Joint feature screening and subsampling in analysis of massive data

by

Kaili Jing

A thesis
submitted to the University of Ottawa
in partial fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy in Mathematics and Statistics[1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The Ph.D. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

In modern scientific research, collecting large amounts of data has become increasingly prevalent. For example, in genomics, there have been more than 500, 000 microarrays that are publicly available with each array containing tens of thousands of expression values of molecules; In biomedical engineering, there have been tens of thousands of terabytes of fMRI images with each image containing more than 50, 000 voxel values [36]. Compared to traditional datasets, massive datasets usually hold great promises for extracting crucial information and discovering subtle patterns. However, their characteristics such as huge volume (large sample size and high dimensionality), heterogeneity (multiple latent subpopulations), and corruption (complicated noise) pose substantial challenges to traditional data analysis. These features require an application to deal with not only huge volumes but also complex structures and uncertainties in the data. There is no doubt that Big Data are something of double-edged sword, which has been fundamentally changing and transforming the way we live and think especially in scientific research. Its emergence has spawned new research paradigms to solve some of the general analytical difficulties, which will be addressed here.

To ease the analytical difficulties, one viable route is to remove most redundant information from a massive dataset before analyzing it. In this spirit, this dissertation develops effective statistical tools to screen out variables (features) and observations that are less relevant to the analysis. Specifically, the dissertation consists of the following three self-contained research projects.

1. Distributed hard screening for massive data.

   Massive datasets with a huge number of features are commonly encountered in many scientific areas. Nevertheless, existing classic screening methods are inefficient or even infeasible due to the high computational burden for a large-N-large-p dataset, where N is the sample size and p is the features size. When both N and p are large, this raises the so-called large-N-large-p regime, which creates opportunities for extracting crucial information and poses substantial challenges. Developing effective tools to manage and utilize large-scale data is an attractive and important topic in statistics and some related fields. To address this research problem, I develop a distributed screening method for the large-N-large-p regime. The new method is built upon an ADMM updating procedure for fitting l_0-constrained consensus regression, where data are processed at m manageable segments by multiple local computers. Compared to the existing methods, the new method can not only fit the need of computational efficiency for processing massive data, but also screen out most non-influential features while retaining influential ones with high probability. Under mild

conditions, I show that the new method is convergent and leads to an effective screening when the starting value is appropriate specified. The promising performance of the method is supported by extensive numerical and real data studies.

2. Joint feature screening for mixture regression.

   Finite mixtures of regression models are ubiquitous for analyzing complex data. They aim to detect heterogeneity in the effects of a set of covariates on a response over a finite number of latent classes. When the number of covariates is large, a direct fitting of mixture regressions can be numerically costly and often leads to a poor interpretive value. One practical strategy is to screen out most irrelevant covariates before an in-depth analysis. Such a strategy is referred to as feature screening, which has attracted a great deal of attention in the past decade. Despite the surge of research on feature screening, the existing methods are mainly developed based on certain dependence measure between the response and features over the entire population. In finite mixture of regressions, however, the sets of relevant features in the model are class-specific. This makes feature screening in mixture regression a challenging task that has not been carefully studied. To address this research problem, I propose a novel method for covariate screening in ultrahigh-dimensional Gaussian mixture regressions. The new method is built upon a sparsity-restricted expectation-approximation-maximization algorithm, which allows a precise control of the number of covariates to be retained in each latent class. In the screening process, the joint effects between covariates are naturally accounted and the class-specific screening results are produced without ad hoc steps. These merits give the new method an edge to outperform the existing screening methods. The promising performance of the method is supported by both theory and numerical examples including real data analysis.

3. Low-gradient based subsampling for corrupted massive data.

   Subsampling is one of basic methodologies used for coupling with massive data, among which "the gradient of loss" based subsampling is accepted as the most effective strategy. In past studies, the large-gradient subsampling is normally suggested, which assigns the subsampling probability directly proportional to the gradient of loss. Such a strategy is effective for uncontaminated massive data but not for corrupted massive data, which is unavoidably encountered in real-world applications. We thus must deal with those corrupted massive data cases. Based on an observation that the use of large-gradient brings the danger of assigning higher sampling probabilities to the heavily corrupted observations, I propose a low-gradient subsampling procedure for corrupted massive data. The new procedure iteratively draws "low-gradient

observations" and updates sampling probabilities according to the new prediction performance. In this way, the procedure excludes the most corrupted observations and leads to a robust subsampling strategy. I show that the procedure is statistically consistent with achieving a nearly optimal estimation precision. The promising performance of the procedure is supported by a variety of numerical simulations and real data applications.

The effective statistical methods I develop in this dissertation are not only significant attempts but also abreast of explorations to tackle the challenges in Big Data. Based on the core idea of removing redundant information that contributes insignificantly to the analysis, this dissertation provides promising results and reliable guidance to further the frontiers of research analysis of massive data.

Acknowledgements

Now is the time to write this note of appreciation to whom I am greatly thankful during these challenging six years of my Ph.D. study at University of Ottawa.

First and foremost, I would like to thank the University of Ottawa to give me this chance and scholarships to pursue my degree. My deepest gratitude also goes to my supervisor, Dr. Chen Xu, for his endless patience and inspiring suggestions during my study period and research work. Your enthusiasm for statistics, your wealth of knowledge, and your productiveness inspires me to learn more and work harder, in an attempt to emulate your achievements. I am also truly thankful for his financial support throughout my studies. Then a special thanks to my fellow colleagues in the laboratory, Xingxiang Li, Qianxiang Zang and Wei Dong. I am benefited much from their collaboration and discussion.

Secondly, special thanks to my thesis committee, Dr. Mayer Alvo, Dr. Maia Fraser, Dr. Song Cai and Dr. Peijun Sang, for their valuable feedback and comments, which have led to significant improvement of my thesis.

Thirdly, I want to thank all the staff in our department including Prof. Benoit Dionne, Prof. Maia Fraser, Prof. Gilles Lamothe, Prof. Burkett Kelly, Prof. Mayer Alvo, Prof. Raluca Balan, Mayada El Maalouf, Diane Demers, Janick Rainville, Carolynne Roy, etc.

Sincerely thank my parents and family members, it is their support, understanding, and patience that encourage me along this challenging way.

I would not be who am I today without you all.

Dedication

This thesis is dedicated to my beloved parents, Fenli Li and Wenfeng Jing. Thank you for unconditional support during my challenging graduate life abroad.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, the rapid development of data generation and acquisition have led to the explosive growth of the size and number of available data sets. Big Data have rapidly become a hot topic that draws large attention from almost every industry and business area. The world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s [50]. It can be asserted that Big Data have integrated into every area of industry and business in 21 century and has played an important role in production. O'Reilly Media, an American learning company and online learning platform, affirmed that "the future belongs to the companies and people that turn data into products" [75]. Big data analysis is becoming more and more prominent in modern research.

In Wikipedia, Big Data are "fields that treats ways to analyze, systematically extract information from, or otherwise, deal with data sets that are too large or complex to be dealt with by traditional data-processing application software". Fox [42] in 2018 also states "Big Data are where parallel computing tools are needed to handle data", and notes, "This represents a distinct and clearly defined change in the computer science used, via parallel programming theories, and losses of some of the guarantees and capabilities made by Codd's relational model." There is no doubt that Big Data are very significant and have been fundamentally changing and transforming the way we live and think especially in scientific research. Companies use Big Data in their systems to improve operations, provide better customer service, create personalized marketing campaigns and take other actions that, ultimately, can increase revenue and profits. Businesses that use it effectively hold a potential competitive advantage over those that don't because they're able to make faster and more informed business decisions. Bioscientists could also use Big Data to identify disease signs and risk genes to help diagnose illnesses and medical conditions in patients. Moreover, a combination of data from electronic health records, social media

sites, the web, and other sources gives healthcare organizations and government agencies up-to-date information on infectious disease threats or outbreaks. In short, Big Data have already been an integral part of many areas of our lives, which brings opportunities for scientific community to re-examine its methodology of scientific research and trigger a revolution in scientific thinking and methods. Abreast of the opportunities of Big Data, scientific researchers still face some challenges involved in performing inferences on Big Data: 1) Massive dataset (large sample size) combined with high dimensionality creates issues such as heavy computational cost and algorithmic instability; 2) The massive samples in Big Data are potentially collected from multiple sources at different time points using different technologies. This creates issues of heterogeneity, in which the behaviors of a statistical estimator or procedure may vary from one class to another; 3) The massive data may be unavoidably contaminated by heavy corruption or high level of noise. The presence of corruption and noise in Big Data can be inevitably caused by accidental outliers, transmission loss, or even adversarial data attacks. Directly analyzing such data without pre-processing procedure may degrade the accuracy and the efficiency of the algorithm.

To handle these challenges of Big Data, we need new statistical tools and computational methods. One viable route is to remove most redundant information from a massive dataset before analyzing it. For high dimensional dataset, the assumption that only a small portion of all features are predictive of the response has been widely adopted. It entails that we only need to identify influential features and build an interpretive model to link them to the response variable. Therefore, there should be a pre-processing procedure to first screen out a large number of noninfluential features before an elaborative analysis. This preprocessing step is referred to as feature screening. With dimensionality reduced from high to low, analytical difficulties are reduced drastically. For corrupted massive data, subsampling is a natural strategy to remove the most non-informative observations. This strategy draws useful informative observations to infer instead of using the whole data, which provides a way to tackle an untractable problem with a confined limited computation resource. This dissertation focuses on three self-contained research topics related to the challenges I discussed above. Particularly, I develop three effective statistical tools to screen out variables (features) and observations that are less relevant to the analysis. All these methods hold great promises for handling the challenges of Big Data, in addition to balancing statistical accuracy and computational efficiency.

1.1 Research Topics

Valid statistical analysis for Big Data is becoming increasingly important. To handle the challenges of Big Data, we need new statistical thinking and computational methods. For

example, many existing traditional methods are designed and perform well for moderate sample size. When the sample size is large, they are significantly hampered due to computational efficiency, statistical accuracy, and algorithmic stability. Similarly, many statistical methods that perform well for low dimensional datasets are facing significant challenges in analyzing high or even ultra-high dimensional datasets. Specifically, large sample size and dimensionality may bring heterogeneity to massive datasets, which are often created via aggregating many data sources from different sub-populations. Each sub-population might exhibit some unique features which are not shared by others. Another severe challenge is the high cost incurred from data storage. Their huge volume poses substantial challenges to traditional data analysis, where a dataset is typically processed on a single machine. From the perspective of noise, almost all of the existing studies are only for massive data without contamination or with relatively low levels of noise. Nevertheless, heavy corruption or high level of noise is unavoidably encountered in real-world applications. Such a corruption effect is especially severe in Big Data and may even dominate the true signals, which leads to systematic biases in inference and interpretation. In short, these features of Big Data make traditional statistical methods less efficient or even invalid. In this section, the major research topics I consider are stated as follows:

1.1.1 Distributed Feature Screening for Massive Data

In modern scientific research, collecting data with huge size are increasingly frequent. Massive data sets with a huge number of features are commonly encountered in many scientific areas. For instance, customs scan information of thousands of shipments every week. In health care research thousands of personal health or drug-usage records are collected. Thousands of keywords in a URL are analyzed in Internet studies. Given a set of observations in these applications, statisticians are confronted with the task of exploring the relationship between a univariate response and a p-dimensional covariate vector.

When the dimensionality is high, one practical assumption that only a small number of predictors contribute to the response is frequently adopted and deemed useful. This strategy aims to screen out a large number of noninfluential features before an elaborative analysis. By doing so, an interpretive model can be built to link the influential features with the response variable. This step is called feature screening, which has been extensively studied in the literature. Specifically, the seminal work of Fan and Lv [38] proposed a marginal screening method called sure independence screening (SIS) for ultrahigh-dimensional linear models. This method has been extended to generalized linear and additive models [35, 40], proportional hazards models [46], and quantile regression [107]. There also is vast research on joint screening methods such as iterative sure independence screening

(ISIS) [39], forward regression (FR) [102], sparsity restricted maximum likelihood estimator (SMLE) [108], and high dimensional ordinary least squares projection (HOLP) [104]. These methods improve the marginal approaches by incorporating the joint effects among features into the screening process. However, almost all of these works are designed for the small-N-large-p regime. Large-N-large-p regime challenges these methods in terms of both theoretical optimality and practical feasibility. For example, when a DC-SIS [68] is used in a regression with ten thousand observations and features, it can be computationally infeasible and unstable in the screening process. In other words, new statistical procedures with these issues in mind are crucially needed. Developing effective tools to manage and utilize large-scale data is a recent focus in statistics and some related fields. In addition, to overcome the storage bottleneck, the divide-and-conquer framework was developed for distributed data storage and parallel processing. With the aid of the divide-and-conquer framework, many machine learning methods have been adapted to a distributed version for analyzing big data [67, 77, 81, 114, 115]. To apply the divide-and-conquer framework to feature screening, one may consider approaches based on aggregation. This distributive learning framework divides full data into m manageable segments and m sub-estimates are generated from m independent implementations. Then the m local machines transfer their sub-estimates to a master machine. The final estimator is obtained by aggregating the local outputs via an aggregating function. However, there are still several important challenges that are not appropriately handled concurrently: 1) Choosing an appropriate aggregating function for divide-and-conquer framework is not straightforward; 2) Neglecting the influence of joint effects; 3) Lacking efficient communications between local and master computers. All these characteristics make distributed feature screening for massive data as an interesting topic to be considered.

1.1.2 Joint Feature Screening for Mixture Regression

In modern scientific research, data analysts frequently deal with high-dimensional complex data. One challenge in the analysis of such data is that observations are often created by aggregating many data sources corresponding to different sub-populations. Each sub-population might exhibit some unique features not shared by others and the behaviors of a statistical estimator or procedure may vary from one class to another. These unique features make traditional statistical procedures inefficient or even inappropriate. How to systematically explore the underlying heterogeneity and unveil the commonality across different sub-populations remains an active research area. On the other hand, in a traditional setting where the sample size is moderate, observations from small sub-populations are generally treated as outliers and it is hard to systematically model them due to a lack of information. However, in big data, the large sample size enables us to better interpret

heterogeneity, shedding light on studies such as exploring the association between certain covariates (e.g., genes or SNPs) and rare outcomes (e.g., rare diseases or diseases in small populations) and understanding why certain treatments (e.g. chemotherapy) benefit a subpopulation and harm another subpopulation. In short, the main advantage brought by Big Data is to understand the heterogeneity of sub-populations such as the benefits of certain personalized treatments, which are infeasible when the sample size is small or moderate.

In regression analysis, we could assume that multiple relationships exist between a response and a set of covariates within a population. To this end, the finite mixture of regression models (FMR) have been widely used to obtain the class-specific regression coefficients for better inference and interpretation [5, 64, 83]. In the last decades, increasing studies have been conducted on this topic and all kinds of algorithms have been designed [30, 60, 93]. However, while these methods are inspiring, when dimensionality p is of the exponential order in the sample size, those methods can be significantly hampered due to several implementation difficulties such as computational efficiency, statistical accuracy, and algorithmic stability. To ease the implementation difficulties for the heterogeneous large p situation, one can also consider using feature screening to screen out most irrelevant features before an elaborate analysis. With dimensionality reduced from ultrahigh to low, analytical difficulties are reduced drastically. Despite the surge of research on feature screening, the existing methods are mainly developed based on certain dependence measure between the response and features over the entire population. These kinds of methods are not suitable for finite mixture of regression models since the sets of relevant features in the model are often class-specific. For example, in structural biology, each protomer of a protein structure may contribute a different set of residues to the binding interface; in marketing, the behaviors of different types of customers may be influenced by different social or economical factors. Feature screening in mixture regression especially in Big Data is still a challenging task that has not been carefully studied.

1.1.3 Subsampling for Corrupted Massive Data

Nowadays, the presence of noise and corruption in real-world data have become ubiquitous. Due to the uncontrolled collecting process in big data, heavy corruption or high level of noise is unavoidably encountered in real-world applications. The presence of such noise and corruption in real-world massive data might be ineluctably caused by many factors such as accidental outliers [90] or adversarial data attacks [24]. All these factors raise a complicated corruption pattern, where the proportion of corruption could be large, the corruption level could be high, and even exists on both covariates and response simultaneously. This amounts that the data contain many anomalies and unwanted information

which cannot be utilized or can be known as noise. The main effects of heavy noise in data are that it reduces the regression/classification accuracy, increases the computation time, and results in the interpretability of the model. We thus must deal with those corrupted massive data cases. Generally, to overcome the burden of limited computation resources, researchers adopt two basic methodologies. One is the divide-and-conquer strategy and the other is the subsampling strategy. The divide-and-conquer strategy solves the problem by partitioning the massive data into several segments, runing a specific algorithm for each segment in parallel and then aggregating these individual outputs into a final estimate. The subsampling strategy solves the problem by carrying out an intended calculation on a representative subset drawn from the full data. These methods draw useful informative observations to infer instead of using the whole data, which provides a way to tackle an untractable problem with a confined limited computation resource. For corrupted massive data, the subsampling strategy naturally gears the idea that removing most redundant information from a massive dataset before analyzing it. Therefore, subsampling can be considered as a pre-processing procedure to first screen out a large number of observations that are less relevant to the analysis. I observe, however, that almost all of the statistical procedures are based on unrealistic assumptions that can not be validated by data. These methods are basically designed for the massive data without contamination or with simple corruption patterns. When the corruption of massive data is complicated, a large proportion of both covariates and response are contaminated by exogenous noise, making it infeasible to apply these methods. Other robust methods such as robust regression, outlier detection and self-paced learning are seldom practical to set the parameter or computationally expensive. Hence, analysis of the corrupted massive data calls for new statistical methodologies and theories.

1.2 Review on Feature Screening and Subsampling

1.2.1 Feature Screening

Feature screening is a pre-processing procedure to first screen out a large number of noninfluential features before an elaborative analysis. It implies to approximate the (ultra)high-dimensional model coefficients on a designated low-dimensional subspace. With dimensionality reduced from high to low, analytical difficulties are reduced drastically. Meanwhile, an accurate estimation can be obtained by using well-developed lower dimensional methods. Many authors have developed various procedures for feature screening.

Before I pursue further, I should introduce some notation first. Throughout this chapter, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ is an independent and identically distributed sample collected from a

population $\{\boldsymbol{x}, y\}$, where $\boldsymbol{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$ is a vector of p features, and y is a real-valued response variable. Denote $Y = (y_1, \ldots, y_n)^{\mathrm{T}}$ and $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^{\mathrm{T}}$, the $N \times p$ observed feature matrix.

1.2.1.1 Marginal screening methods

Linear models

Consider a linear regression model

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \epsilon, \tag{1.1}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^{\top}$ is a $p \times 1$ vector of parameters. When the dimensionality p is greater than the sample size N, the least squares estimator of $\boldsymbol{\beta}$ is not well defined due to the fact that $\boldsymbol{X}^{\top}\boldsymbol{X}$ is singular. The ridge regression, defined by

$$\widehat{\beta}_{\lambda} = (\boldsymbol{X}^{\top}\boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1}\boldsymbol{X}^{\top}\boldsymbol{Y}$$

with λ being a ridge parameter, is a useful technique to deal with the singularity of the design matrix in the classical linear regression. It can be seen that if $\lambda \to 0$, $\widehat{\beta}_{\lambda}$ tends to be the least squares estimator; if $\lambda \to \infty$, $\lambda\widehat{\beta}_{\lambda}$ tends to $\boldsymbol{X}^{\top}\boldsymbol{Y}$. This result implies that $\widehat{\beta}_{\lambda} \propto \boldsymbol{X}^{\top}\boldsymbol{Y}$. If all covariates are marginally standardized, $\boldsymbol{X}^{\top}\boldsymbol{Y}/N$ becomes the vector consisting of the sample version of Pearson correlations between the response and individual covariates. Motivated by this, we can use Pearson correlation as a marginal utility for feature screening. Specifically, denote

$$\omega_j = \frac{1}{N}\boldsymbol{X}_j^{\top}Y, \ \text{for} j = 1, \ldots, p.$$

ω_j is the sample correlation between the jth covariate and the response variable. Fan and Lv [38] suggest ranking all covariates according to $|\omega_j|$ and select the top s covariates as the strongly correlated covariates with the response. Specifically, the selected submodel with top s ranked covariates can be obtained by

$$\mathcal{M}_s = \{1 \leq j \leq p : |\omega_j| \text{is among the first } s \text{ largest of all}\}. \tag{1.2}$$

In this way, one can reduce the high dimensionality p down to a relatively moderate scale s, and then an elaborative analysis can be implemented for further modeling. The screening

procedure defined in (1.2) is referred to as sure independence screening (SIS) in Fan and Lv [38]. They also show that, under mild conditions,

$$P(\mathcal{M}_* \subset \mathcal{M}_s) = 1 - O\left(\exp\left(-\frac{Cn^{1-2\kappa}}{\log N}\right)\right) \to 1, \tag{1.3}$$

where $\mathcal{M}_* = \{1 \le j \le p : \beta_j \neq 0\}$ is the true model and C is some positive constant. The property (1.3) is referred to as sure screening property. It ensures that the submodel would not miss any true influential covariate with an overwhelming probability.

Generalized linear models

Recall that given the predictor vector $\boldsymbol{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$, the conditional distribution of y belongs to an exponential family, whose density function has the canonical form

$$f(y|\boldsymbol{x}) = \exp\{y\theta(\boldsymbol{x}) - b(\theta(\boldsymbol{x})) + c(y)\},$$

for some known functions $b(\cdot)$ and $c(\cdot)$. Parameter θ is called the natural parameter and set $\Theta = \{\theta : \int f(y|\boldsymbol{x})d\nu < \infty\}$ is the natural parameter space, where ν is a σ-finite measure. We assume that $\theta \in \Theta$ with Θ denoting a compact subspace of Θ and $b(\cdot)$ is twice continuously differentiable. Further assume $E(y|\boldsymbol{x}) = b'\{\theta(\boldsymbol{x})\}$. Denote the negative log-likelihood of i-th observation $\{\boldsymbol{x}_i, y_i\}$ to be $\ell(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}, y_i)$. Note that the minimizer of the negative log-likelihood $\sum_{i=1}^N \ell(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}, y_i)$ is not well defined when $p > N$.

Assume that each predictor is standardized to have mean zero and standard deviation one, and define the maximum marginal likelihood estimator (MMLE) $\widehat{\beta}_j^M$ for the j-th predictor x_j as

$$\widehat{\beta}_j^M = \arg\min_{\beta_j} \sum_{i=1}^N \ell(x_{ij}\beta_j, y_i).$$

It is reasonable to consider the magnitude of $\widehat{\beta}_j^M$ as a marginal utility to rank the importance of x_j and select a submodel for a given pre-specified threshold γ,

$$\widehat{\mathcal{M}}_\gamma = \{1 \le j \le p : |\widehat{\beta}_j^M| \ge \gamma\}.$$

This is proposed for feature screening in generalized linear model by Fan and Song [40]. To establish the theoretical properties of MMLE, Fan and Song [40] define the population version of the marginal likelihood maximizer as

$$\beta_j^M = \arg\min_{\beta_j} E\ell(x_j^{\mathrm{T}}\beta_j, y).$$

They first show that the marginal regression parameters $\beta_j^M = 0$ if and only if $\text{cov}(y, x_j) = 0$ for $j = 1, \ldots, p$. Thus, when the influential variables are correlated with the response, $\beta_j^M \neq 0$. Under some technical assumptions, Fan and Song [40] prove that the MMLEs are uniformly convergent to the population values and established the sure screening property of the MMLE screening procedure. That is, for any $c_1 > 0$, there exists some $c_2 > 0$ such that

$$P\big(\max_{1\le j\le p} |\widehat{\beta}_j^M - \beta_j^M| \ge c_1 n^{-\kappa}\big) \le p\{\exp\big(-c_2 n^{1-2\kappa}/(k_n K_n)^2 + nm_1 \exp(-m_0 K_n^{\alpha})\big)\}$$

with $\kappa > 0$, where $k_n = b'(K_n B + B) + m_0 K_n^{\alpha}/s_0$, B is the upper bound of the true value of β_j^M, K_n is the supremum norm of $\boldsymbol{x}$.

In addition, under some technical assumptions, the following inequality holds,

$$P(\mathcal{M}_* \subset \widehat{\mathcal{M}}_\gamma) \ge 1 - s\{\exp\big(-c_2 n^{1-2\kappa}/(k_n K_n)^2 + nm_1 \exp(-m_0 K_n^{\alpha})\big)\} \to 1,$$

as $n \to \infty$. It further implies that the MMLE can handle the ultrahigh-dimensionality as high as $\log p = o(N^{1-2\kappa})$ for the logistic model with bounded predictors, and $\log p = o(N^{(1-2\kappa)/4})$ for the linear model without the joint normality assumption.

