\begin{aligned}\sum_{i\in S_2}|(\gamma\mathbf{x}^* + \nabla\mathcal{L}(\mathbf{x}^*) + \nabla\mathcal{H}_{\lambda,b}(\mathbf{x}^*) + \lambda Z^*)_i| \cdot |(\mathbf{x}^* - \widehat{\mathbf{x}})_i| &\le 3\lambda\sum_{i\in S_2}|(\mathbf{x}^* - \widehat{\mathbf{x}})_i| \\ &\le 3\lambda\sqrt{s_2}\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2.\end{aligned} \tag{3.4.46}$$

We sum (3.4.41), (3.4.44) and (3.4.46), substitute it into (3.4.39) and get

$$\left(\frac{\gamma}{2} - \zeta_-\right)\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2^2 \le C\sqrt{\frac{s_1}{n}}\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2 + 3\lambda\sqrt{s_2}\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2, \tag{3.4.47}$$

which is equivalent to

$$\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2 \le \frac{2C}{\gamma - 2\zeta_-}\sqrt{\frac{s_1}{n}} + \frac{6\sqrt{s_2}\lambda}{\gamma - 2\zeta_-}.$$

That is

$$\|\mathbf{x}^* - \widehat{\mathbf{x}}\|_2 \le \frac{2C}{\gamma - 2\zeta_-}\sqrt{\frac{s_1}{n}} + \frac{6C\sqrt{s_2}}{\gamma - 2\zeta_-}\sqrt{\frac{\log d}{n}}.$$

□

Remark 3.4.12. Theorem 3.4.11 characterizes the sample complexity of the proposed estimator for general signals where the magnitude assumption does not hold necessarily. We can see that for strong signals, we have $|\mathbf{x}_i^*| \ge \nu$ for all $i \in S$, thus $s_2 = 0$. Then our recovery error is just $O(\sqrt{s/n})$, which is equivalent to a sample complexity of $O(s/\epsilon^2)$. This is also consistent to the results in Theorem 3.4.8.

In the worst case, $|\mathbf{x}_i^*| < \nu$ for all $i \in S$, thus $s_2 = s$, and our recovery error is $O(\sqrt{s\log d/n})$. This yields the worst sample complexity of $O(s\log d/\epsilon^2)$. For more general case, the sample complexity is between $O(s\log d/\epsilon^2)$ and $O(s/\epsilon^2)$, which is also a significant improvement.

3.5 Experiments

In this section, I will present the numerical experiments to backup my theory. I apply the proposed algorithm to the recovery of both general and strong signals.

For each recovery task, we will tune C by cross validation and select λ according to Theorem 3.4.3 for strong signals and Theorem 3.4.11 for general signals. For each parameter setting, we present the average results of 100 trials of our method and four other methods:

- Passive: the passive algorithm proposed in [7], the best previous result on sample complexity.
- Convex: the convex programming approach proposed in [10].
- BIHT and BIHT-ℓ_2 proposed in [6]

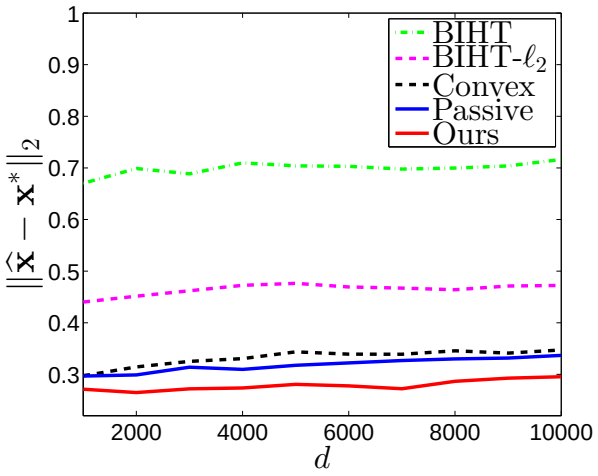
3.5.1 Approximate Vector Recovery for General Signals

In this subsection, we will show our experimental results on general signals, i.e., no magnitude assumption guaranteed. The support of the signal vector is uniformly randomly selected from the entries, and the entry values are drawn from a standard normal distribution. The elements in the matrix $\mathbf{U}$ are also drawn from standard normal distribution and are independent from the signal $\mathbf{x}^*$. We choose the noisy setting in [10] by flipping the signs of measurements with a probability of 0.1.

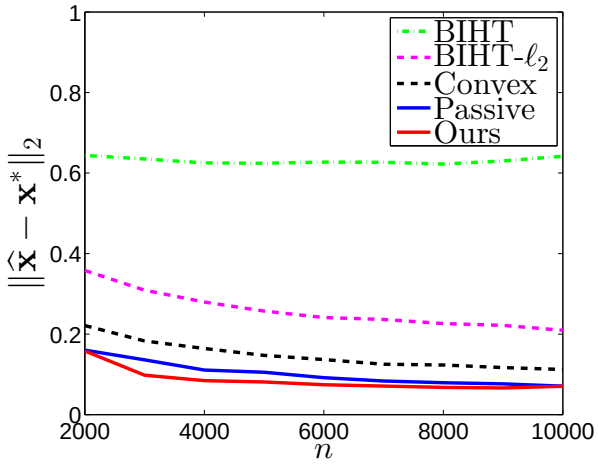
Figure 3.1(a) shows the recovery error against the dimensionality of signals d. We can see that our proposed method outperforms all the other algorithms with a remarkable margin. As the dimensionality of signal d goes up, the recovery error grows slowly, because the dependency on d is logarithmic by Theorem 3.4.11. We can also see that in this noisy setting, the more vulnerable BIHT and BIHT-ℓ_2 consistently perform worse than the other methods.

Figure 3.1(b) shows the recovery error against the number of measurements n. Our method consistently achieves the best performance. The passive algorithm also performs reasonably well, but our method outperforms it in a wide range of n.

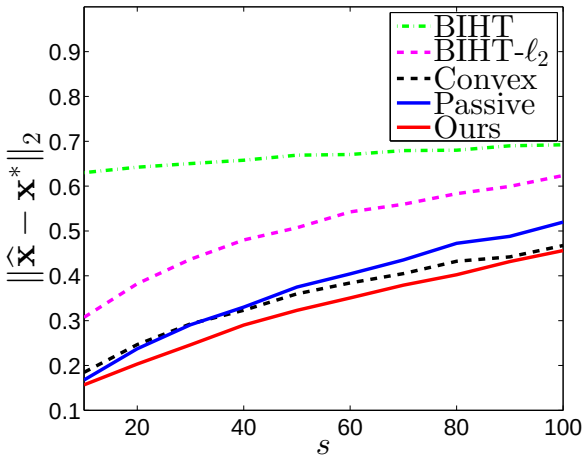
Figure 3.1(c) shows the recovery error against the sparsity of signals s. We can see that for all the algorithms except BIHT, the error goes up quickly when s becomes larger. Our algorithm is still consistently the best among all. Note that the dependency on s is not logarithmic, therefore, the error grows much faster than the case of varying d. We choose number of measurements $n = 3000$ here, which is larger than the signal dimension d. This is practical in one-bit compressed sensing, because the one-bit measurements can be generated at very high rates. To sum up, our method can improve recovery accuracy in different parameter settings even with noise.



(a) $s = 10$, $n = 1000$.



(b) $s = 10$, $d = 1000$.



(c) $d = 1000$, $n = 3000$.

Figure 3.1: Recovery error for general signals

3.5.2 Approximate Vector Recovery for Strong Signals

Now we present results of our recovery algorithm for strong signals. We will first generate unit sparse signals with random support, and set all nonzero entries to $1/\sqrt{s}$. Noise is added in the same way with section 3.5.1.

Figure 3.2 shows the recovery error of strong signals. According to Theorem 3.4.8, our error rate does not depend on dimensionality d, which is verified by the results. Our recovery error stays on the same level, while the errors of all the other algorithms go up with increasing d. Note that the error of BIHT is much higher than the other algorithms. For better illustration and scaling the behavior of the other methods, we omit it in the figure here.

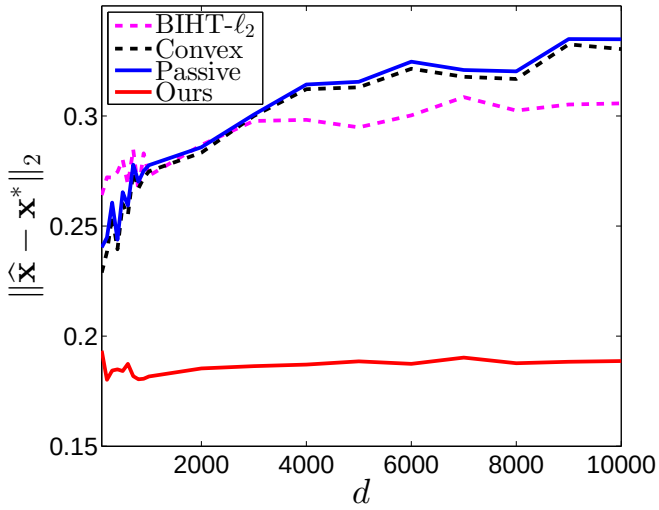


Figure 3.2: Recovery error of strong signals against d when $s = 10$, $n = 1000$.

3.5.3 Support Recovery

We are now going to investigate the problem of support recovery. According to Theorem 3.4.3, our estimator enjoys oracle property for strong signals. We generate the signals in the same way as section 3.5.2 and present the F_1 score of support recovery in different d and n settings. F_1 score is defined as the harmonic mean of precision and recall,

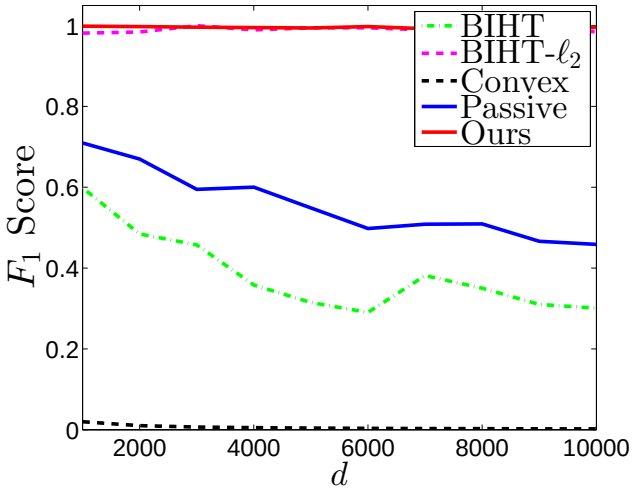
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \ \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \ F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

where

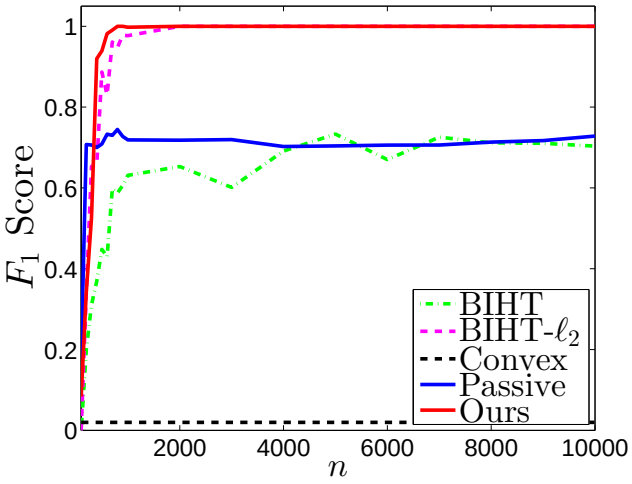
$$\text{TP} = \sum_{i=1}^{d} \mathbf{1}(\widehat{\mathbf{x}}_i \neq 0, \mathbf{x}_i^* \neq 0), \quad \text{FP} = \sum_{i=1}^{d} \mathbf{1}(\widehat{\mathbf{x}}_i \neq 0, \mathbf{x}_i^* = 0),$$
$$\text{TN} = \sum_{i=1}^{d} \mathbf{1}(\widehat{\mathbf{x}}_i = 0, \mathbf{x}_i^* = 0), \quad \text{FN} = \sum_{i=1}^{d} \mathbf{1}(\widehat{\mathbf{x}}_i = 0, \mathbf{x}_i^* \neq 0).$$

Note that our method is different from best previous work on support recovery. We do not need to construct specific measurement matrix as [5, 9], nor do we depend on dynamic range or adaption of the measurement process as [8]. Therefore, their methods are not directly comparable with ours.

Figure 3.3(a) shows the F_1 score against signal dimension d. We can see that as the assumption in Theorem 3.4.3 is satisfied, our algorithm can achieve exact support recovery with very high probability. Our method and BIHT-ℓ_2 outperform the other algorithms with notable margins. In addition, Theorem 3.4.3



(a) $s = 10$, $n = 1000$.



(b) $s = 10$, $d = 1000$.

Figure 3.3: F_1 score for support recovery

indicates that the support recovery of our method does not depend on d, which is also validated by the experiments. While for the other algorithms, the performance of the passive algorithm drops significantly as d goes up; BIHT is not effective either, nor can it achieve a stable performance. Note that for the convex optimization method, there is no ℓ_0 constraints on the signal. Therefore, most of the entries in the estimator are nonzero, resulting in very low precision. This explains the observation that convex optimization method always have a F1-score close to zero.

In Figure 3.3(b), we can find the F_1 score against number of measurements n. For the same reason, the convex optimization method still suffers very low F_1 score close to 0. For the other four methods, when there are not enough measurements, they perform poorly on support recovery. As the number of measurements goes up, the passive algorithm is the fatest to boost the performance. However, the F_1 score will stop increasing around 0.7 in spite of the increase of measurements. For BIHT, the performance is less stable, but F_1 score will still converge around 0.7 with increasing measurements. Compared with the passive algorithm, our algorithm needs a bit more measurements to converge in terms of F_1 score. Moreover, when n is larger than 500, our algorithm can achieve very good performance, almost recover the support with probability 1. BIHT-ℓ_2 has a similar behavior as our algorithm with enough measurements, but our method requires fewer measurements.

3.5.4 Oracle Property

We will further study the oracle property of our estimator. We plot the difference between proposed estimator and the oracle estimator in (3.4.1). By Theorem 3.4.3, the two should be the same with high probability. In

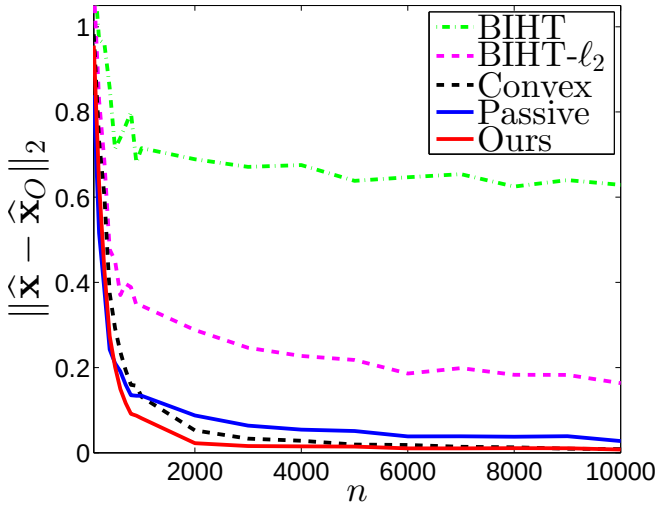


Figure 3.4: Difference between estimators and oracle estimators against n when $s = 10$, $d = 1000$.

Figure 3.4, we can see that when the number of measurements goes up, the difference between our estimator and oracle estimator converges to zero very quickly. For BIHT and BIHT-ℓ_2, the differences are large; for the passive algorithm, the difference is still discernable, and the support recovery is not satisfying; for the convex optimization algorithm, although the norm of the difference is converging, it cannot recover the support. Therefore, our estimator is the only one that enjoys oracle property.

3.6 Summary

We proposed a novel algorithm [14] for the problem of one-bit compressed sensing, which is able to achieve both vector recovery and support recovery with strong theoretical guarantees. We introduce the nonconvex sparsity-inducing penalty functions to this problem for the first time.

More specifically, our main contributions are summarized as follows:

- We propose to incorporate sparsity-inducing penalty functions into one-bit compressed sensing, and derive an algorithm to efficiently solve the resulting problem. To the best of our knowledge, this is the first work on one-bit compressed sensing that utilizes nonconvex penalty functions.

- We prove that our proposed method improves sample complexity from previous best results $O(s \log d/\epsilon^2)$ to $O(s/\epsilon^2)$ for strong signals. And for general signals, our algorithm attains a sample complexity between $O(s \log d/\epsilon^2)$ and $O(s/\epsilon^2)$.

- We prove that our proposed method can achieve exact support recovery of the signal under mild

magnitude assumptions on the signal.

- We verify the effectiveness of our method by thorough numerical experiments.

Even we have not specifically focused on the computational complexity of the proposed algorithm, it is worth noting that our algorithm only involves sorting and analytic form calculation. So it is still very efficient albeit looking involved.

3.7 Proofs and Technical Details

We provide the detailed proofs of the theoretical results in Section 3.4.

3.7.1 Proof of Lemma 3.3.1

Proof. First, we can easily see that $\widehat{x}$ and y must not have opposite signs. If $y = 0$, obviously there is $\widehat{x} = 0$. Assume $y > 0$ and $\widehat{x} < 0$, then it is easy to find that $x' = -\widehat{x}$ has smaller objective function value, which leads to contradiction.

In the case of $b \leq 1$ and without the loss of generality we assume $y > 0$ and $x \geq 0$. Then

$$\frac{1}{2}(x-y)^2 + g_{\lambda,b}(|x|) = \begin{cases} \frac{1}{2}(x-y)^2 + \lambda x - \frac{x^2}{2b}, & \text{if } 0 \leq x \leq b\lambda, \\ \frac{1}{2}(x-y)^2 + \frac{b\lambda^2}{2}, & \text{if } x > b\lambda. \end{cases} \tag{3.7.1}$$

- When $0 \leq x \leq b\lambda$,

$$\frac{1}{2}(x-y)^2 + g_{\lambda,b}(|x|) = \frac{1}{2}(x-y)^2 + \lambda x - \frac{x^2}{2b} = -(\frac{1}{2b} - \frac{1}{2})x^2 + (\lambda - y)x + \frac{1}{2}y^2. \tag{3.7.2}$$

If $b = 1$, then we get

$$\widehat{x} = \begin{cases} 0, & \text{if } 0 \leq y \leq \lambda, \\ b\lambda, & \text{if } y > \lambda. \end{cases} \tag{3.7.3}$$

If $0 < b < 1$, this is a quadratic objective function with negative quadratic term coefficient. So we can just compare the function values of $x = 0$ and $x = b\lambda$ to decide $\widehat{x}$. With some derivation, we have in this

case

$$\widehat{x} = \begin{cases} 0, & \text{if } 0 \le y \le \dfrac{(1+b)\lambda}{2}, \\ b\lambda, & \text{if } y > \dfrac{(1+b)\lambda}{2}. \end{cases} \tag{3.7.4}$$

- When $x > b\lambda$,

$$\frac{1}{2}(x-y)^2 + g_{\lambda,b}(|x|) = \frac{1}{2}(x-y)^2 + \frac{b\lambda^2}{2}. \tag{3.7.5}$$

We can easily get

$$\widehat{x} = \begin{cases} b\lambda, & \text{if } 0 \le y \le b\lambda, \\ y, & \text{if } y > b\lambda. \end{cases} \tag{3.7.6}$$

To sum the two cases up by comparing the function values, we have

If $b = 1$,

$$\widehat{x} = \begin{cases} 0, & \text{if } 0 \le y \le \lambda, \\ y, & \text{if } y > \lambda. \end{cases} \tag{3.7.7}$$

If $0 < b < 1$, we need to compare the function values of $x = 0$ and $x = b\lambda$ when $0 \le y \le b\lambda$, the function values of $x = 0$ and $x = y$ when $b\lambda < y \le (1+b)\lambda/2$ and the function values of $x = y$ and $x = b\lambda$ when $y > (1+b)\lambda/2$. With some derivation, we have

$$\widehat{x} = \begin{cases} 0, & \text{if } 0 \le y \le \sqrt{b}\lambda, \\ y, & \text{if } y > \sqrt{b}\lambda. \end{cases} \tag{3.7.8}$$

We can sum up the two cases of $0 < b < 1$ and $b = 1$ and get when $0 < b \le 1$,

$$\widehat{x} = \begin{cases} 0, & \text{if } 0 \le y \le \sqrt{b}\lambda, \\ y, & \text{if } y > \sqrt{b}\lambda. \end{cases} \tag{3.7.9}$$

The above derivation can be directly applied to the case of $y < 0$, by symmetry we can get the final expression

of $\widehat{x}$

$$\widehat{x} = \begin{cases} 0, & \text{if } 0 \le |y| \le \sqrt{b}\lambda, \\ y, & \text{if } |y| > \sqrt{b}\lambda. \end{cases} \tag{3.7.10}$$

This completes the proof of Lemma 3.3.1. □

3.7.2 Proof of Lemma 3.3.2

Proof. For Lemma 3.3.2, we have

$$\begin{aligned} \frac{1}{2}(x-y)^2 + g_{\lambda,b}(|x|) + \frac{\tau}{2}x^2 &= \frac{1+\tau}{2}x^2 - xy + \frac{y^2}{2} + g_{\lambda,b}(|x|) \\ &= \frac{1+\tau}{2}\left(x - \frac{y}{1+\tau}\right)^2 + g_{\lambda,b}(|x|) + \frac{\tau y^2}{2(1+\tau)} \\ &= \frac{1+\tau}{2}\left[\left(x - \frac{y}{1+\tau}\right)^2 + g_{\frac{2\lambda}{1+\tau},\frac{(1+\tau)b}{2}}(|x|)\right] + \frac{\tau y^2}{2(1+\tau)}. \end{aligned} \tag{3.7.11}$$

The last term of (3.7.11) does not depend on x, and the optimization of the first term can be directly deduced from the proof of Lemma 3.3.1. □

3.7.3 Derivation of Algorithm 1

In this case, for all i that $|v_i| > \lambda\sqrt{2\mu b}$, we will have $|v_i| > 2\mu\lambda b$. We have

$$\begin{aligned} f(\mu) &= 2\mu\Bigg\{ \sum_{i:|v_i|\ge\lambda\sqrt{2\mu b}} p_{\lambda/(2\mu),2\mu b}\Big(\frac{|v_i|}{2\mu}\Big) + \sum_{i:|v_i|<\lambda\sqrt{2\mu b}} \Big[\frac{1}{2}\Big(-\frac{v_i}{2\mu}\Big)^2\Big]\Bigg\} - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu \\ &= 2\mu\Bigg\{ \sum_{i:|v_i|\ge\lambda\sqrt{2\mu b}} \frac{1}{2}(2\mu b)\left(\frac{\lambda}{2\mu}\right)^2 + \sum_{i:|v_i|<\lambda\sqrt{2\mu b}} \Big[\frac{1}{2}\Big(-\frac{v_i}{2\mu}\Big)^2\Big]\Bigg\} - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu \\ &= \sum_{i:|v_i|\ge\lambda\sqrt{2\mu b}} \frac{1}{2}b\lambda^2 + \sum_{i:|v_i|<\lambda\sqrt{2\mu b}} \frac{v_i^2}{4\mu} - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu \\ &= \sum_{i:|v_i|\ge\lambda\sqrt{2\mu b}} \Big(\frac{1}{2}b\lambda^2 - \frac{v_i^2}{4\mu}\Big) - \mu. \end{aligned} \tag{3.7.12}$$

To find the optimal dual solution μ^*, we would first sort $|v_i|$ in an ascending order, i.e., $|v_{(1)}| \le |v_{(2)}| \le ... \le |v_{(n)}|$. Since the feasible region of $(0,\infty)$, we would cut this into $n+1$ intervals, $(0, v_{(1)}^2/2b\lambda^2]$, $(v_{(1)}^2/2b\lambda^2, v_{(2)}^2/2b\lambda^2]$, ... , $(v_{(n)}^2/2b\lambda^2, \infty)$. We define $v_{(0)} = 0$ and $v_{(n+1)} = \infty$ to cover the boundary cases.

Within each interval $(v_{(i)}^2/2b\lambda^2, v_{(i+1)}^2/2b\lambda^2]$, we have

$$\begin{aligned} f(\mu) &= \sum_{i:|v_i|\geq\lambda\sqrt{2\mu b}} \Big(\frac{1}{2}b\lambda^2 - \frac{v_i^2}{4\mu}\Big) - \mu = \sum_{j=i}^{n} \Big(\frac{1}{2}b\lambda^2 - \frac{v_{(j)}^2}{4\mu}\Big) - \mu \\ &= \frac{(n-i+1)b\lambda^2}{2} - \frac{\sum_{j=i}^{n} v_{(j)}^2}{4\mu} - \mu. \end{aligned} \tag{3.7.13}$$

The optimal value for μ should be $\sqrt{\sum_{j=i}^{n} v_{(j)}^2}/2$ if there are no other constraints, however, here we have $\mu \in (v_{(i)}^2/2b\lambda^2, v_{(i+1)}^2/2b\lambda^2]$. Therefore, the optimal μ in this interval, μ_i, should be

$$\mu_i = \begin{cases} \sqrt{\sum_{j=i}^{n} v_{(j)}^2}/2, & \text{if } \sqrt{\sum_{j=i}^{n} v_{(j)}^2}/2 \in (v_{(i)}^2/2b\lambda^2, v_{(i+1)}^2/2b\lambda^2], \\ v_{(i+1)}^2/2b\lambda^2, & \text{if } \sqrt{\sum_{j=i}^{n} v_{(j)}^2}/2 \notin (v_{(i)}^2/2b\lambda^2, v_{(i+1)}^2/2b\lambda^2]. \end{cases}$$

After we have got all $\mu_i, i = 0, 1, ..., n$, we will find the one that maximizes $f(\mu)$.

3.7.4 Derivation of Algorithm 2

In this case, following (3.3.7) and using Lemma 3.3.1 , we get

$$\begin{aligned} f(\mu) &= 2\mu\Big\{ \sum_{i:|v_i|\geq 2\mu\lambda b} p_{\lambda/(2\mu),2\mu b}\Big(\frac{|v_i|}{2\mu}\Big) + \sum_{i:|v_i|<2\mu\lambda b} \Big[\frac{1}{2}\Big(\frac{S(\frac{v_i}{2\mu},\frac{\lambda}{2\mu})}{1-1/(2\mu b)} - \frac{v_i}{2\mu}\Big)^2 + p_{\lambda/(2\mu),2\mu b}\Big(\Big|\frac{S(\frac{|v_i|}{2\mu},\frac{\lambda}{2\mu})}{1-1/(2\mu b)}\Big|\Big)\Big]\Big\} \\ &\quad - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu \\ &= 2\mu\Big\{ \sum_{i:|v_i|\geq 2\mu\lambda b} p_{\lambda/(2\mu),2\mu b}\Big(\frac{|v_i|}{2\mu}\Big) + \sum_{i:|v_i|<2\mu\lambda b} \Big[\frac{1}{2}\Big(\frac{S(v_i,\lambda)}{2\mu-1/b} - \frac{v_i}{2\mu}\Big)^2 + p_{\lambda/(2\mu),2\mu b}\Big(\Big|\frac{S(v_i,\lambda)}{2\mu-1/b}\Big|\Big)\Big]\Big\} \\ &\quad - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu. \end{aligned} \tag{3.7.14}$$

In this case, we will have $\lambda < 2\mu\lambda b$. Therefore, we will need to consider the case where $\lambda < |v_i| < 2\mu\lambda b$

$$\begin{aligned} f(\mu) &= 2\mu\Bigg\{ \sum_{i:|v_i|\geq 2\mu\lambda b} p_{\lambda/(2\mu),2\mu b}\Big(\frac{|v_i|}{2\mu}\Big) + \sum_{i:|v_i|<2\mu\lambda b} \Bigg[\frac{1}{2}\Big(\frac{S(v_i,\lambda)}{2\mu-1/b} - \frac{v_i}{2\mu}\Big)^2 \\ &\quad + p_{\lambda/(2\mu),2\mu b}\Bigg(\Big|\frac{S(v_i,\lambda)}{2\mu-1/b}\Big|\Bigg)\Bigg]\Bigg\} - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu \\ &= 2\mu\Bigg\{ \sum_{i:|v_i|\geq 2\mu\lambda b} \frac{1}{2}(2\mu b)\Big(\frac{\lambda}{2\mu}\Big)^2 + \sum_{i:|v_i|\leq\lambda} \frac{1}{2}\Big(-\frac{v_i}{2\mu}\Big)^2 \\ &\quad + \sum_{i:\lambda<|v_i|<2\mu\lambda b} \underbrace{\Bigg[\frac{1}{2}\Big(\frac{S(v_i,\lambda)}{2\mu-1/b} - \frac{v_i}{2\mu}\Big)^2 + p_{\lambda/(2\mu),2\mu b}\Bigg(\Big|\frac{S(v_i,\lambda)}{2\mu-1/b}\Big|\Bigg)\Bigg]}_{\text{(i)}}\Bigg\} - \frac{\|\mathbf{v}\|_2^2}{4\mu} - \mu. \end{aligned} \tag{3.7.15}$$

Note that term (i) can be written as

$$\begin{aligned} &\frac{1}{2}\left(\frac{S(v_i,\lambda)}{2\mu-1/b} - \frac{v_i}{2\mu}\right)^2 + p_{\lambda/(2\mu),2\mu b}\left(\left|\frac{S(v_i,\lambda)}{2\mu-1/b}\right|\right) \\ &= \frac{1}{2}\left(\frac{\text{sign}(v_i)(|v_i|-\lambda)}{2\mu-1/b} - \frac{\text{sign}(v_i)|v_i|}{2\mu}\right)^2 + p_{\lambda/(2\mu),2\mu b}\left(\frac{|v_i|-\lambda}{2\mu-1/b}\right) \\ &= \frac{1}{2}\left(\frac{(|v_i|-\lambda)}{2\mu-1/b} - \frac{|v_i|}{2\mu}\right)^2 + p_{\lambda/(2\mu),2\mu b}\left(\frac{|v_i|-\lambda}{2\mu-1/b}\right). \end{aligned} \tag{3.7.16}$$

Since we have $2\mu\lambda b > |v_i|$, then

$$\frac{|v_i|-\lambda}{2\mu-1/b} < b\lambda \Leftarrow |v_i|-\lambda < (2\mu-1/b)\lambda b \Leftarrow |v_i|-\lambda < 2\mu\lambda b - \lambda, \tag{3.7.17}$$

which always hold in this case. Therefore, we have

$$p_{\lambda/(2\mu),2\mu b}\Big(\frac{|v_i|-\lambda}{|2\mu-1/b|}\Big) = \frac{\lambda(|v_i|-\lambda)}{2\mu(2\mu-1/b)} - \frac{(|v_i|-\lambda)^2}{4\mu b(2\mu-1/b)^2}. \tag{3.7.18}$$

Substituting (3.7.18) into (3.7.16), we obtain

$$\begin{aligned} \frac{1}{2}\left(\frac{S(v_i,\lambda)}{2\mu-1/b} - \frac{v_i}{2\mu}\right)^2 + p_{\lambda/(2\mu),2\mu b}\left(\left|\frac{S(v_i,\lambda)}{2\mu-1/b}\right|\right) &= \frac{1}{2}\left(\frac{(|v_i|-\lambda)}{2\mu-1/b} - \frac{|v_i|}{2\mu}\right)^2 + p_{\lambda/(2\mu),2\mu b}\left(\frac{|v_i|-\lambda}{2\mu-1/b}\right) \\ &= \frac{1}{2}\left(\frac{(|v_i|-\lambda)}{2\mu-1/b} - \frac{|v_i|}{2\mu}\right)^2 + \frac{\lambda(|v_i|-\lambda)}{2\mu(2\mu-1/b)} - \frac{(|v_i|-\lambda)^2}{4\mu b(2\mu-1/b)^2}. \end{aligned} \tag{3.7.19}$$

We further expand (3.7.19) and combine like terms and obtain

$$\begin{aligned}
&\frac{1}{2}\left(\frac{S(v_i,\lambda)}{2\mu-1/b}-\frac{v_i}{2\mu}\right)^2+p_{\lambda/(2\mu),2\mu b}\left(\left|\frac{S(v_i,\lambda)}{2\mu-1/b}\right|\right)\\
&=\frac{(|v_i|-\lambda)^2}{2(2\mu-1/b)^2}-\frac{|v_i|(|v_i|-\lambda)}{2\mu(2\mu-1/b)}+\frac{v_i^2}{8\mu^2}+\frac{\lambda(|v_i|-\lambda)}{2\mu(2\mu-1/b)}-\frac{(|v_i|-\lambda)^2}{4\mu b(2\mu-1/b)^2}\\
&=\frac{(|v_i|-\lambda)^2}{2(2\mu-1/b)^2}-\frac{(|v_i|-\lambda)^2}{4\mu b(2\mu-1/b)^2}+\frac{v_i^2}{8\mu^2}-\frac{(|v_i|-\lambda)^2}{2\mu(2\mu-1/b)}\\
&=-\frac{(|v_i|-\lambda)^2}{4\mu(2\mu-1/b)}+\frac{v_i^2}{8\mu^2}. \qquad (3.7.20)
\end{aligned}$$

Substituting (3.7.20) into (3.7.15) we obtain

$$\begin{aligned}
f(\mu)&=2\mu\Bigg\{\sum_{i:|v_i|\geq 2\mu\lambda b}\frac{1}{2}(2\mu b)\left(\frac{\lambda}{2\mu}\right)^2+\sum_{i:|v_i|<\lambda}\frac{1}{2}\Big(-\frac{v_i}{2\mu}\Big)^2\\
&\quad+\sum_{i:\lambda<|v_i|<2\mu\lambda b}\left[\frac{1}{2}\Big(\frac{S(v_i,\lambda)}{2\mu-1/b}-\frac{v_i}{2\mu}\Big)^2+p_{\lambda/(2\mu),2\mu b}\left(\left|\frac{S(v_i,\lambda)}{2\mu-1/b}\right|\right)\right]\Bigg\}-\frac{\|\mathbf{v}\|_2^2}{4\mu}-\mu\\
&=2\mu\Bigg\{\sum_{i:|v_i|\geq 2\mu\lambda b}\frac{1}{2}(2\mu b)\left(\frac{\lambda}{2\mu}\right)^2+\sum_{i:|v_i|<\lambda}\frac{1}{2}\Big(-\frac{v_i}{2\mu}\Big)^2+\sum_{i:\lambda<|v_i|<2\mu\lambda b}\left[-\frac{(|v_i|-\lambda)^2}{4\mu(2\mu-1/b)}+\frac{v_i^2}{8\mu^2}\right]\Bigg\}\\
&\quad-\frac{\|\mathbf{v}\|_2^2}{4\mu}-\mu\\
&=\sum_{i:|v_i|\geq 2\mu\lambda b}\frac{1}{2}b\lambda^2+\sum_{i:|v_i|<2\mu\lambda b}\frac{v_i^2}{4\mu}-\sum_{i:\lambda<|v_i|<2\mu\lambda b}\frac{(|v_i|-\lambda)^2}{2(2\mu-1/b)}-\frac{\|\mathbf{v}\|_2^2}{4\mu}-\mu. \qquad (3.7.21)
\end{aligned}$$

To find the optimal μ in this case, we would still sort $|v_i|$ in an ascending order first, i.e., $|v_{(1)}|\leq|v_{(2)}|\leq ...\leq|v_{(n)}|$. For a specific $\mathbf{v}$ and λ, we first assume that $|v_{(l-1)}|\leq\lambda<|v_{(l)}|$. We can define $v_{(0)}=0$ and $v_{(d+1)}=\infty$ to include the boundary cases. Since $2\mu b>1$, we only need to consider the intervals $(1/2\mu,|v_{(l)}|/2\lambda b]$ and $(|v_{(j)}|/2\lambda b,|v_{(j+1)}|/2\lambda b]$, $j=l,...,n$. When $\mu\in(|v_{(j)}|/2\lambda b,|v_{(j+1)}|/2\lambda b]$, where $j\geq l$, we have

$$\begin{aligned}
f(\mu)&=\sum_{i:|v_i|\geq 2\mu\lambda b}\frac{1}{2}b\lambda^2+\sum_{i:|v_i|<2\mu\lambda b}\frac{v_i^2}{4\mu}-\sum_{i:\lambda<|v_i|<2\mu\lambda b}\frac{(|v_i|-\lambda)^2}{2(2\mu-1/b)}-\frac{\|\mathbf{v}\|_2^2}{4\mu}-\mu\\
&=\sum_{i:|v_i|\geq 2\mu\lambda b}\Big(\frac{1}{2}b\lambda^2-\frac{v_i^2}{4\mu}\Big)-\sum_{i:\lambda<|v_i|<2\mu\lambda b}\frac{(|v_i|-\lambda)^2}{2(2\mu-1/b)}-\mu\\
&=\sum_{i=j+1}^{n}\Big(\frac{1}{2}b\lambda^2-\frac{v_{(i)}^2}{4\mu}\Big)-\sum_{i=l}^{j}\frac{(|v_{(i)}|-\lambda)^2}{2(2\mu-1/b)}-\mu. \qquad (3.7.22)
\end{aligned}$$

We denote $S_1 = \sum_{i=j+1}^{n} v_{(i)}^2$ and $S_2 = \sum_{i=l}^{j} (|v_{(i)}| - \lambda)^2$, and (3.7.22) can be written as

$$f(\mu) = \frac{(n-j)b\lambda^2}{2} - \frac{S_1}{4\mu} - \frac{S_2}{2(2\mu - 1/b)} - \mu. \tag{3.7.23}$$

To maximize $f(\mu)$ in each interval, we need to minimize the following objective function within the interval

$$J(\mu) = \frac{S_1}{4\mu} + \frac{S_2}{2(2\mu - 1/b)} + \mu, \tag{3.7.24}$$

which can be easily solved by MATLAB.

3.7.5 Auxiliary Technical Lemmas

In this section, we lay out several definitions and auxiliary lemmas.

Definition 3.7.1. [69] A random variable X is called sub-Gaussian if there exists a positive constant K such that $\mathbb{P}(|X| > t) \leq \exp(1 - t^2/K^2)$ for all $t > 0$.

The sub-Gaussian norm of X, denoted by $\|X\|_{\psi_2}$, is defined as follows

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

Lemma 3.7.2. [69] For Z being sub-Gaussian or sub-exponential, it holds that $\|Z - \mathbb{E}Z\|_{\psi_2} \leq 2 \cdot \|Z\|_{\psi_2}$ or $\|Z - \mathbb{E}Z\|_{\psi_1} \leq 2 \cdot \|Z\|_{\psi_1}$ correspondingly.

Lemma 3.7.3. [69] Let $X_1, \ldots, X_n$ be independent centered sub-Gaussian random variables, and let $K = \max_i \|X_i\|_{\psi_2}$. Then for any $\mathbf{a} = [a_1, \ldots, a_n]^\top \in \mathbb{R}^n$ and every $t > 0$, we have

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

where $C > 0$ is an absolute constant.

Lemma 3.7.4. [69] Let N_ϵ^d be the ϵ-net of a sphere $\mathbb{S}^{d-1}$, that for any $\mathbf{u} \in \mathbb{S}^{d-1}$, there exist a $\mathbf{u}_1 \in N_\epsilon^d$ such that $\|\mathbf{u} - \mathbf{u}_1\|_2 \leq \epsilon$. For any $\epsilon > 0$, we have $|N_\epsilon^d| \leq (1 + \frac{2}{\epsilon})^d$. Moreover, for any vector $\mathbf{a} \in \mathbb{R}^d$, the following inequality holds for $\epsilon \in (0, 1/2)$

$$\|\mathbf{a}\|_2 = \sup_{\mathbf{u} \in \mathbb{S}^d} \langle \mathbf{a}, \mathbf{u} \rangle \leq \frac{1}{1 - \epsilon} \sup_{\mathbf{u} \in N_\epsilon^d} |\langle \mathbf{a}, \mathbf{u} \rangle|.$$

Chapter 4

A Stochastic Gradient EM Algorithm with Improved Computational Complexity

In this chapter, I will introduce my research work on an efficient high dimensional expectation-maximization (EM) algorithm [39]. It is a generic algorithm based on stochastic gradient descent and naturally incorporates the **sparsity structure** of the model parameters. Compared with the existing high dimensional EM algorithms, our algorithm significantly reduces the **computational complexity** and achieves linear convergence rate and the best error bounds up to a logarithmic factor for several common latent variable models.

4.1 Introduction and Background

Expectation Maximization (EM) algorithm is an important method for the estimation of latent variable models.

Let $\boldsymbol{Y} \in \mathcal{Y}$ be an observed random variable and $\boldsymbol{Z} \in \mathcal{Z}$ be a latent random variable. Let $h_{\boldsymbol{\beta}}(\mathbf{y})$ be the probability density function of $\boldsymbol{Y}$ with the model parameter$\boldsymbol{\beta} \in \mathbb{R}^d$. It is given by the marginalization of joint distribution $f_{\boldsymbol{\beta}}(\mathbf{y}, \mathbf{z})$, i.e.,

$$h_{\boldsymbol{\beta}}(\mathbf{y}) = \int_{\mathcal{Z}} f_{\boldsymbol{\beta}}(\mathbf{y}, \mathbf{z}) \, \mathrm{d}\mathbf{z}. \tag{4.1.1}$$

Given the N observations $\{\mathbf{y}_i\}_{i=1}^N$ of $\boldsymbol{Y}$, the EM algorithm aims at maximizing the log-likelihood

$$\ell_N(\boldsymbol{\beta}) = \sum_{i=1}^{N} \log h_{\boldsymbol{\beta}}(\mathbf{y}_i). \tag{4.1.2}$$

It is difficult to directly evaluate $\ell_N(\boldsymbol{\beta})$ due to the unobserved latent variable $\boldsymbol{Z}$. Instead, we turn to focus on the difference between $\ell_N(\boldsymbol{\beta})$ and $\ell_N(\boldsymbol{\beta}')$. Let $p_{\boldsymbol{\beta}}(\mathbf{z} \mid \mathbf{y})$ be the conditional distribution of $\boldsymbol{Z}$ on the observed variable $\boldsymbol{Y} = \mathbf{y}$, i.e.,

$$p_{\boldsymbol{\beta}}(\mathbf{z} \mid \mathbf{y}) = f_{\boldsymbol{\beta}}(\mathbf{y}, \mathbf{z}) / h_{\boldsymbol{\beta}}(\mathbf{y}). \tag{4.1.3}$$

According to (4.1.1) and (4.1.2), we have

$$\frac{1}{N}\big[\ell_N(\boldsymbol{\beta}) - \ell_N(\boldsymbol{\beta}')\big] = \frac{1}{N}\sum_{i=1}^{N}\log\big[h_{\boldsymbol{\beta}}(\mathbf{y}_i)/h_{\boldsymbol{\beta}'}(\mathbf{y}_i)\big] = \frac{1}{N}\sum_{i=1}^{N}\log\left[\int_{\mathcal{Z}}\frac{f_{\boldsymbol{\beta}}(\mathbf{y}_i,\mathbf{z})}{h_{\boldsymbol{\beta}'}(\mathbf{y}_i)}\,\mathrm{d}\mathbf{z}\right]$$
$$= \frac{1}{N}\sum_{i=1}^{N}\log\left[\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}\mid\mathbf{y}_i)\cdot\frac{f_{\boldsymbol{\beta}}(\mathbf{y}_i,\mathbf{z})}{f_{\boldsymbol{\beta}'}(\mathbf{y}_i,\mathbf{z})}\,\mathrm{d}\mathbf{z}\right] \geq \frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}\mid\mathbf{y}_i)\cdot\log\left[\frac{f_{\boldsymbol{\beta}}(\mathbf{y}_i,\mathbf{z})}{f_{\boldsymbol{\beta}'}(\mathbf{y}_i,\mathbf{z})}\right]\,\mathrm{d}\mathbf{z}, \tag{4.1.4}$$

where the third equality comes from (4.1.3) and the inequality is obtained from Jensen's inequality. On the right-hand side of (4.1.4) we have

$$\frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}\mid\mathbf{y}_i)\cdot\log\left[\frac{f_{\boldsymbol{\beta}}(\mathbf{y}_i,\mathbf{z})}{f_{\boldsymbol{\beta}'}(\mathbf{y}_i,\mathbf{z})}\right]\,\mathrm{d}\mathbf{z}$$
$$= \underbrace{\frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}\mid\mathbf{y}_i)\cdot\log f_{\boldsymbol{\beta}}(\mathbf{y}_i,\mathbf{z})\,\mathrm{d}\mathbf{z}}_{\bar{Q}_N(\boldsymbol{\beta};\boldsymbol{\beta}')} - \frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}\mid\mathbf{y}_i)\cdot\log f_{\boldsymbol{\beta}'}(\mathbf{y}_i,\mathbf{z})\,\mathrm{d}\mathbf{z}. \tag{4.1.5}$$

Note that the second term on the right-hand side of (4.1.5) does not depend on $\boldsymbol{\beta}$. We define the first term on the right-hand side of (4.1.5) to be $\bar{Q}_N(\boldsymbol{\beta};\boldsymbol{\beta}')$. Correspondingly, we define its expectation to be $Q(\boldsymbol{\beta};\boldsymbol{\beta}')$. Given some fixed $\boldsymbol{\beta}'$, we can maximize the lower bound function $\bar{Q}_N(\boldsymbol{\beta};\boldsymbol{\beta}')$ over $\boldsymbol{\beta}$ to maximize $\ell_N(\boldsymbol{\beta}) - \ell_N(\boldsymbol{\beta}')$.

For stochastic methods, we often divide the N samples into n mini-batches $\{\mathcal{D}_i\}_{i=1}^n$, and define function $\{q_i\}_{i=1}^n$ on these mini-batches,

$$q_i(\boldsymbol{\beta};\boldsymbol{\beta}') = \frac{1}{|\mathcal{D}_i|}\sum_{j\in\mathcal{D}_i}\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}|\mathbf{y}_j)\cdot\log f_{\boldsymbol{\beta}}(\mathbf{y}_j,\mathbf{z})\,\mathrm{d}\mathbf{z}. \tag{4.1.6}$$

When $n = N$, there are no mini-batches. We further define

$$Q_n(\boldsymbol{\beta};\boldsymbol{\beta}') = \frac{1}{n}\sum_{i=1}^{n} q_i(\boldsymbol{\beta};\boldsymbol{\beta}').$$

Particularly, in the l-th iteration of EM algorithm, we evaluate $\bar{Q}_N(\boldsymbol{\beta};\boldsymbol{\beta}^{(l)})$ in the E-step, and perform maximization of $\bar{Q}_N(\boldsymbol{\beta};\boldsymbol{\beta}^{(l)})$ on $\boldsymbol{\beta}$ in the M-step. For example, in standard gradient ascent implementation of EM algorithm, the M-step is given by

$$\boldsymbol{\beta}^{(l+1)} = \boldsymbol{\beta}^{(l)} + \eta\nabla_1\bar{Q}_N(\boldsymbol{\beta}^{(l)};\boldsymbol{\beta}^{(l)}),$$

where $\nabla_1\bar{Q}_N(\cdot;\cdot)$ denotes the gradient on the first variable and η is the learning rate.

In the high dimensional regime, the dimensionality of the parameter d is comparable with or even larger than N. Therefore, exact maximization of $\bar{Q}_N(\boldsymbol{\beta};\boldsymbol{\beta}^{(l)})$ is often intractable or even not well-defined. In contrast, gradient EM algorithms are more general in this scenario.

It is found in [27] that sparsity has to be enforced and exploited for reliable performance of EM algorithms in high dimensional scenarios. Otherwise, variance and errors will accumulate across all dimensions to significantly perturb the results. Therefore, we assume $\boldsymbol{\beta}^* \in \mathbb{R}^d$ is sparse with $\|\boldsymbol{\beta}\|_0 \leq s^*$. In order to ensure the sparsity of the estimator, we follow [27] to use a truncation step (i.e., T-step) following the M-step. For better reference, we outline the gradient variant of their algorithm in Algorithm 3. To ensure the **sparsity**

Algorithm 3 High Dimensional Gradient EM Algorithm

Parameter: Sparsity Parameter s, Maximum Number of Iterations T, learning rate η
Initialization:
  $\boldsymbol{\beta}^{(0)} = \mathcal{H}_s(\boldsymbol{\beta}^{\text{init}})$,
For $t = 0$ to $T-1$
  E-step:
    Evaluate $\bar{Q}_N\big(\boldsymbol{\beta};\boldsymbol{\beta}^{(t)}\big)$ with the dataset
  M-step:
    $\boldsymbol{\beta}^{(t+0.5)} = \boldsymbol{\beta}^{(t)} + \eta\nabla_1\bar{Q}_N(\boldsymbol{\beta}^{(t)};\boldsymbol{\beta}^{(t)})$,
  T-step:
    $\boldsymbol{\beta}^{(t+1)} = \mathcal{H}_s(\boldsymbol{\beta}^{(t+0.5)})$
End For
Output: $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(T)}$

of the output estimator and overcome the dimensionality issue, we use the hard thresholding operator [70], $\mathcal{H}_s(\mathbf{v}) = \mathbf{v}_{\text{supp}(\mathbf{v},s)}$, which only keeps the largest s entries in magnitude of a vector $\mathbf{v} \in \mathbb{R}^d$. The sparsity parameter s controls the sparsity level of the estimated parameter, and is critical to the estimation error and convergence performance.

4.2 Stochastic Variance Reduced Gradient

One of the key challenges in high dimensional and big data scenario is that the evaluation and optimization of $\bar{Q}_N(\boldsymbol{\beta};\boldsymbol{\beta}')$ can be computationally prohibitive. We cannot afford the iterations of summations and gradient evaluation on the whole enormous dataset. Therefore, stochastic gradient descent is widely adopted as a workaround here to reduce the computational complexity.

However, while stochastic gradient method is popular for large scale and high dimensional optimization, the intrinsic variance of the algorithm harms the convergence rate significantly [33, 71, 72]. In [33], the authors proposed stochastic variance reduced gradient (SVRG), which is proved to be effective on convex and smooth functions.

Specifically, the main idea of SVRG is computing an average gradient on the dataset for one pass, and using such average to overcome the variance of stochastic gradient. In particular, they let

$$\widetilde{\mu} = \nabla_1 \bar{Q}_N(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}) = \frac{1}{N}\sum_{i=1}^{N} q_i(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}),$$

where $\widetilde{\boldsymbol{\beta}}$ is a reasonable estimate of true model parameter $\boldsymbol{\beta}^*$. Then their gradient ascent process is

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \eta\big(\nabla_1 q_i(\boldsymbol{\beta}^{(t)}; \boldsymbol{\beta}^{(t)}) - \nabla_1 q_i(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}) + \widetilde{\mu}\big),$$

where $q_i(\cdot;\cdot)$ is a stochastic gradient based on a data point or a mini-batch.

However, even this framework has been applied to several optimization problems, its efficacy is still remained to be seen for the problem of high dimensional EM due to the complexity, bivariate structure and strong model dependency of this problem.

4.3 Semi-stochastic Gradient EM with Variance Reduction

In this section, we will present our proposed algorithm. We will start with two latent variable models, and then describe our method.

4.3.1 Latent Variable Models

We now introduce two popular latent variable models we use to illustrate the efficacy of our proposed method, sparse Gaussian Mixture Model.

Sparse Gaussian Mixture Model: The random variable $\boldsymbol{Y} \in \mathbb{R}^d$ is given by

$$\boldsymbol{Y} = Z \cdot \boldsymbol{\beta}^* + \boldsymbol{V},$$

where Z is a random variable with $\mathbb{P}(Z = 1) = \mathbb{P}(Z = -1) = 1/2$, and $\boldsymbol{V} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ is a Gaussian random vector, with $\boldsymbol{\Sigma}$ being the covariance matrix, $\boldsymbol{V}$ and Z are independent, and $\|\boldsymbol{\beta}^*\|_0 \leq s^*$. We assume $\boldsymbol{\Sigma}$ is known for simplicity.

Sparse Mixture of Linear Regression: We assume that $Y \in \mathbb{R}$ and $\boldsymbol{X} \in \mathbb{R}^d$ satisfy

$$Y = Z \cdot \boldsymbol{X}^\top \boldsymbol{\beta}^* + V, \tag{4.3.1}$$

where $\boldsymbol{X} \sim N(\mathbf{0}, \mathbf{I}_d)$, $V \sim N(0, \sigma^2)$ and Z is a Rademacher random variable. Here $\boldsymbol{X}$, V and Z are independent. In the high dimensional regime, we also assume $\boldsymbol{\beta}^* \in \mathbb{R}^d$ is sparse. We also assume that σ here is known.

4.3.2 Semi-stochastic Variance Reduced Gradient EM

We have applied a semi-stochastic variance reduced gradient to high dimensional EM, which naturally incorporates the **sparsity structure** in the model parameters. In this section, we introduce this proposed gradient structure and the algorithm.

This semi-stochastic structure is specifically designed for the bivariate structure of the Q-function in EM algorithms. Specifically, we propose two layers of iterations to update the estimator, and for each outer iteration, we perform the E-step, i.e., evaluate $Q_n(\cdot;\cdot)$ and compute the average gradient $\widetilde{\mu}$. Since our algorithm is stochastic, we divide the N data points into n mini-batches. Without the loss of generality, we assume $N = nb$ and b is an integer denoting the mini-batch size. Therefore, given $q_i(\boldsymbol{\beta};\boldsymbol{\beta}') = (1/b)\sum_{j\in\mathcal{D}_i}\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}|\mathbf{y}_j)\cdot\log f_{\boldsymbol{\beta}}(\mathbf{y}_j,\mathbf{z})\,\mathrm{d}\mathbf{z}$, it is easy to verify that

$$
\begin{aligned}
Q_n(\boldsymbol{\beta};\boldsymbol{\beta}') &= \frac{1}{n}\sum_{i=1}^{n} q_i(\boldsymbol{\beta};\boldsymbol{\beta}') = \frac{1}{nb}\sum_{i=1}^{n}\sum_{j\in\mathcal{D}_i}\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}|\mathbf{y}_j)\cdot\log f_{\boldsymbol{\beta}}(\mathbf{y}_j,\mathbf{z})\,\mathrm{d}\mathbf{z} \\
&= \frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Z}} p_{\boldsymbol{\beta}'}(\mathbf{z}|\mathbf{y}_i)\cdot\log f_{\boldsymbol{\beta}}(\mathbf{y}_i,\mathbf{z})\,\mathrm{d}\mathbf{z} = \bar{Q}_N(\boldsymbol{\beta};\boldsymbol{\beta}').
\end{aligned}
$$

Therefore, the maximization of $\bar{Q}_N(\cdot;\cdot)$ is equivalent to the maximization of $Q_n(\cdot;\cdot)$.

In the M-step, we have the inner iterations. We first determine the number of inner iterations, which is randomly selected from $[T]$ uniformly. Specifically, we design a novel semi-stochastic gradient on mini-batches to update the estimator, which is given by

$$
\mathbf{v}^{(t)} = \nabla_1 q_i(\boldsymbol{\beta}^{(t)};\widetilde{\boldsymbol{\beta}}) - \nabla_1 q_i(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}}) + \widetilde{\mu},
$$

which fixes the second variable within each outer iteration for the sake of convergence guarantee. $q_i(\cdot;\cdot)$ here is a stochastic gradient based on a mini-batch given by (4.1.6). While the standard gradient implementation of EM algorithm [27] uses $\nabla_1\bar{Q}_N(\boldsymbol{\beta}^{(t)};\boldsymbol{\beta}^{(t)})$ to update the parameter at each iteration, our newly designed semi-stochastic gradient EM is proved to better reduce the variance and attain a lower computational complexity.