Additive models

Fan et al. [35] also propose a nonparametric independence screening (NIS) for the ultrahigh-dimensional additive model

$$y = \sum_{j=1}^{p} m_j(x_j) + \epsilon,$$

where $\{m_j(x_j)\}_{j=1}^p$ have mean 0 or identifiability. An intuitive population level marginal screening utility is $E(f_j^2(x_j))$, where $f_j(x_j) = E(y|x_j)$ is the projection of y onto x_j. Suppose that $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ are sample from $(\boldsymbol{x}, y)$, $f_j(\boldsymbol{x})$ can be estimated via a normalized B-spline basis $B_j(\boldsymbol{x}) = \{B_{j1}(\boldsymbol{x}), \ldots, B_{jd_N}(\boldsymbol{x})\}^{\mathrm{T}}$: $\widehat{f}_{Nj}(\boldsymbol{x}) = \widehat{\boldsymbol{\beta}}_j^{\mathrm{T}} B_j^{\boldsymbol{x}}$, $1 \le j \le p$, where $\widehat{\boldsymbol{\beta}}_j = (\beta_{j1}, \ldots, \beta_{jd_N})^{\mathrm{T}}$ is obtained through the componentwise least squares regression:

$$\widehat{\boldsymbol{\beta}}_j = \arg\min_{\boldsymbol{\beta}_j \in \mathbb{R}^{d_N}} \sum_{i=1}^{N} (y_i - \boldsymbol{\beta}_j^{\mathrm{T}} B_j(x_{ij})).$$

Thus the screened model index set is

$$\widehat{\mathcal{M}}_\nu = \{1 \le j \le p : \|\widehat{f}_{Nj}\|_N^2 \ge \nu_N\}$$

for some predefined threshold value ν_N with $\|\widehat{f}_{Nj}\|_N^2 = N^{-1}\sum_{i=1}^N \widehat{f}_{Nj}(x_{ij})^2$.

Fan et al. [35] advocate the sure screening property of NIS of including the true model $\mathcal{M}_* = \{j : Em_j(x_j)^2 > 0\}$ based on a set of regularity conditions. Specifically, they show that

$$P(\mathcal{M}_* \subset \widehat{\mathcal{M}}_\nu) \to 1$$

for $p = \exp\{N^{1-4\kappa}d_N^{-3} + Nd_N^{-3}\}$. In addition, if $\text{var}(Y) = O(1)$, then the cardinality of $\mathcal{M}_*$ is bounded by the polynomial order of N and the false selection rate is under control.

Other model setups

The method of sure independence screening has also been extended to other models. Gorst-Rasmussen and Scheike [46] introduce a SIS method for survival data based on the notably simple FAST statistic. They show that FAST-SIS may admit the formal SIS property within a class of single-index hazard rate models. Wu and Yin [107] propose a conditional quantile screening method, which could select features that contribute to the conditional quantile of the response given the covariates. Their method can naturally handle censored data by incorporating a weighting scheme through redistribution of the mass to the right. They also establish sure independent screening properties for both the complete and the censored response cases. In the same spirit, a series of model-free screening methods have been developed solely based on marginal utilities between the response and features. Zhu et al. [116] propose a novel feature screening procedure under a unified model framework, which covers a wide variety of commonly used parametric and semiparametric models. Their method does not require imposing a specific model structure on regression functions, and thus is particularly appealing to ultrahigh-dimensional regressions, where there are a huge number of candidate predictors but little information about the actual model forms. They demonstrate that the new procedure possesses consistency in ranking, which is useful in leading to consistency. Li et al. [68] develop a sure independence screening procedure based on the distance correlation. Unlike the Pearson correlation coefficient, rank correlation coefficient and generalized correlation coefficient, which all are defined for two random variables, the distance covariance are defined from two random vectors which are allowed to have different dimensions. DC-SIS is also a model-free feature screening methods which does not require model specification (e.g., linear model or generalized linear model) for responses or predictors.

1.2.1.2 Joint screening methods

Despite the convenience of implementation, the marginal screening methods are often found to be unreliable in practice, as the joint effects among features are ignored. Features

with significant joint effects but showing weak marginal effects are likely to be wrongly screened out. Researchers have made attempts at improving the marginal approaches by incorporating the joint effects among features into the screening process.

Iterative sure independence screening

The SIS methodology may break down if a feature is marginally unrelated, but jointly related with the response, or if a feature is jointly uncorrelated with the response but has higher marginal correlation with the response than some important features. In the former case, the important feature has already been screened at the first stage, whereas in the latter case, the unimportant feature is ranked too high by the independent screening technique. Fan and Lv [38] propose an iterative version of SIS procedure (ISIS) by iteratively replacing the response with the residual obtained from the regression of the response on selected covariates in the previous step. ISIS seeks to overcome these difficulties by using more fully the joint covariate information while retaining computational expedience and stability as in SIS.

In the first step, they apply SIS to pick a set $\widehat{\mathcal{A}}_1$ of indices of size k_1, and then employ a penalized (pseudo)-likelihood method such as Lasso, SCAD, MCP or the adaptive Lasso to select a subset $\widehat{\mathcal{M}}_1$ of these indices. This is the initial estimate of the set of indices of important features. The screening stage solves only bivariate optimizations and the fitting part solves only a optimization problem

$$\ell(\boldsymbol{\beta}) = N^{-1}\sum_{i=1}^{N} L(y_i, \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(|\beta_j|)$$

with moderate size k_1, where $p_\lambda(\cdot)$ is a penalty function and $\lambda > 0$ is a regularization parameter. This is an attractive feature in ultrahigh dimensional statistical learning.

Then, for each $j \in \{1, \ldots, p\}/\widehat{\mathcal{M}}_1$, they compute

$$L_j^{(2)} = \min_{\boldsymbol{\beta}_{\widehat{\mathcal{M}}_1}, \beta_j} = N^{-1}\sum_{i=1}^{N} L(y_i, \boldsymbol{x}_{i,\widehat{\mathcal{M}}_1}^{\mathrm{T}}\beta_{\widehat{\mathcal{M}}_1} + x_{ij}\beta_j).$$

$L_j^{(2)}$ can be considered as the additional contribution of the j-th predictor given the existence of predictors in $\widehat{\mathcal{M}}_1$. Then, after ordering $\{L_j^{(2)} : j \in \widehat{\mathcal{M}}_1^c\}$, we form the set $\widehat{\mathcal{A}}_2$ consisting of the indices corresponding to the smallest k_2 elements. After the prescreening step, they employ a penalized (pseudo-)likelihood method such as the LASSO and SCAD

on the combined set $\widehat{\mathcal{M}}_1 \cup \widehat{\mathcal{A}}_2$ to select an updated selected set $\widehat{\mathcal{M}}_1$, i.e., use the penalized likelihood to obtain

$$\widehat{\boldsymbol{\beta}}_2 = \arg \min_{\boldsymbol{\beta}_{\widehat{\mathcal{M}}_1}, \boldsymbol{\beta}_{\widehat{\mathcal{A}}_2}} N^{-1} \sum_{i=1}^{N} L(y_i, \boldsymbol{x}_{i,\widehat{\mathcal{M}}_1}^{\mathrm{T}} \beta_{\widehat{\mathcal{M}}_1} + \boldsymbol{x}_{i,\widehat{\mathcal{A}}_2}^{\mathrm{T}} \beta_{\widehat{\mathcal{A}}_2} + \sum_{j \in \widehat{\mathcal{M}}_1 \cup \widehat{\mathcal{A}}_2} p_\lambda(|\beta_j|).$$

The indices of $\widehat{\boldsymbol{\beta}}_2$ that are non-zero yield a new estimated set $\mathcal{M}_2$ of active indices. The process, which iteratively recruits and deletes features, can then be repeated until obtaining a set of indices $\mathcal{M}_l$ which either has reached the prescribed size, or satisfies $\mathcal{M}_l = \mathcal{M}_{l-1}$. This iterative method could better control the false negative rate in the finite sample case than the one-step screening procedures.

Forward regression

Another way to cope with the situation with some variables being jointly correlated but marginally uncorrelated with the response is forward regression, which is proposed by Wang [102] in 2009. It is a stepwise regression approach that begins with an empty model and at each step gradually adds variables to the regression model to find a model that best explains the data. This procedure builds a sequence of models by successively including one covariate at a time up to a pre-specified number of steps or until all the covariates are included. The forward regression algorithm is summarized below. Recall that $\boldsymbol{X}_{\mathcal{M}}$ is the subdesign matrix corresponding to $\mathcal{M}$. Wang [102] studies the property of forward regression with the ultrahigh-dimensional case. He also proposes using the extended BIC to determine the size of the active variable set. However, forward regression has been found to be unstable in the screening process; a small change in data could cause a very different screening result. The reason resulting in this fact is that once a covariate has been added to the model at any step in the screening procedure, it is never removed from the final model. Therefore, for complex and massive data, it can easily lead to a local optimal model.

Sparsity-restricted maximum likelihood estimator

The iterative feature screening procedure significantly improves the simple marginal screening such as SIS in that it can select weaker but still significant predictors and delete inactive predictors which are spuriously marginally correlated with the response. However, Xu and Chen [108] claims that the computational cost and increased complexity are higher than the gain of the iterative procedure. To this end, they propose the sparsity-restricted MLE (SMLE) method for the generalized linear models and demonstrate that SMLE enjoys sure

Algorithm 1.1 The forward regression algorithm.

Input: Data $\{X, Y\}$, $\mathcal{F} = \{1, \ldots, p\}$.

Output: a solution path $\mathbb{S}$.

1: (Initialization). Set $\mathcal{S}^{(0)} = \emptyset$.

2: (Forward regression). In the k-th step ($k \geq 1$), we are given $\mathcal{S}^{(k-1)}$. Then, for every $j \in \mathcal{F}/\mathcal{S}^{(k-1)}$, we construct a candidate model $\mathcal{M}_j^{(k-1)} = \mathcal{S}^{(k-1)} \cup \{j\}$. We then compute

$$\mathrm{RSS}_j^{(k-1)} = Y^{\mathrm{T}}(I - H_j^{(k-1)})Y,$$

where

$$H_j^{(k-1)} = X_{\mathcal{M}_j^{(k-1)}}(X_{\mathcal{M}_j^{(k-1)}}^{\mathrm{T}} X_{\mathcal{M}_j^{(k-1)}})^{-1} X_{\mathcal{M}_j^{(k-1)}}^{\mathrm{T}}$$

is a projection matrix and I is the identity matrix.

3: We then find

$$a_k = \arg \min_{j \in \mathcal{F}/\mathcal{S}^{(k-1)}} \mathrm{RSS}_j^{(k-1)}$$

and update $\mathcal{S}^{(k)} = \mathcal{S}^{(k-1)} \cup \{a_k\}$

4: Repeat for n times, which leads a total of n nested candidate models. We then collect those models by a solution path $\mathbb{S} = \{\mathcal{S}^{(k)} : 1 \leq k \leq n\}$ with $\mathcal{S}^{(k)} = \{a_1, \ldots, a_k\}$.

screening property as the iterative procedure in a conceptually simpler and computationally cheaper manner.

Suppose that a random sample was collected from an ultrahigh-dimensional generalized linear model and its log-likelihood function is

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} \{(\boldsymbol{x}_i^{\top}\boldsymbol{\beta})Y_i - b(\boldsymbol{x}_i^{\top}\boldsymbol{\beta})\}.$$

Xu and Chen [108] define the SMLE of $\boldsymbol{\beta}$ to be

$$\widehat{\boldsymbol{\beta}}_s = \arg\max_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}), \text{ subject to} \|\boldsymbol{\beta}\|_0 \leq s. \tag{1.4}$$

where $\|\boldsymbol{\beta}\|_0$ is the number of nonzero entries of $\boldsymbol{\beta}$. The SMLE method can be considered as a joint-likelihood-supported feature screening procedure. Of course, the maximization problem (1.4) can not be solved directly when dimensionality p is large. To tackle the computation issue, Xu and Chen [108] propose the following approximation of the joint likelihood function,

$$h(\boldsymbol{\gamma}; \boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + (\boldsymbol{\gamma} - \boldsymbol{\beta})^{\top} S(\boldsymbol{\beta}) - \frac{u}{2}\|\boldsymbol{\gamma} - \boldsymbol{\beta}\|_2^2, \tag{1.5}$$

for some step parameter $u > 0$, where $\|\cdot\|_2$ is the L_2 norm and $S(\boldsymbol{\beta}) = \ell'(\boldsymbol{\beta})$ is the score function. The first two terms in (1.5) come from the first order Taylor's expansion of $\ell(\boldsymbol{\beta})$, while the last term is a quadratic penalty to avoid $\boldsymbol{\gamma}$ too far away from $\boldsymbol{\beta}$. One advantage of this approximation is that there is a closed-form solution for $\boldsymbol{\beta}$ at each step. Specifically, given the tth sparse estimate $\boldsymbol{\beta}^{(t)}$, $\boldsymbol{\beta}^{(t+1)}$ can be obtained by

$$\boldsymbol{\beta}^{(t+1)} = \arg\max_{\boldsymbol{\gamma}} h(\boldsymbol{\gamma}; \boldsymbol{\beta}), \text{ subject to} \|\boldsymbol{\gamma}\|_0 \leq s,$$

whose solution has an explicit form $\boldsymbol{\beta}^{(t+1)} = H_k(\tilde{\boldsymbol{\gamma}})$, where

$$\tilde{\boldsymbol{\gamma}} = \boldsymbol{\beta}(t) + u^{-1}\boldsymbol{x}^\top(\boldsymbol{Y} - b'(\boldsymbol{x}\boldsymbol{\beta}^{(t)}))$$

and $H_k(\tilde{\boldsymbol{\gamma}})$ is the s largest components of $\tilde{\boldsymbol{\gamma}}$ in absolute value. Xu and Chen [108] refer to this algorithm as the iterative hard thresholding (IHT) algorithm. They further show that the IHT algorithm has the non-decreasing property, i.e. $\ell_n(\boldsymbol{\beta}^{(t+1)}) \geq \ell_n(\boldsymbol{\beta}^{(t)})$. This property features the SMLE a promising method for feature screening. Therefore, one can start with an initial $\boldsymbol{\beta}^{(0)}$ and carry out the iteration until the convergence criteria are met. Xu and Chen [108] further demonstrate that the SMLE method enjoys the sure screening property.

High-dimensional ordinary least-squares projection

As aforementioned, sure screening property strongly relies on the assumption that the important variables in the model have large marginal correlations with the response. To overcome this, Wang and Leng [104] propose a novel and simple screening technique called the high-dimensional ordinary least-squares projection (HOLP). Recall the ridge regression estimate

$$\widehat{\boldsymbol{\beta}}(r) = (rI + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}Y,$$

where r is the ridge parameter. By letting $r \to \infty$, it is seen that $r\widehat{\boldsymbol{\beta}}(r) \to \boldsymbol{X}^{\mathrm{T}}Y$. Let $r \to 0$, the ridge estimator $\widehat{\boldsymbol{\beta}}(r)$ becomes

$$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{+}\boldsymbol{X}^{\mathrm{T}}Y,$$

where A^+ denotes the Moore-Penrose generalized inverse. An application of the Sherman-Morrison-Woodbury formula gives

$$(rI + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}Y = \boldsymbol{X}^{\mathrm{T}}(rI + \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}Y.$$

Then letting $r \to 0$ gives that

$$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{+}\boldsymbol{X}^{\mathrm{T}}Y = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}Y.$$

They name this estimator $\widehat{\boldsymbol{\beta}} = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}Y$ the High-dimensional Ordinary Least-squares Projection (HOLP) due to the similarity to the classical ordinary least-squares estimate. For variable screening, they follow a very simple strategy by ranking the components of $\widehat{\boldsymbol{\beta}}$ and selecting the largest ones. More precisely, let d be the number of the predictors retained after screening. They choose a submodel $\mathcal{M}_d$ as

$$\mathcal{M}_d = \{x_j : |\widehat{\beta}_j| \text{are among the largest } d \text{ of all } |\widehat{\beta}_j| s\} \text{or } \mathcal{M}_\gamma = \{x_j : |\widehat{\beta}_j| \geq \gamma\}$$

for some γ. HOLP gives a diagonally dominant projection matrix $\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{X}$, such that the product of this matrix and $\boldsymbol{\beta}$ would be more likely to preserve the rank order of the entries in $\boldsymbol{\beta}$. In contrast, tilting and forward regression both rely on some goodness-of-fit measure of the selected variables, aiming to minimize the distance between fitted $\widehat{Y}$ and Y. The HOLP estimator can be seen as the other extreme of the ridge regression estimator by letting $r \to 0$, as opposed to the marginal screening operator $\boldsymbol{X}^{\mathrm{T}}Y$ in Fan and Lv [38] by letting $r \to \infty$. Wang and Leng [104] also show that HOLP possesses the sure screening property and gives consistent variable selection without the strong correlation assumption, and has a low computational complexity.

Feature screening vs. feature reduction

The concept of feature screening is often confused with feature reduction techniques [1, 71, 72]. While both methods can be used to simplify a high-dimensional feature set, they intrinsically serve for different research purposes. Feature screening aims to obtain a refined set by removing redundant features; the refined feature set is a subset of the original feature set and the retained features maintain their original meanings. Feature reduction aims to find a low-dimensional approximation of a high-dimensional feature space; it typically needs to construct new representative features from the the original feature space without keeping the original meanings. As such, feature screening is often preferred in the interpretation-based research, where the goal is to identify influential factors to a response of interest. Feature reduction is more popular in the tasks, where accurate prediction is the major concern.

1.2.2 Subsampling

Data with enormous sizes and complex frames under limited computational resources is another issue of big data which has received much attention. When the sample size is enormous, the assumption that not all observations are useful and informative has been widely adopted. Subsampling motivates us to a solution for processing the corrupted massive

data. Specifically, by defining a suitable subsample strategy, subsampling method draws useful informative observations to infer instead of using the whole data, which provides a way to tackle a large-scale and corrupted data analysis problem with a confined limited computation resource.

Subsampling method draws a subdata set from the full dataset and estimates the interested parameters by the chosen subdata. The fundamental step of the subsampling method is how to select the subdata. The more informative observations we choose, the better approximation performance we could expect. Uniform subsampling is a basic and straightforward method but is not preferred because every observation is treated equally without considering how much information one observation carries. For linear regression, leverage sampling has been discussed by [32] [31] [91]. The subsamples obtained by this method are drawn from the full dataset with replacement based on the normalized leverage scores or their variants. The asymptotic normality and asymptotic unbiasedness of the leveraging sampling estimator are studied in Ma et al. [76]. Besides the probabilistic method for linear models, a deterministic method named information-based optimal subdata selection (IBOSS) is proposed by Wang et al. [100] aiming at finding a subdata that has maximal information matrix under D-optimality. The IBOSS approach is extended to include the logistic regression in Cheng et al. [25].

Optimal subsampling method is a probabilistic approach, where subsamples are expected to be drawn based on the optimal subsampling probabilities that are derived by minimizing the asymptotic covariance matrix of the random sampling-based estimators under certain optimality criteria. The optimal subsampling method for logistic regression is introduced by Wang et al. [101], which formulates the optimal subsampling probabilities by minimizing the asymptotic mean squared error (MSE) of the subsample estimator. Since the expressions of the optimal subsampling probabilities involve the maximum likelihood estimator (MLE) of the full data, the authors proposed a two-stage adaptive algorithm that uses a pilot sample estimator to substitute the full data MLE. This method is named as optimal subsampling methods motivated by the A-optimality criterion (OSMAC) and is improved in Wang [99] by adopting unweighted target functions for subsamples and Poisson subsampling.

For the corrupted massive data case, Camponovo et al. [14] propose a robust subsampling procedure for the linear regression model by elaboratively selecting the subsampling size which is derived by a formula for the breakdown point of subsampling quantiles. Dalalyan and Thompson [28] consider the case where the response is contaminated by at most τ adversarial outliers. They prove that the sampled L_1-penalized Huber's M-estimator based on n subsamples can approximate the M-estimator with full data at an estimation precision $O((s/n)^{\frac{1}{2}} + (\tau/n))$, where s is the sparsity of the problem. McWilliams

et al. [84] study the case of the covariates corruption and assumed that the corruption can be described in an additive noise model. They suggest a subsampling algorithm called the influence-weighted subsampling and they show that the algorithm can yield an estimator approximating the oracle $\boldsymbol{\beta}^*$ at precision $O(C\sqrt{\frac{p\log p}{n}} + \epsilon\sqrt{p}||\boldsymbol{\beta}^*||)$, where ϵ is the proportion of the corruption. Gong et al. [45] also consider the linear regression in the highly noisy cases and they developed a robust gradient-based Markov subsampling(GMS) procedure. They prove that the GMS estimator can approximate the oracle β^* at an optimal estimation precision $O(\sqrt{\frac{p\log p}{n}})$.

Optimal subsampling methods under the A-optimality criterion

Suppose that $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ are N independent and identically distributed observations, where $\boldsymbol{x}_i \in \mathbb{R}^p, i = 1, \ldots, p$, are covariates and $y_i, i = 1, \ldots, p$, are responses. For a logistic regression, $y_i \in \{0, 1\}$ is a binary variable. Given $\boldsymbol{x}_i$, the response y_i satisfies that

$$P(y_i = 1 \mid \boldsymbol{x}_i) = \frac{\exp\big(\boldsymbol{x}_i^\top \boldsymbol{\beta}\big)}{1 + \exp\big(\boldsymbol{x}_i^\top \boldsymbol{\beta}\big)}, \ i = 1, \ldots, N,$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is the known regression coefficient and can be estimated by the MLE $\widehat{\boldsymbol{\beta}}_{MLE}$ which is the maximizer of

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} [y_i \boldsymbol{x}_i^\top \boldsymbol{\beta} - \log\{1 + \exp\big(\boldsymbol{x}_i^\top \boldsymbol{\beta}\big)\}].$$

This optimization problem can be solved by the Newton-Raphson method in $O(\eta N p^2)$ time where η is the number of iterations for the Newton-Raphson method to converge. To reduce the computational burden when N is large, an optimal subsampling method named OSMAC targeting at approximating the full data MLE $\widehat{\boldsymbol{\beta}}_{MLE}$ is proposed in Wang et al. [101]. Different from the traditional arbitrary subsampling algorithm (denote the corresponding estimator by $\widehat{\boldsymbol{\beta}}_{sub}$), the underlying idea of the OSMAC is to find the optimal subsampling probabilities which minimize the asymptotic variance-covariance matrix of $\widehat{\boldsymbol{\beta}}_{sub} - \widehat{\boldsymbol{\beta}}_{MLE}$. To compare matrices, the A-optimality criterion is adopted, which minimizes the trace of this asymptotic variance-covariance matrix. The optimal subsampling probabilities under the A-optimality criterion are

$$\pi_i^{optA} = \frac{|y_i - p(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}_{MLE})| \|M_L^{-1} \boldsymbol{x}_i\|}{\sum_{j=1}^{N} |y_j - p(\boldsymbol{x}_j, \widehat{\boldsymbol{\beta}}_{MLE})| \|M_L^{-1} \boldsymbol{x}_j\|}, \tag{1.6}$$

where $M_L = N^{-1}\sum_{i=1}^{N} p(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}_{MLE})(1 - p(\boldsymbol{x}_i, \widehat{\boldsymbol{\beta}}_{MLE}))\boldsymbol{x}_i^\top \boldsymbol{x}_i$. A-optimal subsampling probabilities depend on the response and covariates, and contain $\widehat{\boldsymbol{\beta}}_{MLE}$, which is the quantity that we are approximating. To solve this problem, a pilot sample estimator is used to substitute $\widehat{\boldsymbol{\beta}}_{MLE}$. The pilot sample can be drawn from the full dataset by uniform subsampling or case-control subsampling whose subsampling probabilities are $\pi_i^0 = N^{-1}$ and $\pi_i^0 = (2\sum_{i=1}^{N} y_i)^{-y_i}(2N-2)^{y_i - 2\sum_{i=1}^{N} y_i}$, respectively. Furthermore, M_L can be approximated by the pilot sample to reduce the computational complexity. It takes $O(Np^2)$ time to compute π_i^{optA}. Theoretical results in Wang et al. [101] show that the asymptotic normality of the estimator obtained by OSMAC conditionally on the full data and the pilot sample estimator and the convergence rate is at the order of $n_1^{-1/2}$, where n_1 is the subsample size in OSMAC and is not related to the full data size.

Influence-weighted subsampling for corrupted massive data

Influence-weighted subsampling uses the effective impact of an individual datum on the overall estimate in order to detect and avoid sampling corrupted points. This method is robust to corrupted observations and exhibits reduced bias compared with other subsampling estimators.

Consider a variant of the standard linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \epsilon, \tag{1.7}$$

where $\epsilon \in \mathbb{R}^N$ is a noise term independent of $\boldsymbol{X} \in \mathbb{R}^{N\times p}$. However, rather than directly observing $\boldsymbol{X}$ we instead observe $\boldsymbol{Z}$ where

$$\boldsymbol{Z} = \boldsymbol{X} + U\boldsymbol{W}. \tag{1.8}$$

$U = \text{diag}(u_1, \ldots, u_n)$ and u_i is a Bernoulli random variable with probability π of being 1. $\boldsymbol{W} \in \mathbb{R}^{N\times p}$ is a matrix of measurement corruptions. The rows of $\boldsymbol{Z}$ therefore are corrupted with probability π and not corrupted with probability $(1 - \pi)$.

We consider the specific instance of the linear corrupted observation model in (1.7) and (1.8), where $\boldsymbol{X}, \boldsymbol{W}$ and ϵ are sub-Gaussian with parameters $(\sigma_x^2/N, \Sigma_x/N)$, $(\sigma_\omega^2/N, \Sigma_\omega/N)$ and $(\sigma_\epsilon^2/N, \sigma_\epsilon^2\boldsymbol{I}/N)$, respectively.