After finishing all the inner iterations, we use the output from the last inner iteration as the updated estimator of this outer iteration. Finally, we use the output from the last outer iteration as the final

estimator. We have outlined our algorithm in Algorithm 4.

Algorithm 4 Accelerated Stochastic Variance Reduced Gradient EM Algorithm (VRGEM)

Parameter: Sparsity Parameter s, Maximum Number of Outer Iterations m, Number of Inner Iterations T, learning rate η
Initialization:
  $\widetilde{\boldsymbol{\beta}}^{(0)} = \mathcal{H}_s(\boldsymbol{\beta}^{\text{init}})$,
For $l = 0$ to $m - 1$
  E-step:
    Evaluate $Q_n\big(\boldsymbol{\beta}; \widetilde{\boldsymbol{\beta}}^{(l)}\big)$ with the dataset
    $\widetilde{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(l)}, \quad \widetilde{\mu} = \nabla_1 Q_n(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}})$
  M-step:
    $\boldsymbol{\beta}^{(0)} = \widetilde{\boldsymbol{\beta}}$
    Randomly select j_l uniformly from $\{0, \ldots, T-1\}$
    For $t = 0$ to j_l
      Randomly select i from $[n]$ uniformly
      $\mathbf{v}^{(t)} = \nabla_1 q_i\big(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}\big) - \nabla_1 q_i\big(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}\big) + \widetilde{\mu}$,
      $\boldsymbol{\beta}^{(t+0.5)} = \boldsymbol{\beta}^{(t)} + \eta \mathbf{v}^{(t)}$,
      T-step: $\boldsymbol{\beta}^{(t+1)} = \mathcal{H}_s(\boldsymbol{\beta}^{(t+0.5)})$
    End For
  $\widetilde{\boldsymbol{\beta}}^{(l+1)} = \boldsymbol{\beta}^{(j_l)}$
End For
Output: $\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(m)}$

4.4 Main Theory

In this section, we show the main theory on the theoretical guarantee of our proposed Algorithm 4. We also present the implications of our algorithm applied to two models described in Section 4.3.1. Specifically, we first provide the theoretical guarantee of the estimation error bound, and then give the implications on two latent variable models as examples. We also analyze the computational complexity and show the advantage of the proposed method.

The estimation error of our estimator in the l-th iteration $\|\widetilde{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}^*\|_2$ can be decomposed into **optimization error** and **statistical error**, which are given by

$$\underbrace{\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2}_{\text{estimation error}} \leq \underbrace{\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2}_{\text{optimization error}} + \underbrace{\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2}_{\text{statistical error}}, \tag{4.4.1}$$

where $\widehat{\boldsymbol{\beta}}$ is our final estimator.

Suppose our algorithm can always converge to a reasonable local optima, then $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ denotes the distance between the current estimator to the final estimator. This error reduces with the optimization

iterations proceed. In fact, our proposed algorithm achieves a linear convergence rate, i.e.,

$$\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2 \leq \rho \|\widetilde{\boldsymbol{\beta}}^{(l-1)} - \widehat{\boldsymbol{\beta}}\|_2,$$
$$\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2 \leq \rho^l \|\widetilde{\boldsymbol{\beta}}^{\text{init}} - \widehat{\boldsymbol{\beta}}\|_2$$

which means that the optimization error decays geometrically.

The second part is the statistical error, which features the distance between our final estimator and the true model parameter. This error largely depends on the latent variable model and the problem setting, e.g. sparsity parameter s, model dimensionality d and number of samples N. Therefore, this error features a statistical rate of convergence for the estimators. For example, the statistical rate of convergence for Gaussian mixture model is $O(s \log d \log N/N)$, which characterizes how their estimator convergences to the true model parameter given s, d and N.

4.4.1 Technical Conditions

Before we layout our main theoretical results, we give some technical conditions for functions $q_i(\cdot;\cdot)$ and $Q_n(\cdot;\cdot)$ which are necessary for our analysis. It is worth noting that these conditions are mild and hold for most of the latent variable models. We will verify these conditions for the two models we use, GMM and MLR.

Condition 4.4.1 (Smoothness)**.** For any $\boldsymbol{\beta}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$, where $p \in (0,1)$ is a model-dependent constant, for any $i \in [n]$, $q_i(\cdot;\cdot)$ in Algorithm 4 satisfies the smoothness condition with respect to the first variable with parameter L:

$$\big\|\nabla_1 q_i(\boldsymbol{\beta}_1; \boldsymbol{\beta}) - \nabla_1 q_i(\boldsymbol{\beta}_2; \boldsymbol{\beta})\big\|_2 \leq L\big\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\big\|_2.$$

Condition 4.4.1 says that the gradient of $q_i(\cdot;\cdot)$ we use in each inner iteration is Lipschitz continuous with respect to the first variable when the first and second variables are within the ball $\mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$. This condition is widely used in high dimensional EM studies [26, 27, 28], and holds for a variety of latent variable models.

Condition 4.4.2 (Concavity)**.** For all $\boldsymbol{\beta}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathcal{B}(p\|\boldsymbol{\beta}^*\|_2; \boldsymbol{\beta}^*)$, where $p \in (0,1)$ is a model-dependent constant, the function $Q_n(\cdot;\cdot)$ in Algorithm 4 satisfies the strong concavity condition with parameter μ:

$$\big[\nabla_1 Q_n(\boldsymbol{\beta}_1; \boldsymbol{\beta}) - \nabla_1 Q_n(\boldsymbol{\beta}_2; \boldsymbol{\beta})\big]^\top (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2) \leq -\mu\|\boldsymbol{\beta}_2 - \boldsymbol{\beta}_1\|_2^2.$$

Condition 4.4.2 requires $Q_n(\cdot;\cdot)$ to be strongly concave with respect to the first variable when the first and second variables are within the ball $\mathcal{B}(p\|\boldsymbol{\beta}^*\|_2;\boldsymbol{\beta}^*)$. We will show this is a reasonable requirement when $N = nb$ is large enough. Variants of this condition are widely used in previous work [26, 27].

Condition 4.4.3 (First-order stability)**.** For the true model parameter $\boldsymbol{\beta}^*$ and any $\boldsymbol{\beta} \in \mathcal{B}(p\|\boldsymbol{\beta}^*\|_2;\boldsymbol{\beta}^*)$, where $p \in (0,1)$ is a model-dependent constant, $Q_n\big(\cdot;\cdot\big)$ satisfies the first-order stability for parameter γ:

$$\big\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}) - \nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_2 \leq \gamma\big\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\big\|_2.$$

Condition 4.4.3 requires that the gradient $\nabla_1 Q_n(\boldsymbol{\beta}^*;\cdot)$ is stable with regard to the second variable, with the second variable within the ball $\mathcal{B}(p\|\boldsymbol{\beta}^*\|_2;\boldsymbol{\beta}^*)$. There are actually various versions of this condition in previous work [26, 28] on population version $Q(\cdot;\cdot) = \mathbb{E}[Q_n(\cdot,\cdot)]$. Here we impose the condition on the sample Q-function, i.e., $Q_n(\cdot,\cdot)$, because our proof technique directly analyzes the sample Q-function. Intuitively, when the sample size N is sufficiently large, $Q_n(\cdot;\cdot)$ and $Q(\cdot;\cdot)$ should be close. Therefore, this condition should hold for $Q_n(\cdot;\cdot)$ as well.

Definition 4.4.4. We let $\kappa = L/\mu$ be the **condition number** where L is the parameter in the smoothness condition and μ is the parameter in the strong concavity condition.

As the Q-function is model-dependent, these technical conditions and the condition number also need to be verified specifically for the latent variable models instead of generally. We will validate these conditions along with the implications of our theory for specific models in later sections.

4.4.2 General Theory

In this section, we layout the general theory that characterize the performance of our proposed algorithm. By "general", we mean that the theory shown in this section is not model-dependent.

Following the convention of previous work on high dimensional EM algorithms [27, 28], we first present a resampling variant of our proposed method. For such resampling version, we split the dataset into several non-overlapping subsets, and use just one of them for each outer iteration. This is for the decoupling the correlation of the data between consecutive outer iterations. The resampling version is only used here to facilitate the theoretical analysis, and in practice including the numerical experiments, we still use Algorithm 4. We outline the resampling version in Algorithm 5.

With the technical conditions introduced in Section 4.4.1 holding, we have the following theorem featuring the estimation error of our proposed estimator $\widetilde{\boldsymbol{\beta}}^{(r)}$ in Algorithm 5.

Theorem 4.4.5. Suppose $q_i(\cdot;\cdot)$ satisfies Conditions 4.4.1 and $Q_n(\cdot;\cdot)$ satisfies Condition 4.4.2 4.4.3. We also assume that $\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\|_2 \leq p\|\boldsymbol{\beta}^*\|_2$, where $p \in (0,1)$. If $\eta \leq \mu/(8L^2)$, and T and s are chosen such that

$$\rho = \frac{1}{T(1-\tau)} + \frac{2\alpha\eta\big[\eta L^2 + (2\eta + L/\mu^2)\gamma^2\big]}{1-\tau} < 1,$$

where $\tau = \alpha(1 - \eta\mu + 2\eta^2 L^2)$ and $\alpha = 1 + \sqrt{s^*}/\sqrt{s - s^*}$, then the estimator $\widetilde{\boldsymbol{\beta}}^{(r)}$ from Algorithm 5 satisfies

$$\mathbb{E}\|\widetilde{\boldsymbol{\beta}}^{(r)} - \boldsymbol{\beta}^*\|_2 \leq \rho^{r/2}\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\|_2 + \sqrt{\frac{2\widetilde{s}\alpha\eta(2\eta + L/\mu^2)}{(1-\tau)(1-\rho)}}\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\|_\infty, \tag{4.4.2}$$

where $\widetilde{s} = 2s + s^*$.

Proof. We provide the proof sketch here. To derive the linear convergence rate in optimization error, we need to first characterize the relationship between estimation errors in consecutive outer iterations. Since we have both the M-step and the T-step to update the estimator, we first have the following lemma for the T-step in Algorithm 4.

Lemma 4.4.6. [34] Let $\boldsymbol{\beta}^* \in \mathbb{R}^d$ is a sparse vector with $\|\boldsymbol{\beta}^*\|_0 \leq s^*$. For any vector $\boldsymbol{\beta} \in \mathbb{R}^d$, we let $\widehat{\mathcal{S}} = \text{supp}(\boldsymbol{\beta}, s)$. We have

$$\|\text{trunc}(\boldsymbol{\beta}, \widehat{\mathcal{S}}) - \boldsymbol{\beta}^*\|_2^2 \leq \left(1 + \frac{2\sqrt{s^*}}{\sqrt{s - s^*}}\right)\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2^2$$

Intuitively, the truncation step introduces extra estimation error. Lemma 4.4.6 indicates that this error can be bounded by the error before truncation. When the truncation parameter s gets larger, we keep more components of $\boldsymbol{\beta}$ and the error brought by truncation will reduce.

Now, we are ready to prove our main results. Our goal is to characterize the relationship between $\|\widetilde{\boldsymbol{\beta}}^{(s+1)} - \boldsymbol{\beta}^*\|_2$ and $\|\widetilde{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}^*\|_2$. We will first analyze the inner iterations, i.e., finding the relationship between $\|\widetilde{\boldsymbol{\beta}}^{(t+1)} - \boldsymbol{\beta}^*\|_2$ and $\|\widetilde{\boldsymbol{\beta}}^{(t)} - \boldsymbol{\beta}^*\|_2$. Then, by telescoping sum, we can get the relation ship between $\|\widetilde{\boldsymbol{\beta}}^{(s+1)} - \boldsymbol{\beta}^*\|_2$ and $\|\widetilde{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}^*\|_2$.

We define $\boldsymbol{v}^{(t)} = \nabla_1 Q_{i_t}\big(\boldsymbol{\beta}^{(t)};\widetilde{\boldsymbol{\beta}}\big) - \nabla_1 Q_{i_t}\big(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}}\big) + \widetilde{\mu}$, and $\mathcal{S} = \text{supp}(\boldsymbol{\beta}^*) \cup \text{supp}(\boldsymbol{\beta}^{(t)}) \cup \text{supp}(\boldsymbol{\beta}^{(t+1)})$. Obviously, we have $|\mathcal{S}| \leq \widetilde{s} = 2s + s^*$. For simplicity, we use $\nabla_{\mathcal{S}} = \big[\nabla_1\big]_{\mathcal{S}}$, e.g., $\nabla_{\mathcal{S}} Q_n(\boldsymbol{\beta}_1;\boldsymbol{\beta}_2) = \big[\nabla_1 Q_n(\boldsymbol{\beta}_1;\boldsymbol{\beta}_2)\big]_{\mathcal{S}}$. For each inner iteration, we have from Lemma 4.4.6

$$\|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^*\|_2^2 \leq \left(1 + \frac{2\sqrt{s^*}}{\sqrt{s - s^*}}\right)\|\boldsymbol{\beta}^{(t)} + \eta\boldsymbol{v}_{\mathcal{S}}^{(t)} - \boldsymbol{\beta}^*\|_2^2.$$

Note that $\mathbb{E}_t\big[\boldsymbol{v}_{\mathcal{S}}^{(t)}\big] = \nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}\big)$, and further we have

$$\mathbb{E}_t\big\|\boldsymbol{\beta}^{(t)} + \eta \boldsymbol{v}_{\mathcal{S}}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 \le \big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 + \underbrace{2\eta(\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*)^\top \nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}\big)}_{I_1} + \underbrace{\eta^2 \mathbb{E}_t \|\boldsymbol{v}_{\mathcal{S}}^{(t)}\|_2^2}_{I_2}. \tag{4.4.3}$$

For I_1, we have

$$\begin{aligned} I_1 &= 2\eta(\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*)^\top \nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}\big) \\ &= \underbrace{2\eta(\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*)^\top \big[\nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}\big) - \nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}}\big)\big]}_{I_{1,1}} + \underbrace{2\eta(\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*)^\top \nabla_{\mathcal{S}} Q_n(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}})}_{I_{1,2}}. \end{aligned}$$

For the first term, we apply Lemma C.2 in [73]

$$\begin{aligned} I_{1,1} &\le -\frac{\eta}{L}\big\|\nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}\big) - \nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}}\big)\big\|_2^2 - \eta\mu\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2, & (4.4.4)\\ &\le -\frac{\eta\mu^2}{L}\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 - \eta\mu\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2, & (4.4.5) \end{aligned}$$

where the second inequality uses the concavity of $Q_n\big(\cdot; \widetilde{\boldsymbol{\beta}}\big)$ For $I_{1,2}$, we have

$$\begin{aligned} I_{1,2} &= 2\eta(\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*)^\top \nabla_{\mathcal{S}} Q_n(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}}) \\ &\le \frac{\eta\mu^2}{L}\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 + \frac{\eta L}{\mu^2}\big\|\nabla_{\mathcal{S}} Q_n(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}})\big\|_2^2 \\ &\le \frac{\eta\mu^2}{L}\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 + \frac{2\eta L}{\mu^2}\big\|\nabla_{\mathcal{S}} Q_n(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}} Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\big\|_2^2 + \frac{2\eta L}{\mu^2}\big\|\nabla_{\mathcal{S}} Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\big\|_2^2 \\ &\le \frac{\eta\mu^2}{L}\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 + \frac{2\eta L\gamma^2}{\mu^2}\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 + \frac{2\eta L}{\mu^2}\big\|\nabla_{\mathcal{S}} Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\big\|_2^2, \end{aligned} \tag{4.4.6}$$

where the first inequality holds because $2\boldsymbol{a}^\top \boldsymbol{b} \le \beta\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2/\beta$ for any $\beta > 0$.

For I_2, we have

$$\begin{aligned} I_2 &= \eta^2 \mathbb{E}_t \|\nabla_{\mathcal{S}} Q_{i_t}\big(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}\big) - \nabla_{\mathcal{S}} Q_{i_t}\big(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}\big) + \nabla_{\mathcal{S}} Q_n\big(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}\big)\|_2^2 \\ &\le 2\eta^2 \mathbb{E}_t\big\|\nabla_{\mathcal{S}} Q_{i_t}(\boldsymbol{\beta}^{(t)}; \widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}} Q_{i_t}(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}})\big\|_2^2 + 2\eta^2 \mathbb{E}_t\big\|\nabla_{\mathcal{S}} Q_{i_t}(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}} Q_{i_t}(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}} Q_n(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}})\big\|_2^2 \\ &\le 2\eta^2 L^2\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 + \underbrace{2\eta^2 \mathbb{E}_t\big\|\nabla_{\mathcal{S}} Q_{i_t}(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}} Q_{i_t}(\boldsymbol{\beta}^*; \widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}} Q_n(\widetilde{\boldsymbol{\beta}}; \widetilde{\boldsymbol{\beta}})\big\|_2^2}_{I_{2,1}}, \end{aligned} \tag{4.4.7}$$

where the first inequality comes from $\|\boldsymbol{a} + \boldsymbol{b}\|_2^2 \le 2\|\boldsymbol{a}\|_2^2 + 2\|\boldsymbol{b}\|_2^2$ and the second inequality comes from the

smoothness of $Q_{i_t}(\cdot;\cdot)$. For $I_{2,1}$ in 4.4.7, we have

$$\begin{aligned}
I_{2,1} &= 2\eta^2\mathbb{E}_t\big\|\nabla_{\mathcal{S}}Q_{i_t}(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}}Q_{i_t}(\boldsymbol{\beta}^*;\widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}}Q_n(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}})\big\|_2^2 \\
&= 2\eta^2\mathbb{E}_t\big\|\nabla_{\mathcal{S}}Q_{i_t}(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}}Q_{i_t}(\boldsymbol{\beta}^*;\widetilde{\boldsymbol{\beta}})\big\|_2^2 - 4\eta^2\nabla_{\mathcal{S}}Q_n(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}})^\top\big[\nabla_{\mathcal{S}}Q_n(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}}Q_n(\boldsymbol{\beta}^*;\widetilde{\boldsymbol{\beta}})\big] \\
&\qquad + 2\eta^2\big\|\nabla_{\mathcal{S}}Q_n(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}})\big\|_2^2 \\
&\le 2\eta^2L^2\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 + 4\eta^2\nabla_{\mathcal{S}}Q_n(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}})^\top\nabla_{\mathcal{S}}Q_n(\boldsymbol{\beta}^*;\widetilde{\boldsymbol{\beta}}) - 2\eta^2\big\|\nabla_{\mathcal{S}}Q_n(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}})\big\|_2^2 \\
&\le 2\eta^2L^2\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 + 2\eta^2\big\|\nabla_{\mathcal{S}}Q_n(\boldsymbol{\beta}^*;\widetilde{\boldsymbol{\beta}})\big\|_2^2 \\
&\le 2\eta^2L^2\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 + 4\eta^2\big\|\nabla_{\mathcal{S}}Q_n(\boldsymbol{\beta}^*;\widetilde{\boldsymbol{\beta}}) - \nabla_{\mathcal{S}}Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_2^2 + 4\eta^2\big\|\nabla_{\mathcal{S}}Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_2^2 \\
&\le 2\eta^2L^2\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 + 4\eta^2\gamma^2\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 + 4\eta^2\big\|\nabla_{\mathcal{S}}Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_2^2 \qquad (4.4.8)
\end{aligned}$$

where the first inequality comes from the smoothness of $Q_{i_t}(\cdot;\cdot)$; the second inequality holds because $2\boldsymbol{a}^\top\boldsymbol{b} \le \|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2$; the third inequality comes from $\|\boldsymbol{a}+\boldsymbol{b}\|_2^2 \le 2\|\boldsymbol{a}\|_2^2 + 2\|\boldsymbol{b}\|_2^2$, and last inequality uses the first order stability of $Q_n(\boldsymbol{\beta}^*;\cdot)$.

Combining (4.4.7) and (4.4.8), we get

$$I_2 \le 2\eta^2L^2\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 + 2\eta^2L^2\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 + 4\eta^2\gamma^2\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 + 4\eta^2\big\|\nabla_{\mathcal{S}}Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_2^2 \qquad (4.4.9)$$

We let $\alpha = 1 + 2\sqrt{s^*}/\sqrt{s-s^*}$. Plugging (4.4.4), (4.4.6) and (4.4.9)into (4.4.3) and applying Lemma 4.4.6, we obtain

$$\begin{aligned}
\mathbb{E}_t\big\|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^*\big\|_2^2 \le \alpha(1-\eta\mu+2\eta^2L^2)\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 + 2\alpha\eta\big[\eta L^2 + (2\eta + L/\mu^2)\gamma^2\big]\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 \\
+ 2\alpha\eta(2\eta + L/\mu^2)\big\|\nabla_{\mathcal{S}}Q_n\big(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*\big)\big\|_2^2.
\end{aligned}$$

We will let $\tau = \alpha(1-\eta\mu+2\eta^2L^2)$. By summing the above inequality over $t = 0, 1, \ldots, T-1$ and taking expectation to all i_t's, we have

$$\begin{aligned}
\mathbb{E}\big\|\boldsymbol{\beta}^{(T)} - \boldsymbol{\beta}^*\big\|_2^2 - \big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 \le (\tau-1)\sum_{t=0}^{T-1}\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 + 2T\alpha\eta\big[\eta L^2 + (2\eta + L/\mu^2)\gamma^2\big]\big\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2^2 \\
+ 2T\alpha\eta(2\eta + L/\mu^2)\big\|\nabla_{\mathcal{S}}Q_n\big(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*\big)\big\|_2^2. \qquad (4.4.10)
\end{aligned}$$

Note that $\widetilde{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^l = \boldsymbol{\beta}^{(0)}$ and we uniformly choose j_l from $[T]$ and let $\widetilde{\boldsymbol{\beta}}^{(l+1)} = \boldsymbol{\beta}^{(j_l)}$, so we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\big\|\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^*\big\|_2^2 = \mathbb{E}\big\|\widetilde{\boldsymbol{\beta}}^{(l+1)} - \boldsymbol{\beta}^*\big\|_2^2.$$

Plugging in this into (4.4.10), we have

$$\begin{aligned} T(1-\tau)\mathbb{E}\big\|\widetilde{\boldsymbol{\beta}}^{(l+1)} - \boldsymbol{\beta}^*\big\|_2^2 \le \big[1 + 2T\alpha\eta[\eta L^2 + (2\eta + L/\mu^2)\gamma^2)]\big]\big\|\widetilde{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}^*\big\|_2^2 \\ + 2T\alpha\eta(2\eta + L/\mu^2)\big\|\nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*\big)\big\|_2^2. \end{aligned} \tag{4.4.11}$$

Let

$$\rho = \frac{1 + 2T\alpha\eta[\eta L^2 + (2\eta + L/\mu^2)\gamma^2]}{T(1-\tau)}, \quad \text{and} \quad \zeta = \frac{2\alpha\eta(2\eta + L/\mu^2)\big\|\nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*\big)\big\|_2^2}{1-\tau},$$

we obtain

$$\mathbb{E}\big\|\widetilde{\boldsymbol{\beta}}^{(l+1)} - \boldsymbol{\beta}^*\big\|_2^2 \le \rho\big\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\big\|_2^2 + \zeta,$$

which immediately yields

$$\mathbb{E}\big\|\widetilde{\boldsymbol{\beta}}^{(m)} - \boldsymbol{\beta}^*\big\|_2^2 \le \rho^m\big\|\widetilde{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^*\big\|_2^2 + \frac{(1-\rho^m)\zeta}{1-\rho} \le \rho^m\big\|\widetilde{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^*\big\|_2^2 + \frac{\zeta}{1-\rho}.$$

We take square root on both sides of the inequality and get

$$\begin{aligned} \mathbb{E}\big\|\widetilde{\boldsymbol{\beta}}^{(m)} - \boldsymbol{\beta}^*\big\|_2 \le \sqrt{\mathbb{E}\big\|\widetilde{\boldsymbol{\beta}}^{(m)} - \boldsymbol{\beta}^*\big\|_2^2} &\le \rho^{m/2}\big\|\widetilde{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^*\big\|_2 + \sqrt{\frac{\zeta}{1-\rho}} \\ &= \rho^{m/2}\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 + \sqrt{\frac{2\alpha\eta(2\eta + L/\mu^2)}{(1-\tau)(1-\rho)}}\big\|\nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*\big)\big\|_2. \end{aligned} \tag{4.4.12}$$

Recall we have $|\mathcal{S}| \le \widetilde{s} = 2s + s^*$. Therefore, $\big\|\nabla_{\mathcal{S}} Q_n\big(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*\big)\big\|_2 \le \sqrt{\widetilde{s}}\big\|\nabla_1 Q_n\big(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*\big)\big\|_\infty$. Insert this into (4.4.12) and we complete the proof of (4.4.2).

For the convergence coefficient $\rho^{1/2}$, we have

$$\rho = \frac{1 + 2T\alpha\eta[\eta L^2 + (2\eta + L/\mu^2)\gamma^2]}{T(1-\tau)} = \underbrace{\frac{1}{T(1-\tau)}}_{\rho_1} + \underbrace{\frac{2\alpha\eta\big[\eta L^2 + (2\eta + L/\mu^2)\gamma^2\big]}{1-\tau}}_{\rho_2},$$

where $\tau = \alpha(1 - \eta\mu + 2\eta^2 L^2)$. We need to confine

$$\tau = \alpha(1 - \eta\mu + 2\eta^2 L^2) < 1,$$

and since ρ_1 can be bounded sufficiently small by T, we need

$$\rho_2 = \frac{2\alpha\eta\big[\eta L^2 + (2\eta + L/\mu^2)\gamma^2\big]}{1 - \tau} < 1.$$

We insert $\eta = \mu/8L^2$ to obtain

$$\tau = \alpha(1 - \eta\mu + 2\eta^2 L^2) = \alpha\bigg(1 - \frac{3\mu^2}{32L^2}\bigg).$$

We let $\tau < 1$, which means

$$\alpha < \frac{1}{1 - 3\mu^2/32L^2},$$

and further gives us

$$s > \bigg[\frac{4(32L^2 - 3\mu^2)^2}{9\mu^4} + 1\bigg]s^*. \tag{4.4.13}$$

We can also obtain

$$\rho_2 = \frac{\alpha\mu^2/32L^2 + \alpha\mu^2\gamma^2/16L^4 + \alpha\gamma^2/4L\mu}{1 - \alpha(1 - 3\mu^2/32L^2)},$$

which means we need

$$\alpha\bigg(1 - \frac{5\mu^2}{96L^2} + \frac{\mu^2\gamma^2}{12L^4} + \frac{\gamma^2}{3L\mu}\bigg) < 1$$

to make $\rho_2 < 3/4$. This further requires

$$-\frac{5\mu^2}{96L^2} + \frac{\mu^2\gamma^2}{12L^4} + \frac{\gamma^2}{3L\mu} < 0,$$

which gives us

$$\gamma < \sqrt{\frac{5\mu^3 L^2}{8\mu^3 + 32L^3}}. \tag{4.4.14}$$

This is our requirement on γ. Finally, we need

$$\alpha < \frac{1}{1 - 5\mu^2/96L^2 + \mu^2\gamma^2/12L^4 + \gamma^2/3L\mu},$$

to guarantee $\rho_2 < 3/4$. This requires

$$s > \left[\frac{4(1-K)^2}{K^2} + 1\right] s^*, \tag{4.4.15}$$

where

$$K = \frac{5\mu^2}{96L^2} - \frac{\mu^2\gamma^2}{12L^4} - \frac{\gamma^2}{3L\mu} > 0.$$

We then let $\rho_1 < 1/8$, which means

$$\frac{1}{T[1 - \alpha(1 - 3\mu^2/32L^2)]} < \frac{1}{8}.$$

We further get

$$T > \frac{8}{1 - \alpha(1 - 3\mu^2/32L^2)} = \frac{256\kappa^2}{3\alpha - 32(\alpha - 1)\kappa^2}.$$

We make $32(\alpha - 1)\kappa^2 < 3$ by choosing sufficiently large value for s, and get

$$T > \frac{256\kappa^2}{3(\alpha - 1)}.$$

Finally, combining (4.4.13), (4.4.15) and (4.4.14), we have the convergence coefficient $\rho = \rho_1 + \rho_2 < 7/8$, with $\eta = \mu/8L^2$, and

$$\begin{aligned}
&\gamma < \sqrt{\frac{5\mu^3 L^2}{8\mu^3 + 32L^3}} \\
&s > \max\left[\left[\frac{4(1-K)^2}{K^2} + 1\right] s^*, \left[\frac{4(32L^2 - 3\mu^2)^2}{9\mu^4} + 1\right] s^*\right],
\end{aligned}$$

where

$$K = \frac{5\mu^2}{96L^2} - \frac{\mu^2\gamma^2}{12L^4} - \frac{\gamma^2}{3L\mu} > 0.$$

□

Remark 4.4.7. As suggested in Theorem 4.4.5 that by choosing appropriate learning rate η, a sufficiently large number of inner iterations T, and sparsity parameter s such that $\rho < 1$, we can achieve linear convergence rate. Here we give an example to show that such ρ is achievable. If we choose step size $\eta = \mu/(8L^2)$, and truncation parameter s satisfies

$$s > \left[\frac{4(1-K)^2}{K^2} + 1\right] s^*,$$

where

$$K = \frac{5\mu^2}{96L^2} - \frac{\mu^2\gamma^2}{12L^4} - \frac{\gamma^2}{3L\mu} > 0.$$

Then, we can get

$$\alpha < \frac{1}{1 - 5\mu^2/96L^2 + \mu^2\gamma^2/12L^4 + \gamma^2/3L\mu},$$

and the contraction parameter ρ in Theorem 4.4.5 can be simplified as

$$\rho \leq \frac{1}{T(1-\tau)} + \frac{3}{4}.$$

Therefore, if we choose $T \geq 256\kappa^2/\big(3(\alpha-1)\big)$, we can obtain $\rho \leq 7/8$, ensuring the linear convergence rate as in [27, 28].