The key challenge is that even when π and the magnitude of the corruptions, σ_ω are relatively small, the standard linear regression estimate is biased and can perform poorly. Sampling methods that are not sensitive to corruption in the observations can perform even worse if they somehow subsample a proportion $rN > \pi N$ of corrupted points. Furthermore, the corruption may not be large enough to be detected via leverage based techniques

alone. McWilliams et al. [84] propose influence-weighted subsampling (IWS-LS) to handle the issue that ordinary least squares and the random approximation to ordinary least squares obtain poor predictions in the corrupted observation model. IWS-LS subsamples points according to the distribution, $p_i = c/d_i$ where c is a normalizing constant so that $\sum_{i=1}^{N} p_i = 1$ and d_i is the influence defined by expected mean squared error. Ordinary least squares are then estimated on the subsampled data. The sampling procedure ensures that points with high influence are selected infrequently and so the resulting estimate is less biased than the full ordinary least squares solution.

The influence-weighted subsampling is described in the Algorithm 1.2 as below. In

Algorithm 1.2 Influence-weighted subsampling(IWS-LS).

Input: Data $\boldsymbol{Z}$, $\boldsymbol{y}$
Output: $\widehat{\boldsymbol{\beta}}_{IWS}$

1: Solve $\widehat{\boldsymbol{\beta}}_{OLS} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\beta}\|^2$
2: **for** $t = 1$ to N **do**
3: $\quad e_i = y_i - \boldsymbol{z}_i \widehat{\boldsymbol{\beta}}_{OLS}$
4: $\quad l_i = \boldsymbol{z}_i^\top (\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} z_i$
5: $\quad d_i = e_i^2 l_i / (1 - l_i)^2$
6: **end for**
7: Sample rows $(\tilde{\boldsymbol{Z}}, \tilde{\boldsymbol{y}})$ of $(\boldsymbol{Z}, \boldsymbol{y})$ proportional to $1/d_i$
8: Solve $\widehat{\boldsymbol{\beta}}_{IWS} = \arg\min_{\boldsymbol{\beta}} \|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{Z}}\boldsymbol{\beta}\|^2$

practice, McWilliams et al. [84] also propose an approximation version of Algorithm 1.2 which combines the approximate leverage computation of [31] and the randomized least squares approach of [80].

Outlier-robust estimation of a sparse linear model

Outlier-robust estimation of a sparse linear model is another framework for handling corrupted data. The problem of outlier-robust estimation considers how to estimate a p-dimensional s-sparse vector in a linear model with Gaussian design and additive noise. In the case where the labels are contaminated by at most o adversarial outliers. Such problem is relatively similar to the issue we want to cope with for corrupted massive data. By defining the sparsity of contamination term $\boldsymbol{\theta}^* = (\theta_1^*, \ldots, \theta_N^*)$, they build a connection between sparsity of sample (the assumption that only a small number o of labels are contaminated by outliers implies that the vector $\boldsymbol{\theta}^*$ is o-sparse) and subsampling from a corrupted dataset. The model they consider is the case where the response is contaminated

(the adversarial corrupted response model), which is also a special case of the corrupted data model we consider. Thus, it is worthy to introduce their models and methods here. More details please check Chapter 4.

Let $\mathcal{D}_N^o = \{(\boldsymbol{x}_i, y_i^o); i = 1, \ldots, N\}$ be i.i.d feature-label pairs such that $\boldsymbol{x}_i \in \mathbb{R}^p$ is Gaussian with zero mean and covariance matrix Σ and y_i^o are defined by the linear model

$$y_i^o = \boldsymbol{x}_i^\top \boldsymbol{\beta}^* + \xi_i, \ i = 1, \ldots, N,$$

where the random noise ξ_i, independent of $\boldsymbol{x}_i$, is Gaussian with zero mean and variance σ^2. Instead of observing the "clean" data $\mathcal{D}_N^o$, we have access to a contaminated version of it, $\mathcal{D}_N = \{(\boldsymbol{x}_i, y_i); i = 1, \ldots, N\}$, in which a small number $o \in \{1, \ldots, N\}$ of labels y_i^o are replaced by an arbitrary value. Setting $\boldsymbol{\theta}_i^* = (y_i - y_i^o)/\sqrt{N}$, and using the matrix-vector notation, the described model can be written as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta}^* + \sqrt{N}\boldsymbol{\theta}^* + \boldsymbol{\xi}, \tag{1.9}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1^\top; \ldots; \boldsymbol{x}_N^\top]$ is the $N \times p$ design matrix, $\boldsymbol{Y} = (y_1, \ldots, y_N)^\top$ is the response vector, $\boldsymbol{\theta}^* = (\theta_1^*, \ldots, \theta_N^*)$ is the contamination and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)^\top$ is the noise vector. They aim to estimate the vector $\boldsymbol{\beta}^*$. The dimensionality p is assumed to be large, possibly larger than the sample size N but, for some small value $1 \leq s \ll p$, the vector $\boldsymbol{\beta}^*$ is assumed to be s-sparse: $\|\boldsymbol{\beta}^*\|_0 \leq s$. In such a setting, if the clean data $\mathcal{D}_N^o$ is known and measure the quality of an estimator $\widehat{\boldsymbol{\beta}}$ by the Mahalanobis norm $\|\Sigma^{1/2}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)\|_2$, the optimal rate is

$$\sigma\Big(\frac{s\log(p/s)}{N}\Big)^{1/2}.$$

In the outlier-contaminated setting, if we only access the corrupted data $\mathcal{D}$, the minimax-optimal-rate [19] is

$$\sigma\Big(\frac{s\log(p/s)}{N}\Big)^{1/2} + \frac{\sigma o}{N}.$$

The assumption that only a small number o of labels are contaminated by outliers implies that the vector $\boldsymbol{\theta}^*$ in (1.9) is o-sparse. In order to take advantage of the sparsity of both $\boldsymbol{\beta}^*$ and $\boldsymbol{\theta}^*$ while ensuring computational tractability of the resulting estimator, a natural approach studied in several papers [85] [27] is to use some version of the L_1-penalized empirical risk minimization. This corresponds to defining

$$\widehat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^N} \min_{\boldsymbol{\theta}\in\mathbb{R}^N} \Big\{\frac{1}{2N}\|\boldsymbol{Y} - \boldsymbol{X}^\top\boldsymbol{\beta}\sqrt{N}\boldsymbol{\theta}\|_2^2 + \lambda_s\|\boldsymbol{\beta}\|_1 + \lambda_o\|\boldsymbol{\theta}\|_1\Big\}, \tag{1.10}$$

where λ_s, λ_o are tuning parameters. This estimator is very attractive from a computational perspective, since it can be seen as the Lasso for the augmented design matrix $M = [\boldsymbol{X}, \sqrt{N}\boldsymbol{I}]$. The best known rate for this type of estimator is

$$\sigma\big(\frac{s\log(p)}{N}\big)^{1/2} + \sigma\big(\frac{o}{N}\big)^{\frac{1}{2}},$$

which is sub-optimal, obtained by Nguyen and Tran [86] under some restrictions on (N, p, s, o). Dalayan and Thompson [28] show that this sub-optimality is not an intrinsic property of the estimator (1.10), but rather an artefact of previous proof techniques. By using a refined argument, they prove that $\widehat{\boldsymbol{\beta}}$ defined by (1.10) does attain the optimal rate under very mild assumptions. They refer $\widehat{\boldsymbol{\beta}}$ as L_1-penalized Hubers M-estimator and provide its rate-optimality, up to logarithmic terms that can be avoided, in the setting of robust linear regression with adversarial contamination.

1.3 Contributions

The major research contributions in this dissertation can be summarized as follows:

- **Distributed hard screening for massive data**

  To address feature screening in large-N-large-p data, I Develop a new distributed feature screening approach DHS. The new method is built upon an ADMM updating procedure of L_0-constrained consensus regression, where data are processed in m manageable segments by multiple local computers. Specifically, at each communication round, local computers communicate learning information with the master computer; between two rounds, local computers work on their own to update the local vectors in a parallel manner. The joint effects between features are also accounted for naturally in the screening process.

  Unlike other distributed methods, it provides a computationally viable and reliable route for screening features with big data. The new design of the divide-and-conquer framework makes new methods dig much more information and knowledge efficiently when the data size becomes large. From the prospective of theoretical analysis, I provide a convergence property of DHS procedure relative to non-convex, non-smooth and non-Lipschitz continuous L_0 norm which enriches the existing ADMM convergence theory. In addition, I prove that DHS enjoys the sure screening property which provides an effective result for correctly identifying true models. I also provide guidance on the sure screening property under finite step set-up, which gives a good

way to implement the DHS algorithm in practice. In addition to establishing the theoretical analysis, The proposed DHS procedure is evaluated with four competing methods in both synthetic and real-world datasets. The results demonstrate that the new approach consistently outperforms existing methods in screening features, delivering a competitive running time.

- **Joint feature screening in ultrahigh-dimensional finite mixture of regressions**

  To address feature screening in ultrahigh-dimensional finite mixture of regressions, I propose a new method sEAM for covariate screening. The new method is built upon a sparsity-restricted expectation-approximation-maximization procedure, which simultaneously detects varying sets of important features for multiple latent classes. It effectively screens out a large number of redundant features that contribute insignificantly to the joint mixture likelihood. Different from the existing screening methods, sEAM allows users to precisely control the number of features to be retained for each latent class. In the screening process, sEAM naturally incorporates the joint effects among features and classes, and flags the sets of key features that receive the most support from the joint mixture likelihood.

  From the prospective of theoretical analysis, I prove that the penalized log-likelihood function of sEAM procedure is non-decreased after every iteration, which results in the general convergence of the sEAM procedure. Especially, the estimation converges to a local maximum of penalized likelihood function subject to sparsity constraint. Also, I show that the sEAM procedure enjoys sure screening property. This finding gives a good way to implement the sEAM algorithm in practice. I conduct several simulations and real-world data studies to compare the sEAM procedure with three methods in terms of accuracy and computation time. The results show that the new approach consistently outperforms existing methods in terms of retaining capacity and positive selection rate.

- **Low gradient-based subsampling for corrupted massive data**

  To address corrupted massive data, I formulate a model for corrupted massive datasets where both covariate and response are contaminated. The proposed model considers contamination on both input data X and response Y. This corrupted observation model may be more realistic with respect to real data which may be subject to many different sources of measurement noise or heterogeneity in the datasets, especially in the massive cases. Compared to systematic errors in measurement error, where biases in measurement lead to a situation wherein the mean of many separate measurements differs significantly from the actual value of the measured attribute in one

direction, the new model considers the case where points may be contaminated in different directions which may result in a more complicated case.

Based on the new model, I develop a new gradient based subsampling procedure which can degrade influence of the heavy corruption in large extent. Based on an observation that the use of "large-gradient of loss" brings very often the danger of assigning higher sampling probabilities to the heavily noised observations, I show that "the low-gradient of loss" corresponds to the uncontaminated data in high probability and therefore, it may exclude the possibility of assigning high sampling probability to the heavy corrupted data. I establish the error bounds of the sampled estimator at each step from the oracle, which shows that the proposed low-gradient subsampling is statistical consistent with an nearly optimal precision. Also, I conduct extensive experiments for performance evaluations. The new method is evaluated on both synthetic data and real-world datasets with various corruption settings. I compare the new algorithm with four methods in the linear and logistic models. The results demonstrate that the proposed approach runs efficiently and consistently outperforms the best of the existing methods along multiple metrics.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 introduces a novel feature screening method for the large-N-large-p situation, where a massive dataset is naturally or manually partitioned into segments. The new method is shown to be convergent and enjoys sure screening within a finite number of iterations. Chapter 3 depicts a new method for feature screening in ultrahigh-dimensional finite mixture of regression models. Different from the existing screening methods, the new method allows users to directly control the number of features to be retained for each latent class. I justify the method by showing that it enjoys all-class sure screening property and is computationally efficient. Chapter 4 presents the corruption pattern of data and formulates the low-gradient subsampling procedure. I establish the error bounds of the sampled estimator to the estimator based on the true model and justify the statistical consistency of the new procedure. Chapter 5 gives a summary of the dissertation and puts forward a few directions for future research.

Chapters 2-4 are self-contained with chapter-specific notations. A reference list is provided at the end of the dissertation. The proofs of theorems are given in the Appendices.

Chapter 2

Distributed Hard Screening for Massive Data

2.1 Introduction

In modern scientific research, collecting large amounts of data has become increasingly prevalent. Massive datasets with a huge number of features are commonly encountered in many scientific areas. For instance, customs scan information of thousands of shipments every week. In health care research thousands of personal health or drug-usage records are collected. Thousands of key words in a URL are analyzed in Internet studies. Given a set of observations in these applications, statisticians are confronted with the task of exploring the relationship between a univariate response and a p-dimensional covariate vector.

The assumption that only a small portion of all features are predictive of the response has been widely adopted in high dimensional models. It entails that we only need to identify influential features and build an interpretive model to link them to the response variable. Therefore, there should be a pre-processing procedure to first screen out a large number of noninfluential features before an elaborative analysis. This preprocessing step is called feature screening, which has been extensively studied in the literature. For example, Fan and Lv [38] proposed a strategy to quickly compute the marginal correlation between each of the p features and the response. Features with top absolute correlations are then retained for an elaborative second-stage analysis. This procedure is referred to as the sure-independent-screening (SIS, for short). Since then, there have been further developments in SIS-based screening methods, such as sure independent ranking and screening [116], a SIS procedure based on the distance correlation [68] and iterative sure independence

screening (ISIS, for short) by [39]. Wang [102] considered feature screening via forward regression. Xu and Chen [108] proposed a screening approach based on a sparse restricted maximum likelihood estimator. Huang et al. [54] and Reese et al. [89] investigated feature screening for ultra-high dimensional categorical data. Barut et al. [6] proposed conditional sure independence screening (CSIS, for short) assuming that prior knowledge about some influential variables is available.

Nevertheless, almost all of the existing works are designed for the small-N-large-p regime, where the number of observations N often remains small or at most moderately sized, whereas the features size p is exceedingly large. When both N and p are large, this raises the so-called large-N-large-p regime, where the high computational demand becomes a major issue. For example, it takes around 60 hours to run DC-Screening [68] on a dataset with $p = N = 10000$ on a computer with a 3.2 GHz processor and 32GB memory.

On the one hand, massive data create opportunities for extracting crucial information and discovering subtle patterns. On the other hand, their huge volume poses substantial challenges to traditional data analysis, where a dataset is typically processed on a single machine. In particular, a severe challenge is the high cost incurred from data storage and computing algorithms. Developing effective tools to manage and utilize large-scale data is a recent focus in statistics and some related fields. To overcome the storage bottleneck, the Hadoop system was developed for distributed data storage and parallel processing. It utilizes the divide-and-conquer framework, which divides a large problem into several manageable subproblems and aggregates all sub-outputs to yield a final output. With the aid of Hadoop, many machine learning methods have been adapted to a distributed version for analyzing big data. For examples, McDonald et al. [81] proposed a distributed training approach for the structured perceptron. Recently, similar ideas have been applied to point estimation [67], kernel ridge regression [114], matrix factorization [77] and principal component analysis [115].

To apply the divide-and-conquer framework to feature screening, simple aggregation may be the easiest way. Compared with the aforementioned screening methods, this distributive learning framework leverages the computing power of multiple machines to avoid operations directly on the original full dataset. Thus it fits the need of efficiently processing big data very well. Panel (a) in Figure 2.1 in Section 2.2 illustrates the simple aggregation framework. Particularly, a full dataset is first divided into m manageable segments and m sub-estimates are generated from m independent implementations. Then the m local machines transfer their sub-estimates to a master machine. The final estimator is obtained by aggregating the local outputs via an aggregating function. One example of simple aggregation is screening based on marginal correlation measures. We first calculate the marginal correlation between each feature and the response for each data segment,

and then take a specific aggregating function to obtain final marginal correlations. Lastly feature screening is conducted by choosing features with the top largest correlations in absolute value. Actually, SIS can be regarded as a distributed screening procedure with simple averaging. However, choosing an appropriate aggregating function is not straightforward for non-linear correlation measures. Another drawback with respect to SIS is that it considers marginal effects of features only while ignoring their joint effects. Distributed screening via majority voting (MV) is another simple aggregation method, which differs from the SIS-based methods at the aggregation step. Chen and Xie [21] developed a distributed MV approach based on the frequency of each feature being retained at every local machine. Both methods, as two examples of simple aggregation, are easy to implement and computationally efficient, but their drawbacks should be noted as well. One particular drawback is that local computers do not interact with each other until the final aggregation step. In addition, the performance of the final estimator obtained from simple aggregation is usually unstable and highly sensitive to the segmentation number [109].

One way to improve simple aggregation is to enhance interactions between local computers. This can be achieved by allowing for more rounds of communication between a master computer and all local computers. Information stored at local computers gets further exchanged from interactions; consequently, the final output may extract more information contained in the full data. In light of this fact, Huang and Huo [53] proposed a distributed estimator with one extra round of communication after simple averaging. Jordan et al. [58] proposed a unified communication-efficient surrogate likelihood (CSL) framework through more rounds of communication. This framework provides a communication-efficient surrogate to the full likelihood function in the sense that it yields an estimator with the same convergence rate as the maximum likelihood estimator (MLE) in regular parametric models or the penalized MLE in high-dimensional models.

In this chapter, we aim to develop a reliable screening algorithm originated from a consensus regression model with an l_0-constraint. To fit the constrained consensus model, we propose to combine the augmented Lagrangian method with the alternating direction method of multipliers (ADMM). We refer to our Distributed Hard Screening procedure as the DHS. At each communication round of DHS, local computers communicate learning information with the master computer; between two rounds, local computers work on their own to update the local vectors in a parallel manner. Computational efficiency is thus greatly enhanced. It should be noted that direct communication only occurs at the master machine. More specifically, the master machine first aggregates all of the results from the local machines and then broadcasts the aggregated result as guidance to each local machine, as shown in panel (b) of Figure 2.1. The DHS procedure is superior to the existing methods in several aspects. First, it can not only fit the need of computational efficiency for

processing massive data, but also screen out most non-influential features while retaining influential ones with high probability. Second, it naturally takes into consideration the joint effects of features and enables indirect communications between local computers. Therefore, it has a good potential to provide more reliable screening results. Compared with the averaging method, DHS leverages much more information in the dataset through more communication between local machines to screen features. Furthermore, accounting for joint effects of features allows DHS to identity features that are marginally uncorrelated but jointly dependent of the response. In addition to developing numerical implementations, we also establish convergence of the DHS algorithm and show that it enjoys the sure screening property in the sense of Fan and Lv [38] in the large-N-large-p regime. We further show that the sure screening property can be achieved within finite iterations of DHS. It is worth noting that the ADMM procedure in DHS involves a non-convex, non-smooth and non-Lipschitz continuous constraint, which makes it different from classic ADMM convergence theories. The promising performance of DHS is further demonstrated by extensive numerical studies conducted by a real distributed computing platform Spark.

2.2 Methodology

2.2.1 Problem Setup

Let y denote a scalar response and $\boldsymbol{x} = (x_1, \ldots, x_p)^\intercal$ denote a p-dimensional covariate. We postulate a linear regression model between y and $\boldsymbol{x}$: $y = \boldsymbol{x}^\intercal \beta + \epsilon$, where ϵ is the random error following a normal distribution with mean 0 and variance σ^2. Given N independently and identically distributed (i.i.d) observations $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, N\}$ on $(\boldsymbol{x}, y)$, statistical inference on β is usually the main focus. This may be not feasible without imposing any extra conditions on β when p is ultrahigh. In the literature, it is a common practice to assume that the underlying true model is sparse, namely, a large portions of $\boldsymbol{x}$ is not influential on y. For example, when analyzing the shipment risk, $\boldsymbol{x}$ represents millions of shipment records while y denotes risk of the shipment. Usually, only a small number of record features have meaningful influences on the risk, thus feature screening is desirable in this application. Nevertheless, serious challenges from both analytical and computational aspects would be encountered when we conduct feature screening in the large-N-large-p regime. This is the main problem we aim to address in this chapter.

For an arbitrary subset of $\{1, \ldots, p\}$, denoted by s, let $\boldsymbol{x}_{is} = \{x_{ij} : j \in s\}$ and $\beta_s = \{\beta_j : j \in s\}$. We use $\|A\|_0$ to denote the number of nonzero components of an arbitrary vector A. Let $s^* = \{j : \beta_j^* \neq 0\}$ denote the index set of all influential features in the true model.

A feature screening problem amounts to estimating s^* from $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, N\}$. When both N and p are excessively large, addressing this problem with one single machine may be not computationally feasible. To facilitate feature screening, distributed algorithms with several machines are desirable. In particular, we randomly split the whole dataset into m segments of equal size n, i.e., $N = mn$. For $l = 1, \ldots, m$, let $\boldsymbol{X}_l \in \mathbb{R}^{n \times p}$ and $\boldsymbol{Y}_l \in \mathbb{R}^n$ denote the observed feature matrix and the response vector in the lth segment, respectively. With a prespecified screening bound k ($k < p$), also called sparsity, we consider the following distributed consensus sparse regression model with an l_0 constraint:

$$\widehat{\theta} = \underset{\alpha_1, \ldots, \alpha_m, \beta}{\arg\min} \frac{1}{2} \sum_{l=1}^{m} \|\boldsymbol{Y}_l - \boldsymbol{X}_l \alpha_l\|_2^2, \quad \text{such that} \quad \|\beta\|_0 \leq k, \alpha_l - \beta = 0, \ l = 1, \ldots, m, \quad (2.1)$$

where $\theta^\intercal = (\alpha_1^\intercal, \ldots, \alpha_m^\intercal, \beta^\intercal)$, and α_l is the auxiliary vector at the lth local machine. Note that α_l's from different local machines are required to be identical in (2.1). That is why it is called distributed consensus sparse regression. The resulting estimator of β minimizes sum of squared residuals across all segments and indicates which features are retained.

2.2.2 Distributed Hard Screening Procedure

In this subsection, we introduce the DHS algorithm to solve the minimization problem (2.1).

To facilitate parallel computing, we apply the ADMM algorithm to the minimization problem with a nonconvex and non-Lipschitz continuous constraint. More specifically, the augmented Lagrangian function for (2.1) is given by

$$\begin{aligned} L_\eta(\alpha, \beta, \rho) = \frac{1}{2} \sum_{l=1}^{m} \|\boldsymbol{Y}_l - \boldsymbol{X}_l \alpha_l\|_2^2 + \sum_{l=1}^{m} \rho_l^\intercal (\alpha_l - \beta) + \frac{\eta}{2} \sum_{l=1}^{m} \|\alpha_l - \beta\|_2^2 \qquad (2.2) \\ \text{subject to} \quad \|\beta\|_0 \leq k, \end{aligned}$$

where $\alpha^\intercal = (\alpha_1^\intercal, \ldots, \alpha_m^\intercal)$, $\rho^\intercal = (\rho_1^\intercal, \ldots, \rho_m^\intercal)$ denotes m Lagrange multiplier vectors and $\eta \in \mathbb{R}$ is a step parameter. We update the parameter estimates in (2.2) iteratively as follows:

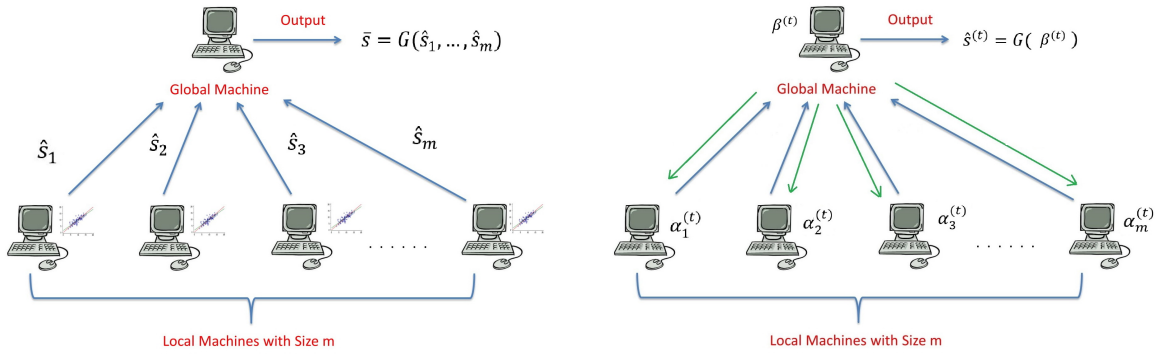
$$\begin{cases} \beta^{(t+1)} = \arg\min_\beta L_\eta \left(\alpha^{(t)}, \beta, \rho^{(t)}\right), \quad \text{subject to} \quad \|\beta\|_0 \leq k \\ \alpha_l^{(t+1)} = \arg\min_{\alpha_l} L_\eta \left(\alpha, \beta^{(t+1)}, \rho^{(t)}\right), \quad l = 1, \ldots, m \\ \rho_l^{(t+1)} = \rho_l^{(t)} + \eta \left\{\alpha_l^{(t+1)} - \beta^{(t+1)}\right\}, \quad l = 1, \ldots, m. \end{cases}$$

For the β update, we employ the hard thresholding scheme. Consequently, the closed-form

solution to the three minimization problems above is

$$\begin{cases} \beta^{(t+1)} = H_k\left(\frac{1}{m\eta}\sum_{l=1}^{m}\left\{\eta\alpha_l^{(t)} + \rho_l^{(t)}\right\}\right), \\ \alpha_l^{(t+1)} = \left\{\frac{1}{\eta}I_p - \frac{1}{\eta^2}\boldsymbol{X}_l^{\intercal}(I_n + \frac{1}{\eta}\boldsymbol{X}_l\boldsymbol{X}_l^{\intercal})^{-1}\boldsymbol{X}_l\right\}\left(\boldsymbol{X}_l^{\intercal}\boldsymbol{Y}_l + \eta\beta^{(t+1)} - \rho_l^{(t)}\right), \quad l = 1, \ldots, m \\ \rho_l^{(t+1)} = \rho_l^{(t)} + \eta\left\{\alpha_l^{(t+1)} - \beta^{(t+1)}\right\}, \quad l = 1, \ldots, m, \end{cases} \tag{2.3}$$

where $H_k(x)$ is the hard thresholding function retaining the k largest components of vector x in absolute value. The ADMM-based iterative updating procedure bypasses involved operations such as massive matrix inversion. Particularly, to update α, we only need to find the inverse of the $n \times n$ matrix $I_n + \eta^{-1}\boldsymbol{X}_l^{\intercal}\boldsymbol{X}_l$, whose computational complexity is $O(n^3)$. With the aid of the ADMM, computational efficiency of the DHS framework is greatly enhanced.



(a) (b)

Figure 2.1: Simple aggregating (Left) vs. Communication-enhanced screening (Right)

Furthermore, the auxiliary vector α_l and the Lagrange multiplier vector ρ_l are updated in a parallel manner, that is, each local machine only utilizes the segment-specific information to update these two vectors. Nevertheless, the update of β, which is conducted only at the master machine, aggregates results obtained from all local machines, and then the updated β is broadcast as guidance to update local parameters at each local machine in the next iteration. Such an updating procedure entails more effective communications among all machines. Figure 2.1 illustrates this essential difference between the simple aggregating framework and our proposed DHS framework. Additionally, in contrast to SIS, which considers marginal effects of all features only, DHS accounts for their joint effects in parameter

updates. Therefore, DHS is a communication-enhanced feature screening method and is able to capture joint effects of features.

It should be noted that DHS achieves a trade off between computational efficiency and approximation accuracy of the solution to the minimization problem (2.1). The closed-form solution to (2.2) is not necessarily identical to that of the distributed consensus sparse regression in (2.1). However, we are only concerned about the sure screening property of the final estimator, and this DHS procedure via the ADMM indeed satisfies our need. Though it is appealing to find the global minimum of (2.1), developing a computationally efficient method to find an estimated β that retains all influential feature is more desirable. In next section we show that the estimator obtained from DHS enjoys the sure screening property under some mild conditions. These arguments justify our motivation to implement DHS to find an approximate solution to (2.1). The details of parameter updates are summarized in Algorithm 2.1.

Algorithm 2.1 The DHS Procedure

Input: Training dataset $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, step parameter η, data segment number m, initialization $\{(\alpha_l^{(0)}, \beta^{(0)}, \rho_l^{(0)}) : l = 1, \ldots, m\}$.

Output: The index set of influential feature $\widehat{s}_t$.

1: Data preprocessing: randomly split training data into m parts, and store them on m local machines; initialization: $\{(\alpha_l^{(0)}, \beta^{(0)}, \rho_l^{(0)}) : l = 1, \ldots, m\}$ at $t = 0$;

2: **for** $t = 0$ to end until some stopping criterion is satisfied. **do**

3: Master machine updates variable $\beta^{(t+1)}$:

$$\beta^{(t+1)} = H_k \left\{ \frac{1}{\eta m} \sum_{l=1}^{m} \left(\eta \alpha_l^{(t)} + \rho_l^{(t)} \right) \right\}.$$

Each local machine updates variables $\alpha_l^{(t+1)}$ and $\rho_l^{(t+1)}$:

$$\alpha_l^{(t+1)} = \left\{ \frac{1}{\eta} I_p - \frac{1}{\eta^2} \boldsymbol{X}_l^\intercal (I_n + \frac{1}{\eta} \boldsymbol{X}_l \boldsymbol{X}_l^\intercal)^{-1} \boldsymbol{X}_l \right\} \left(\boldsymbol{X}_l^\intercal \boldsymbol{Y}_l + \eta \beta^{(t+1)} - \rho_l^{(t)} \right), \quad l = 1, \ldots, m,$$

$$\rho_l^{(t+1)} = \rho_l^{(t)} + \eta \left(\alpha_l^{(t+1)} - \beta_l^{(t+1)} \right).$$

4: **end for**

5: **return** $\widehat{s}_t = \{\text{indices of nonzero components of } \beta^{(t)}\}$

Remark 1. *All numerical implementations and the following theoretical results are based on the assumption of equally sized data segments. This entails that each local machine*

receives the same amount of information. However, this assumption can be easily violated in practice, especially when the entire dataset is naturally gathered from different sites. To tackle this problem, we suggest assigning different weights to each local segment. This part will be left for future research. In addition, due to the block mechanism of Spark, getting equally sized data segments is not always feasible. The numerical studies shown in Section 2.4 demonstrate superior performance of the DHS procedure for unequal data segments.

2.3 Theoretical Analysis

In this section we study theoretical properties of the DHS algorithm, including convergence and sure screening of the resulting estimator.

2.3.1 Convergence of DHS Algorithm

Although we are not interested in finding the exact solution to (2.1), it is critical to verify whether the sequence of updated parameters in the DHS procedure can converge eventually. Theorem 2.1 below shows that the sequences of both the augmented Lagrangian functions and the primal and dual residuals are able to converge.

Theorem 2.1. *In Algorithm 2.1, let* $\lambda = \max_{l=1,\ldots,m} \lambda_{\max}(\boldsymbol{X}_l^\top \boldsymbol{X}_l)$*, where* $\boldsymbol{X}_l$ *is the observed feature matrix in the* l*th segment. If*

$$\eta > \max\left\{2, (1/2)m\lambda^2 + \sqrt{(1/4)m^2\lambda^4 + m\lambda^2}\right\},$$

then the sequence $\{L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)})\}_t$ *is convergent, and the primal and dual residues are convergent, i.e.,* $\|\boldsymbol{\alpha}_l^{(t+1)} - \boldsymbol{\alpha}_l^{(t)}\| \to 0$, $\|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}\| \to 0$ *and* $\|\boldsymbol{\rho}_l^{(t+1)} - \boldsymbol{\rho}_l^{(t)}\| \to 0$ *for* $l = 1, \ldots, m$ *as* $t \to \infty$.

The proof of Theorem 2.1 relies on the monotonically decreasing property of L_η. To establish this property, we find bounds on the descent of L_η after each complete update of α_l, β and ρ_l. Specifically, we show that L_η may increase or decrease after one update of $\boldsymbol{\alpha}_l$, $\boldsymbol{\beta}$, $\boldsymbol{\rho}_l$ or the hard thresholding function but will eventually decrease after each complete update of all vectors.

Theorem 2.1 states that, L_η is convergent with an appropriate scale parameter η, and the changes of β and ρ in two consecutive iterations, which determine the primal and dual residuals in Algorithm 2.1, converge to zero eventually. Hence we can terminate the

updates in Algorithm 2.1 when the difference in two consecutive iterations is sufficiently small.

Theorem 2.1 is different from other convergence results established for the ADMM [18, 74, 92, 105]. The convergence property of the ADMM in convex problems was firstly established in [44]. When the objective function is nonconvex, some recent results [57, 79, 110] imposed assumptions directly on the updates of α, β and ρ. Hong et al. [52] and Li and Pong [66] studied theoretical properties of the ADMM in nonconvex optimization problems. In the aforementioned literature, both L_η and constraints are assumed to be Lipschitz-differentiable, which leads to an analytical expression of $\rho^{(t+1)}$ in terms of $\beta^{(t)}$ and $\beta^{(t+1)}$. But in this chapter, we develop a convergence property of DHS with an l_0 constraint, which is non-convex, non-smooth and non-Lipschitz continuous. Hence we provide new insights into theoretical frameworks of the ADMM.

2.3.2 Sure Screening of DHS Algorithm

We now provide theoretical results about feature screening based on the DHS procedure.

Suppose a screening procedure retains k out of p features such that $\|s^*\|_0 = q < k$. Then we define

$$S_+^k = \{s : s^* \subset s : \|s\|_0 \leq k\} \quad \text{and} \quad S_-^k = \{s : s^* \not\subset s : \|s\|_0 \leq k\}$$

as collections of subsets containing all influential features and subsets missing some influential features, respectively. Let $\beta^{(t)}$ be the updated β at the tth iteration from DHS. The following assumptions are needed to establish asymptotic properties of $\beta^{(t)}$.

Assumption 1. $\log p = O(N^a)$ *for some* $0 \leq a < 1$.

Assumption 2. *There exist constants* $\omega_1 > 0$, $\omega_2 > 0$, $\tau_1 > 0$ *and* $\tau_2 > 0$ *such that* $\tau_1 + \tau_2 \in (0, (1-a)/2)$,

$$\min_{j \in s^*} |\beta_j^*| \geq \omega_1 N^{-\tau_1} \quad \textit{and} \quad q < k \leq \omega_2 n^{\tau_2}.$$

Assumption 3. *There exists some constant* $c_1 > 0$, $c_2 > 0$, *such that* $|x_{ij}| \leq c_2$ *and for any data segment number* $l \in \{1, \ldots, m\}$,

$$\lambda_{\min}(n^{-1} \boldsymbol{X}_{ls}^\intercal \boldsymbol{X}_{ls}) \geq c_1$$

where $\lambda_{\min}$ *denotes the smallest eigenvalue of a matrix,* $s \in S_+^{2k}$, *and* $\boldsymbol{X}_{ls} \in \mathbb{R}^{n \times k}$ *represents the matrix consisting of features in the* l*th data segment indexed by* s *with cardinality* k.

Assumption 1 shows that dimensionality p grows at an exponential rate compared to the sample size N. In this case, the number of features can be substantially larger than the sample size. Assumption 2 entails that the minimal true nonzero coefficient does not decay too fast, such that all true signals can be distinguished from noninfluential features; This assumption also specifies a proper range of k to ensure the satisfactory performance of $\widehat{s}$. Assumption 3 plays the same role as the UUP condition in [15], which has been commonly adopted as a mild condition in the literature of high-dimensional regression models; see [16] and [102] and references therein. We establish the sure screening property for the proposed DHS algorithm under such assumptions.

Theorem 2.2. *Suppose that Assumptions 1 and 3 are met. Let $\widehat{s}$ be the set of features obtained from the DHS with step parameter $\eta \geq c_1 n$. Then, we have*

$$P(s^* \subset \widehat{s}) \to 1, as\ N \to \infty.$$

Theorem 2.2 claims that, under Assumption 1-3 and appropriate conditions of η and λ, the DHS procedure enjoys sure screening property with an overwhelming probability. Namely, it could identify important feature set which contains true model as a subset. However, Theorem 2.2 provides us a theoretical justification that DHS has such screening property under the assumption $t \to \infty$. Although it implies the feasibility of the proposed procedure for sufficient large iteration steps, it is computationally unaffordable in practice. In aspect of practical use, it is encouraged to have a sure screening property within finite iteration step. Theorem 2.3 and Corollary 2.1 below provide such a result.

Theorem 2.3. *Under the assumptions of Theorem 2.1, if DHS is initiated with $\beta^{(0)}$ satisfying*

$$\left\|\beta^{(0)} - \beta^*\right\|_2 \leq \frac{\omega_1}{6} m^{1-\tau_1} \log p \cdot n^{-\tau_1}, \tag{2.4}$$

and the number of iterations T satisfies

$$T \geq T_0 = C \log m$$

for some constant $C > 0$, then we have

$$P(s^* \subset \widehat{s}_T) \to 1,\ as\ N \to \infty.$$

Theorem 2.3 shows sure screening of DHS after $T \geq T_0 = C \log m$ iterations, where m is the number of data segments. This property assures that s^* can be correctly retained through the DHS procedure with an appropriately chosen initial value of $\boldsymbol{\beta}$. It is interesting to show that the communication bound T_0 we derived matches those required in [103] and

[114], and slightly less than the bound $O(\log(\log N/\log n))$ in [58] when $\log m \geq (\log N + (\log^2 N - 4\log N)^{1/2})/2$ and $\log N \geq 4$. These show that the proposed DHS procedure possesses the sure screening property while keeping communication cost low. From the prospect of initialization, a natural questions arises whether the proposed DHS procedure with an arbitrary initial value necessarily leads to $\widehat{s}$ that enjoys the sure screening property in practice. Unfortunately, due to the complexity of the distributed consensus sparse regression model, there is no guarantee in practice that $\widehat{s}$ obtained through an arbitrary initial value enjoys sure screening. When DHS starts with $\beta^{(0)} = 0$, the update at the first iteration corresponds to the ridge learning with regularization parameter η^{-1}. When will $\beta^{(0)} = 0$ be a suitable choice of initial values that leads to a desirable screening eventually? Theorem 2.3 provides a sufficient condition: $\max_{j\in s^*} |\beta_j^*| \leq \omega_1/6 \cdot q^{-1/2} m^{1-\tau_1} \log p \cdot n^{-\tau_1}$. Intuitively speaking, when the true model consists of adequately weak signals, taking the initial value $\beta^{(0)} = 0$ is more likely to yield an estimate with the sure screening property, as in such situations $\beta^{(0)}$ and β^* are reasonably close. It should be noted that, the relative error upper bound of an appropriate initialization $\boldsymbol{\beta}^{(0)}$ in Theorem 2.3 is $O(m^{1-\tau_1} \log p \cdot n^{-\tau_1})$, which is relative loose since $\tau_1 \leq 1/2$ from our Assumption 2. Such a relaxed assumption of initial leads to an edge obtaining initialization in a crude way such as LASSO on one data segment, where the estimation precision is $O(\sqrt{\log p} n^{-1/2})$(see [43] and [48]). Thus, in prospect of practical use, it is encouraged for user to choose LASSO-type initial value on one data segment as a crude but good enough initialization. Our simulation studies suggest that the penalty parameter in the LASSO that gives rise to $\|\beta^{(0)}\|_0 = n - 1$ is usually a good choice.

While a fine tuning of $\beta^{(0)}$ is not practically needed for DHS, an accurate $\beta^{(0)}$ may help to further boost the screening efficiency of $\widehat{s}^{(t)}$. The sure screening of the well-initialed DHS is stated as the following corollary. Specifically, we show that, with stronger restrictions on the initial value, the DHS procedure does enjoy the sure screening property within any finite number of iterations.

Corollary 2.1. *Under the setting of Theorem 2.3, if $\beta^{(0)}$ further satisfies*

$$\left\|\beta^{(0)} - \beta^*\right\|_2 \leq \frac{\omega_1}{6} \log p \cdot N^{-\tau_1}, \tag{2.5}$$

then we have

$$P(s^* \subset \widehat{s}^{(t)}) \to 1$$

for any finite $t \geq 1$, as $n \to \infty$.

Corollary 2.1 shows that the DHS procedure enjoys sure screening even with an one-step iteration if $\beta^{(0)}$ is taken delicately with an error bound of $O(\log p \cdot N^{-\tau_1})$. In this corollary,

the initialization of $\boldsymbol{\beta}^{(0)}$ is required to be appropriate to guarantee the effectiveness of DHS. Such relative error upper bound of $\boldsymbol{\beta}^{(0)}$ is achievable but stronger than that in Theorem 2.3. It worth to note that, the l_2-norm, $\|\widehat{\beta} - \beta^*\|_2$, where $\widehat{\beta}$ is the LASSO estimate under a Gaussian linear regression model, is bounded above by $C_1 N^{-1}\sqrt{\log p}$ for some positive constant C_1. Thus the constraint (2.5) is still automatically fulfilled when $\beta^{(0)}$ is set to be the LASSO estimate. By Corollary 2.1, the well-LASSO-initialed estimator of β ensures sure screening within finite iterations of DHS.

Nevertheless, Theorem 2.3 and Corollary 2.1 do not provide guidelines on the choice of sparsity k for implementing DHS in practice. Generally speaking, sure screening is more likely to ensue from a larger k. Inspired by [39], we set $k = aN^{1/3}\log N$ for some constant a in our numerical studies. It turns out this choice works quite well. Actually, we consider various choices of k in the DHS algorithm and compare their performances in Section **??**. It turns out that the performance of the DHS is robust to the choice of k, as long as it is reasonably large.

Remark 2. *Although we focus on feature screening in linear regression and study its theoretical properties, the proposed framework can be extended to generalized linear regression with a convex loss function.*

2.4 Numerical Studies

In this section, we assess the finite sample performance of the DHS algorithm through simulated examples and a real-world application. In particular, we compare DHS with CSL, MV, SIS, and ISIS. The performances of these methods depend on the correlation structure between features, the number of (influential) features, sample size, level of noise, and the number of data segments. For an honest comparison between different algorithms, we consider various simulation settings, and in each setting 200 independent runs are conducted for each algorithm, except for the case of $(N, p) = (10000, 30000)$. Due to its heavy computational burden, we only repeat 50 simulation runs for this particular case. All numerical studies are implemented on a Spark distributed computing platform consisting of 30 computers, each of which is equipped with a 12-core CPU and 64GB memory.

In numerical experiments, we choose the step parameter η in DHS adaptively as suggested by [49]. More specifically,

$$\eta^{(t+1)} = \begin{cases} h^{\text{incr}}\eta^{(t)}, & \|r^{(t)}\|_2 > \mu\|s^{(t)}\| \\ \frac{\eta^{(t)}}{h^{\text{decr}}}, & \|r^{(t)}\|_2 < \mu\|s^{(t)}\| \\ \eta^{(t)}, & \text{otherwise} \end{cases} \tag{2.6}$$

where the parameters satisfy $\mu > 1, h^{\text{incr}} > 1$ and $h^{\text{decr}} > 1$, and $r^{(t)} = \sum_{l=1}^{m} \eta(\rho_l^{(t)} - \rho_l^{(t-1)})$ and $s^{(t)} = -\eta(\beta^{(t)} - \beta^{(t-1)})$ denote the primal residual and the dual residual, respectively. This gives rise to an adaptive choice of η throughout the DHS algorithm. In the simulations, we take a relatively small η as its initial value. Then we follow the rule in (2.6) to update η at each iteration until the algorithm converges; that is, the difference in two consecutive steps is sufficiently small. This adaptive choice of η works well in the numerical studies.

2.4.1 Summary of Simulation Settings

We examine the five screening methods under linear models, while we do not include ISIS under one special design (the case of a small N/p) due to its high computational cost. Three correlation structures between candidate features are considered: the first one considers i.i.d. features; the second one considers features with an auto-correlation structure, where the neighboring features are correlated but distant ones are virtually uncorrelated; the last one sets every feature either influential or correlated with some influential features. Obviously, the first structure is the most straightforward for feature selection, while the last one is the most challenging.

More specifically, the correlation structures under consideration are listed as follows. They actually have been investigated in many relevant works; see [39] and [102] for instance. We specify other parameter values like N, p and m and generate β_j's for each model in the following subsections.
C1: Candidate features $x_1, \ldots, x_p$ are independently generated from $N(0, 1)$.
C2: Candidate features $x_1, \ldots, x_p$ follow multivariate normal with mean 0 and covariance matrix $\Sigma = (\sigma_{ij})$, where $\sigma_{jj} = 1$, $\sigma_{j,j-1} = 2/3$, $\sigma_{j,j-2} = 1/3$ and $\sigma_{jh} = 0$ for $|j - h| \geq 3$.
C3: Candidate features $x_1, \ldots, x_p$ follow multivariate normal with mean 0 and covariance matrix $\Sigma = (\sigma_{ij})$, where $\sigma_{jh} = 0.15$ for $j, h \in s^*$ and $\sigma_{jh} = 0.3$ for either j or $h \in \{1, \ldots, p\} \backslash s^*$.

The five screening methods are compared in terms of retaining capacity (RC), positive selection rate (PSR), minimal selection rate (MSR) and computational time. These metrics are defined as follows:

$$\text{RC} = \frac{\sum_{h=1}^{200} \mathbb{1}(s^* \subseteq \widehat{s}_h)}{200}, \ \text{PSR} = \frac{\sum_{h=1}^{200} \|s^* \cap \widehat{s}_h\|_0}{200\|s^*\|_0}, \ \text{MSR} = \frac{\sum_{h=1}^{200} \mathbb{1}(s^*_{\min} \subseteq \widehat{s}_h)}{200},$$

where $\widehat{s}_h$ denotes the set of features retained in the hth replicate by any of these five methods, $s^*_{\min}$ denotes the weakest signals (in absolute value) in the true model and $\mathbb{1}(\cdot)$ denotes the indicator function. RC measures the capability that a method can retain all influential features. PSR and MSR characterize selection performance from two perspectives: a high

PSR indicates that most of the influential features are retained, while a high MSR entails that features with weakest effects are retained. A good screening method should be high in both of them. Besides these selection metrics, we also report the averaged computational time (in seconds) of running each algorithm.

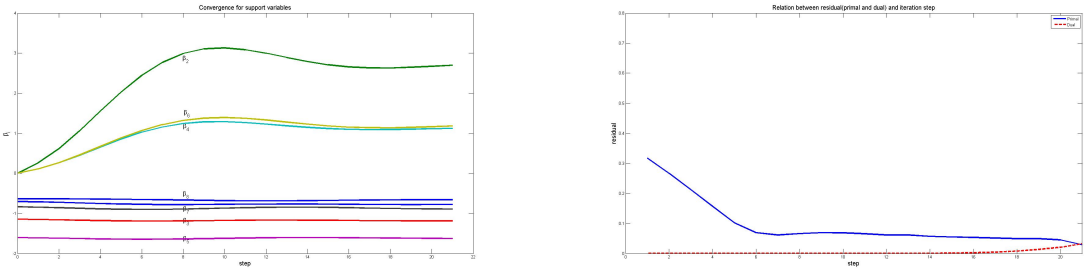
2.4.1.1 Data Generation under Four Different Designs

In this example, N i.i.d observations $(\boldsymbol{x}_i, y_i)_i$ are generated from the linear model $y = \boldsymbol{x}^\top \beta + \varepsilon$, where ε follows a mixture Gaussian $0.8N(0, \sigma_1^2) + 0.2N(0, \sigma_2^2)$ $(\sigma_1 < \sigma_2)$. We treat the 20% of data, namely, the second mixture component, as outliers to demonstrate the performance of our algorithm. The model parameters in each of the three correlation structures (C1-C3) are enumerated below:
S1: $(N, p, \sigma_1^2, \sigma_2^2) = (10000, 10000, 5, 20)$. The index set of true influential features s^* is randomly chosen from $\{1, \ldots, p\}$ with $\|s^*\|_0 = 8$. For $j \in s^*$, β_j is generated independently from $U(4\sqrt{n}\log n + |Z|)$, where U is a binary random variable with $P(U = 1) = 0.6$ and $P(U = -1) = 0.4$ and Z follows $N(0, 1)$. For $j \notin s^*$, $\beta_j = 0$.
S2: $(N, p, \sigma_1^2, \sigma_2^2) = (10000, 10000, 5, 20)$ and $s^* = \{1, 3, 5, 7, 9\}$. For $j \in s^*$, β_j is generated the same as S1 and $\beta_{(s^*)^c} = 0$, where $(s^*)^c = \{1, \ldots, p\} \setminus s^*$ is the complement of set s^*.
S3(a): $(N, p, \sigma_1^2, \sigma_2^2) = (10000, 10000, 5, 20)$ and $s^* = \{1, 2, 3, 4\}$. For $j \in s^*$, β_j is generated the same as S1 and $\beta_{(s^*)^c} = 0$.
S3(b): $(N, p, \sigma_1^2, \sigma_2^2) = (10000, 10000, 5, 20)$ and $s^* = \{1, 2, 3, 4\}$. β_{s^*} ={1, 1, 1, -0.45} and $\beta_{(s^*)^c} = 0$. The correlation structure is the same as S3(a), but X_4 is generated to be marginally uncorrelated but jointly dependent of Y.

Setting S1 is taken from Example 1 of Wang [102] with the correlation structure C1. Under setting S2, β_j's are generated in the same manner as under S1 but features are generated from the correlation structure C2. In S3(a) and S3(b), candidate features are both generated from the structure C3. But in S3(b), we fix β_{s^*} to be some specific value to ensure X_4 is marginally uncorrelated but jointly dependent of Y, which renders it the the most challenging setting for feature screening. A similar setting was considered in Example II of Fan and Lv [38]. But Fan and Lv [38] ruled out this possibility in their theoretical results. Fan and Song [40] asserted that SIS is likely to miss features that have weak marginal correlations with Y, but strong joint effects on Y.

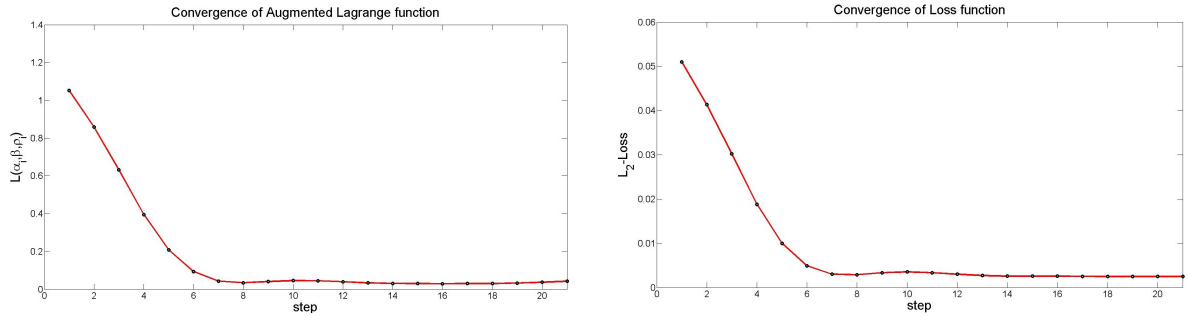
In our experiments, following the suggestion in Fan et al. [39], we choose the sparsity $k = (1/4)N^{1/3}\log N$, which is 50 under these settings. When implementing the DHS, $\beta^{(0)}$ is set as the LASSO estimate, which is obtained from applying the fast iterative shrinkage thresholding algorithm [7] to the first data segment with sparsity $n-1$. The DHS procedure is terminated when $\|\beta^{(t+1)} - \beta^{(t)}\|_2^2 < 0.01$.