Remark 4.4.8. The right hand side of (4.4.2) in Theorem 4.4.5 consists of two terms. The first term stands for the optimization error and the second term is the statistical error.

The computational complexity of the optimization process can be formulated by the number of gradients needed to be computed. This is also called the gradient complexity which has been studied in According to Remark 4.4.7, our algorithm is able to ensure linear convergence. Therefore, for any specific error bound $\epsilon > 0$, we actually need $r \geq 2\log_{\rho^{-1}}[\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\|_2/\epsilon]$ iterations to let the optimization error $\rho^{r/2}\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\|_2 \leq \epsilon$, which basically requires $O\big(\log(1/\epsilon)\big)$ outer iterations.

For each outer iteration, we need to compute T gradients of $q_i(\cdot;\cdot)$, and one full gradient. The gradient of a $q_i(\cdot;\cdot)$ depends on b component functions in a mini-batch, and one full gradient takes N component functions. Since we have $T = O(\kappa^2)$, which is suggested in Remark 4.4.7, the gradient complexity of our algorithm would be $O\big((N + b\kappa^2)\cdot\log(1/\epsilon)\big)$. Nevertheless, for the state-of-the-art gradient based high dimensional EM algorithm [27], its gradient complexity is $O\big(\kappa N\log(1/\epsilon)\big)$. As long as $\kappa \leq N/b$, the gradient complexity of our algorithm is less than that of [27].

The second term on the right-hand side of (4.4.2) stands for the upper bound of the statistical error, which depends on specific models as we will introduce later.

Algorithm 5 Accelerated Stochastic Variance Reduced Gradient EM Algorithm With Resampling

Parameter: Sparsity Parameter s, Maximum Number of Outer Iterations m, Number of Inner Iterations T, learning rate η
Initialization:
  $\widetilde{\boldsymbol{\beta}}^{(0)} = \mathcal{H}_s(\boldsymbol{\beta}^{\text{init}})$,
  Split the Dataset into m Subsets of Size N/m
For $l = 0$ to $m-1$
  E-step:
    Evaluate $Q_n\big(\boldsymbol{\beta};\widetilde{\boldsymbol{\beta}}^{(l)}\big)$ with the $(l+1)$-th Subset
    $\widetilde{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(l)}, \quad \widetilde{\mu} = \frac{1}{n/m}\sum_{i=1}^{n/m}\nabla_1 q_i(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}})$
  M-step:
    $\boldsymbol{\beta}^{(0)} = \widetilde{\boldsymbol{\beta}}$
    Randomly select j_l uniformly from $\{0,\ldots,T-1\}$
    For $t = 0$ to j_l
      Randomly select i from [n] uniformly
      $\mathbf{v}^{(t)} = \nabla_1 q_i\big(\boldsymbol{\beta}^{(t)};\widetilde{\boldsymbol{\beta}}\big) - \nabla_1 q_i\big(\widetilde{\boldsymbol{\beta}};\widetilde{\boldsymbol{\beta}}\big) + \widetilde{\mu}$,
      $\boldsymbol{\beta}^{(t+0.5)} = \boldsymbol{\beta}^{(t)} + \eta\mathbf{v}^{(t)}$,
      T-step: $\boldsymbol{\beta}^{(t+1)} = \mathcal{H}_s(\boldsymbol{\beta}^{(t+0.5)})$
    End For
    $\widetilde{\boldsymbol{\beta}}^{(l+1)} = \boldsymbol{\beta}^{(j_l)}$
End For
Output: $\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(m)}$

4.4.3 Implications on Specific Models

In this section, we will apply our general theory to two representative sparse latent variable models, GMM and MLR, described in Section 4.3.1. Specifically, for each model, we first verify the technical conditions, then analyze the bound of statistical error, i.e., $\|Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\|_\infty$, and finally propose a theorem to characterize the optimization error and statistical error of our estimator. We will show that for both GMM and MLR, our algorithm achieves linear convergence rate and optimal statistical rate of convergence up to a logarithmic factor.

4.4.3.1 Sparse Gaussian Mixture Model

Given the sparse GMM introduced in Section 4.3.1, we can obtain that

$$Q_n(\boldsymbol{\beta};\boldsymbol{\beta}') = -\frac{1}{2N'}\sum_{i=1}^{N'} \omega_{\boldsymbol{\beta}}(\mathbf{y}_i)\cdot(\mathbf{y}_i-\boldsymbol{\beta}')^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y}_i-\boldsymbol{\beta}') + \big[1-\omega_{\boldsymbol{\beta}}(\mathbf{y}_i)\big]\cdot(\mathbf{y}_i-\boldsymbol{\beta}')^\top\boldsymbol{\Sigma}^{-1}(\mathbf{y}_i-\boldsymbol{\beta}'), \quad (4.4.16)$$

$$\text{where} \quad \omega_{\boldsymbol{\beta}}(\mathbf{y}) = \frac{1}{1+\exp\big(-2\cdot\mathbf{y}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}\big)}.$$

Since we use resampling and splitting the dataset into m subsets, we use $N' = N/m$ to denote the size of each subset. From (4.4.16), we have the following lemma verifying the technical conditions in Section 4.4.1 for GMM.

Lemma 4.4.9 (Conditions for GMM)**.** Suppose we have $\{\mathbf{y}_i\}_{i=1}^N$ as N i.i.d. realizations of $\boldsymbol{Y}\in\mathbb{R}^d$ given by Gaussian mixture model defined in Section 4.3.1, then Conditions 4.4.1 to 4.4.3 hold with

$$L = \frac{1}{\lambda_{\min}(\boldsymbol{\Sigma})}, \qquad \mu = \frac{1}{\lambda_{\max}(\boldsymbol{\Sigma})},$$
$$\gamma = \frac{20}{\lambda_{\min}(\boldsymbol{\Sigma})}\cdot(\xi^2+\xi+1+\xi^{-2})e^{-\xi^2/64},$$

where $\xi = \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2$ denotes the signal-to-noise ratio (SNR).

Proof of Lemma 4.4.9. Since we have Q-function for GMM given by (4.4.16), we can easily obtain

$$\nabla_1 Q_n(\boldsymbol{\beta}';\boldsymbol{\beta}) = \frac{1}{N'}\sum_{i=1}^{N'}\big[2\cdot\omega_{\boldsymbol{\beta}}(\mathbf{y}_i)-1\big]\cdot\boldsymbol{\Sigma}^{-1}\mathbf{y}_i - \boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}'.$$

Recall L defined in Condition 4.4.1, we know

$$\big\|\nabla_1 q_i(\boldsymbol{\beta}_1;\boldsymbol{\beta}) - \nabla_1 q_i(\boldsymbol{\beta}_2;\boldsymbol{\beta})\big\|_2 = \big\|\boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)\big\|_2 \le \lambda_{\max}(\boldsymbol{\Sigma}^{-1})\big\|\boldsymbol{\beta}_1-\boldsymbol{\beta}_2\big\|_2,$$

to get that $L = \lambda_{\max}(\boldsymbol{\Sigma}^{-1})$, the largest eigenvalue of $\boldsymbol{\Sigma}^{-1}$.

Similarly, we have

$$\big[\nabla_1 Q_n(\boldsymbol{\beta}_1;\boldsymbol{\beta}) - \nabla_1 Q_n(\boldsymbol{\beta}_2;\boldsymbol{\beta})\big]^\top(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2) = (\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2) \le -\lambda_{\min}(\boldsymbol{\Sigma}^{-1})\big\|\boldsymbol{\beta}_1-\boldsymbol{\beta}_2\big\|_2$$

to get $\mu = \lambda_{\min}(\boldsymbol{\Sigma}^{-1})$ in Condition 4.4.2, where $\lambda_{\min}(\boldsymbol{\Sigma}^{-1})$ is the smallest eigenvalue of $\boldsymbol{\Sigma}^{-1}$. For the proof

Condition 4.4.3, we provide the sketch here. The basic idea is using

$$\begin{aligned}&\|\nabla_1 Q_n(\boldsymbol{\beta};\boldsymbol{\beta}_1) - \nabla_1 Q_n(\boldsymbol{\beta};\boldsymbol{\beta}_2)\|_2 \\ \leq\ &\|\nabla_1 Q(\boldsymbol{\beta};\boldsymbol{\beta}_1) - \nabla_1 Q(\boldsymbol{\beta};\boldsymbol{\beta}_2)\|_2 + \|[\nabla_1 Q(\boldsymbol{\beta};\boldsymbol{\beta}_1) - \nabla_1 Q(\boldsymbol{\beta};\boldsymbol{\beta}_2)] - [\nabla_1 Q_n(\boldsymbol{\beta};\boldsymbol{\beta}_1) - \nabla_1 Q_n(\boldsymbol{\beta};\boldsymbol{\beta}_2)]\|_2,\end{aligned}$$

where $Q(\cdot;\cdot) = \mathbb{E}[Q_n(\cdot;\cdot)]$. By dividing it into the expectation, i.e., population version and the difference between the sample and population versions. For the first part, we use the features of $Q(\cdot;\cdot)$ to derive the bound; for the second part, we bound the infinity norm of the difference vector.

Please see Section 4.7 for details. □

After verifying the technical conditions, the following lemma featuring the statistical error of sparse GMM. Specifically, we have extended the work in [27, 28] from identity covariance matrix to general positive definite matrix.

Lemma 4.4.10 (Statistical Error for GMM)**.** We have the following bound for Gaussian mixture model

$$\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\|_\infty \leq C(\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty + \sqrt{\|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}}\sigma)\sqrt{\frac{(\log d + \log(2e/\delta))\log N}{N}} \tag{4.4.17}$$

holding with probability at least $1-\delta$, where C is an absolute constant.

Proof. Please see Section 4.7. □

With the technical conditions in Section 4.4.1 verified and statistical error bounded for GMM, we have the following corollary as the implication of our algorithm on GMM:

Corollary 4.4.11. Under the same conditions of Theorem 4.4.5 and suppose

$$\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 \leq \frac{\sqrt{\lambda_{\min}(\boldsymbol{\Sigma})/\lambda_{\max}(\boldsymbol{\Sigma})}}{4}\big\|\boldsymbol{\beta}^*\big\|_2.$$

Then with probability at least $1-2e/d$, the estimator $\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(m)}$ from Algorithm 5 satisfies

$$\mathbb{E}\big\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\big\|_2 \leq \rho^{m/2}\big\|\boldsymbol{\beta}^{\text{init}} - \boldsymbol{\beta}^*\big\|_2 + C\lambda_{\min}(\boldsymbol{\Sigma})\kappa^{3/2}\big(\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty + \sqrt{\|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}}\sigma\big)\sqrt{\frac{s^*\log d\cdot\log N}{N}} \tag{4.4.18}$$

where $\kappa = L/\mu$ is the condition number and C is an absolute constant.

Proof Sketch. We provide the proof sketch here. For sparse Gaussian mixture model, we have Conditions 4.4.1 to 4.4.3 hold with parameters $L = 1/\lambda_{\min}(\boldsymbol{\Sigma})$, $\mu = 1/\lambda_{\max}(\boldsymbol{\Sigma})$, and $\gamma = 20(\xi^2 + \xi + 1 +$

$\xi^{-2})e^{-\xi^2/64}/\lambda_{\min}(\mathbf{\Sigma})$, where $\xi = \|\mathbf{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2$ denotes the signal-to-noise ratio (SNR). Next, $\widetilde{s} = 2s + s^*$ is of the same order as s^*.

For the term $\|\nabla_1 Q_n(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*)\|_\infty$ in (4.4.2), we apply Lemma 4.4.10 and complete the proof. For the technical details, please see Section 4.7. □

Remark 4.4.12. From Lemma 4.4.9, we can see that the parameters in Condition 4.4.1 and 4.4.2 are determined by covariance matrix $\mathbf{\Sigma}$, which is reasonable because $\mathbf{\Sigma}$ actually denotes the variance of the data. For Condition 4.4.3, we need to introduce the signal-to-noise ratio (SNR). The concept of SNR in parameter estimation is also proposed in [26, 74]. Since we have extended the covariance matrix of noise from identity matrix in previous work to any positive definite matrix, our SNR is also a little bit different from their definition. Generally speaking, for GMM with lower SNR, the variance of the noise makes it harder or even impossible for the algorithm to converge. Therefore, it is always reasonable to have a requirement for the SNR of GMM to be large enough for reliable parameter estimation. Spectral method [40] can be used to match the requirement on initialization for GMM, however, we find that random initialization also performs reasonably well in practice as we will show later.

According to Remark 4.4.7, by choosing appropriate learning rate η, inner iterations T, and sparsity parameter s, we can ensure linear convergence rate of our algorithm. Therefore, from Corollary 4.4.11, we know that after $O\big(\log\big(N/(s^* \log d \log N)\big)\big)$ number of iterations, the output of our algorithm attains $O(\sqrt{s^* \log d \cdot \log N/N})$ statistical error, which matches the best-known error bound [27, 28] for Gaussian mixture model up to a logarithmic factor $\log N$. Note that the extra logarithmic factor is due to the resampling strategy in Algorithm 5.

4.4.3.2 Sparse Mixture of Linear Regression

For sparse MLR, we let $N' = N/m$ be the size of a subset, $y_1, \ldots, y_{N'}$ and $\mathbf{x}_1, \ldots, \mathbf{x}_{N'}$ be the N' realizations of Y and $\boldsymbol{X}$ of mixture of linear regression defined in Section 4.3.1. We have the following $Q_n(\cdot;\cdot)$ function

$$Q_n(\boldsymbol{\beta}'; \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} q_i(\boldsymbol{\beta}'; \boldsymbol{\beta}) = -\frac{1}{2N'}\sum_{i=1}^{N'} \omega_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i)\cdot\big(y_i - \langle \mathbf{x}_i, \boldsymbol{\beta}'\rangle\big)^2 + \big[1 - \omega_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i)\big]\cdot\big(y_i + \langle \mathbf{x}_i, \boldsymbol{\beta}'\rangle\big)^2, \tag{4.4.19}$$

$$\text{where}\quad \omega_{\boldsymbol{\beta}}(\mathbf{x}, y) = \frac{1}{1 + \exp\big(-2y\cdot \mathbf{x}^\top\boldsymbol{\beta}/\sigma^2\big)}. \tag{4.4.20}$$

We further have

$$\nabla_1 Q_n(\boldsymbol{\beta}';\boldsymbol{\beta}) = \frac{1}{N'}\sum_{i=1}^{N'}\Big[\big[2\omega_{\boldsymbol{\beta}}(\mathbf{x}_i,y_i)-1\big]y_i\mathbf{x}_i - \mathbf{x}_i\cdot\mathbf{x}_i^\top\cdot\boldsymbol{\beta}'\Big].$$

Similar to our analysis for GMM, we first verify the technical conditions for MLR by the following lemma.

Lemma 4.4.13 (Conditions for MLR)**.** For mixture of linear regression defined in Section 4.3.1, then Conditions 4.4.1 to 4.4.2 hold with

$$L = 2\lambda_{\max}(\boldsymbol{\Sigma}), \quad \mu = \lambda_{\min}(\boldsymbol{\Sigma})/2, \quad \gamma = \gamma_1\lambda_{\max}(\boldsymbol{\Sigma}),$$

where $\gamma_1 \in (0, 1/3)$ is a constant.

Proof. For the sake of simplicity, we use

$$\widehat{\boldsymbol{\Sigma}}_{N'} = \frac{1}{N'}\sum_{i=1}^{N'}\mathbf{x}_i\cdot\mathbf{x}_i^\top, \quad \widehat{\boldsymbol{\Sigma}}_i = \frac{1}{b}\sum_{j\in\mathcal{D}_i}\mathbf{x}_j\cdot\mathbf{x}_j^\top$$

to denote the sample covariance matrix of $\{\mathbf{x}_i\}_{i=1}^{N'}$ and $\{\mathbf{x}_j\}_{j\in\mathcal{D}_i}$.

With the Q-function on the sample given by (4.4.19), we can obtain

$$\begin{aligned}
\big\|\nabla_1 q_i(\boldsymbol{\beta}_1;\boldsymbol{\beta}) - \nabla_1 q_i(\boldsymbol{\beta}_2;\boldsymbol{\beta})\big\|_2 &= \bigg\|\frac{1}{b}\sum_{j\in\mathcal{D}_i}\mathbf{x}_j\cdot\mathbf{x}_j^\top(\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)\bigg\|_2 = \big\|\widehat{\boldsymbol{\Sigma}}_i(\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)\big\|_2 \\
&\le \big\|(\widehat{\boldsymbol{\Sigma}}_i-\boldsymbol{\Sigma})(\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)\big\|_2 + \big\|\boldsymbol{\Sigma}(\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)\big\|_2 \\
&\le \bigg[C\sqrt{\frac{d}{b}} + \lambda_{\max}(\boldsymbol{\Sigma})\bigg]\big\|\boldsymbol{\beta}_1-\boldsymbol{\beta}_2\big\|_2,
\end{aligned}$$

where the last inequality holds with probability of at least $1 - C/d$, coming from Lemma C.1 in [75]. We let $b > C^2 d/\lambda_{\max}^2(\boldsymbol{\Sigma})$ to get $L = 2\lambda_{\max}(\boldsymbol{\Sigma})$.

For Condition 4.4.2, we have

$$\begin{aligned}
\big[\nabla_1 Q_n(\boldsymbol{\beta}_1;\boldsymbol{\beta}) - \nabla_1 Q_n(\boldsymbol{\beta}_2;\boldsymbol{\beta})\big]^\top(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2) &= \bigg[\frac{1}{N'}\sum_{i=1}^{N'}\mathbf{x}_i\cdot\mathbf{x}_i^\top(\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)\bigg]^\top(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2)\\
&= -(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2)^\top\widehat{\boldsymbol{\Sigma}}_{N'}^\top(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2)\\
&= -(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2)^\top\widehat{\boldsymbol{\Sigma}}_{N'}(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2)\\
&= -(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2)^\top(\widehat{\boldsymbol{\Sigma}}_{N'}-\boldsymbol{\Sigma})(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2) - (\boldsymbol{\beta}_1-\boldsymbol{\beta}_2)^\top\boldsymbol{\Sigma}(\boldsymbol{\beta}_1-\boldsymbol{\beta}_2)\\
&\le \|\widehat{\boldsymbol{\Sigma}}_{N'}-\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}_1-\boldsymbol{\beta}_2\|_2^2 - \lambda_{\min}(\boldsymbol{\Sigma})\|\boldsymbol{\beta}_1-\boldsymbol{\beta}_2\|_2^2\\
&\le \bigg[C\sqrt{\frac{d}{N'}} - \lambda_{\min}(\boldsymbol{\Sigma})\bigg]\|\boldsymbol{\beta}_1-\boldsymbol{\beta}_2\|_2^2,
\end{aligned}$$

where the last inequality holds with probability of at least $1-C/d$ coming from Lemma C.1 in [73]. We can get $\mu = \lambda_{\min}(\boldsymbol{\Sigma})/2$ with $N' > 4C^2d/\lambda_{\min}^2(\boldsymbol{\Sigma})$.

For Condition 4.4.3, we provide the proof sketch here. Similar to GMM, we divide the difference vector on sample into difference vector on population and the difference between sample and population. We provide the technical details in Section 4.7. □

The next lemma characterizes the statistical error of sparse MLR.

Lemma 4.4.14 (Statistical Error for MLR)**.** We have the following bound for mixture of linear regression

$$\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\|_\infty \le C(\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2 + \sqrt{\|\boldsymbol{\Sigma}\|_2}\sigma)\sqrt{\frac{(\log d + \log(6/\delta))\log N}{N}}, \tag{4.4.21}$$

holding with probability at least $1-\delta$, where C is an absolute constant.

Proof. Please see Section 4.7. □

With the verified technical conditions and bounded statistical error, the implication of our main theory for mixture of linear regression is presented in the following corollary.

Corollary 4.4.15. Under the same conditions of Theorem 4.4.5 and suppose

$$\big\|\boldsymbol{\beta}^{\text{init}}-\boldsymbol{\beta}^*\big\|_2 \le \frac{\sqrt{\lambda_{\min}(\boldsymbol{\Sigma})/\lambda_{\max}(\boldsymbol{\Sigma})}}{32}\big\|\boldsymbol{\beta}^*\big\|_2.$$

Then with probability at least $1-2e/d$, the estimator $\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(m)}$ from Algorithm 5 satisfies

$$\mathbb{E}\big\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*\big\|_2 \le \rho^{m/2}\big\|\boldsymbol{\beta}^{\text{init}}-\boldsymbol{\beta}^*\big\|_2 + C\kappa^{3/2}\bigg(\|\boldsymbol{\beta}^*\|_2 + \frac{\sigma}{\sqrt{\lambda_{\max}(\boldsymbol{\Sigma})}}\bigg)\sqrt{\frac{s^*\log d\cdot\log N}{N}},$$

where $\kappa = L/\mu$ is the condition number and C is an absolute constant.

Proof Sketch. For mixture of linear regression, we have Conditions 4.4.1 to 4.4.3 hold with parameters $L = 2\lambda_{\max}(\mathbf{\Sigma})$, $\mu = \lambda_{\min}(\mathbf{\Sigma})/2$, and $\gamma = \gamma_1\lambda_{\max}(\mathbf{\Sigma})$ according to Lemma 4.4.13. We also show that $\widetilde{s}$ and s^* are of the same order in Remark 4.4.7. Next, for the term $\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\|_\infty$ in (4.4.2), we apply Lemma 4.4.14. This completes the proof. For more technical details, please see Section 4.7. □

Remark 4.4.16. According to Remark 4.4.7, our algorithm can achieve linear convergence rate with appropriate learning rate η, inner iterations T, and sparsity parameter s. Thus Corollary 4.4.15 tells us that after $O\big(\log\big(N/(s^*\log d\log N)\big)\big)$ number of outer iterations, the output of our algorithm achieves $O(\sqrt{s^*\log d\cdot\log N/N})$ statistical error, which matches the best-known statistical error [28] for mixture of linear regression up to a logarithmic factor from the resampling strategy. Specifically, the dependence on $\|\boldsymbol{\beta}^*\|_2$ is due to the fundamental limits of EM, which also appears in [26, 28]. There is also spectral method [41] helping the initialization of MLR, but we use random initialization which also performs well in our experiments.

4.5 Experiment Results

In this section, we present the results of numerical experiments to backup our theory. We use Gaussian mixture model and mixture of linear regression for parameter estimation, and compare our proposed accelerated stochastic gradient EM algorithm (VRGEM) with two state-of-the-art high dimensional EM algorithms as baselines:

- (HDGEM) High Dimensional Gradient EM algorithm proposed in [27]: the gradient variant of high dimensional EM method enforcing sparsity structure.
- (HDREM) High Dimensional Regularized EM algorithm proposed in [28]: the method based on decaying regularization.

It is worth noting that the truncation step in our algorithm is designed to enforce sparsity and combat dimensionality. Therefore, our algorithm (VRGEM) is also able to work in the low-dimensional regime naturally by removing the truncation step. However, given that high dimensional scenario is much more challenging, we only compare our algorithm with high dimensional EM algorithms to validate its efficacy.

4.5.1 Experimental Setup

For each latent variable model, we compare both (1) the **optimization error** $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ featuring the convergence of the estimator to the local optima, and (2) the **overall estimation error** $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ featuring the overall estimation accuracy with regard to the true model parameter $\boldsymbol{\beta}^*$. We also show the convergence comparison in terms of training time.

All the comparisons are under two different parameter settings: $s^* = 5$, $d = 256$, $b = 100$, $N = 5000$ and $s^* = 10$, $d = 512$, $b = 200$, $N = 10000$. For VRGEM, we choose $m = 30$, $n = 50$ and $T = 50$ across all settings and models. Besides the comparison of different algorithms, we also verify our statistical rate of convergence by plotting the statistical error $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|$ against $\sqrt{s^* \log d / N}$. Specifically, we fix $d = 512$ and show the plots of three cases $s^* = 5$, $s^* = 10$ and $s^* = 15$ with varying N.

In each experiment setting, we run 100 trials and show the averaged results. The learning rate η is tuned by grid search and s is chosen by cross validation. We use random initialization.

4.5.2 Gaussian Mixture Model

We test VRGEM on sparse Gaussian mixture models introduced in Section 4.3.1. For the sake of simplicity and better matching the problem setting of the baseline methods, the covariance matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{V}$ is chosen to be a diagonal matrix with all elements being 1, except two randomly selected elements set to $\lambda_{\max}(\boldsymbol{\Sigma}) = 10$, and another two randomly selected elements set to $\lambda_{\min}(\boldsymbol{\Sigma}) = 0.1$. For the true model parameter $\boldsymbol{\beta}^*$, we randomly choose s^* out of d entries and assign random values to them. All the other entries are zeros. The results are shown in Figure 4.1.

From Figure 4.1(a), we can see that for both parameter settings, all three algorithms have linear convergence as Corollary 4.4.11 states. VRGEM clearly enjoys a faster convergence rate than the baselines. Moreover, as shown in Figure 4.1(b), the performance on overall estimation error of our algorithm is comparable with HDGEM, which is far better than HDREM. In terms of time consumption, our algorithm also enjoys a remarkable advantage over the baselines as shown in Figure 4.1(c) and 4.1(d).

4.5.3 Mixture of Linear Regression

Similar to the setting for GMM, we use the same covariance matrix $\boldsymbol{\Sigma}$ in Section 4.5.2 for $\boldsymbol{X}$ here. We also use the same way of generating $\boldsymbol{\beta}^*$. For V, we let $\sigma = 1$. We show the results in Figure 4.3.