2.4.1.2 Preliminary Analysis under Setting S1



(a) (b)

Figure 2.2: (a)Convergence for influential features; (b)Updates of primal and dual residuals.



(a) (b)

Figure 2.3: (a)Convergence of the augmented Lagrange function; (b)Convergence of the L_2-Loss function.

We first present some convergence results of DHS under setting S1. As displayed in Figure 2.2, the parameter estimate ($\beta^{(t)}$) and primal and dual residuals become stabilized after fewer than ten iterations. It entails that $\beta_j^{(t)}$, where $j \in s^*$, in the DHS algorithm converges to an estimator of β_j. Additionally, Figure 2.3 depicts both the trace of the augmented Lagrange function L_η defined in (2.2) and that of the associated L_2 loss function. This further justifies that the DHS procedure is convergent.

Figure 2.4 shows the RC of DHS, CSL and MV under different numbers of data segments (m); more detailed results are presented in Table 2.1. For a relatively large m, DHS outperforms both CSL and MV. The CSL algorithm leverages n samples stored at one single

local machine to approximate the global loss function; this explains why its performance is sensitive to n or m. For those no interaction-involved distributed methods like MV, m also plays a critical role: a larger m entails faster computations on local machines, but less accurate results due to fewer samples, while a smaller m ensures more accurate results on local machines but demands more computational costs. Thus m controls the trade-off between accuracy and computational efficiency on local machines for those no interaction-involved distributed methods. In contrast, the DHS algorithm enables more communications between local machines; this particular merit leads to a more robust result. Both Figure 2.4 and Table 2.1 demonstrate that the RC of DHS is less sensitive to the choice of m than MV. It should be noted that computational time is an important concern when choosing m in DHS, since the communication between local machines and the master machine to broadcast and update relevant parameters increases the computational cost, especially in the large-N-large-p regime. In these studies, we find $m = 8$ is a reasonable choice.

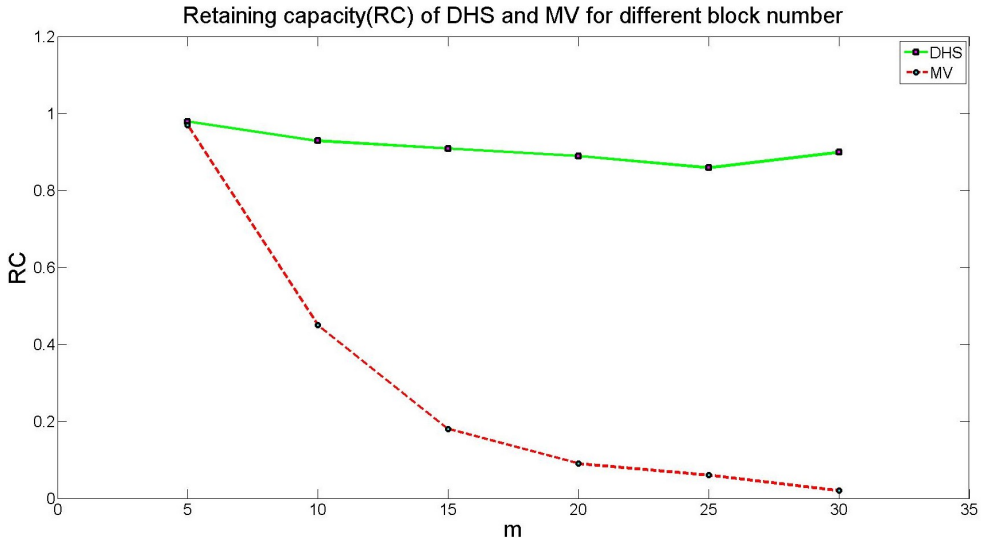


Figure 2.4: Retaining capacity(RC) of DHS and MV for different data segments number m.

We investigate the finite-sample performance of DHS for linear regression under settings S1 - S3(b) in this subsection. Tables 2.2 to 2.5 display simulation results of the five methods under different settings, and in each table we mark the best results in red and the second best in blue.

Features are independent under setting S1. Even though both p and N are large, all screening methods have a relatively high RC, PSR and MSR. The good performances

Table 2.1: Summary of the RC and computation time for the DHS, CSL and MV under different m under setting S1.

S1	m	5	10	15	20	25	30	40	50
DHS	RC	0.98	0.93	0.91	0.89	0.86	0.90	0.88	0.92
	Time	1.81	1.61	1.21	0.93	0.74	0.72	0.70	0.67
CSL	RC	0.97	0.92	0.90	0.88	0.82	0.74	0.62	0.58
	Time	3.59	3.82	3.62	3.26	3.34	3.17	2.92	2.77
MV	RC	0.97	0.45	0.18	0.09	0.06	0.02	0.02	0.01
	Time	1.36	1.62	1.55	1.39	1.47	1.34	1.26	1.21

Table 2.2: Summary of simulation results for S1.

S1		RC	PSR	MSR	Time
$m = 8$	DHS	0.96	0.99	0.97	2.12
	CSL	0.98	0.99	0.98	4.80
	MV	0.96	0.99	0.97	2.09
	SIS	0.78	0.97	0.81	0.29
	ISIS	0.84	0.97	0.89	3.24
$m = 10$	DHS	0.94	0.99	0.95	1.78
	CSL	0.92	0.99	0.92	4.45
	MV	0.46	0.86	0.52	1.88
	SIS	0.78	0.97	0.81	0.29
	ISIS	0.84	0.97	0.89	3.24

are due to the parsimonious correlation structure between features. DHS, CSL and MV outperform SIS and ISIS in terms of the RC and MSR. As for the computational cost, SIS and MV are the best and second among the five methods. There is no communication between local machines for SIS and MV, hence they can be implemented quickly. Though communications are indeed involved in DHS, it can be conducted in a relatively short time. In contrast, long iterations involved in ISIS demands substantial computational time, while a local LASSO problem needs to be solved elaborately in CSL, which makes it time consuming. Taking both the performance in feature screening and the computational cost into consideration, we find MV is superior to the other competitors under setting S1.

Under setting S2 where the features are autocorrelated, the performance of SIS deteriorates obviously. If two features are highly positively correlated, then they have a similar correlation with the response variable. Since SIS only accounts for marginal information of

Table 2.3: Summary of simulation results for S2.

S2		RC	PSR	MSR	Time
$m = 8$	DHS	0.75	0.92	0.82	4.08
	CSL	0.46	0.87	0.54	4.68
	MV	0.63	0.91	0.80	1.99
	SIS	0.37	0.84	0.59	0.24
	ISIS	0.48	0.86	0.68	2.93
$m = 10$	DHS	0.73	0.92	0.81	3.36
	CSL	0.51	0.88	0.56	4.52
	MV	0.37	0.75	0.43	1.95
	SIS	0.37	0.84	0.59	0.24
	ISIS	0.48	0.86	0.68	2.93

Table 2.4: Summary of simulation results for S3(a).

S3(a)		RC	PSR	MSR	Time
$m = 8$	DHS	0.88	0.96	0.90	3.38
	CSL	0.56	0.87	0.56	4.71
	MV	0.47	0.79	0.53	1.99
	SIS	0.20	0.67	0.41	0.27
	ISIS	0.24	0.73	0.39	2.86
$m = 10$	DHS	0.90	0.96	0.92	2.79
	CSL	0.60	0.85	0.62	4.35
	MV	0.32	0.73	0.38	1.87
	SIS	0.20	0.67	0.41	0.27
	ISIS	0.24	0.73	0.39	2.86

each feature, two highly positively correlated features cannot be retained simultaneously with a high probability by SIS. Actually, after retaining one of the two features, the other one only has a weak effect on the residual. The performances of MV and CSL are badly effected by this structure too; thus more information is needed at local machines to achieve a good performance for these two methods. DHS, however, can leverage efficient communications and account for joint effects between features to weaken the negative effect caused by this correlation structure. Consequently, as shown in Table 2.3, SIS and MV have a relative low RC, PSR and MSR, while ISIS only slightly improves the performance of SIS with substantially greater computational costs. Nevertheless, our proposed DHS can still achieve a relatively high RC, PSR and MSR without extra computational costs than under setting S1.

Table 2.5: Summary of simulation results for S3(b).

S3(b)		RC	PSR	MSR	Time
$m = 8$	DHS	0.99	0.99	0.99	3.30
	CSL	0.30	0.82	0.30	4.65
	MV	0.01	0.75	0.01	1.98
	SIS	0.00	0.75	0.00	0.24
	ISIS	0.10	0.65	0.11	2.81
$m = 10$	DHS	0.94	0.97	0.96	2.79
	CSL	0.11	0.79	0.11	4.34
	MV	0.01	0.75	0.01	1.86
	SIS	0.00	0.75	0.00	0.24
	ISIS	0.10	0.65	0.11	2.81

Under setting S3(a), a feature is either influential or correlated with other features. It is rather challenging to distinguish the influential features from non-influential ones, especially when some weak signals are involved. Table 2.4 shows that ISIS slightly outperforms SIS with considerably more computational costs. Nevertheless, MV and CSL are not as seriously affected by this correlation structure as SIS; they achieve almost half RC. We find that DHS displays a remarkably better screening performance with a high RC, PSR and MSR and a negligible increase of computational time.

Under setting S3(b), the strong correlation among x_{s^*} with a specific β_{s^*} causes X_4 to be marginally uncorrelated but jointly dependent of Y; it renders X_4 hardly identified by most screening methods. Appealing theoretical properties of SIS and MV do not apply to this unusual situation. This is further demonstrated in Table 2.5. Since SIS and MV cannot identify X_4 as an influential feature, their RC and MSR are quite low. ISIS shows a marginally better retaining power than that of SIS. CSL compares favorably with these three methods in terms of feature screening at the cost of considerably longer computational time. However, our proposed DHS achieves a remarkable increase of RC, PSR and MSR with a lower computation cost than that of CSL. A critical reason why DHS dominates the four competitors is that it accounts for joint effects of features and allows for interacting information between local machines. This is the essential difference between our algorithm and the others.

In summary, since communications between local machines and the master machine are needed to update parameters over iterations, our DHS algorithm requires slightly more computational costs than MV and SIS. However, this cost is accompanied by a remarkable improvement in feature screening. In practice, a slight increment in the computation time is acceptable for most applications.. For the linear model, DHS is among the best performers

under all settings that we consider. The discrepancy between DHS and its competitors is more distinct under settings S2-S3(b), where features are strongly correlated.

2.4.1.3 Results for logistic regression under settings S1-S3(b)

In this subsection, we examine the finite sample performance of DHS for generalized linear models. In particular, we consider logistic regression models, where the binary response y is generated from a Bernoulli distribution with

$$P(Y = 1|\boldsymbol{x}) = \frac{\exp\left(\boldsymbol{x}^\top \beta\right)}{1 + \exp\left(\boldsymbol{x}^\top \beta\right)}.$$

The model parameters in each of the three correlation structures (C1-C3) are the same as in Section 2.4.1.1 except for the noise setting. With slight abuse of notation, we still use S1 - S3(b) to denote the four settings in this section. We still consider the five methods, CSL, MV, SIS, ISIS and DHS, for comparison. We carried out 200 independent simulation runs for each method under each design.

It should be noted that, for logistic regression, there is no closed-form solution to the α_l-minimization problem, as opposed to (2.3) for linear regression. We instead employ gradient descent to solve the minimization problem. Then we calculate the screening performance measures in the same way as for the linear regression models. The comparison results are summarized in Tables 2.6-2.9.

Table 2.6: Summary of simulation results for logistic regression under S1.

S1		RC	PSR	MSR	Time
$m = 8$	DHS	1.00	1.00	1.00	2.66
	CSL	0.90	0.98	0.95	5.50
	MV	0.87	0.98	0.87	1.88
	SIS	0.99	0.99	0.99	0.24
	ISIS	1.00	1.00	1.00	2.33
$m = 10$	DHS	1.00	1.00	1.00	2.02
	CSL	0.93	0.99	0.96	5.22
	MV	0.82	0.97	0.86	1.65
	SIS	0.99	0.99	0.99	0.24
	ISIS	1.00	1.00	1.00	2.33

Similar to linear regression models, in setting S1, we find that all methods have a good performance in feature screening when features are independent of each other. Moreover,

Table 2.7: Summary of simulation results for logistic regression under S2.

S2		RC	PSR	MSR	Time
$m = 8$	DHS	0.86	0.97	0.92	2.19
	CSL	0.77	0.95	0.80	3.90
	MV	0.73	0.94	0.90	1.79
	SIS	0.60	0.92	0.75	0.24
	ISIS	0.74	0.94	0.82	2.15
$m = 10$	DHS	0.84	0.96	0.87	1.93
	CSL	0.77	0.92	0.82	3.81
	MV	0.58	0.92	0.86	1.36
	SIS	0.60	0.92	0.75	0.24
	ISIS	0.74	0.94	0.82	2.15

Table 2.8: Summary of simulation results for logistic regression under S3(a).

S3(a)		RC	PSR	MSR	Time
$m = 8$	DHS	0.82	0.94	0.85	3.13
	CSL	0.77	0.91	0.84	3.39
	MV	0.42	0.77	0.45	2.16
	SIS	0.38	0.75	0.42	0.06
	ISIS	0.42	0.76	0.45	2.32
$m = 10$	DHS	0.82	0.94	0.85	2.70
	CSL	0.77	0.93	0.83	3.15
	MV	0.30	0.72	0.37	1.67
	SIS	0.38	0.75	0.42	0.06
	ISIS	0.42	0.76	0.45	2.32

DHS performs slightly better than the other four competitors. In setting S2, RC of all methods becomes smaller than that in setting S1. Nevertheless, DHS still dominates the competitors in all of the three metrics. In setting S3, the correlation structure is least favorable to feature screening. Consequently, as observed from Tables 2.8 and 2.9, MV, SIS and ISIS have a low MSR, especially in setting S3(b). The comparison between DHS and CSL is not sharp in terms of retaining capacity. Yet the significant improvement of DHS over CSL under this setting is a lower computational cost.

Table 2.9: Summary of simulation results for logistic regression under S3(b).

S3(b)		RC	PSR	MSR	Time
$m = 8$	DHS	1.00	1.00	1.00	2.54
	CSL	0.97	0.98	0.98	3.18
	MV	0.00	0.75	0.00	1.82
	SIS	0.00	0.75	0.00	0.25
	ISIS	0.00	0.75	0.00	2.34
$m = 10$	DHS	1.00	1.00	1.00	2.08
	CSL	0.94	0.97	0.96	2.79
	MV	0.00	0.75	0.00	1.68
	SIS	0.00	0.75	0.00	0.24
	ISIS	0.00	0.75	0.00	2.14

2.4.1.4 Performance under Other Feature Dimension

To fully assess the proposed DHS algorithm, we vary the ratio of sample size N and feature dimension p. Previous empirical studies have demonstrated the performances of the five screening methods when this ratio is relatively large. In this subsection, we consider three smaller ratios of N and p than that in Section 2.4.1.1. To make results comparable, we remove the outliers of setting S1 in Section 2.4.1.1 and ISIS is not implemented due to its high computational cost in the large-p regime.

Table 2.10: Summary of performances of DHS, CSL, SIS and MV under different feature dimension p (without outliers).

S1	Methods	RC	PSR	MSR	Time
$(N, p) = (1000, 5000)$	DHS	0.97	0.99	0.98	0.13
	CSL	0.76	0.96	0.82	0.99
	MV	0.56	0.93	0.60	0.48
	SIS	0.66	0.94	0.72	0.02
$(N, p) = (1000, 10000)$	DHS	0.88	0.98	0.90	0.15
	CSL	0.66	0.94	0.86	2.12
	MV	0.37	0.89	0.43	1.04
	SIS	0.51	0.92	0.59	0.04
$(N, p) = (1000, 20000)$	DHS	0.81	0.97	0.85	0.24
	CSL	0.58	0.93	0.72	4.42
	MV	0.16	0.81	0.23	2.10
	SIS	0.43	0.90	0.48	0.08

Table 2.10 summarizes the comparison between DHS and CSL, MV and SIS under three feature dimensions: $p = 5000, 10000, 20000$. We take $N = 1000$ and $m = 10$ in all three cases. Under setting S1, we find that CSL, MV and SIS perform worse in feature screening when N/p decreases. More specfically, when this ratio decreases from 1/5 to 1/20, RCs of MV and SIS drop sharply and end up with smaller than one half. CSL compares favorably with MV and SIS, but its RC is only marginally larger than one half when $N/p = 1/20$. However, our DHS algorithm can achieve a high RC, PSR and MSR regardless of the ratio. It indicates that our method is applicable to problems with high-dimension features or a small ratio of N and p. It should be noted that the advantage of DHS in feature screening is at the cost of longer computational time, which is needed for communications between machines. Further, we find that the computational time of DHS increases rapidly with the decay of N/p. Given that we only use $m = 10$ local machines in all three cases, this rapid increase seems reasonable since 10 seems to be a small number when screening $p = 20000$ features. This finding suggests that to meet requirements of computational costs, the number of data segments should be chosen carefully.

2.4.2 Additional Simulation Studies

As mentioned in Section 2.4.1.1, we choose sparsity $k = 1/4N^{1/3} \log N$ in our previous simulations; it closely matches the recommenced k value in Fan et al. [39] for SIS and ISIS. Thus the comparisons we have made so far are pretty fair. To evaluate the impact of sparsity on the performances of these screening methods, we conduct simulation studies under various sparsity: $k = 25, 30, 35, 40, 50$. By definition, RC (PSR, MSR) is positively correlated with k. This assertion is demonstrated in Figure 2.5, which displays the traces of RC obtained from the five algorithms. Obviously, a larger k provides each method with more chances to retain influential features, while a smaller k enforces a small number of retained features. When sparsity of one method is smaller than that of the true model, this method would definitely miss some influential features. Furthermore, regardless of the sparsity we choose, DHS always enjoys the highest RC, and almost attains 0.9 eventually. This reveals the advantage of the proposed DHS algorithm from another perspective.

2.4.3 Real Data Examples

We now apply the proposed DHS to two real world datasets, Buzz (http://archive.ics.uci.edu/ml/datasets/Buzz+in+social+media+) and Million Song (http://archive.ics.uci.edu/ml/datasets/YearPredictionMSD), to further investigate its performance on feature screening. The Buzz dataset consists of 583,250 instances of Twitter discussions on

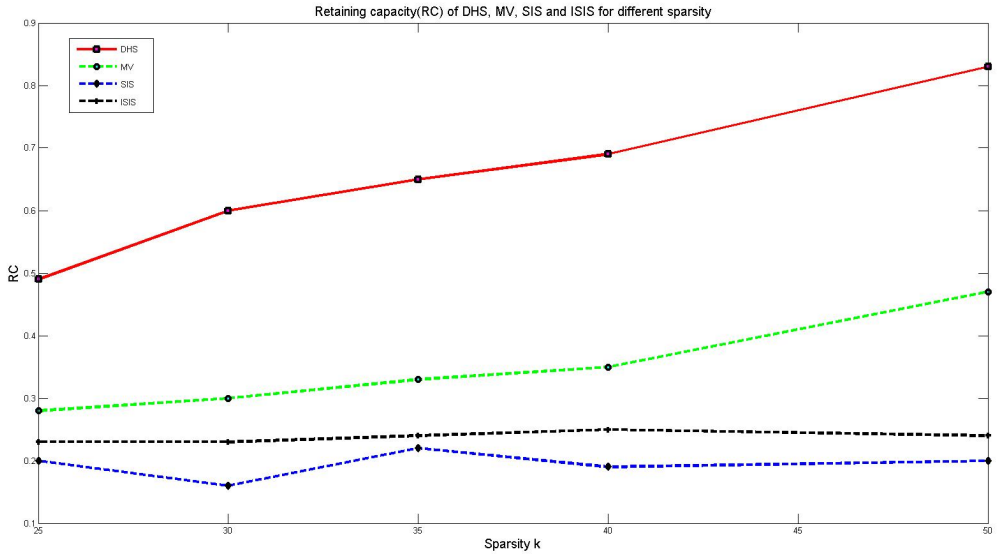
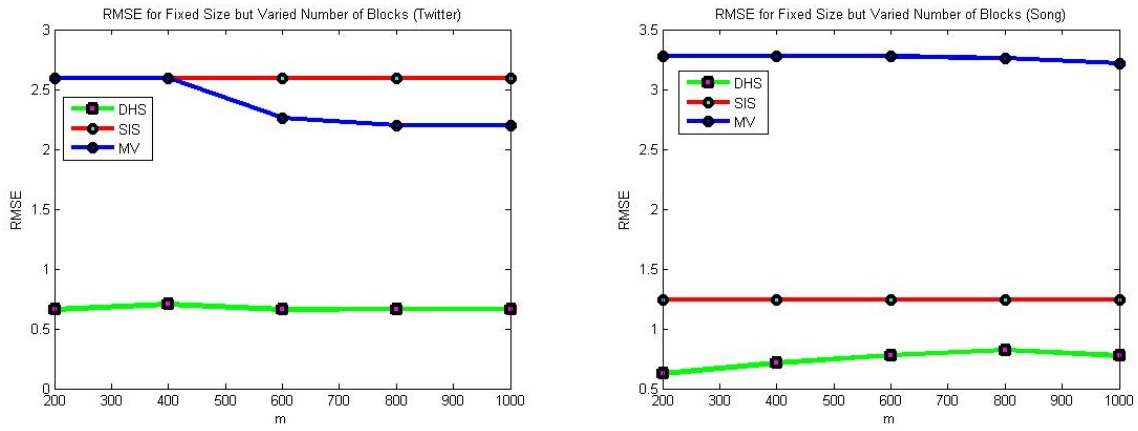


Figure 2.5: Retaining capacity(RC) of DHS, MV, SIS and ISIS with different sparsity k.

topics related to new technologies in 2013. Each instance contains $p = 77$ features related to that discussion. It is of interest to predict the number of active discussions Y based on these features denoted by $\boldsymbol{x}$. To facilitate computations, we only include the instances with $Y \in [20, 200]$ in our analysis. The Million Song dataset consists of 463,715 training instances and 51,630 testing instances. Each instance is a song (track) released between 1922 and 2011, and each song is represented as a 90-dimensional vector of timbre information about that song. In our studies, the training set size is set to be 100,000 and the testing set size is 5,000. To mitigate the effect of the feature magnitude, we standardize each attribute in $\boldsymbol{x}$ such that it has a zero mean and a unit standard deviation.

Note that p is much smaller than N in these two applications. To mimic the large-N-large-p regime, we employ kernel ridge regression (KRR) to generate N features. In particular, we choose the Gaussian kernel with $r = 10$ for the Buzz dataset and $r = 6$ for the Million Song dataset when conducting KRR. We consider various numbers of data segments: $m = 200, 400, 600, 800, 1000$ for both Buzz and Million Song when running DHS, MV and SIS. Due to the heavy computational burden, ISIS is not implemented.

In these two applications, we apply the screening methods to select a smaller number say 100 of relatively "high-quality" samples as the training data; more details about conducting distributed KRR can be found in Xu et al. [109]. Then we conduct KRR on this selected sample and calculate the root mean square error (RMSE) on the testing data. As shown



(a) (b)

Figure 2.6: The generalization performance in terms of RMSE when DHS, MV and SIS applied to (a) Buzz dataset and (b) Million Song dataset.

in Figure 2.6, DHS achieves a much smaller RMSE than MV and SIS when sparsity is 100 regardless of m. This results indirectly demonstrates that, compared with other screening methods, DHS enjoys a greater power of retaining influential features.

2.5 Summary and Conclusion

In this chapter, we develop a new distributed feature screening approach, which is computationally convenient and is therefore suitable for huge data. The proposed algorithm outperforms the existing marginal-information-based distributed methods by taking into consideration joint effects between features. Furthermore, we establish convergence of the algorithm and its sure screening property within a finite number of iterations.

Empirical studies demonstrate that compared with existing screening methods, DHS has a remarkable capacity of retaining relevant features under all designs that we consider. In contrast to simple aggregating methods, DHS enables more communication among all machines to leverage more information contained in the whole dataset. These advantages render the DHS promising for feature screening in the large-N-large-p regime.

Chapter 3

Sparse EM Screening for High-Dimensional Mixture Regression

3.1 Introduction

In modern scientific research, data analysts frequently deal with high-dimensional complex data [37]. One challenge in the analysis of such data is that observations are potentially collected from heterogenous populations that are made up of multiple latent sub-populations (classes); the behaviors of a statistical estimator or procedure may vary from one class to another. In regression analysis, this amounts to varying relationships between a response and a set of covariates (features) across the sub-populations. To tackle this challenge, finite mixture of regressions have been widely used to model population heterogeneity for better inference and interpretation [5, 51, 64, 83].

In many applications, data are often collected on a large number of features at the initial stage of a study. For example, in bacteriology, researchers may hybridize a certain bacterium under different fermentation conditions and investigate its varying vitamin production rate linkable to thousands of genetic makers [93]. In functional genomics, tens of thousands of regulating motifs may be examined in order to identify those that contribute to different sets of genes [2]. When the number of features p is large, fitting a finite mixture of regression to a dataset is numerically challenging and often leads to poor interpretative estimates. To cope with the large p situation, it is often reasonable to assume that only a handful of features are relevant to the analysis. With this assumption,

penalized likelihood methods have been proposed for fitting sparse finite mixture of regressions [30, 60, 93]. While these methods are inspiring, when p is of the exponential order in the sample size (ultrahigh-dimensionality), they can be significantly hampered due to computational efficiency, statistical accuracy, and algorithmic stability [39].

To ease the implementation difficulties for the ultrahigh-dimensional situations, one practical strategy is to screen out most irrelevant features before an elaborate analysis. Such a strategy is referred to as feature screening, which has attracted a great deal of attention in the past decade. With dimensionality reduced from ultrahigh to low, analytical difficulties are reduced drastically. In the literature, there is vast research on feature screening. The seminal work of [38] proposed sure independence screening for ultrahigh-dimensional linear models. This method has been extended to generalized linear and additive models [35, 40], proportional hazards models [46], and quantile regression [107]. In the same spirit, a series of model-free screening methods have been developed solely based on marginal utilities between the response and features [26, 68, 116]. Researchers have made attempts at improving the marginal approaches by incorporating the joint effects among features into the screening process [59, 102, 108]. Recently, the partition-based techniques have been also used in the context of feature screening [59, 70, 94].

Despite the surge of research on feature screening, the existing methods are mainly developed based on certain dependence measure between the response and features over the entire population. In finite mixture of regressions, however, the response-feature-dependence varies across the latent classes; hence, the sets of relevant features in the model are class-specific. This makes feature screening in mixture regression a challenging task that has not been carefully studied. In the spirit of [38], [112] considered a marginal screening method for mixture regressions using Bayesian information criterion. [61] developed a boosting-based screening technique that requires multiple rounds of sequential scanning over the feature space. While these methods are encouraging, both require ad hoc steps to conduct class-specific screening. Also, as to be revealed in Section 3.4, their practical performance can be less satisfactory for data with complex structures.

In this chapter, we propose a novel screening method for ultrahigh-dimensional Gaussian mixture regressions. The new method is built upon a sparsity-restricted expectation-approximation-maximization procedure, which simultaneously detects varying sets of important features for multiple latent classes. It effectively screens out a large number of redundant features that contribute insignificantly to the joint mixture likelihood. Unlike the existing penalization techniques, the new method allows users to directly control the number of features to be retained in each latent class. In the screening process, the joint effects between features are naturally accounted and the class-specific screening results are produced without ad hoc steps. Numerically, the new method seeks to locally fit a mix-

ture of L_0-sparse regressions under a modified expectation-maximization framework. An approximation step is designed to facilitate solving the involved non-convex and discrete optimization problem. In practice, each step of the proposed method can be explicitly carried out without the need of conducting sub-iterations or complex numerical operations. These merits give the new method an edge to potentially outperform the existing screening methods. Under mild conditions, we show that the proposed method enjoys the sure screening property for each latent class in the sense of [38]. The promising performance of the method is supported by a series of numerical examples.

3.2 Screening with sEAM

3.2.1 Model and Problem Setup

Let y be a real-valued response variable and $x = (x_1, \ldots, x_p)^{\mathrm{T}}$ be a vector of p features that may have effect on y. In a Gaussian mixture regression with K components (classes), the conditional distribution of y given x is

$$f(y \mid x, \theta) = \sum_{k=1}^{K} \pi_k \ \phi(y; x^{\mathrm{T}}\beta_k, \sigma_k^2), \tag{3.1}$$

where $\phi(\cdot; \mu, \sigma)$ is the Gaussian density with mean μ and variance σ^2, $\beta_k = (\beta_{k1}, \ldots, \beta_{kp})^{\mathrm{T}}$ is the vector of regression coefficients in the k-th component, and the mixing proportions $\pi_k > 0$ satisfying $\sum_{k=1}^{K} \pi_k = 1$. Here, β_{k1} can be treated as an intercept when $x_1 = 1$. The collection of all $Kp + 2K - 1$ parameters is denoted by $\theta = \{\pi_1, \ldots, \pi_K, \beta_1, \ldots, \beta_K, \sigma_1^2, \ldots, \sigma_K^2\}$. In what follows we assume K is fixed, and we refer to (3.1) as the full model containing all the features.

Suppose that $\{y_i, x_i\}_{i=1}^n$ are collected independently from (3.1), where the sample size n is much smaller than the number of features p. We assume that model (3.1) is sparse in the sense that only a subset of features are influential on y. Specifically, let β_k^* be the true value of β_k and $\|\cdot\|_0$ denote the L_0 norm of a vector indicating the number of its non-zero elements. We assume β_k^* is sparse such that $\|\beta_k^*\|_0 \ll p$, for $1 \le k \le K$. Thus, only features with indices in $\mathcal{M}_k^* = \{j : \beta_{kj}^* \neq 0, 1 \le j \le p\}$ are considered to be relevant for the kth class.

The goal is to design a data-dependent method that screens out most irrelevant features in each class of (3.1). Features after screening form a refined model, on which a subsequent analysis can be carried out with an affordable cost and an improved accuracy. Since $\mathcal{M}_k^*$

may vary across different classes, a desirable screening procedure should produce class-specific screening results for a better interpretative value.

3.2.2 Joint Screening in Mixture Regressions

Feature screening in model (3.1) is challenging since the class memberships of observations are unknown and the effect of a feature may vary across the K classes. Two or more classes may share the same feature, whose class-specific effects can be masked or distorted. Moreover, the features x_j are usually correlated and may have joint effects in some classes of (3.1).

The aforementioned challenges make the conventional single-population-index-based screening methods less effective, which we have also observed in our simulation study. This motivates us to design a new screening method via the joint likelihood based on model (3.1). Specifically, with a random sample $\{y_i, x_i\}_{i=1}^n$, the (conditional) joint log-likelihood of θ is

$$\ell_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log(f(y_i \mid x_i, \theta)). \tag{3.2}$$

For the screening proposes, we consider a sparsity-restricted optimization problem

$$\max_{\theta} \ell_n(\theta) - \sum_{k=1}^{K} \psi(\sigma_k^2), \quad \text{subject to } \|\beta_k\|_0 \le s_k, k = 1, \ldots, K, \tag{3.3}$$

where $s_k > 0$ is a user-specified screening size showing the number of features to be retained after screening in the kth class. The penalty function

$$\psi(\sigma_k^2) = \frac{1}{n}\left\{ \frac{\nu_n^2}{\sigma_k^2} + \log\left(\frac{\sigma_k^2}{\nu_n^2}\right) - 1 \right\}$$

with ν_n^2 being the sample variance of $\{y_i\}_{i=1}^n$ is introduced to protect ℓ_n from approaching to infinity in the extreme cases where $\sigma_k \to 0$, for some k. [17].

The sparsity constraint ensures that a solution to (3.3) in β_k would have a sparse structure with the total number of its non-zero elements controlled by s_k. This amounts to identifying s_k features supported most by the joint likelihood. Hence, these s_k features form a refined feature set, while all other $p - s_k$ features are screened out for the kth class. With a small number of features retained in each class, one can readily obtain a reduced

model from (3.1). Thus, with a $s_k \ll p$, a solution to (3.3) can be viewed as a feature screener, which naturally takes the joint effects among classes or features into account.

While (3.3) is conceptually simple, finding its global solution can be numerically challenging, due to the complexity of ultrahigh-dimensional combinatorial optimization, as well as the compound structure of the mixture density. However, since our goal is feature screening, finding the global solution to (3.3) is not necessary. In fact, it suffices if we can obtain a good local solution that retains all relevant features for different classes. This inspires us to develop a joint feature screener based on a sparse expectation-approximation-maximization updating procedure associated with (3.3), to which we now turn.

Let z_{ik} be a binary indicator with $z_{ik} = 1$ if the ith observation belongs to class k and $z_{ik} = 0$ otherwise, such that $\sum_{k=1}^{K} z_{ik} = 1$. If z_{ik}s were all observed, the log-likelihood of θ based on the so-called complete data $\{y_i, x_i, z_{i1}, \ldots, z_{iK}\}_{i=1}^{n}$ is

$$\ell_n^c(\theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \Big\{ \log(\pi_k) + \log \phi(y_i, x_i^{\mathrm{T}} \beta_k, \sigma_k^2) \Big\}.$$

Of course, z_{ik} are unknown in practice and one typically needs to work with an expected $\ell_n^c(\theta)$. Specifically, with an initial value $\theta^{(0)}$, the proposed screening procedure recursively carries out the following three steps.

Expectation step: Given the current estimate $\theta^{(t)}$, compute the conditional expectation of $\ell_n^c(\theta)$ with respect to z_{ik} given $\{y_i, x_i\}_{i=1}^{n}$ and $\theta^{(t)}$, by

$$E(\ell_n^c(\theta) \mid \{y_i, x_i\}_{i=1}^{n}, \theta^{(t)}) = Q(\theta; \theta^{(t)}) = \sum_{k=1}^{K} g(\beta_k, \sigma_k^2; \theta^{(t)}) + \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik}^{(t)} \log(\pi_k), \quad (3.4)$$

where

$$g(\beta_k, \sigma_k^2; \theta^{(t)}) = \frac{1}{n} \sum_{i=1}^{n} r_{ik}^{(t)} \log\big(\phi(y_i, x_i^{\mathrm{T}} \beta_k, \sigma_k^2)\big)$$

and

$$r_{ik}^{(t)} = E(z_{ik} \mid y_i, x_i, \theta^{(t)}) = \frac{\pi_k^{(t)} \phi(y_i, x_i^{\mathrm{T}} \beta_k^{(t)}, \sigma_k^{2(t)})}{\sum_{k=1}^{K} \pi_k^{(t)} \phi(y_i, x_i^{\mathrm{T}} \beta_k^{(t)}, \sigma_k^{2(t)})}.$$

Approximation step: Approximate $g(\beta_k, \sigma_k^{2(t)}; \theta^{(t)})$ in β_k at $\beta_k = \beta_k^{(t)}$ by

$$h(\beta_k; \theta^{(t)}) = g(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) + (\beta_k - \beta_k^{(t)})^{\mathrm{T}} g'(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) - \frac{u}{2} \|\beta_k - \beta_k^{(t)}\|_2^2, \quad (3.5)$$

where $\|\cdot\|_2$ denotes the L_2 norm, $u > 0$ is a scale parameter, and

$$g'(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) = \frac{\partial g}{\partial \beta_k}|_{\beta_k=\beta_k^{(t)}} = \frac{1}{n}\sum_{i=1}^{n} r_{ik}^{(t)} \sigma_k^{-2(t)}(y_i - x_i^{\mathrm{T}}\beta_k^{(t)})x_i.$$

Maximization step: For $k = 1, \ldots, K$, update the model parameters by

$$\begin{aligned}
\pi_k^{(t+1)} &= \arg\max_{\pi_k} \sum_{i=1}^{n} r_{ik}^{(t)} \log(\pi_k), \\
\sigma_k^{2(t+1)} &= \arg\max_{\sigma_k^2} g(\beta_k^{(t)}, \sigma_k^2; \theta^{(t)}) - \psi(\sigma_k^2), \\
\beta_k^{(t+1)} &= \arg\max_{\beta_k} h(\beta_k; \theta^{(t)}), \quad \text{subject to } \|\beta_k\|_0 \le s_k.
\end{aligned} \tag{3.6}$$

Unlike the expectation-maximization algorithm used in the existing penalization methods, the $\beta_k^{(t)}$-updating here involves solving an ultrahigh-dimensional L_0-constrained optimization problem, which is numerically costly. To facilitate the computation, we design an Approximation step to first approximate $g(\beta_k, \sigma_k^2; \theta^{(t)})$ in (3.4) by a simple quadratic function h in (3.5), which is additive and separable in the elements of β_k. We then update $\beta_k^{(t+1)}$ in the Maximization step by solving (3.6) using h, which serves as a surrogate of g based on the current estimate $\beta_k^{(t)}$. By doing so, we essentially convert a p-dimensional non-convex optimization problem into p univariate problems, each of which can be solved explicitly (see Section 3.2.3).

The form of (3.6) ensures that $\beta_k^{(t)}$ has at most s_k non-zero elements corresponding to a set of s_k features retained for class k; features associated with zero coefficients can be regarded as irrelevant and to be removed from the kth class. Hence, after $t = T$ iterations, the method retains only a set of features in class k with indices in

$$\widehat{\mathcal{M}}_k^{(T)} = \{j : \beta_{kj}^{(T)} \neq 0, 1 \le j \le p\}. \tag{3.7}$$

The class-specific screening is done simultaneously for all classes in model (3.1), as the $\beta_k^{(t)}$-updating can be carried out in parallel with respect to k in the M-step. Also, since the procedure is developed based on a complete joint likelihood $\ell_n^c(\theta)$, the joint effects among classes and features are naturally incorporated into the screening (updating) process.

The technique used in (3.5) is inspired from an iterative hard-thresholding algorithm, which was studied in [9] for numerically solving a compressed sensing problem. We embed the quadratic approximation into an expectation-maximization framework and focus on feature screening in ultrahigh-dimensional mixture regression from a statistical point of view. As mentioned, our goal is to screen out most redundant features for different classes of (3.1) rather than solving the motivating problem (3.3).

3.2.3 Implementation Details

While the proposed method provides an attractive route for joint feature screening in mixture regression, its efficiency depends on a few implementation factors. In this subsection, we discuss how to carry out the proposed method in practice.

Carrying out the Maximization step: It involves solving three simple maximization problems (3.6), each of which has closed-form solution. In particular, one can conveniently use the standard derivative method to obtain the updates

$$\begin{aligned} \pi_k^{(t+1)} &= \frac{1}{n}\sum_{i=1}^{n} r_{ik}^{(t)}, \\ \sigma_k^{2(t+1)} &= \frac{\sum_{i=1}^{n} r_{ik}^{(t)}(y_i - x_i^{\mathrm{T}}\beta_k^{(t)})^2 + 2\nu_n^2}{\sum_{i=1}^{n} r_{ik}^{(t)} + 2}. \end{aligned} \tag{3.8}$$

For the $\beta_k^{(t)}$-updating, the special form of the surrogate function h makes the corresponding L_0 problem analytically solvable. For the convenience of presentation, let $Y = (y_i, \ldots, y_n)^{\mathrm{T}}$ be the $n \times 1$ response vector, $X = (x_1, \ldots, x_n)^{\mathrm{T}}$ be the $n \times p$ data matrix, and $R_k^{(t)}$ be a $n \times n$ diagonal matrix with element (i, i) equaling to $(r_{ik}^{(t)})^{1/2}/\sigma_k^{(t)}$. It can be shown that (3.6) has an explicit solution

$$\beta_k^{(t+1)} = H_{s_k}\left(\beta_k^{(t)} + \frac{1}{nu}(\tilde{X}_k^{(t)})^{\mathrm{T}}(\tilde{Y}_k^{(t)} - \tilde{X}_k^{(t)}\beta_k^{(t)})\right), \tag{3.9}$$

where $\tilde{X}_k^{(t)} = R_k^{(t)}X$, $\tilde{Y}_k^{(t)} = R_k^{(t)}Y$, and $H_{s_k}(\cdot)$ is a truncation operator setting all but the largest (in absolute value) s_k entries of a vector as zero.

It is seen that the implementation of the Maximization step only involves simple numerical operations, which ensure the efficiency of the proposed screening procedure.

Choice of s_k and u: The screening size s_k controls the number of features to be retained for the kth class after screening. The choice of s_k should reflect a user's belief or prior knowledge on the total number of relevant features in the kth class. Intuitively, a larger s_k increases the chance of retaining all relevant features, while a smaller s_k brings more interpretive value and computational convenience for the subsequent in-depth analysis. One practical strategy is to set s_k to be three times larger than the anticipated number of relevant features. In Section 3.4, we implement the method with $s_k = 0.85n^{1/3}\log n$. In fact, the proposed method is able to conduct accurate screening over a wide range of s_k (see Section 3.4.2).

The scale parameter u in (3.5) controls the distance moved from $\beta_k^{(t)}$ to $\beta_k^{(t+1)}$. While a smaller u helps to obtain an improved $\beta_k^{(t+1)}$ in terms of the value of the objective function in (3.3), the procedure may fail to converge when u is overly small. In Section 3.3, we derive a lower bound of u that ensures the convergence of the proposed procedure. In practice, to balance algorithm convergence and iteration efficiency, one may begin with a small u at each iteration and gradually increase its value until obtaining an improved value of the objective function in (3.3). This seems to be an effective way for achieving sufficiently fast convergence in our numerical studies.

Initialization and termination: Since our goal is feature screening, our experience shows that a fine tuning on initialization $\theta^{(0)}$ of the algorithm is not necessary. In our simulation study, for $\{\pi_k^{(0)}\}_{k=1}^K$ we choose a set of random values between 0 and 1 that satisfy $\sum_{k=1}^K \pi_k^{(0)} = 1$, and set $\sigma_k^{(0)} = \nu_n/K$ with ν_n being the sample standard deviation of $\{y_i\}_{i=1}^n$. For $\beta_k^{(0)}$, we use a standard L_1-penalized least squares estimate [95] with a sparsity close to ks_k. These empirical choices seem to work satisfactorily in our examples.

The algorithm can be terminated when the change of objective value in (3.3) is small enough. Alternatively, one may stop the algorithm when the memberships of the non-zero entries in $\beta_k^{(t)}$ for $k = 1, \ldots, K$ remain unchanged in a number of consecutive iterations.

3.3 Theoretical Analysis

In this section, we provide theoretical justification for the proposed screening method. In particular, we aim to answer two key questions: 1) is the expectation-approximation-maximization updating procedure convergent; 2) does the method leads to sure screening for each latent class in the sense of [38].

For the convenience of presentation, we introduce a few additional notations as follows. Let

$$\tilde{\ell}_n(\theta) = \ell_n(\theta) - \sum_{k=1}^K \psi(\sigma_k^2) \tag{3.10}$$

denote the adjusted log-likelihood which is the objective function in problem (3.3). Let $\mathcal{M}_k \subset \{1, 2, \ldots, p\}$ be an arbitrary feature index set for class k; we denote $X_{\mathcal{M}_k}$ as the sub-matrix of X containing the columns that are specified in $\mathcal{M}_k$. We measure the cardinality of a set by $\tau(\cdot)$ and let ρ be the largest eigenvalue of $X^{\mathrm{T}}X/n$. Moreover, let $s_k^* = \tau(\mathcal{M}_k^*)$ denote the number of relevant features in the kth class and let $(\beta_k^*, \sigma_k^{2*}, \pi_k^*)$ be the true value of $(\beta_k, \sigma_k^2, \pi_k)$, for $k = 1, \ldots, K$.

Clearly, the screening performance of the proposed method relies on a stable updating of model parameters in (3.6). It is thus important to first check whether the parameter updating would stabilize over the iterations. In Theorem 3.1 below, we show that the updating procedure is convergent in terms of the objective value of (3.3).

Theorem 3.1. *Let* $\theta^{(t)} = (\pi_1^{(t)}, \ldots, \pi_K^{(t)}, \beta_1^{(t)}, \ldots, \beta_K^{(t)}, \sigma_1^{2(t)}, \ldots, \sigma_K^{2(t)})$ *be the* t*th update of the parameter* θ *based on* (3.4)-(3.6). *With any given* $\theta^{(0)}$, *if* $u > \max_k\{\rho/\sigma_k^{2(t)}\}$, *we have*

$$\tilde{\ell}_n(\theta^{(t+1)}) \geq \tilde{\ell}_n(\theta^{(t)})$$

for $t \geq 1$.

Theorem 3.1 indicates that, when the scale parameter u is large enough, the proposed procedure necessarily improves the adjusted likelihood within the feasible region of θ. The new procedure thus enjoys a similar monotonicity property of the standard expectation-maximization algorithm [82]. Since $\tilde{\ell}_n(\theta)$ is bounded above, $\tilde{\ell}_n(\theta^{(t)})$ will stabilize as $t \to \infty$. At each iteration, $\{\beta_k^{(t)}\}_{k=1}^K$ is updated in parallel to receive a better support from the adjusted joint likelihood, while it fully complies with the requirement of $\|\beta_k^{(t)}\| \leq s_k$. It thus simultaneously leads to reliable and class-specific screening results for all the K classes.

In Theorem 3.1, we derive a lower bound of u to guarantee the improvement of $\tilde{\ell}_n(\theta^{(t)})$. This bound is purely theoretical and may not necessarily provide a practical guidance for the choice of u, as computing ρ can be computationally costly when p is large. Our empirical experience shows that the proposed procedure usually enjoys the monotonicity property with a u below this bound at certain iterations. In practice, one may use an adaptive tuning strategy for u, as discussed in Section 3.2.3.

While Theorem 3.1 justifies the convergence of $\tilde{\ell}_n(\theta^{(t)})$, it does not rule out the possibility that $\theta^{(t)}$ itself does not have a limit; this may hinder us from obtaining a stable screening result. In the next theorem, we give a sufficient condition to ensure the convergence of $\theta^{(t)}$.

Theorem 3.2. *Suppose that the conditions of Theorem 3.1 are satisfied. If* $X_{\mathcal{M}_k}^{\mathrm{T}} X_{\mathcal{M}_k}$ *is positive-definite for any* $\mathcal{M}_k \subset \{1, \ldots, p\}$ *with* $\tau(\mathcal{M}_k) \leq s_k, k = 1, \ldots, K$, *then*

$$\theta^{(t)} \to \tilde{\theta} \quad \textit{as} \quad t \to \infty,$$

where $\tilde{\theta}$ *is a local maximum of* $\tilde{\ell}_n(\theta)$ *subject to* $\|\beta_k\|_0 \leq s_k, k = 1, \ldots, K$.

Theorem 3.2 implies that, when X has a proper structure, $\theta^{(t)}$ converges to a local solution of (3.3) and thus $\widehat{\mathcal{M}}_k^{(t)}$ in (3.7) stabilizes after a sufficient number of iterations. This finding echoes the existing convergence results of a standard expectation-maximization

procedure [11, 82, 97]. The structural requirement of the theorem corresponds to condition 2 of [93] and condition A4 of [16]. With a moderate s_k, this requirement is usually satisfied when the features are not strongly correlated.

By Theorems 3.1 and 3.2, we have gained insights on the convergence of the proposed updating procedure. We now turn to evaluate the screening performance of the method. To this end, we investigate the behavior of $\widehat{\mathcal{M}}_k^{(T)}$ asymptotically, when both the sample size n and the number of features p diverge to infinity. Our investigation is based on the following technical assumptions.

Assumption 1. $\log p = O(n^a)$, *for some* $a \in [0, 1)$.

Assumption 2. *There exists* $\kappa_1, \kappa_2 \geq 0$ *and positive constants* $\omega_1, \omega_2, \omega_3$ *such that*

$$\omega_1 n^{-\kappa_1} \leq | \beta_{kj}^* | \leq \omega_2 n^{\kappa_2} \quad \textit{and} \quad \frac{\|\beta_k^*\|_1}{\sigma_k^*} \geq \omega_3$$

for any $j \in \mathcal{M}_k^*$ *and* $k = 1, \ldots, K$.

Assumption 3. *Suppose that* $E(x_j) = 0$ *and* $| x_j | \leq \omega_4, j = 1, \ldots, p$, *for some* $\omega_4 > 0$. *Let* $\Sigma = cov(x)$ *be the covariance matrix of* x. *There exists positive constants* ω_5, ω_6 *such that*

$$\omega_5 < \lambda_{min}(\Sigma) \leq \lambda_{max}(\Sigma) < \omega_6,$$

where $\lambda_{min}(\Sigma)$ *and* $\lambda_{max}(\Sigma)$ *are the smallest and largest eigenvalues of* Σ, *respectively.*

Assumption 4. *There exist constants* $\omega_7, \kappa_3 > 0$ *such that the screening size* s_k *satisfies*

$$\max\left\{1, \frac{16}{(\eta^{-1} - 1)^2}\right\} s_k^* \leq s_k \leq \omega_7 n^{\kappa_3},$$

where $\eta = 1 - C_1 + \alpha$, *for some* $\alpha \in (0, 1/3)$ *and* $C_1 \in (\alpha, 1)$.

Assumption 1 is typical in ultrahigh-dimensional regression; it allows the number of features to grow up to an exponential rate of n^a. Assumption 2 requires moderate decaying and diverging rates of β_k^* in the asymptotic process; it also sets a lower bound on signal-to-noise ratio so that $\mathcal{M}_k^*$ is detectable. The first part of Assumption 3 corresponds to a boundary condition on the features; it is merely introduced to facilitate the theoretical proofs. This condition may be relaxed to a random x_j with a sub-Gaussian distribution [40, 112]. The second part of Assumption 3 corresponds to a model identifiability condition introduced in [102]. Since $\omega_4, \omega_5, \omega_6$ are arbitrary, this assumption does

not impose additional restrictions for using the proposed screening method in practice. Assumption 4 specifies a proper range of s_k to ensure the satisfactory performance of $\widehat{\mathcal{M}}_k^{(T)}$; this assumption is compatible with the structural requirement of X in Theorem 3.2.

With the assumptions above, we justify the screening effectiveness of the new method in the following theorem.