From Figure 4.3(a), we can see that VRGEM achieves linear convergence which is consistent with Corollary 4.4.15, and our algorithm significantly outperforms the baselines in terms of optimization error. In terms of overall estimation error shown in Figure 4.3(b), VRGEM is as good as HDGEM and beats HDREM

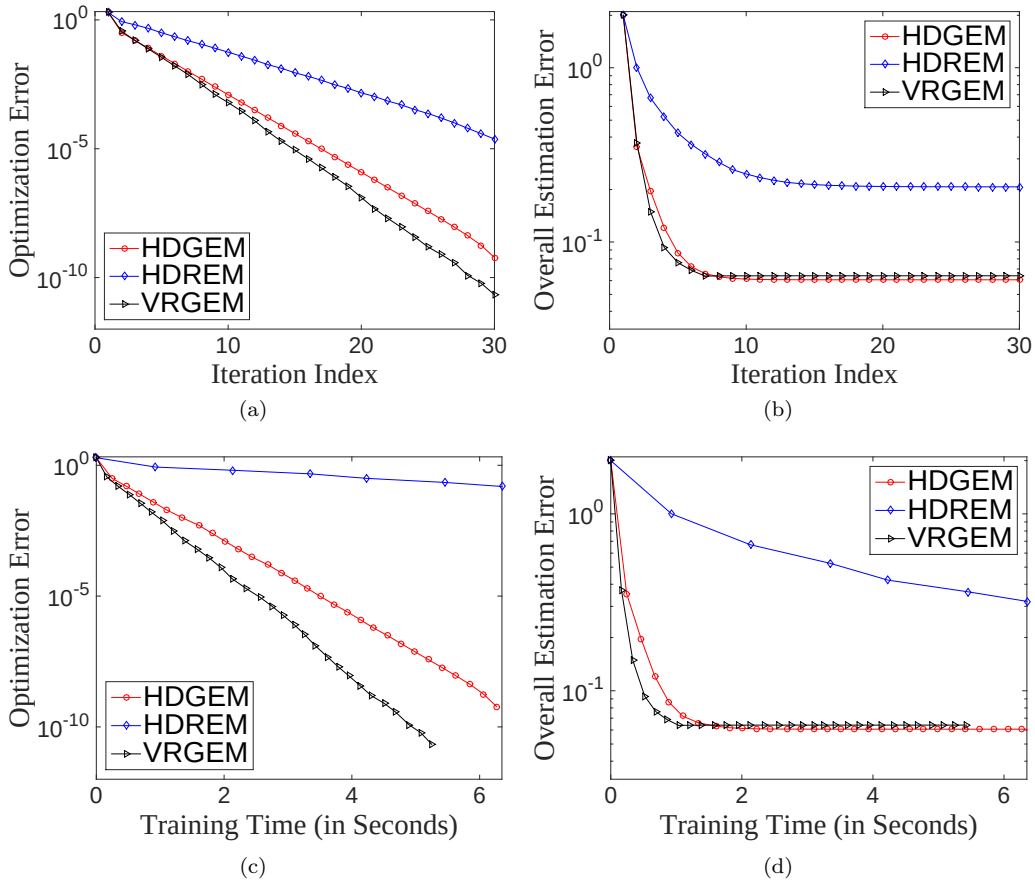


Figure 4.1: Comparison of optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ and overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ for GMM. $s^* = 5$, $d = 256$, $b = 100$, $N = 5000$. (a) (b) errors against iterations, (c) (d) errors against training time.

by a remarkable margin. Our algorithm also beats the baselines in time consumption for convergence as we can see in Figure 4.3(c) 4.3(d). Overall, VRGEM achieves the best performance among all the methods.

4.5.4 Statistical Rate of Convergence

In this section, we look into statistical errors of the models. From Corollary 4.4.11 and 4.4.15, we know that our estimator achieves the optimal statistical rate for GMM and MLR. Note that we do not have the logarithmic factor here since we do not need the resampling process in the experiments. Therefore, for both GMM and MLR, the statistical rate of convergence, i.e., order of statistical error of our estimator should be $O(\sqrt{s^* \log d/N})$

The statistical error results are shown in Figure 4.5. From Figure 4.5(a), we can clearly see that for GMM, the statistical error of VRGEM shows a linear dependency on $\sqrt{s^* \log d/N}$ across different settings of s^*, verifying results in Corollary 4.4.11.

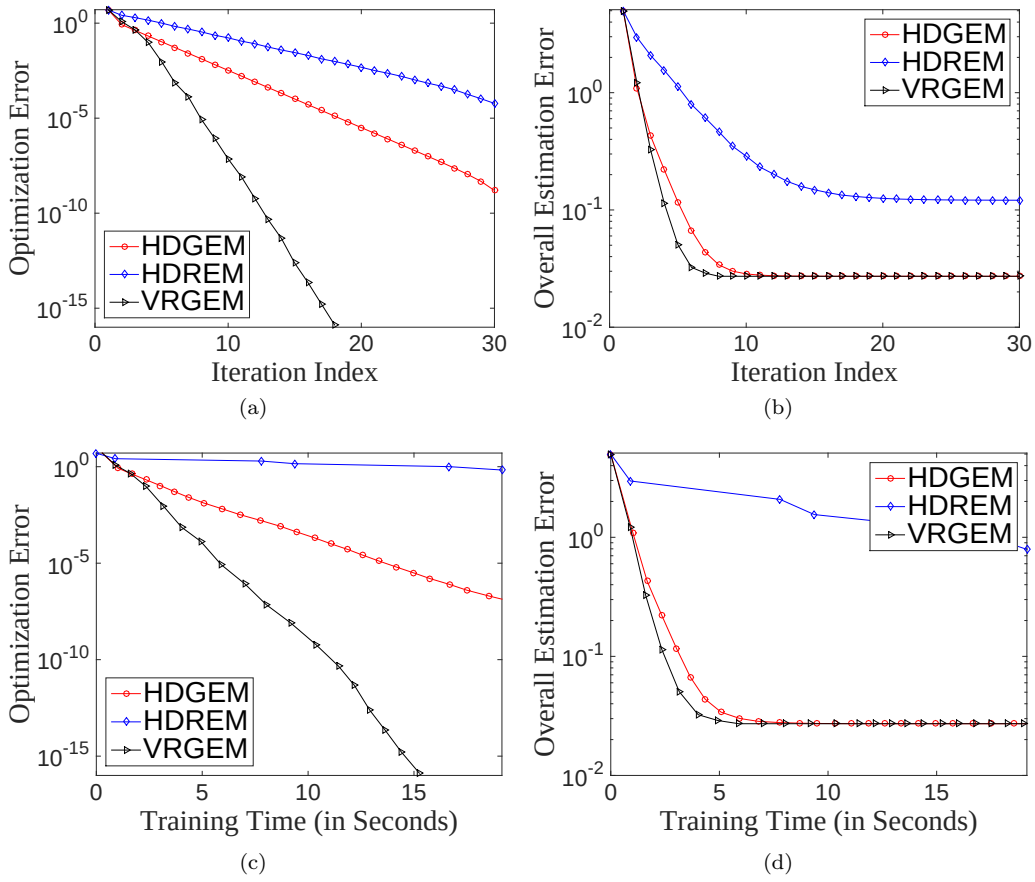


Figure 4.2: Comparison of optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ and overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ for GMM. $s^* = 10$, $d = 512$, $b = 200$, $N = 10000$. (a) (b) errors against iteration, (c) (d) errors against training time.

From Figure 4.5(b), we can see that for MLR, statistical error is also of order $O(\sqrt{s^* \log d/n})$, which supports Corollary 4.4.15.

4.6 Summary

In this work, I propose an efficient semi-stochastic gradient EM algorithm with variance reduction [39]. By incorporating a truncation step (T-step) after the M-step, our algorithm can naturally enforce sparsity in the estimator and work in the challenging high dimensional regime. To the best of our knowledge, this is the first work

We testify our algorithm to two popular latent variable models and thorough numerical experiments are provided to backup our theory. In particular, we summarize our major contributions as follows:

- We propose a novel high dimensional EM algorithm by incorporating variance reduction into stochastic

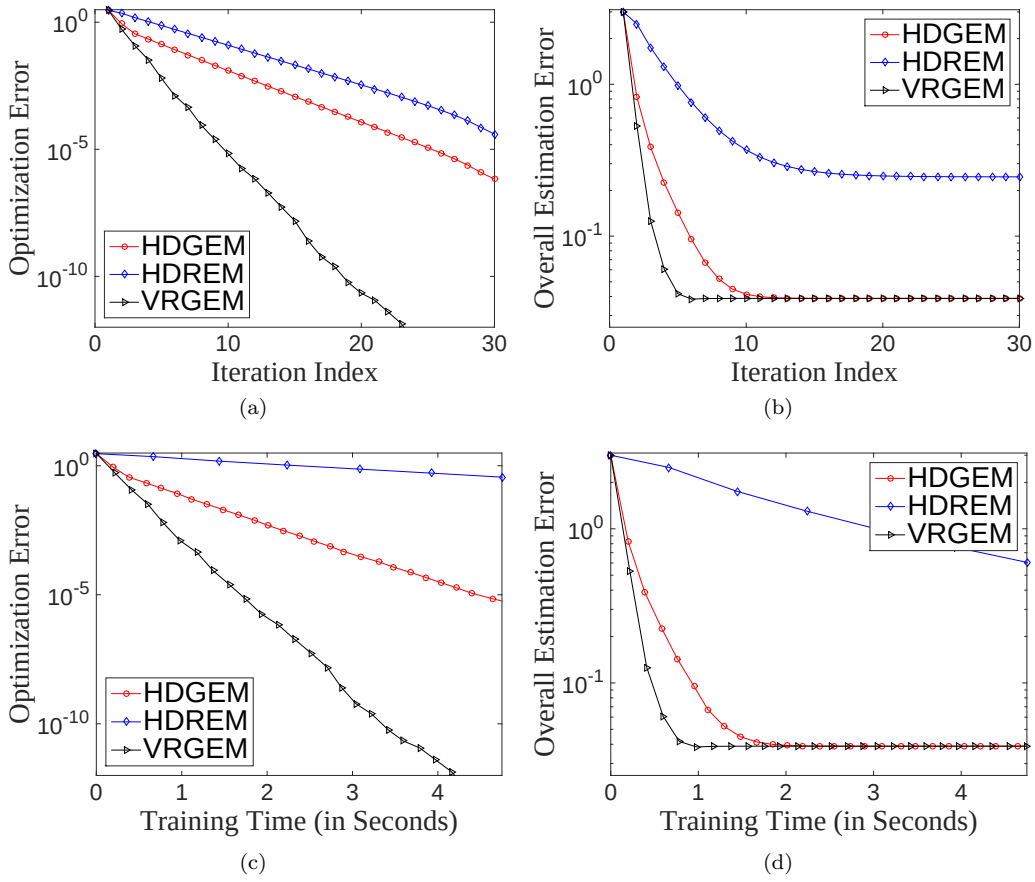


Figure 4.3: Comparison of optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ and overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ for MLR. $s^* = 5$, $d = 256$, $b = 100$, $N = 5000$. (a) (b) errors against iterations, (c) (d) errors against training time.

gradient method for EM. Specifically, we design a novel semi-stochastic gradient tailored to the bivariate structure of the Q-function in the EM algorithm. To the best of our knowledge, this is the first work ever that brings variance reduction into stochastic gradient EM algorithm in the high dimensional scenario.

- We prove that our proposed algorithm converges at a linear rate to the unknown model parameter and achieves the best-known statistical rate of convergence with a mild condition on the initialization.

- We show that the proposed algorithm has an improved overall computational complexity over the state-of-the-art algorithm. Specifically, to achieve an optimization error of ϵ, our algorithm needs $O\big((N + b\kappa^2) \cdot \log(1/\epsilon)\big)$ gradient evaluation[1], where N is the sample size, b is the mini batch size that will be discussed later, and κ is the restricted condition number. In contrast, the gradient complexity

[1]Throughout this thesis, we consider the calculation of the gradient of the Q-function over a data point as a unit gradient evaluation cost. And we use the gradient complexity, i.e., number of gradient evaluation units, to fairly compare different algorithms.

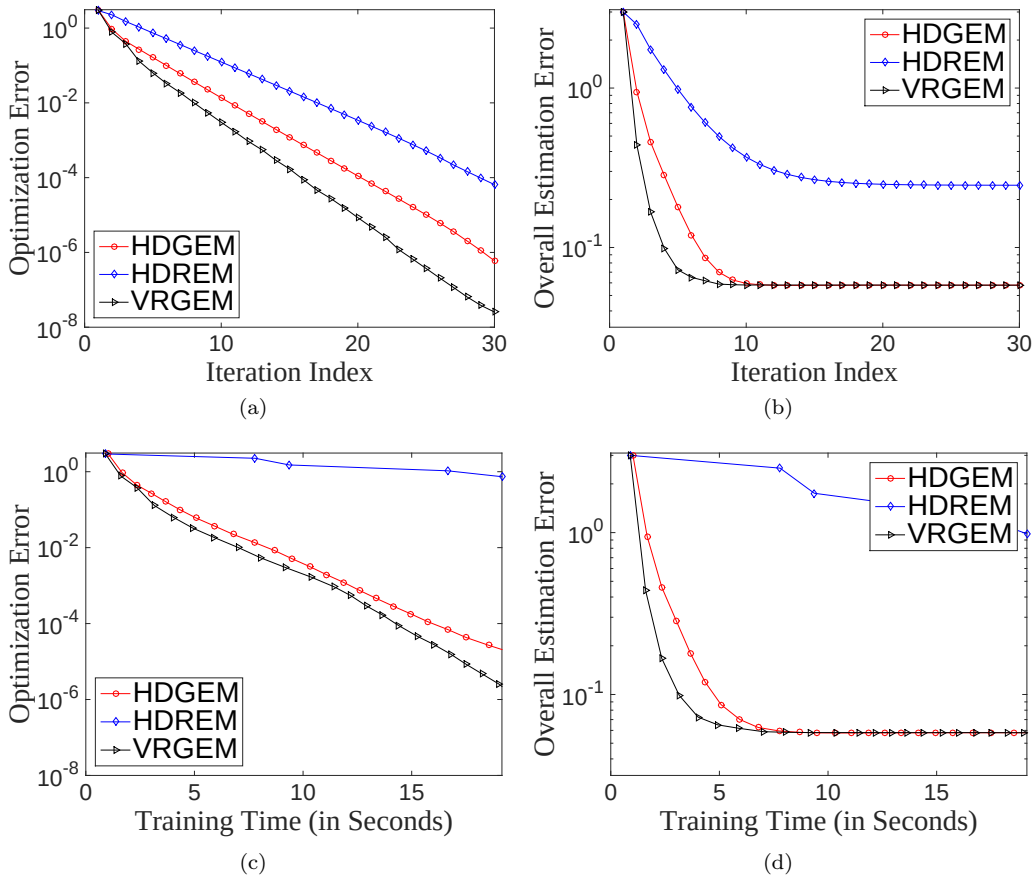


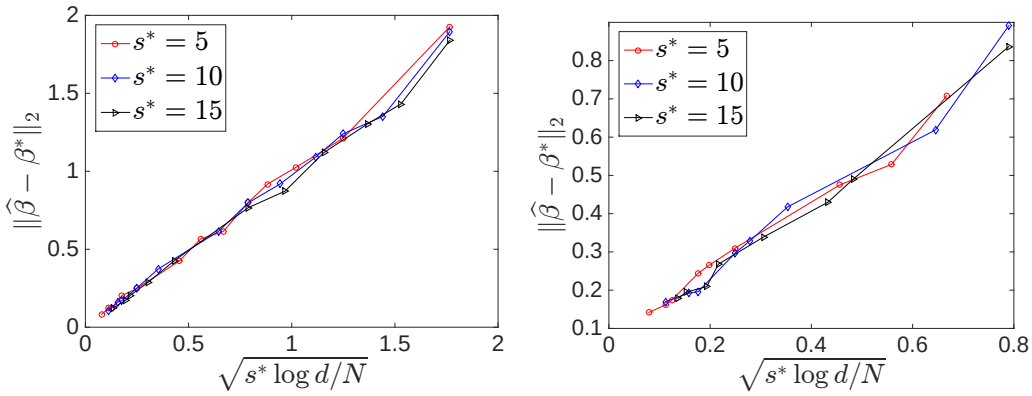
Figure 4.4: Comparison of optimization error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \widehat{\boldsymbol{\beta}}\|_2$ and overall estimation error $\|\widetilde{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}^*\|_2$ for MLR. $s^* = 10$, $d = 512$, $b = 200$, $N = 10000$. (a) (b) errors against iteration, (c) (d) errors against training time.

of the state-of-the-art high dimensional EM algorithm [27] is $O\big(\kappa N \log(1/\epsilon)\big)$. As long as $\kappa \leq N/b$, the overall gradient complexity of our algorithm is less than [27].

- Different from the proof technique used in existing work [26, 27, 28], which analyzes both the population and sample versions of the Q-function, we directly analyze the sample version of the Q-function. Our proof is much simpler and provides a good interface to analyze the semi-stochastic gradient.

4.7 Proofs and Technical Details

This section works as an auxiliary part, which contains the technical details for the lemmas in Section . Specifically, we provide the detailed proof of Condition 4.4.3 in Lemma 4.4.9 and 4.4.13, and statistical errors in Lemma 4.4.10 and 4.4.14.



(a) GMM (b) MLR

Figure 4.5: Statistical error $\|\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*\|_2$ of VRGEM against $\sqrt{s^* \log d/N}$ with fixed $d=512$ and varying s^* and N.

4.7.1 First-order Stability

In this section, we verify the first-order stability condition for GMM and MLR proposed in 4.4.9 and 4.4.13.

4.7.1.1 Proof of Lemma 4.4.9

Condition 4.4.1 and 4.4.2 have already been verified for GMM. The proof of Condition 4.4.3 is directly inspired by [26]. In particular, we extend Lemma 3 in [26] and follow a homogeneous idea in the proof.

Proof. For Condition 4.4.3, we have

$$\big\|\nabla_1 Q_n(\boldsymbol{\beta};\boldsymbol{\beta}_1) - \nabla_1 Q_n(\boldsymbol{\beta};\boldsymbol{\beta}_2)\big\|_2 = \bigg\|\frac{1}{N'}\sum_{i=1}^{N'}\big[2\omega_{\boldsymbol{\beta}_1}(\mathbf{y}_i) - 2\omega_{\boldsymbol{\beta}_2}(\mathbf{y}_i)\big]\cdot\boldsymbol{\Sigma}^{-1}\mathbf{y}_i\bigg\|_2 \le \gamma\big\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\big\|_2.$$

We have

$$\begin{aligned}
&\big\|\nabla_1 Q_n(\boldsymbol{\beta};\boldsymbol{\beta}_1) - \nabla_1 Q_n(\boldsymbol{\beta};\boldsymbol{\beta}_2)\big\|_2 \\
&= \bigg\|\frac{1}{N'}\sum_{i=1}^{N'}\big[2\omega_{\boldsymbol{\beta}}(\mathbf{y}_i) - 2\omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)\big]\cdot\boldsymbol{\Sigma}^{-1}\mathbf{y}_i\bigg\|_2 \\
&\le \underbrace{\big\|\mathbb{E}\big[2\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\big]\big\|_2}_{I_1} + \underbrace{\bigg\|\frac{2}{N'}\sum_{i=1}^{N'}[\omega_{\boldsymbol{\beta}}(\mathbf{y}_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)]\boldsymbol{\Sigma}^{-1}\mathbf{y}_i - \mathbb{E}\big[\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\big]\bigg\|_2}_{I_2}.
\end{aligned}$$

For term I_1, our proof is similar to Lemma 3 under Corollary 1 in [26]. For $u \in [0,1]$, we define $\boldsymbol{\beta}_u = \boldsymbol{\beta}^* + u\Delta$,

where $\Delta = \boldsymbol{\beta} - \boldsymbol{\beta}^*$. Applying Taylor's theorem and taking expectations, we have

$$\begin{aligned}
&\mathbb{E}\big[(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y}))\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\big] \\
&= 2\int_0^1 \mathbb{E}\Bigg[\underbrace{\frac{\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}(\boldsymbol{\Sigma}^{-1}\boldsymbol{Y})^\top}{(\exp(\boldsymbol{Y}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}_u) + \exp(-\boldsymbol{Y}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}_u))^2}}_{\Gamma_u(\boldsymbol{\Sigma}^{-1}\boldsymbol{Y})}\Bigg]\Delta\,\mathrm{d}u \\
&= 2\int_0^1 \mathbb{E}\Bigg[\frac{(\boldsymbol{\Sigma}^{-1/2})^\top(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})^\top\boldsymbol{\Sigma}^{-1/2}}{(\exp((\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})^\top\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u)) + \exp(-(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})^\top\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u))^2}\Bigg]\Delta\dot{u} \\
&= 2\int_0^1 (\boldsymbol{\Sigma}^{-1/2})^\top\mathbb{E}\Bigg[\underbrace{\frac{(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})^\top}{(\exp((\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})^\top\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u)) + \exp(-(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})^\top\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u))^2}}_{\Gamma_u(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})}\Bigg]\boldsymbol{\Sigma}^{-1/2}\Delta\,\mathrm{d}u
\end{aligned}$$

For each choice of $u \in [0,1]$, we can easily get that $\Gamma_u(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y}) = \Gamma_u(-\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})$. Note that the distribution of $\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y}$ is symmetric around zero, we know that $\mathbb{E}[\Gamma_u(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y})] = \mathbb{E}[\Gamma_u(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y}')]$, where $\boldsymbol{Y}' \sim N(\boldsymbol{\beta}^*, \boldsymbol{\Sigma})$. We further have

$$\big\|\mathbb{E}\big[(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y}))\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\big]\big\|_2 \leq 2\sup_{u\in[0,1]}\|\mathbb{E}(\Gamma_u(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y}'))\|_2\|\boldsymbol{\Sigma}^{-1}\|_2\|\Delta\|_2. \tag{4.7.1}$$

Then we go on to bound $\|\mathbb{E}(\Gamma_u(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y}'))\|_2$ uniformly over $u \in [0,1]$. Defining $\widetilde{\boldsymbol{Y}} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{Y}'$, we have $\widetilde{\boldsymbol{Y}} \sim N(\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*, \boldsymbol{I}_d)$. For any fixed value u, we let matrix $\mathbf{R}$ be an orthogonal matrix that $\mathbf{R}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u = \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\mathbf{e}_1$, where $\mathbf{e}_1 \in \mathbb{R}^d$ denotes the first canonical basis vector. Define $\boldsymbol{U} = \mathbf{R}\widetilde{Y}$, and we get that $\boldsymbol{U} \sim N(\mathbf{R}\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*, \boldsymbol{I}_d)$. With this transformation and letting U_1 be the first coordinate of $\boldsymbol{U}$, the operator norm of the matrix $\mathbb{E}[\Gamma_u(\widetilde{\boldsymbol{Y}})]$ is equal to that of

$$\begin{aligned}
D &= \mathbb{E}\Bigg[\frac{\boldsymbol{U}\boldsymbol{U}^\top}{\big(\exp\big(\boldsymbol{U}^\top\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\mathbf{e}_1\big) + \exp\big(-\boldsymbol{U}^\top\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\mathbf{e}_1\big)\big)^2}\Bigg] \\
&= \mathbb{E}\Bigg[\frac{\boldsymbol{U}\boldsymbol{U}^\top}{\big(\exp\big(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big) + \exp\big(-U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big)\big)^2}\Bigg].
\end{aligned}$$

In order to bound the operator norm of D, we define

$$\alpha_1 := \mathbb{E}\bigg[\frac{U_1^2}{\big(\exp\big(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big)+\exp\big(-U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big)\big)^2}\bigg],$$
$$\alpha_2 := \mathbb{E}\bigg[\frac{U_1}{\big(\exp\big(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big)+\exp\big(-U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big)\big)^2}\bigg],$$
$$\alpha_3 := \mathbb{E}\bigg[\frac{1}{\big(\exp\big(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big)+\exp\big(-U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big)\big)^2}\bigg],$$
$$\boldsymbol{f} := \mathbf{R}\Sigma^{-1/2}\boldsymbol{\beta}^*,$$
$$\boldsymbol{g} := [0, f_2, f_3, \ldots, f_d]^\top.$$

We also let $\mathbf{M}$ be the matrix that is identical to $\boldsymbol{I}_d$ except that the first diagonal element of $\mathbf{M}$ is zero. In terms of these quantities, we can write D as

$$\mathbf{D} = \alpha_1 \mathbf{e}_1\mathbf{e}_1^\top + \alpha_2(\mathbf{e}_1\boldsymbol{g}^\top + \boldsymbol{g}\mathbf{e}_1^\top) + \alpha_3(\boldsymbol{g}\boldsymbol{g}^\top + \mathbf{M}).$$

So we have that

$$\|\mathbf{D}\|_2 \le \|\mathbf{D}-\alpha_3\mathbf{M}\|_2 + \|\alpha_3\mathbf{M}\|_2 \le \|\mathbf{D}-\alpha_3\mathbf{M}\|_F + \alpha_3\|\mathbf{M}\|_2 \le \alpha_1 + 2\alpha_2\|\boldsymbol{g}\|_2 + \alpha_3\|\boldsymbol{g}\|_2^2 + \alpha_3\|\mathbf{M}\|_2. \tag{4.7.2}$$

In order to bound α_1, we have

$$\alpha_1 \le \mathbb{E}\bigg[\frac{U_1^2}{\exp(2U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)}\bigg].$$

We define the event $\mathcal{E} = \{U_1 \le \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2/4\}$, we condition on it and its complement to obtain

$$\alpha_1 \le \mathbb{E}\bigg[\frac{U_1^2}{\exp(2U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)}|\mathcal{E}\bigg]\mathbb{P}(\mathcal{E}) + \mathbb{E}\bigg[\frac{U_1^2}{\exp(2U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)}|\mathcal{E}^c\bigg]. \tag{4.7.3}$$

Note that for any $\mu > 0$, the function $f_1(t) = t^2/\exp(\mu t)$ achieves maxima at $t = 2/\mu$ for $t \in [0,\infty]$, and $f_1(t)$ achieves maxima at t^* for $t \in [t^*,\infty]$ and any $t^* > 2\mu$. Provided that $\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2 \cdot \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2 \ge 4$ which means that $\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2/4 \ge 2/\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2$, we then bound the two parts respectively

$$\mathbb{E}\bigg[\frac{U_1^2}{\exp(2U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)}|\mathcal{E}\bigg] \le \frac{1}{e^2\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2^2}, \tag{4.7.4}$$
$$\mathbb{E}\bigg[\frac{U_1^2}{\exp(2U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)}|\mathcal{E}^c\bigg] \le \frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{16\exp(\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2 \cdot \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2/2)}. \tag{4.7.5}$$

Note that for U_1 we have

$$\begin{aligned}\mathbb{E}[U_1] = (R\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*)^\top \mathbf{e}_1 &= (R\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u)^\top \mathbf{e}_1 + [R\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\beta}^* - \boldsymbol{\beta}_u)]^\top \mathbf{e}_1 \\ &\geq \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2 - \|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\beta}_u - \boldsymbol{\beta}^*)\|.\end{aligned}$$

Note that $\|\boldsymbol{\beta}_u - \boldsymbol{\beta}^*\|_2 \leq \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2 \leq \frac{1}{4}\sqrt{\frac{\lambda_{\min}(\boldsymbol{\Sigma})}{\lambda_{\max}(\boldsymbol{\Sigma})}}\|\boldsymbol{\beta}^*\|_2$, and $\sqrt{1/\lambda_{\max}}\|\boldsymbol{x}\|_2 \leq \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}\|_2 \leq \sqrt{1/\lambda_{\min}}\|\boldsymbol{x}\|_2$, we have

$$\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\beta}_u - \boldsymbol{\beta}^*)\|_2 \leq \sqrt{\frac{1}{\lambda_{\min}}}\|\boldsymbol{\beta}_u - \boldsymbol{\beta}^*\|_2 \leq \frac{1}{4}\sqrt{\frac{1}{\lambda_{\max}(\boldsymbol{\Sigma})}}\|\boldsymbol{\beta}^*\|_2 \leq \frac{1}{4}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2. \tag{4.7.6}$$

We also have

$$\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2 = \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^* - \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\beta}^* - \boldsymbol{\beta}_u)\|_2 \geq \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2 - \|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\beta}_u - \boldsymbol{\beta}^*)\|_2 \geq \frac{3}{4}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2, \tag{4.7.7}$$

where the last inequality comes from (4.7.6). Combining (4.7.6) and (4.7.7) we get

$$\mathbb{E}[U_1] \geq \frac{1}{2}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2.$$

Therefore, by standard Gaussian tail bounds we have

$$\mathbb{P}[\mathcal{E}] \leq \exp\left(\frac{-\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{32}\right) \tag{4.7.8}$$

Inserting (4.7.4), (4.7.5), (4.7.7) and (4.7.8) into (4.7.3), we obtain

$$\begin{aligned}\alpha_1 &\leq \frac{16}{9e^2\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}\exp\left(-\frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{32}\right) + \frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{16}\exp\left(-\frac{3\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{8}\right) \\ &\leq \left(\frac{16}{9e^2\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2} + \frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{16}\right)\exp\left(-\frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{32}\right),\end{aligned} \tag{4.7.9}$$

for any $\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}\|_2^2 \geq 16/3$.