Theorem 3.3. *Suppose that Assumptions 1-4 are satisfied with* $\kappa_1 + 2\kappa_2 + 3\kappa_3 < (1-a)/2$ *and procedure* (3.4)-(3.6) *is implemented with a* $\theta^{(0)}$ *such that*

$$\left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \leq \frac{\omega_1}{8}(\log p)^{\frac{1}{2}} n^{-\kappa_1} \quad \text{and} \quad |\ \sigma_k^{2(0)} - \sigma_k^{2*}\ | \leq \omega_8 \sigma_k^{2*}$$

for some $\omega_8 > 0$ *and* $k = 1, \ldots, K$. *If* $\omega_6 \sigma_k^{-2(t)} < u \leq C_1^{-1}\omega_5 \sigma_k^{-2(t)}$ *and* $T \geq C_3 \log\left(C_2^{-1}(n/s_k^*)^{\frac{1}{2}}\right)$ *with constants* $C_2, C_3 > 0$, *then when* n *is sufficiently large, we have*

$$\text{pr}\left(\mathcal{M}_k^* \subset \widehat{\mathcal{M}}_k^{(T)}\right) \geq 1 - 4s_k^* \exp(-C_4 n^a) \tag{3.11}$$

for some $C_4 > 0$ *and any* $k \in \{1, \ldots, K\}$.

Theorem 3.3 implies that, with an appropriate initial value $\theta^{(0)}$ and sufficient iterations T, the proposed procedure removes most irrelevant features while retaining all the relevant features for each class of a mixture regression, with an overwhelming probability. This property is referred to as the sure screening property, which is desired for a good screening method [38]. More specifically, (3.11) indicates that, when the number of iterations is in the same or higher order of $\log(n/s_k^*)^{1/2}$, the probability of missing a relevant feature by $\widehat{\mathcal{M}}_k^{(T)}$ goes to zero at an exponential rate as $n \to \infty$. Our rate here is nearly optimal in the sense that it matches the common screening rate of the conventional feature screening in a non-mixture setup [20, 116]. This justifies both effectiveness and efficiency of the new method.

Theorem 3.3 requires $\beta_k^{(0)}$ to be in an $O((\log p)^{1/2} n^{-\kappa_1})$ neighborhood of β_k^* for some $\kappa_1 \in (0, 0.5)$. This is due to the fact that $\beta_k^{(t)}$ may only lead to a local solution of (3.3) and not all local solutions are good for feature screening. This requirement specifies a sufficient condition to ensure the screening consistency of $\widehat{\mathcal{M}}_k^{(T)}$ in theory. For mixture regressions, it is known that a regularized expectation-maximization estimator of β_k can achieve an error bound of $O((s_k^* \log p/n)^{1/2})$ [111]; it thus serves as a good initial choice for the proposed method. Our experience shows that the new method works reasonably well even with a crude initial value. See Section 3.2.3 for the practical choices of the initial values.

While a fine tuning of $\beta_k^{(0)}$ is not practically needed, an accurate $\beta_k^{(0)}$ may help to further boost the screening efficiency of $\widehat{\mathcal{M}}_k^{(T)}$. We illustrate this point in the following corollary.

Corollary 3.1. *Under the conditions of Theorem 3.3, if* $\|\beta_k^{(0)} - \beta_k^*\|_2 \leq \omega_1(\log p/n)^{1/2}/8$, *for* $k = 1, \ldots, K$, *then we have, for any finite* $t \geq 1$,

$$\text{pr}\left(\mathcal{M}_k^* \subset \widehat{\mathcal{M}}_k^{(t)}\right) \to 1, \text{ as } n \to \infty.$$

Corollary 3.1 implies that, with an accurate $\beta_k^{(0)}$, the proposed method enjoys the sure screening property even with a one-step iteration. This, together with the monotonicity property (Theorem 3.1), suggests that the proposed method can also be used as a sparsity-restricted refining procedure for a regularized estimator or other screening methods designed for mixture regression.

3.4 Numerical Studies

We assess the finite sample performance of the proposed method via a series of numerical examples and a real-world application. All numerical experiments are conducted using software MATLAB on Windows computers with 3.6 GHz CPUs and 32GB memory.

3.4.1 Screening Performance

In order to evaluate the proposed method, we generate $n = 250$ random copies of $\{y, x\}$ based on model (3.1) with $K = 2$, where $x = (x_1, x_2, \ldots, x_p)^{\mathrm{T}}$ follows a $p = 1000$ dimensional multivariate normal distribution with mean zero. We set $\text{var}(x_j) = 1$ for $j = 1, \ldots, p$ and consider three different correlation structures of x as follows.

Correlation 1: $\text{corr}(x_j, x_l) = 0$, for $j \neq l \in \{1, \ldots, p\}$.

Correlation 2: $\text{corr}(x_j, x_l) = 0.35$ when $|j - l| = 1$; $\text{corr}(x_j, x_l) = 0.15$ when $|j - l| = 2$ and $\text{corr}(x_j, x_l) = 0$ elsewhere.

Correlation 3: $\text{corr}(x_j, x_l) = 0.1$ when j or $l \in \mathcal{M}_k^*$, and $\text{corr}(x_j, x_l) = 0.2$ elsewhere.

In Correlation 1, features are independent from each other; feature screening under this setup should be fairly straightforward. In Correlation 2, we increase the level of difficulty by making features correlated with their neighbors. In Correlation 3, we consider a compound symmetry structure, where a correlation presents for any pair of features; in this challenging setup, the true relevance of a feature is likely to be masked or distorted.

For each of the above correlation structures, we consider the index sets $\mathcal{M}_1^*$ and $\mathcal{M}_2^*$ of relevant features under the following two scenarios.

Scenario 1: $\mathcal{M}_1^*$ and $\mathcal{M}_2^*$ are formed by 4 and 6 randomly selected indices from $\{1, \ldots, p\}$, respectively.

Scenario 2: $\mathcal{M}_1^* = \{1, 3, 5, 10\}$ and $\mathcal{M}_2^* = \{1, 2, 3, 4, 7, 25\}$.

In both scenarios, the regression coefficients for features in $\mathcal{M}_1^*$ are $(3.7, -2.9, 3.2, -2.2)$; the regression coefficients for features in $\mathcal{M}_2^*$ are $(3.2, 3.3, 4.5, 2.3, -4.0, 2.7)$. All other regression coefficients are set to zero and the corresponding features are considered irrelevant. In comparison with Scenario 1, Scenario 2 is likely to pose more challenges for sure screening, as the relevant feature x_3 has opposite effects' signs in the two classes; this makes the detection of x_3 harder by the conventional single-index-based methods.

We consider both balanced and unbalanced mixtures. In the balanced case, we set $(\pi_1, \pi_2) = (0.5, 0.5)$ and $(\sigma_1, \sigma_2) = (3, 5)$; in the unbalanced case, we set $(\pi_1, \pi_2) = (0.3, 0.7)$ and $(\sigma_1, \sigma_2) = (3, 3)$.

For each of the simulated dataset, we use the proposed method for feature screening by retaining $s_k = 0.85n^{1/3} \log n \approx 30$ features for each class. We follow the discussion in Section 3.2.3 to initialize the algorithm and terminate the updating after $T = 50$ iterations.

We compare our method with the marginal screening method of [112] and the boosting method of [61]. Since both of them do not conduct the class-specific screening, their generic screening results apply to each of the classes. We also include the penalized method of [93] in our comparison, although it is not mainly designed for ultrahigh-dimensional feature screening. Since the penalized method does not allow the precise control on the number of features to be retained, we follow the tuning strategy in [93] to obtain a sparse penalized estimate of β_k and use it for the screening purposes on class k.

For each method, we assess its screening accuracy in terms of the sure screening rate and the positive selection rate based on $V = 100$ repetitions. Specifically, let $\widehat{M}_k(v)$ denote the index set of the retained features for the kth class at the vth repetition and $\tau(\cdot)$ denote the cardinality of a set. We calculate the two rates for the kth class and over all the classes as below.

$$\mathrm{SSR}_{all} = \frac{1}{V}\sum_{v=1}^{V} I(\mathcal{M}^* \subset \{\cup_k \widehat{\mathcal{M}}_k(v)\}), \quad \mathrm{SSR}_k = \frac{1}{V}\sum_{v=1}^{V} I(\mathcal{M}_k^* \subset \widehat{\mathcal{M}}_k(v)),$$

$$\mathrm{PSR}_{all} = \frac{1}{V}\sum_{v=1}^{V} \frac{\tau\left(\mathcal{M}^* \cap \{\cup_k \widehat{\mathcal{M}}_k(v)\}\right)}{\tau(\mathcal{M}^*)}, \quad \mathrm{PSR}_k = \frac{1}{V}\sum_{v=1}^{V} \frac{\tau\left(\mathcal{M}_k^* \cap \widehat{\mathcal{M}}_k(v)\right)}{\tau(\mathcal{M}_k^*)},$$

where $I(\cdot)$ is a 0-1 indicator function and $\mathcal{M}^* = \cup_k \mathcal{M}_k^*$. We also report the average computational time (in seconds) of each method as a measure of the screening efficiency. The

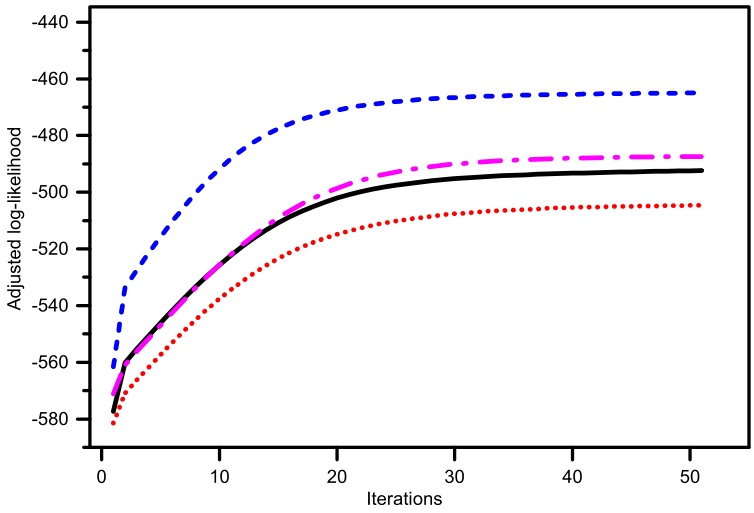


Figure 3.1: Updating paths of the adjusted log-likelihood over iterations of the sparse expectation-approximation-maximization procedure. The four paths shown in the figure (black solid, blue dashed, red doted, fuchsia dot-dashed) are based on four randomly generated samples under Scenario 1 with Correlation 1 and unbalanced classes.

results are summarized in Tables 3.1–3.3 corresponding to Correlations 1–3, respectively. The methods in comparison are marginal screening (green doted), forward screening (black dashed), L_1-penalized method (blue dot-dashed), and sparse expectation-approximation-maximizati Before turning to the analysis of the results on screening performance, it would be of interest to first check the convergence of the proposed procedure. In Figure 3.1, we show the updating paths of the adjusted log-likelihood $\tilde{\ell}_n(\theta^{(t)})$ in (3.10) for four simulated datasets from an unbalanced mixture under Scenario 1 and Correlation 1. It is seen that the adjusted log-likelihood substantially increases within the first 15 iterations and quickly reaches a stable status in about 30 iterations. The high efficiency of the parameter updating echoes Theorem 3.1 in Section 3.2 and lays a foundation of effective feature screening.

From Table 3.1, we observe that, except for the marginal method, all methods work reasonably well when $\pi_1 = \pi_2$ and features are independent with each other. From Tables 3.2 and 3.3, when correlations present among features, the accuracy of both the forward and penalized methods start degrading, while our method still maintains high overall sure screening rate. When $\pi_1 = \pi_2$, the penalized method seems to struggle in Scenario 2, where one relevant feature x_3 has the opposite effects in two classes; in particular, under Correlation 3, its overall sure screening rate drastically drops from 88% in Scenario 1 down

Table 3.1: Screening results for a two-component mixture regression under Correlation 1 with $(n, p) = (250, 1000)$. The methods in comparison are the marginal screening method of [112] (GAUMIX), the forward screening method of [61] (Boost), the L_1-penalized method of [93] (FMRLasso), sparse expectation-approximation-maximization screening (sEAM).

Class	Method	Scenario	SSR_{all}	SSR_1	SSR_2	PSR_{all}	PSR_1	PSR_2	Time(s)
Balance	GAUMIX	1	0.52	0.72	0.73	0.94	0.93	0.95	0.47
		2	0.35	0.45	0.44	0.87	0.84	0.89	0.48
	Boost	1	0.87	0.90	0.92	0.97	0.97	0.98	10.61
		2	0.80	0.82	0.85	0.95	0.95	0.96	10.51
	FMRLasso	1	0.95	0.96	0.98	0.99	0.99	0.99	6.80
		2	0.86	0.91	0.88	0.97	0.97	0.98	6.43
	sEAM	1	0.97	0.98	0.98	0.99	0.99	0.99	1.94
		2	0.96	0.97	0.97	0.99	0.99	0.99	1.93
Unbalance	GAUMIX	1	0.02	0.02	0.87	0.77	0.45	0.98	0.47
		2	0.02	0.03	0.95	0.77	0.46	0.99	0.47
	Boost	1	0.68	0.68	0.99	0.96	0.91	0.99	11.20
		2	0.55	0.55	0.99	0.94	0.85	0.99	11.59
	FMRLasso	1	0.90	0.90	1.00	0.99	0.97	1.00	6.84
		2	0.90	0.90	1.00	0.98	0.97	1.00	6.23
	sEAM	1	0.98	0.98	1.00	0.99	0.99	1.00	2.04
		2	0.96	0.96	1.00	0.99	0.98	1.00	1.76

to 49% in Scenario 2. In comparison with the balanced classes, the setup of unbalanced classes brings more challenges for sure screening of the smaller class. This is reflected by the reduced sure screening rate of class 1 in all three tables. The forward method seems to be particularly sensitive to the class size, as we observe about 20%-30% loss in its overall sure screening rate for the unbalanced cases.

In summary, the proposed method shows a promising performance in all the settings we considered. Even in the most challenging setup, it still leads to a decent overall sure screening rate of 67% which is substantially higher than that of its competitors. As for the computational cost, our method is the second best among the four methods by being about 3 times faster than the penalized method and 5 times faster than the forward method. All these make the proposed method an efficient and reliable tool for feature screening in ultrahigh-dimensional mixture regression.

To further test the performance of the method for a more complex model, we generate

Table 3.2: Screening results for a two-component mixture regression under Correlation 2 with $(n, p) = (250, 1000)$.

Class	Method	Scenario	SSR_{all}	SSR_1	SSR_2	PSR_{all}	PSR_1	PSR_2	Time(s)
Balanced	GAUMIX	1	0.56	0.69	0.84	0.95	0.92	0.97	0.47
		2	0.31	0.57	0.63	0.91	0.88	0.93	0.47
	Boost	1	0.86	0.88	0.92	0.97	0.96	0.98	10.82
		2	0.74	0.85	0.82	0.96	0.96	0.96	10.73
	FMRLasso	1	0.90	0.93	0.95	0.98	0.98	0.99	5.41
		2	0.66	0.77	0.76	0.95	0.94	0.95	6.19
	sEAM	1	0.99	1.00	0.99	0.99	1.00	0.99	2.10
		2	0.97	0.97	0.98	0.99	0.99	0.99	1.92
Unbalanced	GAUMIX	1	0.05	0.05	0.93	0.78	0.47	0.98	0.48
		2	0.05	0.05	0.97	0.79	0.48	0.99	0.47
	Boost	1	0.65	0.65	1.00	0.95	0.89	1.00	11.44
		2	0.47	0.47	1.00	0.93	0.84	1.00	11.05
	FMRLasso	1	0.89	0.89	1.00	0.98	0.97	1.00	5.40
		2	0.57	0.57	1.00	0.95	0.98	1.00	5.45
	sEAM	1	0.99	0.99	1.00	0.99	0.99	1.00	1.95
		2	0.95	0.95	1.00	0.99	0.98	1.00	2.01

$n = 250$ independent copies of (y, x) from model (3.1) with $K = 3$, $p = 1000$, $(\pi_1, \pi_2, \pi_3) = (0.3, 0.3, 0.4)$, and $(\sigma_1, \sigma_2, \sigma_3) = (0.5, 2, 6)$. In this example, Correlation 2 is used and three sets of relevant features are $\mathcal{M}_1^* = \{1, 3, 5, 10\}$, $\mathcal{M}_2^* = \{1, 3, 8, 12\}$, $\mathcal{M}_3^* = \{1, 2, 3, 4, 7, 25\}$ with coefficients $(4.7, -3.9, -4.2, -3.2)$, $(2.2, -3.9, -3.2, 3.2)$, $(4.1, 4.0, -4.3, 3.3, -4.1, 3.7)$, respectively. We summarize the results based on $V = 100$ repetitions in Table 3.4. It is seen that, with the number of classes increased from 2 to 3, the difficulty for all-class sure screening elevates significantly. In this case, the out-performance of the new method is even better observed; it consistently achieves the highest screening accuracy with a moderate cost. We also examined the performance of the methods with $K = 4, 5$, and the results show that the screening task becomes more difficult as K increases while our method is still the overall winner.

Table 3.3: Screening results for a two-component ($K = 2$) mixture regression under Correlation 3 with $(n, p) = (250, 1000)$.

Class	Method	Scenario	SSR_{all}	SSR_1	SSR_2	PSR_{all}	PSR_1	PSR_2	Time(s)
Balanced	GAUMIX	1	0.00	0.00	0.00	0.70	0.50	0.83	0.47
		2	0.00	0.00	0.02	0.79	0.73	0.82	0.47
	Boost	1	0.76	0.78	0.89	0.95	0.92	0.98	11.04
		2	0.68	0.69	0.79	0.93	0.89	0.95	11.75
	FMRLasso	1	0.88	0.94	0.94	0.98	0.98	0.99	6.27
		2	0.49	0.54	0.61	0.90	0.87	0.92	5.57
	sEAM	1	0.96	0.96	1.00	0.99	0.99	1.00	2.01
		2	0.92	0.93	0.96	0.98	0.98	0.99	2.03
Unbalanced	GAUMIX	1	0.00	0.13	0.04	0.81	0.77	0.84	0.47
		2	0.00	0.03	0.01	0.72	0.56	0.83	0.48
	Boost	1	0.43	0.43	1.00	0.93	0.84	1.00	11.48
		2	0.28	0.28	1.00	0.90	0.77	1.00	12.12
	FMRLasso	1	0.51	0.51	1.00	0.94	0.85	1.00	5.37
		2	0.45	0.45	1.00	0.92	0.81	1.00	5.41
	sEAM	1	0.78	0.78	1.00	0.97	0.94	1.00	1.93
		2	0.67	0.67	1.00	0.96	0.91	1.00	2.03

Table 3.4: Screening results for a three-component ($K = 3$) mixture regression under Correlation 2 with $(n, p) = (250, 1000)$.

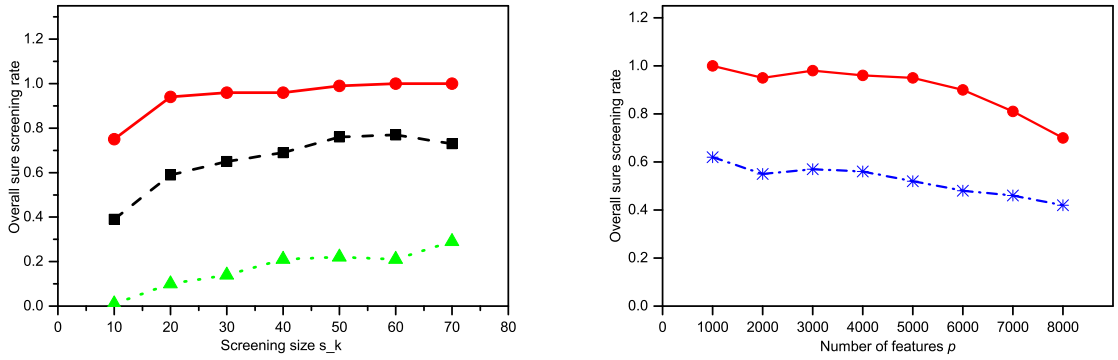
Method	SSR_{all}	SSR_1	SSR_2	SSR_3	PSR_{all}	PSR_1	PSR_2	PSR_3	Time(s)
GAUMIX	0.12	0.47	0.33	1.00	0.90	0.85	0.82	1.00	1.93
Boost	0.18	0.63	0.59	0.22	0.86	0.89	0.87	0.86	12.37
FMRLasso	0.43	0.87	0.50	0.98	0.93	0.95	0.83	0.99	8.27
sEAM	0.92	0.98	0.94	0.99	0.99	0.99	0.98	0.99	1.62

3.4.2 Sensitivity Analysis

We assess how sensitive the proposed method is with respect to the choice of screening size s_k, feature dimensionality p, and model misspecification on the number of classes K. To this end, we re-conduct a few simulations in Section 3.4.1 with the following extended setups. As before, the results are averaged based on 100 repetitions.

To check the impact of s_k, we re-conduct the analysis on a two-class mixture regression

by retaining $s_k = (10, 20, 30, 40, 50, 60, 70)$ key features for each class. We exclude the penalized method of [93] from this experiment as it does not allow users to precisely control the number of features to be retained. In Figure 3.2(a), we show the overall sure screening rate of each screening method over varying screening sizes under Correlation 2 and Scenario 2 with the unbalanced classes. It is clear that a larger s_k helps to improve the chance of retaining all relevant features for all methods. Yet, as discussed in Section 3.2.3, one may not want to make s_k overly large. From Figure 3.2(a), we observe that the proposed method is able to conduct accurate screening over a wide range of s_k. This provides the users a high flexibility of using it in different applications.



(a) (b)

Figure 3.2: (a) Overall sure screening rate of the selected methods over different screening sizes s_k under Scenario 2 with Correlation 2 and the unbalanced classes for $(n, p) = (250, 1000)$. (b) Overall sure screening rate of the selected methods over different dimensionality p under Scenario 2 with Correlation 1 and the unbalanced classes for $n = 100$. The methods in comparison are marginal screening (green doted), forward screening (black dashed), L_1-penalized method (blue dot-dashed), and sparse expectation-approximation-maximization screening (red solid).

To check the impact of p, we re-conduct the analysis on a two-class mixture regression by reducing the sample size down to $n = 100$ and gradually increasing p from 1000 to 8000. We set $s_k = 20 + p/100$ and focus on the case of unbalanced classes with $\sigma_1 = \sigma_2 = 1$. In this experiment, both marginal method of [112] and forward method of [61] are excluded due to their relatively inferior performance in the "small-p" examples shown in Section 3.4. In Figure 3.2(b), we show the overall sure screening rate of the proposed method and the penalized method over different values of p under Correlation 1 and Scenario 2. As anticipated, the higher dimensionality poses more challenges for sure screening. In

comparison with the penalized method, the new method leads to a higher accuracy for all values of p under our consideration. In particular, even when p is as large as 7000, the proposed method still achieves a decent overall sure screening rate of 81% with merely $n = 100$ observations. Thanks to its efficient updating, the proposed method is able to maintain a moderate computational cost over a wide range of p (see Table 3.5). These seem to give the new method an edge on its competitors for the ultrahigh-dimensional applications.

In the previous examples, the proposed screening method is used with a pre-specified number of classes K matching the underlying model. However, in practice, the value of K may be wrongly specified. To check the impact of the potential model misspecification on K, we re-conduct the analysis by running the method with a variety of anticipated values of K in $[1, 9]$ on a mixture regression with three classes. In this experiment, we set $(n, p, s_k) = (200, 1000, 15)$ and use the same model setup as the last example in Section 3.4. In Figure 3.3, we report the overall sure screening rate of the proposed method over different anticipated values of K, where the $K = 3$ case corresponds to the correct specification of model (3.1). It is seen that, when $K < 3$, one under-specifies the number of classes and this leads to insufficient degrees of freedom for the method to retain relevant features in all three classes. When $K \geq 3$, the K is correctly or overly specified and the performance of our method is fairly promising. These observations seem to indicate that the proposed method is relatively robust against the over-specification of K. Therefore, in practice, when one has little knowledge about the number of classes in (3.1), it may be beneficial to use the proposed method with a slightly liberal choice of K.

Table 3.5: The averaged computational time (in seconds) of the penalized method and the proposed method over different p on a dataset of $n = 100$ under Scenario 2 with Correlation 1 and unbalanced classes.

Method	p=1000	2000	3000	4000	5000	6000	7000	8000
FMRLasso	0.9	8.2	21.5	41.0	67.0	94.7	133.1	179.0
sEAM	0.9	2.0	2.9	4.0	5.1	6.3	7.6	9.0

3.4.3 A Real Data Example

We apply the proposed screening method to analyze a genetic dataset, which contains the expression levels of $p = 4088$ genes measured on $n = 71$ samples of genetically engineered mutants of Bacillus Subtilis (a common bacterium in soil). It is believed that the amount

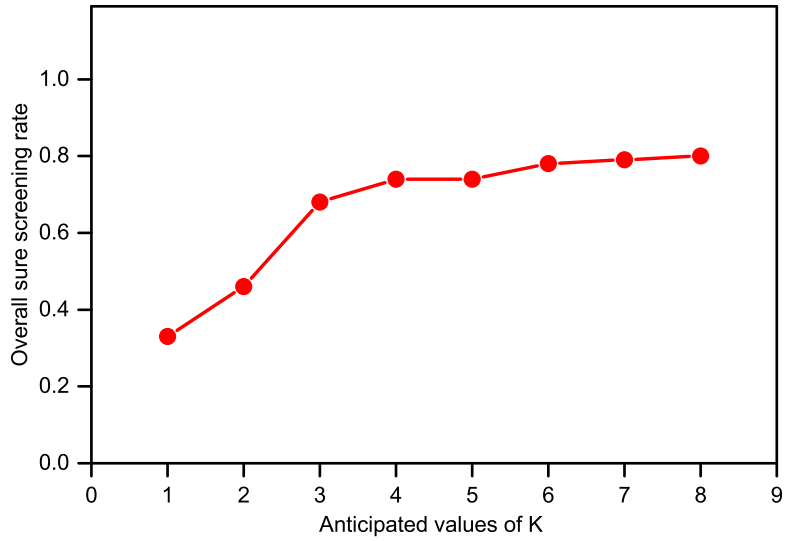


Figure 3.3: Sure screening rate of sparse expectation-approximation-maximization screening over different anticipated values of K on a three-component mixture regression with $(n, p) = (200, 1000)$.

of vitamin B2 produced by Bacillus Subtilis is affected by genetic factors; it is of interest to identify the genes that are influential in this regard. To this end, the analysts measured the vitamin B2 production rate for each of the Bacillus Subtilis samples and attempted to associate the production rate (y) with some of the genes (x). This dataset is available at: `http://www.annualreviews/org`. Readers may refer to [13] for more details.