Similarly, for α_2 we have

$$\alpha_2 = \mathbb{E}\left[\frac{U_1}{\left(\exp\left(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\right) + \exp\left(-U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\right)\right)^2}\right] \tag{4.7.10}$$

$$\leq \sqrt{\mathbb{E}[U_1^2]}\sqrt{\mathbb{E}[(\exp(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2) + \exp(-U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2))^{-4}]}. \tag{4.7.11}$$

We know that $\mathbb{E}[U_1^2] = \mathbb{E}^2[U_1] + \mathrm{Var}(U_1) \leq \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2 + 1$. For the second term, we let $f_2(t) = [\exp(t) + \exp(-t)]^{-2}$. We have $f_2^2(t) \leq 1/16$ for any t and $f_2^2(t) \leq \exp(-4t^*)$, for $t \in [t^*, \infty]$. Therefore, we condition it on $\mathcal{E}$ and obtain

$$
\begin{aligned}
&\mathbb{E}[(\exp(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2) + \exp(-U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2))^{-4}] \\
&= \mathbb{E}[f(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)] \\
&\leq \mathbb{E}[f(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)|\mathcal{E}]\mathbb{P}(\mathcal{E}) + \mathbb{E}[f(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)|\mathcal{E}^c] \\
&\leq \frac{1}{16}\mathbb{P}(\mathcal{E}) + \exp(-\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2) \\
&\leq \frac{1}{16}\exp\Big(-\frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{32}\Big) + \exp(-\frac{3\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{4}) \\
&\leq 2\exp\Big(-\frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{32}\Big).
\end{aligned} \tag{4.7.12}
$$

Inserting (4.7.5) into (4.7.10) we get that

$$
\alpha_2 \leq 2(\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2 + 1)\exp\Big(-\frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{64}\Big). \tag{4.7.13}
$$

For α_3, we have

$$
\begin{aligned}
\alpha_3 &= \mathbb{E}\Big[\frac{1}{\big(\exp\big(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big) + \exp\big(-U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\big)\big)^2}\Big] \\
&= \mathbb{E}[f(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)] \\
&\leq \mathbb{E}[f(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)|\mathcal{E}]\mathbb{P}(\mathcal{E}) + \mathbb{E}[f(U_1\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2)|\mathcal{E}^c] \\
&\leq \frac{1}{4}\mathbb{P}(\mathcal{E}) + \exp\big(\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}_u\|_2\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2/4\big) \\
&\leq \frac{1}{4}\exp\Big(-\frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{32}\Big) + \exp\Big(-\frac{3\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{4}\Big) \\
&\leq 2\exp\Big(-\frac{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2}{32}\Big)
\end{aligned} \tag{4.7.14}
$$

Inserting (4.7.9), (4.7.13) and (4.7.14) into (4.7.2), and using $\xi = \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2$ to denote the signal-to-noise

ratio, we get

$$\begin{aligned}\|\mathbf{D}\|_2 &\le \alpha_1 + 2\alpha_2\|\boldsymbol{g}\|_2 + \alpha_3\|\boldsymbol{g}\|_2^2 + \alpha_3\|\mathbf{M}\|_2 \\ &\le \alpha_1 + 2\alpha_2\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2 + \alpha_3\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2^2 + \alpha_3 \\ &\le \left[\frac{16}{9e^2\xi^2} + \frac{65\xi^2}{16} + 2\xi + 2\right]e^{-\xi^2/64} \\ &\le 5(\xi^2 + \xi + 1 + \xi^{-2})e^{-\xi^2/64}, \end{aligned} \tag{4.7.15}$$

provided that $\xi^2 \ge 16/3$. Combining (4.7.15) and (4.7.1), we have

$$I_1 = \big\|\mathbb{E}\big[2\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\big]\big\|_2 \le 10\lambda_{\min}^{-1}(\boldsymbol{\Sigma}) \cdot (\xi^2 + \xi + 1 + \xi^{-2})e^{-\xi^2/64}\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2, \tag{4.7.16}$$

where $\xi = \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2$ is the signal-to-noise ratio. We can see that as long as signal-to-noise ratio is sufficiently large, the coefficient on the right-hand side of (4.7.16) before $\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2$ will be small enough.

For term (ii), we let $\boldsymbol{\phi} = \frac{2}{N'}\sum_{i=1}^{N'}[\omega_{\boldsymbol{\beta}}(\mathbf{y}_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)]\boldsymbol{\Sigma}^{-1}\mathbf{y}_i - \mathbb{E}[2\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]$. We consider ϕ_j, the $j-$th coordinate of $\boldsymbol{\phi}$, which is given by

$$\phi_j = \frac{2}{N'}\sum_{i=1}^{N'}\Big[[\omega_{\boldsymbol{\beta}}(\mathbf{y}_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)][\boldsymbol{\Sigma}^{-1}\mathbf{y}_i]_j - \mathbb{E}\big[\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big]\Big] \tag{4.7.17}$$

We know $\big\{[\omega_{\boldsymbol{\beta}}(\mathbf{y}_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)][\boldsymbol{\Sigma}^{-1}\mathbf{y}_i]_j - \mathbb{E}\big[2\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big]\big\}_{i=1}^{N'}$ are independent copies of the centered random variable given by

$$\big[\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big][\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j - \mathbb{E}\big[\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big].$$

Note that $\omega_{\boldsymbol{\beta}}(\boldsymbol{Y})$ and $\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})$ are both between 0 and 1, we know $|\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})| \in [0,1]$. Therefore, we obtain

$$\begin{aligned}&\big\|\big[\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big][\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j - \mathbb{E}\big[\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big]\big\|_{\psi_2} \\ \le\ & 2\big\|\big[\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big][\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big\|_{\psi_2} \\ \le\ & 2\|[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\|_{\psi_2} \\ =\ & 2\|Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j + [\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j\|_{\psi_2}, \end{aligned} \tag{4.7.18}$$

where the first inequality comes from $\|\boldsymbol{X} - \mathbb{E}\boldsymbol{X}\|_{\psi_2} \le 2\|\boldsymbol{X}\|_{\psi_2}$, and Z is a Rademacher random variable

and $[\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j \sim N(0, [\Sigma^{-1}]_{j,j}\sigma^2)$. Since $Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j$ and $[\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j$ are both sub-Gaussian variables with $\|Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j\|_{\psi_2} \leq |[\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j|$ and $\|[\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j\|_{\psi_2} \leq \sqrt{[\Sigma^{-1}]_{j,j}}\sigma$. By Lemma 5.9 (rotation invariance) in [69], we have

$$\begin{aligned}\|Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j + [\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j\|_{\psi_2} &\leq \sqrt{\|Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j\|_{\psi_2}^2 + \|[\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j\|_{\psi_2}^2} \\ &\leq \sqrt{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + [\boldsymbol{\Sigma}^{-1}]_{jj}\sigma^2} \\ &\leq \sqrt{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}. \end{aligned} \tag{4.7.19}$$

Combining (4.7.17), (4.7.18) and (4.7.19), and by Lemma 5.5 and Proposition 5.10 in [69], we know that there exists some constant C_1 such that for any $j \in [d]$ and all $t > 0$,

$$\mathbb{P}(|\phi_j| \geq t) \leq e \cdot \exp\bigg(- \frac{C_1 N' t^2}{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}\bigg).$$

By applying the union bound, we obtain

$$\mathbb{P}\bigg(\sup_{j\in[d]} |\phi_j| \geq t\bigg) \leq de \cdot \exp\bigg(- \frac{C_1 N' t^2}{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}\bigg).$$

Setting the right-hand side to be δ, we have the following bound

$$\begin{aligned}&\bigg\| \frac{2}{N'} \sum_{i=1}^{N'} [\omega_{\boldsymbol{\beta}}(\mathbf{y}_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)]\boldsymbol{\Sigma}^{-1}\mathbf{y}_i - \mathbb{E}\big[\big(\omega_{\boldsymbol{\beta}}(\boldsymbol{Y}) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\big)\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\big]\bigg\|_\infty \\ &\qquad \leq C(\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty + \sqrt{\|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}}\sigma)\sqrt{\frac{\log d + \log(e/\delta)}{N'}},\end{aligned}$$

holds with probability at least $1 - \delta$. For any $\boldsymbol{a} \in \mathbb{R}^d$, we know $\|\boldsymbol{a}\|_2 \leq \sqrt{d}\|\boldsymbol{a}\|_\infty$, which means

$$I_2 \leq C(\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty + \sqrt{\|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}}\sigma)\sqrt{\frac{d(\log d + \log(e/\delta))}{N'}}, \tag{4.7.20}$$

where C is an absolute constant. We can see that I_2 will be sufficiently small when N' is large enough. Therefore, we can always make $I_2 \leq I_1$. Combining (4.7.16) and (4.7.20), we have the following γ for Gaussian mixture model

$$\gamma = 20\lambda_{\min}^{-1}(\boldsymbol{\Sigma}) \cdot (\xi^2 + \xi + 1 + \xi^{-2})e^{-\xi^2/64},$$

where $\xi = \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\beta}^*\|_2$ denotes the signal-to-noise ratio.

□

4.7.1.2 Proof of Lemma 4.4.13

We have already verified Condition 4.4.1 and 4.4.2 for MLR. This following proof of Condition 4.4.3 is directly inspired by [26]. Specifically, we extend their Lemma 4 and 5 and follow a similar idea in the proof.

Proof. We have

$$
\begin{aligned}
&\left\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}) - \nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\right\|_2 \\
&= \left\| \frac{1}{N'}\sum_{i=1}^{N'} [2\omega_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i) - 2\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)] y_i \mathbf{x}_i \right\|_2 \\
&\le 2 \underbrace{\left\|\mathbb{E}[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)) Y \boldsymbol{X}]\right\|_2}_{\text{(i)}} \\
&\qquad + 2 \underbrace{\left\| \frac{1}{N'}\sum_{i=1}^{N'} [\omega_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)] y_i \mathbf{x}_i - \mathbb{E}[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)) Y \boldsymbol{X}] \right\|_2}_{\text{(ii)}}.
\end{aligned}
$$

Now we bound the two terms above respectively.

For term (i), we let $\boldsymbol{X}' = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{X}$, $\boldsymbol{\beta}' = \boldsymbol{\Sigma}^{1/2}\boldsymbol{\beta}$ and $\boldsymbol{\beta}^{*\prime} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{\beta}^*$. Note that we have $\|\boldsymbol{\beta}' - \boldsymbol{\beta}^{*\prime}\|_2 \le \|\boldsymbol{\beta}^{*\prime}\|_2/32$ in the problem setting, and $\boldsymbol{X}' \sim N(\mathbf{0}, \boldsymbol{I}_d)$. From Lemma 4 in [26], we have the following bound

$$
\begin{aligned}
\left\|\mathbb{E}[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)) Y \boldsymbol{X}]\right\|_2 &= \left\|\mathbb{E}[(\omega_{\boldsymbol{\beta}'}(\boldsymbol{X}', Y) - \omega_{\boldsymbol{\beta}^{*\prime}}(\boldsymbol{X}', Y)) Y \boldsymbol{\Sigma}^{1/2}\boldsymbol{X}']\right\|_2 \\
&\le \lambda_{\max}(\boldsymbol{\Sigma}^{1/2}) \left\|\mathbb{E}[(\omega_{\boldsymbol{\beta}'}(\boldsymbol{X}', Y) - \omega_{\boldsymbol{\beta}^{*\prime}}(\boldsymbol{X}', Y)) Y \boldsymbol{X}']\right\|_2 \\
&\le \sqrt{\lambda_{\max}(\boldsymbol{\Sigma})}\gamma_1 \|\boldsymbol{\beta}' - \boldsymbol{\beta}^{*\prime}\|_2 \\
&= \sqrt{\lambda_{\max}(\boldsymbol{\Sigma})}\gamma_1 \|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\beta} - \boldsymbol{\beta}^*)\|_2 \\
&\le \lambda_{\max}(\boldsymbol{\Sigma})\gamma_1 \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2, \qquad (4.7.21)
\end{aligned}
$$

with a $\gamma_1 < 1/4$.

For term (ii), our proof goes similar with the proof in Section 4.7.1.1. We let $\boldsymbol{\phi} = \frac{2}{N'}\sum_{i=1}^{N'}[\omega_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i) -$

$\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)]y_i\mathbf{x}_i - 2\mathbb{E}[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y))Y\boldsymbol{X}]$ and consider the j-th coordinate

$$\begin{aligned}\phi_j &= \frac{2}{N'}\sum_{i=1}^{N'}[\omega_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)]y_i x_{ij} - 2\mathbb{E}[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y))YX_j] \\ &= \frac{2}{N'}\sum_{i=1}^{N'}\Big[\big[\omega_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)\big]y_i x_{ij} - \mathbb{E}\big[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y))YX_j\big]\Big]\end{aligned}$$

Note that $\big\{\big[\omega_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i) - \omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)\big]y_i x_{ij} - \mathbb{E}\big[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y))YX_j\big]\big\}_{i=1}^{N'}$ are independent copies of the centered random variable given by

$$\big[\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)\big]YX_j - \mathbb{E}\big[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y))YX_j\big].$$

Note that both $\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y)$ and $\omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)$ are between 0 and 1, we know $|\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)| \in [0, 1]$. We further obtain that $\big[\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)\big]YX_j - \mathbb{E}\big[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y))YX_j\big]$ is a centered sub-exponential random variable with

$$\begin{aligned}&\big\|\big[\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)\big]YX_j - \mathbb{E}\big[(\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y))YX_j\big]\big\|_{\psi_1} \\ &\le 2\big\|\big[\omega_{\boldsymbol{\beta}}(\boldsymbol{X}, Y) - \omega_{\boldsymbol{\beta}^*}(\boldsymbol{X}, Y)\big]YX_j\big\|_{\psi_1} \le 2\|Y\boldsymbol{X}_j\|_{\psi_1} \le 2\|Y\|_{\psi_2}\cdot\|X_j\|_{\psi_2} \\ &\le \sqrt{\|\boldsymbol{\Sigma}\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2^2 + \sigma^2)}.\end{aligned}$$

where the last inequality holds because Y is sub-Gaussian with $\|Y\|_{\psi_2} \le \sqrt{\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2^2 + \sigma^2}$, and X_j is sub-Gaussian with $\|X_j\|_{\psi_2} \le \sqrt{\Sigma_{j,j}} \le \sqrt{\|\boldsymbol{\Sigma}\|_{\infty,\infty}}$. By Proposition 5.16 in [69], we have

$$\mathbb{P}(|\phi_j| \ge t) \le 2\exp\bigg(-\frac{C'N't^2}{\|\boldsymbol{\Sigma}\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2^2 + \sigma^2)}\bigg)$$

for sufficient small t. By applying the union bound we have

$$\mathbb{P}\bigg(\sup_{j\in[d]}|\phi_j| \ge t\bigg) \le 2d\exp\bigg(-\frac{C'N't^2}{\|\Sigma\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2^2 + \sigma^2)}\bigg).$$

Setting the right-hand side to be δ, we have the following bound for some absolute constant C

$$\bigg\|\frac{1}{N'}\sum_{i=1}^{N'}\big[2\cdot\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)\big]y_i\mathbf{x}_i - 2\mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y, \boldsymbol{X})Y\boldsymbol{X}]\bigg\|_\infty \le C(\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2 + \sqrt{\|\boldsymbol{\Sigma}\|_2}\sigma)\sqrt{\frac{\log d + \log(2/\delta)}{N'}} \tag{4.7.22}$$

holds with probability with at least $1-\delta$. We can see that I_2 will be sufficiently small when N' is large enough. Therefore, we can always make $I_2 \leq I_1/3 \leq (\lambda_{\max}(\boldsymbol{\Sigma})\gamma_1/3)\|\boldsymbol{\beta}-\boldsymbol{\beta}^*\|_2$. Combining (4.7.21) and (4.7.22), we have the following γ for mixture of linear regression

$$\gamma = \gamma_1 \lambda_{\max}(\boldsymbol{\Sigma}),$$

where $\gamma_1 \in (0, 1/3)$ is a constant. □

4.7.2 Statistical Error

In this section, we provide the detailed proof of statistical errors for GMM and MLR.

4.7.2.1 Proof of Lemma 4.4.10

This proof is directly inspired by [28]. We extend their Lemma 4.4 and follow a homogeneous idea in the proof.

Proof. In each outer iteration, we have $N' = N/m$ samples. Note that $\boldsymbol{\beta}^*$ is the true model parameter and $\nabla_1 Q(\boldsymbol{\beta}^*; \boldsymbol{\beta}^*) = 0$. For Gaussian mixture model, we have

$$\begin{aligned}
&\big\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_\infty \\
&= \big\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*) - \nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_\infty \\
&= \bigg\| \frac{1}{N'}\sum_{i=1}^{N'} \big[2\cdot\omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i) - 1\big]\cdot\boldsymbol{\Sigma}^{-1}\mathbf{y}_i - \boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^* - \Big[\mathbb{E}\big[(2\cdot\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})-1)\cdot\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\big] - \boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\Big]\bigg\|_\infty \\
&= \bigg\| -\frac{1}{N'}\sum_{i=1}^{N'}\boldsymbol{\Sigma}^{-1}\mathbf{y}_i + \mathbb{E}[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}] + \frac{1}{N'}\sum_{i=1}^{N'}\big[2\cdot\omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)\big]\cdot\boldsymbol{\Sigma}^{-1}\mathbf{y}_i - 2\mathbb{E}[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]\bigg\|_\infty \\
&\leq \underbrace{\bigg\|\frac{1}{N'}\sum_{i=1}^{N'}\boldsymbol{\Sigma}^{-1}\mathbf{y}_i\bigg\|_\infty}_{\text{(i)}} + \underbrace{\bigg\|\frac{1}{N'}\sum_{i=1}^{N'}\big[2\cdot\omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)\big]\cdot\boldsymbol{\Sigma}^{-1}\mathbf{y}_i - 2\mathbb{E}[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]\bigg\|_\infty}_{\text{(ii)}},
\end{aligned}$$

where the last equality holds because $\mathbb{E}[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}] = \boldsymbol{\Sigma}^{-1}\mathbb{E}[\boldsymbol{Y}] = 0$.

For term (i), we let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}\boldsymbol{Y}$, $\boldsymbol{\theta}_i = \boldsymbol{\Sigma}^{-1}\mathbf{y}_i$ and $\boldsymbol{\phi} = \frac{1}{N'}\sum_{i=1}^{N'}\boldsymbol{\Sigma}^{-1}\mathbf{y}_i = \frac{1}{N'}\sum_{i=1}^{N'}\boldsymbol{\theta}_i$. Then ϕ_j,the $j-$th coordinate of $\boldsymbol{\phi}$, is given by

$$\phi_j = \frac{1}{N'}\sum_{i=1}^{N'}\theta_{i,j}.$$

Note that

$$\boldsymbol{\Theta} = Z \cdot \boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^* + \boldsymbol{\Sigma}^{-1}\boldsymbol{V},$$

and $\{\theta_{i,j}\}_{i=1}^{N'}$ are independent and identical copies from random variable given by

$$Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j + [\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j,$$

where Z is a Rademacher random variable and $[\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j \sim N(0, [\Sigma^{-1}]_{j,j}\sigma^2)$. Since $Z\cdot[\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j$ and $[\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j$ are both sub-Gaussian variables with $\|Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j\|_{\psi_2} \leq |[\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j|$ and $\|[\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j\|_{\psi_2} \leq \sqrt{[\Sigma^{-1}]_{j,j}}\sigma$. By Lemma 5.9 (rotation invariance) in [69], we have

$$\begin{aligned}
\|Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j + [\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j\|_{\psi_2} &\leq \sqrt{\|Z \cdot [\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*]_j\|_{\psi_2}^2 + \|[\boldsymbol{\Sigma}^{-1}\boldsymbol{V}]_j\|_{\psi_2}^2} \\
&\leq \sqrt{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + [\boldsymbol{\Sigma}^{-1}]_{jj}\sigma^2} \\
&\leq \sqrt{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}. \qquad (4.7.23)
\end{aligned}$$

By Lemma 5.5 and Proposition 5.10 in [69], we know that there exists some constant C such that for any $j \in [d]$ and all $t > 0$,

$$\mathbb{P}(|\phi_j| \geq t) \leq e \cdot \exp\bigg(- \frac{CN't^2}{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}\bigg).$$

By applying the union bound, we obtain

$$\mathbb{P}\bigg(\sup_{j\in[d]} |\phi_j| \geq t\bigg) = \mathbb{P}\bigg(\bigg\|\frac{1}{N'}\sum_{i=1}^{N'} \boldsymbol{\Sigma}^{-1}\mathbf{y}_i\bigg\|_\infty \geq t\bigg) \leq de \cdot \exp\bigg(- \frac{CN't^2}{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}\bigg).$$

Setting the right-hand side to be $\delta/2$, we have the following bound

$$\bigg\|\frac{1}{N'}\sum_{i=1}^{N'} \boldsymbol{\Sigma}^{-1}\mathbf{y}_i\bigg\|_\infty \leq C_1(\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty + \sqrt{\|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}}\sigma)\sqrt{\frac{\log d + \log(2e/\delta)}{N'}}, \qquad (4.7.24)$$

holds with probability at least $1 - \delta/2$ where C_1 is an absolute constant.

For term (ii), we now let $\boldsymbol{\theta}_i = \big[\omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)\big] \cdot \Sigma^{-1}\mathbf{y}_i - \mathbb{E}[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]$, $\boldsymbol{\phi} = \frac{1}{N'}\sum_{i=1}^{N'}\big[2 \cdot \omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)\big] \cdot \Sigma^{-1}\mathbf{y}_i -$

$2\mathbb{E}[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}] = \frac{2}{N'}\sum_{i=1}^{N'}\boldsymbol{\theta}_i$, and consider the $j-$th coordinate ϕ_j

$$\phi_j = \frac{2}{N'}\sum_{i=1}^{N'}\theta_{i,j} = \frac{2}{N'}\sum_{i=1}^{N'}\Big[\omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)[\boldsymbol{\Sigma}^{-1}\mathbf{y}_i]_j - \mathbb{E}\big[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big]\Big].$$

We know that $\{\theta_{i,j}\}_{i=1}^{N'}$ are independent copies of random variable $\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j - \mathbb{E}\big[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big]$. Note that $\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y}) \in [0,1]$ and we have

$$\mathbb{P}(|\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j| \geq t) \leq \mathbb{P}(|[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j| \geq t) \leq \exp(1 - Ct^2/\|\Sigma^{-1}\boldsymbol{Y}\|_{\psi_2}^2),$$

for some absolutely constant C by Definition 5.7 and Example 5.8 in [69]. Thus by Lemma 5.5 in [69] and (4.7.23) we know that $\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j$ is sub-Gaussian with $\|\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\|_{\psi_2} \leq \|[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\|_{\psi_2} \leq \sqrt{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}$. Using Remark 5.18 in [69], we obtain

$$\big\|\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j - \mathbb{E}\big[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big]\big\|_{\psi_2} \leq 2\big\|\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big\|_{\psi_2} \leq 2\sqrt{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}.$$

By Lemma 5.5 and Proposition 5.10 in [69], we have

$$\begin{aligned}\mathbb{P}(|\phi_j| \geq t) &= \mathbb{P}\bigg(\bigg|\frac{2}{N'}\sum_{i=1}^{N'}\big[\omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)[\boldsymbol{\Sigma}^{-1}\mathbf{y}_i]_j - \mathbb{E}\big(\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})[\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]_j\big)\big]\bigg| \geq t\bigg)\\ &\leq e\cdot\exp\bigg(-\frac{C'N't^2}{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}\bigg),\end{aligned}$$

where C' is an absolute constant. By applying the union bound, we obtain

$$\mathbb{P}\bigg(\sup_{j\in[d]}|\phi_j| \geq t\bigg) \leq de\cdot\exp\bigg(-\frac{C'N't^2}{\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty^2 + \|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}\sigma^2}\bigg).$$

Setting the right-hand side to be $\delta/2$, we have the following bound

$$\begin{aligned}&\bigg\|\frac{1}{N'}\sum_{i=1}^{N'}\big[2\cdot\omega_{\boldsymbol{\beta}^*}(\mathbf{y}_i)\big]\cdot\Sigma^{-1}\mathbf{y}_i - 2\mathbb{E}[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{Y})\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}]\bigg\|_\infty\\ &\leq C_2(\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}^*\|_\infty + \sqrt{\|\boldsymbol{\Sigma}^{-1}\|_{\infty,\infty}}\sigma)\sqrt{\frac{\log d + \log(2e/\delta)}{N'}}\end{aligned} \tag{4.7.25}$$

holds with probability at least $1-\delta/2$, where C_2 is an absolute constant.

Note that $N' = N/m$, and from Remark 4.4.12 we know $m = O(\log N)$. Therefore, we have $N' = O(N/\log N)$. Combining (4.7.24), (4.7.25) and $N' = O(N/\log N)$, we can get Lemma 4.4.10. □

4.7.2.2 Proof of Lemma 4.4.14

This proof is directly inspired by [28]. Specifically, we extend their Lemma 4.9 and follow a similar idea in the proof.

Proof. For our algorithm with resampling, we have $N' = N/m$ samples in each outer iteration. For mixture of linear regression, recall that

$$\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*) = \frac{1}{N'}\sum_{i=1}^{N'}\big[\big(2\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i,y_i)-1\big)y_i\mathbf{x}_i - \mathbf{x}_i\cdot\mathbf{x}_i^\top\cdot\boldsymbol{\beta}^*\big].$$

We have

$$\begin{aligned}
&\big\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_\infty \\
&= \big\|\nabla_1 Q_n(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*) - \nabla_1 Q(\boldsymbol{\beta}^*;\boldsymbol{\beta}^*)\big\|_\infty \\
&= \Big\|\frac{1}{N'}\sum_{i=1}^{N'}\big[\big(2\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i,y_i)-1\big)y_i\mathbf{x}_i - \mathbf{x}_i\cdot\mathbf{x}_i^\top\cdot\boldsymbol{\beta}^*\big] - \big[\mathbb{E}[(2\cdot\omega_{\boldsymbol{\beta}^*}(\boldsymbol{X},Y)-1)Y\boldsymbol{X}] - \boldsymbol{\Sigma}\boldsymbol{\beta}^*\big]\Big\|_\infty \\
&= \big\| - \frac{1}{N'}\sum_{i=1}^{N'}\mathbf{x}_i\cdot\mathbf{x}_i^\top\cdot\boldsymbol{\beta}^* + \boldsymbol{\Sigma}\boldsymbol{\beta}^* + \frac{1}{N'}\sum_{i=1}^{N'}\big[2\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i,y_i)y_i\mathbf{x}_i\big] - 2\mathbb{E}[\omega_{\boldsymbol{\beta}^*}(\boldsymbol{X},Y)Y\boldsymbol{X}] - \frac{1}{n}\sum_{i=1}^{N'}y_i\mathbf{x}_i\big\|_\infty \\
&\le \underbrace{\Big\|\frac{1}{N'}\sum_{i=1}^{N'}y_i\mathbf{x}_i\Big\|_\infty}_{I_1} + \underbrace{\Big\|\frac{1}{N'}\sum_{i=1}^{N'}\big[2\cdot\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i,y_i)\big]y_i\mathbf{x}_i - 2\mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y,\boldsymbol{X})Y\boldsymbol{X}]\Big\|_\infty}_{I_2} + \underbrace{\Big\|\frac{1}{N'}\sum_{i=1}^{N'}\mathbf{x}_i\cdot\mathbf{x}_i^\top\cdot\boldsymbol{\beta}^* - \boldsymbol{\Sigma}\boldsymbol{\beta}^*\Big\|_\infty}_{I_3}.
\end{aligned}$$

For term I_1, we let $\boldsymbol{\phi} = \frac{1}{N'}\sum_{i=1}^{N'}y_i\mathbf{x}_i$ and consider the $j-$th coordinate

$$\phi_j = \frac{1}{N'}\sum_{i=1}^{N'}y_i x_{i,j}.$$

We know $\{y_i x_{i,j}\}_{i=1}^{N'}$ are independent copies of random variable $(Z\cdot\boldsymbol{X}^\top\boldsymbol{\beta}^*+V)X_j$, where Z is a Rademacher random variable, $\boldsymbol{X}\sim N(0,\boldsymbol{\Sigma})$, $X_j\sim N(0,\Sigma_{j,j})$ and $V\sim N(0,\sigma^2)$. By Lemma 5.9 in [69], $Z\cdot\boldsymbol{X}^\top\boldsymbol{\beta}^*+V$ is a sub-Gaussian random variable with $\|Z\cdot\boldsymbol{X}^\top\boldsymbol{\beta}^*+V\|_{\psi_2}\le\sqrt{\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2^2+\sigma^2}$, and $\|X_j\|_{\psi_2}\le\sqrt{\Sigma_{j,j}}$. Therefore, we obtain that $(Z\cdot\boldsymbol{X}^\top\boldsymbol{\beta}^*+V)X_j$ is sub-exponential random variable with

$$\|(Z\cdot\boldsymbol{X}^\top\boldsymbol{\beta}^*+V)X_j\|_{\psi_1}\le\sqrt{\Sigma_{j,j}(\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2^2+\sigma^2)}\le\sqrt{\|\boldsymbol{\Sigma}\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2\cdot\|\boldsymbol{\beta}^*\|_2^2+\sigma^2)},$$

by Definition 5.13 in [69]. Further by Proposition 5.16 in [69], we have

$$\mathbb{P}(|\phi_j| \geq t) \leq 2\exp\bigg(-\frac{CN't^2}{\|\boldsymbol{\Sigma}\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2 \cdot \|\boldsymbol{\beta}^*\|_2^2 + \sigma^2)}\bigg)$$

for sufficient small t. By applying the union bound we have

$$\mathbb{P}\bigg(\sup_{j\in[d]}|\phi_j| \geq t\bigg) \leq 2d\exp\bigg(-\frac{CN't^2}{\|\boldsymbol{\Sigma}\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2 \cdot \|\boldsymbol{\beta}^*\|_2^2 + \sigma^2)}\bigg).$$

Setting the right-hand side to be $\delta/3$ and noting $\|\boldsymbol{\Sigma}\|_{\infty,\infty} \leq \|\boldsymbol{\Sigma}\|_2$, we have the following bound

$$\bigg\|\frac{1}{N'}\sum_{i=1}^{N'} y_i\mathbf{x}_i\bigg\|_\infty \leq C_1(\|\boldsymbol{\Sigma}\|_2 \cdot \|\boldsymbol{\beta}^*\|_2 + \sqrt{\|\boldsymbol{\Sigma}\|_2}\sigma)\sqrt{\frac{\log d + \log(6/\delta)}{N'}} \tag{4.7.26}$$

holds with probability with at least $1-\delta/3$ for some absolute constant C_1.

For term I_2, we now let $\boldsymbol{\phi} = \frac{2}{N'}\sum_{i=1}^{N'}[\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)]y_i x_{i,j} - \mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y,\boldsymbol{X})Y\boldsymbol{X}]$ and the $j-$th coordinate is given by

$$\phi_j = \frac{2}{N'}\sum_{i=1}^{N'}[\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)]y_i x_{i,j} - \mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y,\boldsymbol{X})YX_j].$$

We know $\{\omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)y_i x_{i,j} - \mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y,\boldsymbol{X})Y\boldsymbol{X}]\}_{i=1}^{N'}$ are independent copies of random variable $\omega_{\boldsymbol{\beta}^*}(\boldsymbol{X},Y)YX_j - \mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y,\boldsymbol{X})YX_j]$, where $\omega_{\boldsymbol{\beta}^*}(Y,\boldsymbol{X}) \in [0,1]$, and we further obtain that $\omega_{\boldsymbol{\beta}^*}(\boldsymbol{X},Y)YX_j - \mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y,\boldsymbol{X})YX_j]$ is a centered sub-exponential random variable with

$$\begin{aligned}\|\omega_{\boldsymbol{\beta}^*}(\boldsymbol{X},Y)YX_j - \mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y,\boldsymbol{X})YX_j]\|_{\psi_1} &\leq 2\|\omega_{\boldsymbol{\beta}^*}(\boldsymbol{X},Y)YX_j\|_{\psi_1} \leq 2\|YX_j\|_{\psi_1} \leq 2\|Y\|_{\psi_2}\|X_j\|_{\psi_2} \\ &\leq \sqrt{\|\boldsymbol{\Sigma}\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2^2\|\boldsymbol{\beta}^*\|_2^2 + \|\boldsymbol{\Sigma}\|_2)}.\end{aligned}$$

By Proposition 5.16 in [69], we have

$$\mathbb{P}(|\phi_j| \geq t) \leq 2\exp\bigg(-\frac{C'N't^2}{\|\boldsymbol{\Sigma}\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2 \cdot \|\boldsymbol{\beta}^*\|_2^2 + \sigma^2)}\bigg)$$

for sufficient small t. By applying the union bound we have

$$\mathbb{P}\bigg(\sup_{j\in[d]}|\phi_j| \geq t\bigg) \leq 2d\exp\bigg(-\frac{C'N't^2}{\|\Sigma\|_{\infty,\infty}(\|\boldsymbol{\Sigma}\|_2 \cdot \|\boldsymbol{\beta}^*\|_2^2 + \sigma^2)}\bigg).$$

Setting the right-hand side to be $\delta/3$, we have the following bound for some absolute constant C_2

$$\left\| \frac{1}{N'} \sum_{i=1}^{N'} \big[2 \cdot \omega_{\boldsymbol{\beta}^*}(\mathbf{x}_i, y_i)\big] y_i \mathbf{x}_i - 2\mathbb{E}[\omega_{\boldsymbol{\beta}^*}(Y, \boldsymbol{X}) Y \boldsymbol{X}] \right\|_\infty \leq C_2(\|\boldsymbol{\Sigma}\|_2 \cdot \|\boldsymbol{\beta}^*\|_2 + \sqrt{\|\boldsymbol{\Sigma}\|_2}\sigma)\sqrt{\frac{\log d + \log(6/\delta)}{N'}} \tag{4.7.27}$$

holds with probability with at least $1 - \delta/3$.

For term (iii), we now let $\boldsymbol{\phi} = \frac{1}{N'} \sum_{i=1}^{N'} \mathbf{x}_i \cdot \mathbf{x}_i^\top \cdot \boldsymbol{\beta}^* - \boldsymbol{\Sigma}\boldsymbol{\beta}^*$ and the $j-$th coordinate is given by

$$\phi_j = \frac{1}{N'} \sum_{i=1}^{N'} x_{i,j} \cdot \mathbf{x}_i^\top \boldsymbol{\beta}^* - [\boldsymbol{\Sigma}\boldsymbol{\beta}^*]_j.$$

Since $\{x_{i,j} \cdot \mathbf{x}_i^\top \boldsymbol{\beta}^* - [\boldsymbol{\Sigma}\boldsymbol{\beta}^*]_j\}_{i=1}^{N'}$ are independent copies of random variable $X_j \cdot \boldsymbol{X}^\top \boldsymbol{\beta}^* - [\boldsymbol{\Sigma}\boldsymbol{\beta}^*]_j$, which we know is a centered sub-exponential random variable with

$$\begin{aligned} \|X_j \cdot \boldsymbol{X}^\top \boldsymbol{\beta}^* - [\boldsymbol{\Sigma}\boldsymbol{\beta}^*]_j\|_{\psi_1} &\leq 2\|X_j \cdot \boldsymbol{X}^\top \boldsymbol{\beta}^*\|_{\psi_1} \leq 2\|X_j\|_{\psi_2} \cdot \|\boldsymbol{X}^\top \boldsymbol{\beta}^*\|_{\psi_2} \\ &\leq 2\sqrt{\Sigma_{j,j}}\sqrt{\|\boldsymbol{\Sigma}\|_2}\|\boldsymbol{\beta}^*\|_2 \leq 2\|\boldsymbol{\Sigma}\|_2 \cdot \|\boldsymbol{\beta}^*\|_2, \end{aligned}$$

where the inequalities come from Remark 5.18, Definition 5.13 in [69] and the fact that $\|\boldsymbol{\Sigma}\|_2 \geq \|\boldsymbol{\Sigma}\|_{\infty,\infty}$. Therefore, for sufficiently small t we have

$$\mathbb{P}(|\phi_j| \geq t) \leq 2\exp\bigg(- \frac{C'' N' t^2}{\|\boldsymbol{\Sigma}\|_2^2 \|\boldsymbol{\beta}^*\|_2^2}\bigg),$$

for some absolute constant C''. By applying the union bound we obtain

$$\mathbb{P}\bigg(\sup_{j \in [d]} |\phi_j| \geq t \bigg) \leq 2d\exp\bigg(- \frac{C'' N' t^2}{\|\boldsymbol{\Sigma}\|_2^2 \|\boldsymbol{\beta}^*\|_2^2}\bigg).$$

Setting the right-hand side term to be $\delta/3$ we have the following bound

$$\left\| \frac{1}{N'} \sum_{i=1}^{N'} \mathbf{x}_i \cdot \mathbf{x}_i^\top \cdot \boldsymbol{\beta}^* - \boldsymbol{\Sigma}\boldsymbol{\beta}^* \right\|_\infty \leq C_3 \|\boldsymbol{\Sigma}\|_2 \cdot \|\boldsymbol{\beta}^*\|_2 \sqrt{\frac{\log d + \log(6/\delta)}{N'}} \tag{4.7.28}$$

holds with probability of at least $1 - \delta/3$ for some absolute constant C_3.

Note that $N' = N/m$, and from Remark 4.4.16 we know $m = O(\log N)$. Therefore, $N' = O(N/\log N)$. Combining (4.7.26), (4.7.27) and (4.7.28) together with $N' = O(N/\log N)$, we get (4.4.21). □

Chapter 5

Event Detection with Noise Reduction

In this chapter, we present our work in event detection, serving as the improvement focusing on the aspect of noise reduction. Specifically, we pioneeringly utilize the topic distribution, of which the temporal divergence can be a very good indicator of emerging events. We then propose a novel method, *TopicDiver* [76], to address the event detection problem. Specifically, we apply **sparsity-inducing** longitudinal regularization to overcome the **noises** effectively. The experimental results demonstrate that *TopicDiver* outperforms the baseline models in the measures for accuracy across various settings.

5.1 Topic Distribution

The input of a typical topic model is the text corpus, and the output includes:

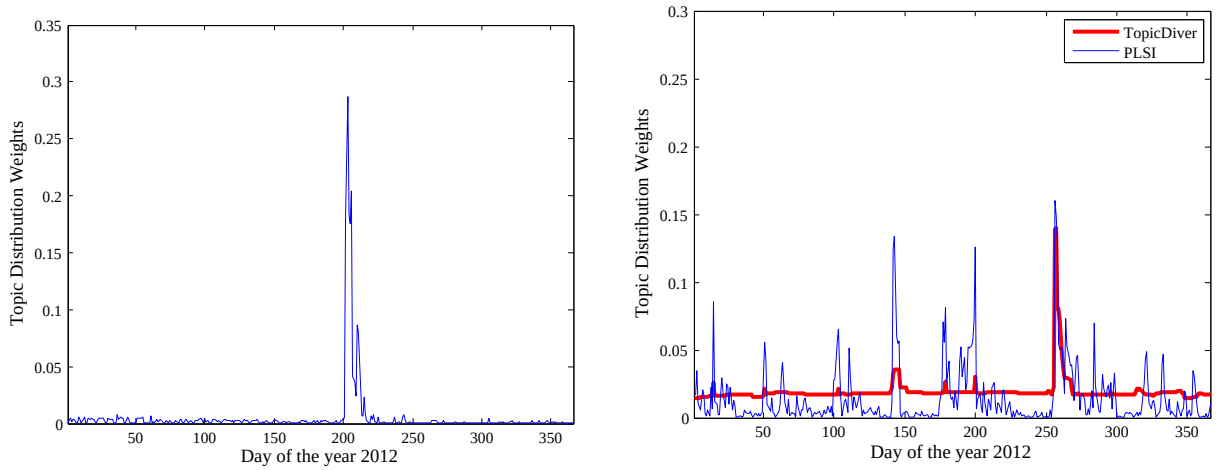
- *Word distribution* of topics $p(w|z)$: given a topic z, its probability of generating a word w in vocabulary. We have $\sum_{i=1}^{|V|} p(w|z) = 1$.
- *Topic distribution* of documents $p(z|d)$: given a document d in corpus, the probability it's about a topic z. We have $\sum_{k=1}^{K} p(z_k|d) = 1$.

For example, for a specific topic z_k about computer industry, the words with the highest probability may be $P(\text{“}computer\text{”}|z_k) = 0.05, P(\text{“}software\text{”}|z_k) = 0.04$ and $P(\text{“}technology\text{”}|z_k) = 0.02$. If a specific document d is about computer industry, then z_k may be the most relevant topic with the highest $P(z_k|d)$.

Our proposed algorithm is based on the afore-neglected topic distribution of documents over time. In this section, we will give two examples to demonstrate that utilizing this information for event detection is both promising and challenging.

Motivating Example. On July 20, 2012, a mass shooting occurred in a theater in Aurora, Colorado[1]. The topic model PLSI [53] is applied on our CNN dataset. The daily topic distribution corresponding to mass shooting around the event date is exhibited in Figure 5.1(a). The topic distribution on a specific day is

[1] https://en.wikipedia.org/wiki/2012_Aurora_shooting



(a) Topic on Aurora Shooting

(b) Topic on Libya Situations

Figure 5.1: Temporal Distribution for Two Topics

the average of the topic distributions of all documents on the day. The top words associated with this topic include "Aurora", "gunman", "theater", and "victim". The peak in Figure 5.1(a) corresponds to the exact date of the event. As the coverage of the event lingers a few days after the event date, this peak gradually disappears as the coverage goes down. Therefore, the divergence of topic distributions between adjacent time stamps may precisely indicate the occurrence of a new event.

Challenge. We highlight that capturing such temporal divergence of topic distributions is challenging. To illustrate, Figure 5.1(b) depicts the daily distribution of the topic on the situation in Libya over time. The thin blue line denotes the topic distribution of PLSI generated the same way as the mass shooting topic, where numerous peaks can be observed. However, most of these peaks are not related to events that are noteworthy. According to our manual annotation, only one event (the peak in the red curve in Figure 5.1(b), generated by *TopicDiver*) corresponds to the topic of Libya's situation. On most of the other "peaks" in the figure, the divergence is caused by updates of status, follow-ups of events, or general discussions. Different from the mass shooting topic that is about a single emergency, the Libya's situation topic is broader with multiple aspects that evolve over time. Therefore, the divergence of the Libya's situation topic distributions over adjacent time may be affected by other non-event factors. We refer to such non-event divergence as **noises** since it hampers the detection of real events. Such noise is actually very common in the detection of important events, as most of the significant events involve effects in multiple aspects, cause different follow-ups that last a long time period.

Our Contribution

We take the initiative towards exploiting the hitherto-undiscovered temporal divergence of topic distributions for event detection. Intuitively, when an event takes place, there will be a lot of documents discussing it. Therefore, the average topic distribution of the documents on the topic corresponding to the event will go up. The more significant the event is, the larger this increase should be. Compared with the word distributions of topics which are more complicated and affected by more factors, the topic distribution serves as a more straightforward sign of the change in corpus themes. While quantifying the distance of word distributions is always involved with complex measure such as KL-divergence, another advantage of topic distribution is that the difference is much simpler and easier to use.

Specifically, our contributions in this work are summarized as follows:

- We pioneeringly study the topic distributions of documents and find that their temporal divergence is a potentially useful indicator of real events.
- We propose longitudinal regularization for noise reduction in the divergence of topic distribution and propose a novel event detection algorithm, *TopicDiver*.
- We show our proposed method can effectively overcome the noise challenge and outperform the state-of-the-art methods consistently, especially at the detection of significant events.

5.2 Problem Formulation

In this section, we will formalize our event detection problem. The input for the event detection problem is a time-stamped text stream, represented by a collection of documents over a set of time stamps. These time stamps can be at any reasonable granularity based on the data. For example, if we want to analyze the scientific literature in computer science over the past few decades, year should be an appropriate time unit here; but if our data is Twitter stream generated at a very high rate, we might use hour as the time unit. The time stamps are denoted by $t_1, t_2, \ldots, t_T$, and the collection of documents published on t_i is denoted by $C_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,N_i}\}$, where $d_{i,l}$ is the l-th document on this time stamp. The input collection is denoted by $D = \{C_1, \ldots, C_T\}$.

The output should be a set of detected *events*, where each event is denoted by one of the following two:

- A *first story* $d_{t,k}$, the first document discussing the event in D.
- A set of words $\{w_1, w_2, \ldots w_M\}$ that can describe the event.

As we will show later, *TopicDiver* can be conveniently adapted into both settings. We have two settings of event detection:

- Retrospective: we have the entire collection D available and want to detect events on all the time stamps $\{t_1, \ldots, t_T\}$.
- Online: we have the documents up to time t_i, *i.e.* $C_1, \ldots, C_i$ available when the detection time is t_i.

From the above description, we can see that if the time granularity is fine enough, both settings are at the document level, *i.e.* each C_i has only one document and we determine whether it is a new event right on its published time.

5.3 Proposed Method

In this section, we will describe our proposed method, *TopicDiver*. We will start with PLSI, one of the most popular topic models, and then build *TopicDiver* on top of it.

5.3.1 Probabilistic Latent Semantic Indexing (PLSI)

PLSI is a widely-used model analyzing the hidden topics of text corpus, featured by latent variables. Specifically, given a co-occurrence of a word, document pair (w, d), the probability of the pair is modeled as the mixture of K different topics:

$$P(w, d) = P(d) \sum_{k=1}^{K} P(w|z_k) P(z_k|d),$$

where each z_k is a hidden topic, and $P(w|z_k)$, $P(z_k|d)$ are what we refer as word distribution of topic z_k and topic distribution of document d respectively.

While most existing methods look into the divergence of $P(w|z_k)$ at adjacent time stamps to detect events, we use $P(z_k|d)$ over time. For example, when generating the curve in Figure 5.1(a) for retrospective event detection, we first run PLSI on the whole corpus to generate $P(z_k|d)$ for all document d. For each time stamp t_i, we compute $P(z_k|t_i)$ as the average of $P(z_k|d_{i,l})$ and plot it against time. We then use the criterion $P(z_k|t_i) > \mu P(z_k|t_{i-1})$ to easily check if there is an event on t_i corresponding to topic z_k, where μ is a predefined threshold parameter.

5.3.2 *TopicDiver*: A Longitudinal Regularized Mixture Model

We now describe *TopicDiver* as a two-step extension of PLSI, starting from a mixture model for text streams, and then introducing a longitudinal regularization.

A Mixture Model for Text Streams

Recall that the input of the problem is a collection D. Since our method utilizes the topic distributions over time, we concatenate all the documents published on the same time stamp to form a *super document*. For example, documents in C_i will form a *super document* S_i. Obviously, when the time granularity is fine enough, the *super documents* S_i are just the documents d_i. The vocabulary of S_i is denoted by V_i, and the vocabulary of the collection is $V = \cup_{i=1}^{T} V_i$. We use $f(w, d)$ to denote the count of a certain word w in a certain document or *super document* d.

Given a collection of *super documents* $S = \{S_1, S_2, \ldots, S_T\}$, the log-likelihood of S is given by the mixture model:

$$\log P(S) = \sum_{i=1}^{T} \log P(S_i) = \sum_{i=1}^{T} \sum_{j=1}^{|V_i|} f(w_j, S_i) \log P(w_j|t_i), \tag{5.3.1}$$

where $P(w_j|t_i)$ is denoted as the mixture of K topics $\{z_1, z_2, \ldots, z_K\}$,

$$P(w_j|t_i) = \sum_{k=1}^{K} P(w_j|z_k)P(z_k|t_i).$$

We use $\boldsymbol{\beta}$ to denote the matrix of word distributions of topics, *i.e.* $\beta_{k,j} = P(w_j|z_k)$ and $\boldsymbol{\theta}$ to denote the matrix of topic distributions over time, *i.e.* $\theta_{i,k} = P(z_k|t_i)$. The divergence of $\boldsymbol{\theta}$ over time is the key of *TopicDiver*.

Longitudinal Regularization

From [53], we know that the direct maximization of document likelihood in (5.3.1) is the process of PLSI on *superdocuments*. However, PLSI deals with static vocabulary with no temporal information considered. Moreover, directly using the topic distributions generated by a conventional topic model will bring much noise for precise event detection, as we will show in the experiments in Section 5.5. Inspired by the idea of *fused lasso* [77], we apply ℓ-1 regularization on the successive differences of topic distributions. Formally,

our framework is given by

$$
\begin{aligned}
(\boldsymbol{\theta}^*, \boldsymbol{\beta}^*) = \arg\min_{\boldsymbol{\theta},\boldsymbol{\beta}} & -\sum_{i=1}^{T}\sum_{j=1}^{|V_i|} f(w_j, S_i)\log\sum_{k=1}^{K}\theta_{i,k}\beta_{k,j} \\
& +\lambda\sum_{i=2}^{T}\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\|_1, \qquad (5.3.2) \\
\text{subject to} \quad & \sum_{k=1}^{K}\theta_{i,k} = 1, \quad i = 1, \ldots, T.
\end{aligned}
$$

where $\boldsymbol{\theta}_i$ denotes the i-th row of $\boldsymbol{\theta}$.

From (5.3.2), we know that the regularization parameter λ is indicating the regularization strength. When λ goes to infinity, we will allow no divergence and $\boldsymbol{\theta}_i$ will be constant along the time; when λ is zero, our framework will become conventional PLSI on the *super documents*.

To sum up, the key differences of *TopicDiver* and conventional PLSI are two-fold:

- We introduce the time variable and apply mixture model on time-stamped text streams instead of documents.
- We add longitudinal regularization on topic distributions of adjacent time stamps.

5.4 Optimization Algorithm

Now we describe our algorithm to solve the optimization problem in (5.3.2). We discuss retrospective and online settings respectively. In retrospective setting, the complete collection D is available, so the vocabulary V is also known. We can directly set $V_i = V$ and use a coordinate descent over $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. Note that for the constraint, we introduce $\widetilde{\boldsymbol{\theta}}$ and let

$$
\theta_{i,k} = e^{\widetilde{\theta}_{i,k}} / \sum_{k'=1}^{K} e^{\widetilde{\theta}_{i,k'}}, \quad i = 1, \ldots, T, \quad k = 1, \ldots, K.
$$

We use L to denote the objective function in (5.3.2). Then the gradient is given by:

$$\begin{aligned}\frac{\partial L}{\partial \widetilde{\theta}_{i,k}} &= -\sum_{j=1}^{V} f(w_j, S_i)\Big[\frac{e^{\widetilde{\theta}_{i,k}}\beta_{k,j}}{\sum_{k'=1}^{K} e^{\widetilde{\theta}_{i,k'}}\beta_{k',j}} - \frac{e^{\widetilde{\theta}_{i,k}}}{\sum_{k'=1}^{K} e^{\widetilde{\theta}_{i,k'}}}\Big] \\ &\quad + \lambda \mathbf{sign}(\widetilde{\theta}_{i,k} - \widetilde{\theta}_{i-1,k}), \\ \frac{\partial L}{\partial \beta_{k,j}} &= -\sum_{i=1}^{T} f(w_j, S_i)\frac{e^{\widetilde{\theta}_{i,k}}}{\sum_{k'=1}^{K} e^{\widetilde{\theta}_{i,k'}}\beta_{k',j}}.\end{aligned}$$

We update $\widetilde{\boldsymbol{\theta}}$ and $\boldsymbol{\beta}$ as following:

$$\widetilde{\boldsymbol{\theta}}^{(n+1)} = \widetilde{\boldsymbol{\theta}}^{(n)} - \gamma_1 \frac{\partial L}{\partial \widetilde{\boldsymbol{\theta}}}, \quad \boldsymbol{\beta}^{(n+1)} = \boldsymbol{\beta}^{(n)} - \gamma_2 \frac{\partial L}{\partial \boldsymbol{\beta}}. \tag{5.4.1}$$

The algorithm stops when L converges.

In the online setting, the vocabulary evolves over time. Therefore, we need to fold in the new words and documents in a streaming fashion. We use the method in [78] folding in new words and documents. Specifically, we run a topic model on the first *super document* S_1 to get $\boldsymbol{\beta}$ and $\boldsymbol{\theta}_1$. After finishing detection on t_{i-1}, we do the following steps for detection on t_i:

1. Fold in new documents. For each $d_{i,l}$ in C_i we initialize all $P(z_k|d_{i,l})$ randomly. We adopt the EM algorithm to compute

$$P(z_k|w_j, d_{i,l}) = \frac{P(w_j|z_k)P(z_k|d_{i,l})}{\sum_{k'=1}^{K} P(w_j|z_{k'})P(z_{k'}|d_{i,l})}, \tag{5.4.2}$$

$$P(d_{i,l}|z_k) = \frac{\sum_{j=1}^{|V_i|} f(w_j, d_{i,l})P(z_k|w_j, d_{i,l})}{\sum_{l=1}^{N_i}\sum_{j=1}^{|V_i|} f(w_j, d_{i,l})P(z_k|w, d_{i,l})}. \tag{5.4.3}$$

2. Fold in new words. We use w_{new} to denote the new words in V_i that are not in $V_1, \ldots, V_{i-1}$, and compute

$$P(z_k|w_{new}, d_{i,l}) = \frac{P(d_{i,l}|z_k)P(z_k|w_{new})}{\sum_{k'=1}^{K} P(d_{i,l}|z_{k'})P(z_{k'}|w_{new})}, \tag{5.4.4}$$

$$P(z_k|w_{new}) = \frac{\sum_{l=1}^{N_i} f(w_{new}, d_{i,l})P(z_k|w_{new}, d_{i,l})}{\sum_{l=1}^{N_i} f(w_{new}, d_{i,l})}. \tag{5.4.5}$$

Then we will use (5.4.1) to do coordinate descent.

After we get $\boldsymbol{\beta}^*$ and $\boldsymbol{\theta}^*$, we can use the divergence in $\boldsymbol{\theta}^*_{i-1}$ and $\boldsymbol{\theta}^*_i$. We will use the simple but effective

rule $\theta^*_{i,k} > \mu\theta^*_{i-1,k}$ to determine if there is enough divergence to indicate an event in topic k emerged on t_i, where μ is a predefined constant. When such divergence is detected, the top words associated with the topic can be directly used to describe the event. After we get the description keywords for events from the diverging topic, we then rank the documents on this bursty time stamp. The ranking function for a document is defined as

$$r(d, t_i, z_k) = P(z_k|d) \cdot e^{-\eta n_d}, \tag{5.4.6}$$

where $P(z_k|d)$ denotes how relevant the document d is to the k-th topic, n_d is the temporal order of d in C_i, and $\eta = 0.5$ is the decaying rate penalizing later documents. After ranking all the documents, we set a threshold value to identify all the first story documents. We show the outline of the online version of *TopicDiver* algorithm in Algorithm 6.

Algorithm 6 *TopicDiver*: Online Event Detection from Text Streams

Input: the text corpus D with time stamps $\{t_1, \ldots, t_T\}$, number of topics K, threshold parameter μ
Output: A set of detected events, each featured by a set of keywords and top document(s).
Initialize $\boldsymbol{\beta}$ and $\boldsymbol{\theta}_1$ from topic model on S_1, the set of detected event documents $\mathcal{E} \leftarrow \emptyset$, the set of detected event keyword sets $\mathcal{W} \leftarrow \emptyset$
for $i = 2$ to T **do**
  Concatenate all documents from D on t_i to get S_i.
  for each document $d_{i,l}$ **do**
    Fold in $d_{i,l}$ using (5.4.2) and (5.4.3)
  end for
  for each new word w_{new} **do**
    Fold in w_{new} using (5.4.4) and (5.4.5)
  end for
  Gradient Descent using (5.4.1)
  for each topic z_k **do**
    if $\theta^*_{i,k} > \mu\theta^*_{i-1,k}$ **then**
      Find top documents $d_{i,k}$ related to z_k by (5.4.6), and top terms $w_{i,k}$ by $P(w|z_k)$
      $\mathcal{E} = \mathcal{E} \cup \{d_{i,k}\}$
      $\mathcal{W} = \mathcal{W} \cup \{w_{i,k}\}$
    end if
  end for
end for
return $\mathcal{E}$, $\mathcal{W}$

5.5 Experiments

In this section, we evaluate *TopicDiver* on datasets from news articles and social media. For all the quantitative evaluation metrics, we use first story documents as output. We also use keywords to qualitatively illustrate the example events we have detected. We show that *TopicDiver* outperforms other state-of-the-art methods, and is especially good at detecting significant events.

5.5.1 Datasets

We use three datasets for our experiments, two from newswire and one from social media. We first testify our algorithm on news datasets, which are standard TDT5 dataset and CNN TV transcripts before moving to the social media dataset from Twitter.

TDT5 Dataset The standard TDT5 dataset is the benchmark dataset widely used in several TDT contests. It consists of news articles from various news media in multiple languages. We will use only the English part, containing 126 events labeled with first story in 221306 documents spanning 183 days from April to September 2003, with a vocabulary size of 87790 after preprocessing. Each day is used as a time stamp.

CNN TV Transcripts We collect transcripts of several CNN TV shows from 2009 to 2012. Transcripts are the on-screen text during programs, which are good description of the events covered by the program. We manually label events with the transcripts of the first programs covering them. There are 33593 documents and 50 events in total, with the vocabulary size 28670. The time stamp for this dataset is also day.

Twitter Dataset We collected 26 millions Tweets with over 180 million tokens from March 1st to 20th, 2016, using Apollo System [2]. The total size of the dataset is 98.9GB. Since tweets are generated at a very high rate, we use hour as our time stamp to match the pace. Even though hashtags and special characters such as at signs would be potential indicators of the Tweet content [50], we remove all the hashtags and at signs in the tweets to maintain the generality of our method. We also only select the tweets in English.

5.5.2 Evaluation Metrics

Due to the different natures of the datasets, we will now introduce the evaluation metrics for newswire and Twitter data respectively.

Newswire Datasets. For the TDT5 dataset, we follow the ***official*** TDT evaluation plan [45] using *minimal normalized cost*, which is the most popular metric for detection problems. For CNN data, we use both *minimal normalized cost* and F_1 score. We first introduce the basic measures:

- $Precision$: fraction of detected documents that are events.
- $Recall$: fraction of events that are detected.
- $False\ Alarm\ (FA)$: fraction of non-event documents that are detected as events.
- $Miss$: fraction of events that are not detected.
- $F_1 = 2 \cdot Precision \cdot Recall/(Precision + Recall)$.

[2] http://apollo3.cs.illinois.edu/

Table 5.1: Retrospective event detection on news data. A smaller $\mathbf{C_{min}}$ or a larger $\mathbf{F_1}$ is better.

	TDT5			**CNN**					
Method	False Alarm	Miss	$\mathbf{C_{min}}$	Precision	Recall	$\mathbf{F_1}$	False Alarm	Miss	$\mathbf{C_{min}}$
PLSI	0.026	0.468	0.595	0.041	0.740	0.078	0.026	0.260	0.386
TopicDiver	0.011	0.492	**0.546**	0.248	0.700	**0.366**	0.003	0.300	**0.316**

We now introduce *minimal normalized cost*. First, we define *detection cost* C_{det} as

$$C_{det} = Miss \cdot C_{miss} \cdot P_{target} + FA \cdot C_{FA} \cdot P_{non\text{-}target},$$

According to the official TDT evaluation plan [45], we set $C_{miss} = 1$ as the cost of missing an event; $C_{FA} = 0.1$ as the cost of detecting a non-event document as an event, $P_{target} = 0.02$ and $P_{non\text{-}target} = 0.98$ as the prior probability of an event document in the corpus. We can easily see that $C_1 = C_{FA} \cdot P_{non\text{-}target}$ and $C_2 = C_{miss} \cdot P_{target}$ are the costs of declaring all documents events and non-events respectively. The normalized detection cost is defined as

$$C_{norm} = \frac{C_{det}}{\min\{C_1, C_2\}}.$$

Finally, different parameter values will lead to different miss and false alarm values. In [45], the authors do a parameter sweep on the threshold. In our case, λ is an important parameter controlling the strength of regularization, and thus the number of events detected. Therefore, we use grid search to determine the best value of λ minimizing C_{norm}. The *minimal normalized cost* C_{min} is the minimum of C_{norm}.

Twitter Dataset. The evaluation metrics for Twitter dataset is different. Given the vast volume and rapid generating rate of Twitter data, it is not practical either to label all the tweets or to choose an event and find the first tweet mentioning it. We evaluate methods on the tweets detected instead of the whole collection, which is the method used in many other works [43, 46, 51]. For evaluation on Twitter dataset, we use *precision*, which is the fraction of selected tweets related to events (not necessarily the earliest), and *recall*, which is now the number of unique events detected on a daily basis [51]. Since Twitter data is often overwhelmed with noises potentially undermining the event detection, we will also use number of detections to check if the model can generate both precise and concise results.

Table 5.2: Online event detection on news data. A smaller $\mathbf{C_{min}}$ or a larger $\mathbf{F_1}$ is better.

	TDT5			**CNN**					
Method	False Alarm	Miss	$\mathbf{C_{min}}$	Precision	Recall	$\mathbf{F_1}$	False Alarm	Miss	$\mathbf{C_{min}}$
UMASS	0.042	0.492	**0.696**	0.132	0.660	0.220	0.007	0.340	0.372
LSH	0.044	0.492	0.707	0.112	0.660	0.191	0.008	0.340	0.379
TopicDiver	0.037	0.524	0.703	0.165	0.700	**0.267**	0.005	0.300	**0.326**

Table 5.3: Labeled Events in 2012 CNN Transcripts

News Event	**Date**	**Keywords**
Death of Whitney Houston	Feb 11, 2012	'Whitney', 'Houston', 'death'
Shooting of Trayvon Martin	Feb 27, 2012	'Trayvon', 'Martin', 'Zimmerman'
Jerry Sandusky's Trial	Jun 12, 2012	'Sandusky','child', 'scandal'
Aurora Shooting	Jul 20, 2012	'Aurora', 'victims', 'gun'
London Olympics	Jul 28, 2012	'Olympic', 'London', 'medal'
Hurricane Isaac	Aug 21, 2012	'hurricane', 'storm', 'Louisiana'
Benghazi Attack	Sept 11, 2012	'Benghazi', 'attack', 'arm'
Hurricane Sandy	Oct 22, 2012	'flood', 'hurricane', 'storm'
Presidential Election	Nov 6, 2012	'Obama', 'election', 'president'
Sandy Hook Shooting	Dec 14, 2012	'shooting', 'connecticut', 'elementary'

Table 5.4: Labeled Events in TDT5 Dataset

News Event	**Date**	**Keywords**
London Marathon	Apr 13, 2003	'London', 'marathon', 'competition', 'Radcliffe'
Bombing in Riyadh, Saudi Arabia	May 12, 2003	'Riyadh', 'explosion', 'Arabia', 'terrorist'
Hu Jintao meets Bush	Jun 01, 2003	'president','Bush', 'China', 'Korea'
U.S. Helicopter Crashed in Kosovo	Jun 08, 2003	'helicopter', 'Kosovo', 'crash'
Two Britons among terror suspects	Jul 04, 2003	'Abbasu', 'Begg'[3], 'Cuba'
2003 World Swimming Championship	Jul 20, 2003	'swim', 'record', 'champion', 'Thorpe'
Wildfire in Portugal	Aug 09, 2003	'Portugal', 'forest', 'fire', 'flame'
Wu Bangguo visits Manila	Aug 30, 2003	'Chinese', 'Philippines', 'policy'
Earthquake in Japan	Sept 26, 2003	'Hokaido', 'Japan', 'earthquake'
First Nigerian satellite in space	Sept 27, 2003	'Nigerian', 'launch', 'satellite'

5.5.3 Experiment Design

We now introduce and verify the design of our experiment. We want to test and show the following aspects through our experiments:

The effect of longitudinal regularization. First of all, recall that PLSI is a special case of *TopicDiver* where $\lambda = 0$. Therefore, we want to compare *TopicDiver* and PLSI to demonstrate the effect of our longitudinal regularization. Since PLSI is a static topic model and online variants are not directly related to *TopicDiver*, we only compare *TopicDiver* and PLSI on retrospective event detection of newswire datasets

[3]Abbasu and Begg are people names.

to see the effect of longitudinal regularization alone.

The efficacy of *TopicDiver* on newswire datasets. We want to testify the efficacy of *TopicDiver* on newswire datasets. In newswire datasets, the documents are well-written news articles in formal language. Note that the events labeled in CNN dataset are mostly significant ones with extensive coverage, and the events in TDT5 dataset also include some less important ones with less and short coverage. Since the online setting is more challenging and important in real application, we will only use this setting and show the comparison between *TopicDiver* with the baselines including the UMASS system [45] which performed best in several TDT competitions, and the improved algorithm based on LSH with variance reduction, proposed in [46].

The efficacy of *TopicDiver* on social media. As we have mentioned earlier, social media is very different from newswire data, with a rapid generating rate and a lot more informal language, meaningless babbles and personal conversation. Due to the rapid pace and timeliness of social media, retrospective event detection on Twitter is far less meaningful. Therefore, only online event detection is conducted on Twitter data. Since the UMASS system is not designed to work on web scale, we replace it with IPLSI introduced in [78]. By grid search, we set number of topics 80 for *TopicDiver* and IPLSI.

5.5.4 Experimental Results

We now show the experiment results to testify the efficacy of longitudinal regularization and *TopicDiver*. To reduce the variance, all results shown are the mean values of ten runs of the systems. First of all, we look into the comparison between *TopicDiver* and PLSI on retrospective event detection. From Table 5.1, we can observe that PLSI suffers from the noises and *TopicDiver* improves precision by and false alarm, thus F_1 and C_{min} greatly. The advantage of *TopicDiver* is especially remarkable on CNN data, which is expected because the events there are more important, and longitudinal regularization can effectively filter out noises without hurting the more significant divergence points caused by real events. This further validates our idea of adding longitudinal regularization on top of PLSI.

Secondly, we verify *TopicDiver* on online event detection from news data. From Table 5.2, we can see that *TopicDiver* again has a remarkable advantage on CNN data. This is also a demonstration of the effect of the longitudinal regularization. We are achieving comparable performance with the baselines on TDT5 data, with better false alarm and a slightly higher miss. This is because that the events in TDT5 dataset contain some minor ones with less coverage, and the divergence they cause can be smoothed out mistakenly by our regularization. However, for the more important events, we are actually still better than the baselines.

Table 5.5: Event detection on Twitter data. A larger precision or recall is better.

Method	No. of Claimed Detections	Precision	Recall
IPLSI	752	0.331	**45**
LSH	188	0.601	37
TopicDiver	147	**0.755**	42

For qualitative evaluation, we list all the labeled and detected events throughout the year 2012 in the CNN dataset, and ten of the detected events in the TDT5 dataset for comparison, in Table 5.3 and Table 5.4. The dates are from the divergence points we have detected, and the keywords are from the top words of the topic of which the temporal distribution is diverging. We can see that all the events in the CNN dataset are important ones which will attract most of the coverage at the time of its emergence. In the contrary, the events in the TDT5 data may be less significant.

Finally, we look at results on Twitter data. From Table 5.5, we observe that IPLSI claims far more events than the other two. This is also expected, because IPLSI is designed as an online variant of PLSI, and it suffers the similar problem with PLSI. With regularization to filter out the noise, *TopicDiver* has a great advantage on precision over IPLSI with only minor loss on recall, and also outperforms LSH remarkably both on precision and recall.

TopicDiver is also efficient in terms of complexity and runtime. For retrospective event detection, the optimization problem of our framework can be easily delivered by a stochastic gradient descent, which is much more efficient than EM algorithm of PLSA. In the online setting, since *TopicDiver* folds in new words and documents incrementally, it's also efficient compared to document based methods such as the UMASS system. In our experiments, *TopicDiver* is comparably efficient with LSH method, faster than IPLSI, and far more efficient than PLSI and the UMASS system.

5.5.5 Parameter Setting

As we have mentioned in previous sections, the most important parameters of *TopicDiver* is the regularization parameter λ. Recall that λ in our method controls the regularization strength and thus the number of detected events by our algorithm.

Intuitively, the larger λ is, the more regularization we put onto *TopicDiver* and the more rigorous we are on the events we detect. In this way, we are more likely to detect significant events causing really large changes in topic distributions. On the other hand, we expect more detections including some minor events if λ is smaller. When $\lambda = 0$, our model will become the conventional PLSI model. Therefore, we can flexibly choose the value of λ based on both our information needs and the data.

We show the number of detected events and the detection accuracy on CNN dataset in retrospective mode

Table 5.6: Effect of Regularization Parameter λ

λ	# Detections	False Alarm	Miss	$\mathbf{C_{norm}}$
0	900	0.026	0.260	0.386
0.02	819	0.023	0.260	0.374
0.05	524	0.014	0.280	0.351
0.1	235	0.006	0.300	0.329
0.2	145	0.003	0.300	**0.316**
0.3	93	0.002	0.340	0.349
0.4	65	0.002	0.420	0.426
0.6	36	0.001	0.700	0.703

against the value of lambda in Table 5.6, with the best value in bold. We can see that the value of λ has a large impact on the detection results, with larger λ causing higher miss and lower false alarm, and vice versa. However, the optimal values may largely depend on the datasets. Therefore, we use grid search in our experiments to determine the best parameter values.

5.6 Summary

In this chapter, I introduce my work on event detection. I look into the undiscovered temporal divergence of topic distributions for event detection from time-stamped text streams. Since both true events and non-event factors cause such divergence, and the latter is often dominant, real events are always very sparse in the divergence. Such sparsity needs to be enforced by appropriate noise reduction, and direct event detection without noise reduction can be extremely noisy and inaccurate.

I propose a framework that detects this temporal divergence and enhances its sparsity simultaneously by regularization. The proposed framework is built on a PLSI-like topic mixture over time-stamped text data, and inspired by the fused lasso, we add longitudinal regularization on the difference between adjacent topic distributions. Such regularization is proved to effectively wipe the noise off the real events and boost the detection accuracy.

The proposed framework is able to work well in both respective and online settings. Specifically for the online setting, my algorithm folds in the new terms to an evolving vocabulary and folds in the new documents in a streaming fashion. Extensive experiments on both newswire and Twitter datasets validate the efficacy of the proposed algorithm.

Chapter 6

Conclusion and Future Work

In this thesis, I present my research work on machine learning in big data. Specifically, I exploit sparsity to address the three challenges: (1) sample complexity, (2) computational complexity and (3) noise reduction, and propose algorithms that need less training examples, less computational resmyces to learn and are more robust to noise.

Sample complexity is one of the most important issues of the problem of one-bit compressed sensing, which focuses on the recovery of sparse signals with just a few linear measurements. I propose a novel and efficient algorithm with close-form solution for universal measurement matrices. My framework is based on nonconvex penalty functions, which are untouched for this problem in previous work. We also propose an algorithm to solve the resulting optimization problem. My algorithm improves the best sample complexity for vector recovery from $O(s \log d/\epsilon^2)$ to $O(s/\epsilon^2)$ for signals with a mild magnitude condition, and achieves exact support recovery at the same time even in noisy settings. This improvement is especially important for high dimensional and big data scenarios. In my work, we show that the sparsity in signals can be effectively utilized to improve sample complexity, i.e., reduce the number of training examples needed.

In terms of computational complexity, I study the EM algorithms in the high dimensional regime for sparse latent variable models. It is proved that sparsity structure must be enhanced for desirable convergence for high dimensional models, otherwise the noises and errors will accumulate across all dimensions and cause unstable performance. In addition, exact maximization can be intractable due to the dimensionality and gradient variants bring huge computational challenge. I present a novel semi-stochastic variance reduced gradient method, which is the first work to introduce variance reduction to high dimensional EM algorithms. Specifically, I propose a unique semi-stochastic gradient matching the bivariate structure of EM, and truncation step is applied after gradient ascent to enforce sparsity. Such gradient is based on mini-batches to reduce the computation complexity. My algorithm has a linear convergence towards the local optimal, and also achieves minimax optimal statistical rate of convergence up to a logarithmic factor.

For noise reduction, I study event detection from text corpus as noise has been an increasingly significant challenge for more accurate event detection. I have discovered that temporal divergence of topic distributions

can be an important indicator for real events. To filter out the divergence caused by non-event factors, longitudinal regularization is applied to enforce sparsity in such divergence. I also propose variants of my algorithm to work in both retrospective and online settings. As the corresponding optimization problem in my algorithm can be solved using simple stochastic gradient descent, my algorithm can work efficiently even on the scale of social media such as Twitter.

Machine learning in big data is a broad research topic with a lot of applications in various fields, and exploiting sparsity will continue to play an important part. The new algorithms developed in this thesis are general and thus can be applied to many different applications in big data.

The work of this thesis can be further extended in multiple directions. First, the work on one-bit compressed sensing can be further extended to relax or even clear out the magnitude assumption. Even this is a mild assumption, extending my method to general sparse signals is still meaningful.

Another potential improvement for my proposed EM algorithm is generalizing the choice of sparsity parameter. Currently in practice, this is done by cross validation. A more data-driven choice of this parameter will be desirable. The data-splitting technique is used for decorrelation brings in the logarithmic factor which can also be improved.

Since my proposed semi-stochastic variance-reduced gradient is general for any bivariate framework, we can further apply it to other problems in machine learning of similar structure.

For event detection, since social media is getting increasingly important in this task, its features can be exploited more to aid the content-based methods. Such features may include network structure and communities, temporal and spatial information, and hashtags and keywords. It is worth noting that sparsity also widely lies in these features and has great potential for event detection from these perspectives.

References

[1] J. G. Proakis, *Companders*. Wiley Online Library, 2001.

[2] Y. Plan and R. Vershynin, "One-bit compressed sensing by linear programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1275–1297, 2013.

[3] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *Proc. Conf. Inform. Science and Systems (CISS)*, Princeton, NJ, March 19-21 2008.

[4] P. Boufounos, "Reconstruction of sparse signals from distorted randomized measurements," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 3998–4001.

[5] S. Gopi, P. Netrapalli, P. Jain, and A. V. Nori, "One-bit compressed sensing: Provable support and vector recovery," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 2013, pp. 154–162.

[6] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," *IEEE Trans. Info. Theory*, vol. 59, no. 4, April 2013.

[7] L. Zhang, J. Yi, and R. Jin, "Efficient algorithms for robust one-bit compressive sensing," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, T. Jebara and E. P. Xing, Eds. JMLR Workshop and Conference Proceedings, 2014, pp. 820–828.

[8] A. Gupta, R. Nowak, and B. Recht, "Sample complexity for 1-bit compressed sensing and sparse classification," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, June 2010, pp. 1553–1557.

[9] J. Haupt and R. Baraniuk, "Robust support recovery using sparse compressive sensing matrices," in *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, March 2011, pp. 1–6.

[10] Y. Plan and R. Vershynin, "Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach," *Information Theory, IEEE Transactions on*, vol. 59, no. 1, pp. 482–494, Jan 2013.

[11] A. Ai, A. Lapanowski, Y. Plan, and R. Vershynin, "One-bit compressed sensing with non-gaussian measurements," *Linear Algebra and its Applications*, vol. 441, no. 0, pp. 222 – 239, 2014, special Issue on Sparse Approximate Solution of Linear Systems.

[12] A. Movahed, A. Panahi, and M. Reed, "Recovering signals with variable sparsity levels from the noisy 1-bit compressive measurements," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 6454–6458.

[13] X. Zeng and M. Figueiredo, "Robust binary fused compressive sensing using adaptive outlier pursuit," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 7674–7678.

[14] R. Zhu and Q. Gu, "Towards a lower sample complexity for robust one-bit compressed sensing," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 739–747.

[15] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[16] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Ann. Statist.*, vol. 38, no. 2, pp. 894–942, 2010.

[17] Z. Wang, H. Liu, and T. Zhang, "Optimal computational and statistical rates of convergence for sparse nonconvex learning problems," *The Annals of Statistics*, vol. 42, no. 6, pp. 2164–2201, 12 2014.

[18] Q. Gu, Z. Wang, and H. Liu, "Sparse pca with oracle property," in *Advances in Neural Information Processing Systems*, 2014, pp. 1529–1537.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 39, no. 1, pp. 1–38, 1977.

[20] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 03 1983.

[21] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[22] P. Tseng, "An analysis of the EM algorithm and entropy-like proximal point methods," *Mathematics of Operations Research*, vol. 29, no. 1, pp. 27–44, 2004.

[23] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[24] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2014.

[25] L. Xu and M. I. Jordan, "On convergence properties of the em algorithm for gaussian mixtures," *Neural computation*, vol. 8, no. 1, pp. 129–151, 1996.

[26] S. Balakrishnan, M. J. Wainwright, and B. Yu, "Statistical guarantees for the EM algorithm: From population to sample-based analysis," *arXiv preprint arXiv:1408.2156*, 2014.

[27] Z. Wang, Q. Gu, Y. Ning, and H. Liu, "High dimensional expectation-maximization algorithm: Statistical optimization and asymptotic normality," *arXiv preprint arXiv:1412.8729*, 2014.

[28] X. Yi and C. Caramanis, "Regularized em algorithms: A unified framework and statistical guarantees," in *Advances in Neural Information Processing Systems*, 2015, pp. 1567–1575.

[29] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[30] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 69–77.

[31] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 116.

[32] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 807–814.

[33] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in Neural Information Processing Systems*, 2013, pp. 315–323.

[34] X. Li, T. Zhao, R. Arora, H. Liu, and J. Haupt, "Stochastic variance reduced optimization for nonconvex sparse learning," *arXiv preprint arXiv:1605.02711*, 2016.

[35] J. Chen and Q. Gu, "Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization," in *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2016, pp. 132–141.

[36] D. Garber and E. Hazan, "Fast and simple pca via convex optimization," *arXiv preprint arXiv:1509.05647*, 2015.

[37] S. J. Reddi, A. Hefny, S. Sra, B. Póczós, and A. Smola, "Stochastic variance reduction for nonconvex optimization," *arXiv preprint arXiv:1603.06160*, 2016.

[38] Z. Allen-Zhu and E. Hazan, "Variance reduction for faster non-convex optimization," *arXiv preprint arXiv:1603.05643*, 2016.

[39] R. Zhu, L. Wang, C. Zhai, and Q. Gu, "Accelerated stochastic gradient expectation-maximization algorithm," in *In submission to the 34th International Conference on Machine Learning (ICML-17)*, 2017.

[40] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *Journal of Machine Learning Research*, vol. 15, pp. 2773–2832, 2014.

[41] A. T. Chaganty and P. Liang, "Spectral experts for estimating mixtures of linear regressions," *arXiv preprint arXiv:1306.3729*, 2013.

[42] J. Kleinberg, "Bursty and hierarchical structure in streams," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: ACM, 2002, pp. 91–101.

[43] C. Li, A. Sun, and A. Datta, "Twevent: Segment-based event detection from tweets," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, ser. CIKM '12. ACM, pp. 155–164.

[44] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. ACM, 2010, pp. 851–860.

[45] J. Allan, V. Lavrenko, D. Malin, and R. Swan, "Detections, bounds, and timelines: Umass and tdt-3," in *Proceedings of topic detection and tracking workshop*, 2000, pp. 167–174.

[46] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 181–189.

[47] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 181–192.

[48] G. P. C. Fung, J. X. Yu, H. Liu, and P. S. Yu, "Time-dependent event hierarchy construction," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 300–309.

[49] Q. He, K. Chang, and E.-P. Lim, "Analyzing feature trajectories for event detection," in *Proceedings of the 30th ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 207–214.

[50] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 3. IEEE, 2010, pp. 120–123.

[51] J. Weng and B.-S. Lee, "Event detection in twitter." *Internation AAAI Conference on Web and Social Media ICWSM*, vol. 11, pp. 401–408, 2011.

[52] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 1155–1158.

[53] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[54] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[55] X. Wang and A. McCallum, "Topics over time: a non-markov continuous-time model of topical trends," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 424–433.

[56] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 113–120.

[57] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *2008 8th IEEE international conference on data mining*. IEEE, pp. 3–12.

[58] J. H. Lau, N. Collier, and T. Baldwin, "On-line trend analysis with topic models:\# twitter trends detection topic model online." in *COLING*, 2012, pp. 1519–1534.

[59] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[60] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[61] T. Zhang, "Multi-stage convex relaxation for learning with sparse regularization," in *Advances in Neural Information Processing Systems*, 2009, pp. 1929–1936.

[62] T. Zhang, "Adaptive forward-backward greedy algorithm for sparse learning with linear models," in *Advances in Neural Information Processing Systems*, 2009, pp. 1921–1928.

[63] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via l0-regularized intensity and gradient prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2901–2908.

[64] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," *Advances in neural information processing systems*, vol. 19, p. 801, 2007.

[65] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted 1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5, pp. 877–905, 2008.

[66] T. Zhang et al., "Some sharp performance bounds for least squares regression with l1 regularization," *The Annals of Statistics*, vol. 37, no. 5A, pp. 2109–2144, 2009.

[67] D. Donoho, I. Johnstone, and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1993.

[68] P. Breheny and J. Huang, "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection," *The Annals of Applied Statistics*, vol. 5, no. 1, pp. 232–253, 03 2011.

[69] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," *arXiv preprint arXiv:1011.3027*, 2010.

[70] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[71] N. L. Roux, M. Schmidt, and F. R. Bach, "A stochastic gradient method with an exponential convergence _rate for finite training sets," in *Advances in Neural Information Processing Systems*, 2012, pp. 2663–2671.

[72] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical programming*, vol. 127, no. 1, pp. 3–30, 2011.

[73] L. Wang, X. Zhang, and Q. Gu, "A unified computational and statistical framework for nonconvex low-rank matrix estimation," *arXiv preprint arXiv:1610.05275*, 2016.

[74] S. Dasgupta and L. Schulman, "A probabilistic analysis of EM for mixtures of separated, spherical Gaussians," *Journal of Machine Learning Research*, vol. 8, pp. 203–226, 2007.

[75] L. Wang, X. Ren, and Q. Gu, "Precision matrix estimation in high dimensional gaussian graphical models with faster rates," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 177–185.

[76] R. Zhu, A. Zhang, J. Peng, and C. Zhai, "Exploiting temporal divergence of topic distributions for event detection," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016.

[77] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[78] T.-C. Chou and M. C. Chen, "Using incremental plsi for threshold-resilient online event analysis," *IEEE transactions on Knowledge and Data Engineering*, vol. 20, no. 3, pp. 289–299, 2008.