The dataset is known to be fairly heterogenous, as the Bacillus Subtilis samples were hybridized repeatedly during a fed-batch fermentation process under different conditions. As such, we build a Gaussian mixture regression of the vitamin B2 rate on all the 4088 genes and use the proposed screening method to remove the irrelevant genes. We follow [93] to set the number of classes $K = 3$ and adopt a "leave-one-out" screening strategy as follows. For each $i \in \{1, 2, \ldots, 71\}$, we create a training set to retain $(s_1, s_2, s_3) = (10, 8, 6)$ key genes for the three classes, respectively. The retained genes together with the final $\beta_k^{(t)}$-update in (3.6) form a refined mixture regression model, which is used to predict the vitamin B2 rate of the ith sample.

In Figure 3.4, we show the top 10 genes in terms of their relative frequencies of being retained for different classes over the 71 repetitions. Accordingly, our method suggests 7 influential genes for all classes of Bacillus Subtilis, while gene 4006, gene 942, gene 946

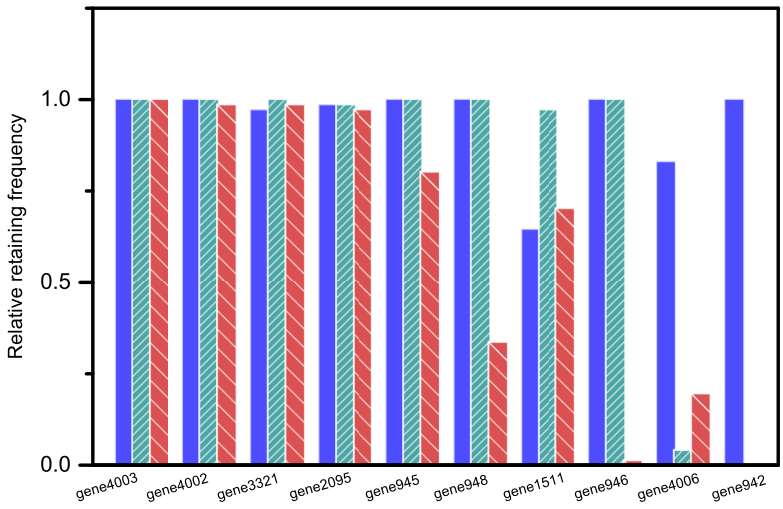


Figure 3.4: Relative frequencies for the top 10 genes that are retained most by the proposed screening method in the real data example. The bars in blue, turquoise, and red represent the three latent classes in the analysis.

seem to be influential only for one or two classes. To justify the obtained screening results, we show in Figure 3.5 the boxplots of the prediction errors for the models refined by the proposed method. For comparison, we also report the results obtained by the forward and penalized methods. It is seen that the proposed method leads to the best predictive model among the three methods. This seems to indicate the effectiveness of the new screening method.

3.5 Concluding Remarks

In this chapter, we proposed a new method for feature screening in ultrahigh-dimensional finite mixture of regression models. Different from the existing screening methods, our method allows users to directly control the number of features to be retained for each latent class. In the screening process, the proposed method naturally incorporates the joint effects among features and flags the small sets of features that receive the most support from the joint mixture likelihood. The new method enjoys the all-class sure screening property and is computationally attractive in practice. Its promising performances were illustrated in extensive numerical examples.

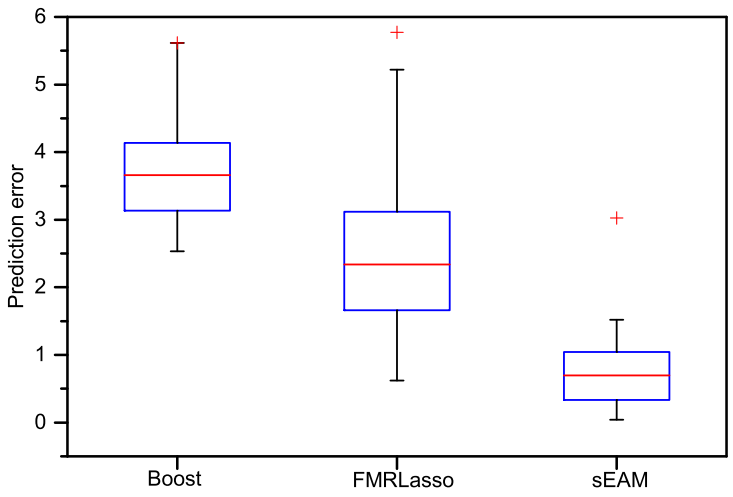


Figure 3.5: Boxplots of the prediction errors in the real data example. The methods in comparison are forward screening (Boost), L_1 penalized method (FMRLasso), and sparse expectation-approximation-maximization screening (sEAM).

The current work focuses on Gaussian mixture regressions. One may naturally extend the proposed method to mixture of generalized regression models. Also, in the proposed method, we assume that the number of classes K is pre-specified. Our simulation study seems to indicate that the proposed method is relatively robust against the over-specification of K. Therefore, in practice, when K is unknown, one may apply the method with a more liberal choice of K. A careful analysis of this situation is a promising topic for future research.

Chapter 4

Low Gradient-based Subsampling for Corrupted Massive Data

4.1 Introduction

Nowadays, massive data are encountered frequently in many scientific areas, which brings unprecedented opportunities and challenges to machine learning and statistics. Although many traditional methods are still valid, it is computationally infeasible to implement in typical personal computers due to a limited computation resource. To overcome the burden, researchers adopt generally two basic methodologies. One is the divide-and-conquer strategy and the other is the subsampling strategy. The divide-and-conquer strategy solves the problem by partitioning the massive data into several segments, runing a specific algorithm for each segment in parallel and then aggregating these individual outputs into a final estimate. Such a strategy makes the untractable problem into a tractable problem and has been generalized to dealing with the distributively stored data [33, 58], forming a so-called distributed machine learning and statistics [10, 36]. The subsampling strategy solves the problem by carrying out an intended calculation on a representative subset drawn from the full data [29, 41]. These methods draw useful informative observations to infer instead of using the whole data, which provides a way to tackle an untractable problem with a confined limited computation resource. In this dissertation, we focus on a study on subsampling strategy for corrupted massive data case.

The key idea of subsampling is to select the most informative observations so that the sampled data with small size convey most of the information contained in the full data. Subsampling normally adapts three different schemes. The first scheme is to draw

observations according to a deterministic pre-specific probability model, in which uniform subsampling is the most common way. It is most straightforward but difficult and even impossible to set an appropriate probability model. The second scheme of subsampling is drawing observations based on an informative score system, where each observation is given a different chance to be selected from a pre-fixed score system or according to an influence function. One representative informative score based subampling is the leverage score subsampling, which assigns nonuniform sampling probabilities according to the empirical statistical leverage scores of covariates [31, 32, 91]. The leverage score scheme performs fast but does not take it into account the relationship between covariates and the response variable and therefore, results in often insufficient utilization of data. The third scheme of subsampling is the task-driven subsampling, which connects the sampling probabilities with the prediction performance of a pilot estimate. The gradient-based subsampling is the typical example of such scheme, where the sampling probabilities are proportional to the gradient of loss associated with the pilot estimate [117]. It differs from the leverage-based sampling in that the sampling probabilities are allowed to hinge upon the covariates as well as the response. Those three subsampling schemes are all extensively studied and applied, and especially, the gradient-based subsampling has been accepted as one of the most efficient strategies to gain performance improvement. We observe, however, that almost all of the studies are only for the massive data without contamination or with relatively low level of noise. In all the cases, the large-gradient subsampling has been suggested in use, namely, the sampling probability for each datum is directly proportional to the gradient of loss.

Nevertheless, heavy corruption or high level of noise is unavoidably encountered in real-world application. The presence of noise and corruption in real-world massive data might be ineluctably caused by many factors such as accidental outliers [90] or adversarial data attacks [24]. All these factors raise a complicated corruption pattern, where the proportion of corruption could be large, the corruption level could be high and even exists on both covariates and response simultaneously. We thus must deal with those corrupted massive data cases. To tackle this issue, many works have been done for robust regression [47,55,56,98], compressed sensing [69,86] and high dimensional spare regression [23,28,73]. The subsampling in the corrupted massive data case has also attracted attention in recent years. Camponovo et al. [14] proposed a robust subsampling procedure for the linear regression model through elaboratively selecting subsampling size which is derived by a formula for the breakdown point of subsampling quantiles. Dalalyan and Thompson [28] considered the case where the response is contaminated by at most o adversarial outliers. They proved that the sampled L_1-penalized Huber's M-estimator based on n subsamples can approximate the M-estimator with full data at an estimation precision $O((s/n)^{\frac{1}{2}} + (o/n))$, where s is the sparsity of the problem. McWilliams et al. [84] studied the case of the

covariates corruption and assumed that the corruption can be described in an additive noise model. They suggested a subsampling algorithm called the influence weighted subsampling and they showed that the algorithm can yield an estimator approximating the oracle $\boldsymbol{\beta^*}$ at precision $O(C\sqrt{\frac{p\log p}{n}} + \epsilon\sqrt{p}||\boldsymbol{\beta^*}||)$, where ϵ is the proportion of the corruption. Gong et al. [45] also considered the linear regression in highly noisy case and they developed a robust gradient-based Markov subsampling(GMS) procedure. They proved that the GMS estimator can approximate the oracle β^* at an optimal estimation precision $O(\sqrt{\frac{p\log p}{n}})$.

In this dissertation, we consider a more general realistic corrupted data case, namely, that the collected data could be corrupted with very complicated noise (say, with larger proportion and higher level of corruption) on both covariates and response. We will develop a new gradient based subsampling procedure which can degrade influence of the heavy corruption in large extent. Based on an observation that the use of "large-gradient of loss" brings very often the danger of assigning higher sampling probabilities to the heavily noised observations, we propose a low-gradient subsampling procedure, that assigns the subsampling probability to each datum inversely proportional to the gradient value of loss at the datum. We show that "the low-gradient of loss" corresponds to the uncontaminated data in high probability and therefore, it may exclude the possibility of assigning high sampling probability to the heavy corrupted data. The proposed procedure is iterative in nature. The basic steps are as follows: we begin with a given crude pilot $\boldsymbol{\beta^{(0)}}$ and perform the low-gradient subsampling, yielding a first sampled dataset $\widetilde{\mathcal{D}}_0$. Then, we update the pilot by optimizing a memorized cost function on $\widetilde{\mathcal{D}}_0$, resulting the updated pilot $\boldsymbol{\beta^{(1)}}$, and we perform again the low-gradient subsampling based on $\boldsymbol{\beta^{(1)}}$. This process is repeated until a stopping criterion is met. We establish the error bounds of the sampled estimator (pilot) at each step from the oracle, which shows that the proposed low-gradient subsampling is statistical consistent with an estimation precision $O(C\sqrt{\frac{p\log p}{n}})$, where C depends on the pattern of corruption. We provide a series of numerical simulations and real data applications to demonstrate the effectiveness and efficiency of the proposed procedure. In particular, we show the much improved robustness of the new procedure as compared with the large-gradient based subsampling.

4.2 Methodology

4.2.1 Problem Set up

Let y be a response variable depending on p covariates $\boldsymbol{x} \in \mathbb{R}^p$. Given N independent observations $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^N$, where $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^{1\times p}, y_i \in \mathbb{R}$. Let $\boldsymbol{X} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_N})^T \in \mathbb{R}^{N\times p}$, $\boldsymbol{Y} = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$. We postulate the standard regression model between response y and covariates $\boldsymbol{x}$ as follows.

$$y = f(\boldsymbol{x}, \boldsymbol{\beta^*}) + \xi, \tag{4.1}$$

where $\boldsymbol{\xi} \in \mathbb{R}^N$ is a noise term independent of $\boldsymbol{x}$, f is a unknown regression function which has a linear or logistic form. Due to uncontrolled process of data acquisition, we can only obtain the corrupted observations $\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{Y}}$ instead of the idea observations $\boldsymbol{X}$ and $\boldsymbol{Y}$. We assume the corrupted and idea observations obey to the following model,

$$\begin{aligned} \widetilde{\boldsymbol{X}} &= \boldsymbol{X} + a\boldsymbol{U}\boldsymbol{W}, \\ \widetilde{\boldsymbol{Y}} &= \boldsymbol{Y} + b\boldsymbol{U}\boldsymbol{O}, \end{aligned} \tag{4.2}$$

where $\boldsymbol{U} = \mathrm{diag}(u_1, u_2, \ldots, u_N)$ and u_i is a Bernoulli random variable with parameter ϵ that takes values 1 when $\boldsymbol{x_i}$ is corrupted, and 0 otherwise. The rows of $\widetilde{\boldsymbol{X}}$ and $\widetilde{\boldsymbol{Y}}$ therefore are corrupted with probability ϵ and not corrupted with probability $1-\epsilon$. $\boldsymbol{W} \in \mathbb{R}^{N\times p}, \boldsymbol{O} \in \mathbb{R}^{N\times 1}$ are the matrices of corruption whose form defines the corruption pattern. Parameters a,b control the corruption level, where large a, b indicate that the observation is corrupted heavily and $a = 0$, $b = 0$ mean the observation is not corrupted.

The model described in (4.2) summarizes many realistic corrupted data cases. For example, when $a = 0$, $b \neq 0$, the model is the case called the adversarial corrupted response model discuessed in [8], [28] and [113]. When $a \neq 0$, $b = 0$, the model degenerates the corrupted covariate model studied in [84]. When $a = 0$, $b \neq 0$ and U is of sparsity, the model is the case discussed in [28] for high dimensional robust regression. We will discuss the more general case when $a \neq 0$, $b \neq 0$ and without sparsity assumption. Our aim is to develop an effective subsampling procedure for approximating the oracle $\boldsymbol{\beta^*}$ under such heavily corrupted data case.

4.2.2 A Low-Gradient Subsampling Procedure

In this subsection, we formulate a new gradient based subsampling procedure for the heavily corrupted massive data. Suppose $\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)\}_{i=1}^N$ is the corrupted data with $\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}$

being corrupted covariates, $\widetilde{y}_i$ being corrupted response, $\boldsymbol{\beta}^*$ is the p-dimensional oracle and $L(f(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \beta), \widetilde{y}_i)$ is the prediction discrepancy of f measured by the loss L. We define the average loss of an estimator $\boldsymbol{\beta}$ on a subset $\widetilde{\mathcal{D}}_s$ by

$$L_{\widetilde{\mathcal{D}}_s}(\boldsymbol{\beta}) = |\widetilde{\mathcal{D}}_s|^{-1} \sum_{\widetilde{\mathcal{D}}_s} L(f(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \boldsymbol{\beta}), \widetilde{y}_i)$$

and define the gradient matrix of $\boldsymbol{\beta}$ by

$$\boldsymbol{G}(\boldsymbol{\beta}) = \begin{bmatrix} \frac{\partial L(f(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \boldsymbol{\beta}), \widetilde{y}_1)}{\partial \beta_1} & \cdots & \frac{\partial L(f(\widetilde{\boldsymbol{x}}_{\boldsymbol{1}}, \boldsymbol{\beta}), \widetilde{y}_1)}{\partial \beta_p} \\ \vdots & & \vdots \\ \frac{\partial L(f(\widetilde{\boldsymbol{x}}_{\boldsymbol{N}}, \boldsymbol{\beta}), \widetilde{y}_N)}{\partial \beta_1} & \cdots & \frac{\partial L(f(\widetilde{\boldsymbol{x}}_{\boldsymbol{N}}, \boldsymbol{\beta}), \widetilde{y}_N)}{\partial \beta_p} \end{bmatrix} = (G_{ij})_{N \times p}. \tag{4.3}$$

Let g_i be the i-th row of $\boldsymbol{G}$. Different from the traditional large-gradient based subsampling, in which the subsampling probability is directly proportional to the gradient value, we propose to use the low-gradient subsampling, that is, the subsampling probability is inversely proportional to the gradient value. Thus, if $p_i \propto ||g_i||^{-1}$ is a probability corresponding to $(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)$ with the convention that $p_i = 0$ if $||g_i|| = 0$, we can assign p_i as the subsampling probability for $(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)$ in which the observations with lower gradient values are assigned larger p_i and the observations with larger gradient values assigned smaller p_i. We will show in the next subsection that such low-gradient subsampling does can express the effect of corruption of data, in particular, the values of gradient of $\boldsymbol{\beta}$ at $(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)$ can give an indication of corruption extent of the data. More precisely, the large-gradient corresponds to the highly corrupted data and the low-gradient corresponds to the uncontaminated data. This indication holds in probability and therefore, purely ignoring the use of large-gradient data might lose some information contained in the large-gradient data. As a compromise, we propose the following rule of assigning probability for subsampling,

$$\pi_i = \frac{\alpha}{N} + (1 - \alpha) p_i \tag{4.4}$$

where $\alpha < \frac{1}{2}$ is a non-negative parameter, which means that we mainly subsample data according to the low-gradient of loss while the large-gradient data are also sampled in small probabilities.

Note that assigning p_i in (4.4) depends on the estimator $\boldsymbol{\beta}$. A delicate choice of the estimator is the key to achieve a satisfying performance. In heavily corrupted data case, it is unlikely to specify a desirable initial pilot estimator for guidance of the assignment of probability. Thus, in this chapter, we propose to update the estimators recursively and use the timely updated estimator to guide the low-gradient subsampling. Given the $(t-1)$-th

estimator $\boldsymbol{\beta^{(t-1)}}$, we then can get the subsampled dataset $\widetilde{\mathcal{D}}_t$ of size n by the rule (4.4) from $\widetilde{\mathcal{D}}$. The t-th estimator $\boldsymbol{\beta^{(t)}}$ will be updated by minimizing the following memorized regularization function

$$\boldsymbol{\beta^{(t)}} = \underset{\boldsymbol{\beta}}{argmin}[L_{\widetilde{\mathcal{D}}_t}(\boldsymbol{\beta}) + g(\boldsymbol{\beta}, \boldsymbol{\beta^{(t-1)}})] \tag{4.5}$$

where $L_{\widetilde{\mathcal{D}}_t}(\boldsymbol{\beta})$ is the average loss defined on subset $\widetilde{\mathcal{D}}_t$ and

$$g(\boldsymbol{\beta}, \boldsymbol{\beta^{(t-1)}}) = \sum_{j=1}^{p} \nu_j (\beta_j - \beta_j^{(t-1)})^2 \tag{4.6}$$

is the memorized regularization function, within which

$$\nu_j = c \left[\frac{t}{log(1 + |\mu_j^G|)} \right]^d, \quad \mu_j^G = N^{-1} \sum_{i=1}^{N} G_{ij} \tag{4.7}$$

is the regularization parameter. This regularization function $g(\boldsymbol{\beta}, \boldsymbol{\beta^{(t-1)}})$ is a weighted measurement on discrepancy of the current estimator $\boldsymbol{\beta}$ from the previous estimator $\boldsymbol{\beta^{(t-1)}}$ in all covariates and, ν_j is a quantity direct proportional to step t and inversely proportional to μ_j^G, a quantity characterizing the prediction performance of $\beta_j^{(t-1)}$ (the smaller the better performance) on all observations. We take the regularization function as (4.6) is for the purpose that we tend to consist in the previous estimator $\boldsymbol{\beta^{(t-1)}}$ whenever the estimator already has good performance, otherwise we prefer to get a new estimator by drawing new observations.

By the procedure (4.5), we can recursively get better and better pilot estimators that then can be used to provide more and more reliable guidance for subsampling. We will show that such an iterative subsampling procedure is statistically consistent and empirically effective and efficient. The proposed subsampling procedure can be summarized as Algorithm 4.1 below.

Algorithm 4.1 LGS: A Low-Gradient Subsampling Procedure

Input: The observated dataset $\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}_i, \widetilde{y}_i)\}_{i=1}^N$, a pilot estimator $\boldsymbol{\beta^{(0)}}$, the subsampling size $n < N$, the hyperparameters $0 < \alpha < 1$, c, d and a stopping criterion.

Output: The output estimator $\widehat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta^*}$.

1: Initialize parameters $\boldsymbol{\nu} = \mathbf{0} \in \mathbb{R}^p$, compute $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{\beta^{(0)}})$ by (4.3).
2: **for** t=1 until the stopping criterion is met **do**
3: Compute probability π_i using G;
4: Draw a subset $\widetilde{\mathcal{D}}_t$ of size n from $\widetilde{\mathcal{D}}$ according to probability π_i;
5: Set $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta^{(t)}} = argmin_{\boldsymbol{\beta}} \left[L_{\widetilde{\mathcal{D}}_t}(\boldsymbol{\beta}) + \sum_{j=1}^p \nu_j (\beta_j - \beta_j^{(t-1)})^2 \right]$;
6: Update $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{\beta^{(t)}})$;
7: Compute $\mu_j^G = N^{-1} \sum_{i=1}^N G_{ij}$ for $j = 1, \ldots, p$;
8: Update

$$\nu_j = c \left[\frac{t}{log(1 + |\mu_j^G|)} \right]^d .$$

9: **end for**

Some remarks on Algorithm 4.1 are as follows.

Remark 3. *(i). The pilot $\boldsymbol{\beta^{(0)}}$ in Algorithm 4.1 can be simply taken as the estimator through n uniformly sampled observations from $\widetilde{\mathcal{D}}$. Certainly, other more delicated estimators can be applied. Due to the recursive feature of the suggested procedure, such choice of pilot is not so sensitive by our simulations. The subsampling size n is a crucial variable determining the computation complexity and prediction performance of the estimator. It can however be conveniently set according to the available computing resources in hand(say, the storage space limitation) and approximation precision requirement. The stopping criterion can be taken either as the maximum number T of update steps or as the tolerance ε between successive two estimators(say, $||\boldsymbol{\beta^{(t)}} - \boldsymbol{\beta^{(t-1)}}|| < \varepsilon$).*

(ii). There are many possible ways to implement the subsampling to draw subset $\widetilde{\mathcal{D}_t}$ in step 4. The sampling with replacement and Poisson sampling both can be simply used. Though Poisson sampling has slight advantages in computational load for massive data, there is no significant difference in prediction performance by our experiments. Therefore, we suggest the use of sampling with replacement.

(iii). Algorithm 4.1 can be assuredly terminated in finite steps since regularization parameter ν_j approaches to infinity as t tends to, which enforces the regularization term to dominate the cost function in step 5. If Algorithm 4.1 terminates within T steps, it can be easily seen that the computation complexity of the algorithm is around $O(TNp)$.

Such complexity can be further reduced if we adapt a K-Nearest-Neighbor technique(that is, we compute $\boldsymbol{g_i}$ confined to K neighboring observations of $\boldsymbol{x_i}$). However, for the sake of justifying feasibility of the proposed subsampling for the heavily corrupted data case, we consider the slightly higher complexity form (Algorithm 4.1) only in this chapter.

4.2.3 Why Low-gradient?

We provide a further explanation on rationality of using low-gradient when dealing with the heavily corrupted massive data case. We take the linear model as an example(that is, $f(\boldsymbol{x}, \boldsymbol{\beta}) = \boldsymbol{x}\boldsymbol{\beta}$).

Observe that the gradient of the i-th observation $(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)$ is given by

$$g_i = \frac{\partial L(f(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \beta), \widetilde{y}_i)}{\partial \boldsymbol{\beta}} = \widetilde{\boldsymbol{x}}_{\boldsymbol{i}}(\widetilde{y}_i - \widetilde{\boldsymbol{x}}_{\boldsymbol{i}}\boldsymbol{\beta}). \tag{4.8}$$

Thus, $||g_i|| = |\widetilde{y}_i - \widetilde{\boldsymbol{x}}_i\boldsymbol{\beta}|||\widetilde{x}_i||$. This shows that the magnitude of g_i is controlled by $|\widetilde{y}_i - \widetilde{\boldsymbol{x}}_i\boldsymbol{\beta}|$ and $||\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}||$. This first quantity $|\widetilde{y}_i - \widetilde{\boldsymbol{x}}_i\boldsymbol{\beta}|$ reflects the prediction deviation of $\boldsymbol{\beta}$ at $(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)$ and the second term $||\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}||$ contains the corruption information of datum. With reference to (4.1), we define the residual of oracle $\boldsymbol{\beta^*}$ at i-th observation as

$$\theta_i^* = \widetilde{y}_i - \widetilde{\boldsymbol{x}}_{\boldsymbol{i}}\boldsymbol{\beta^*} - \xi_i \tag{4.9}$$

which implies that $\theta_i^* = 0$ means the exact coincidence of the observation $(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)$ with the oracle relation. We can accept the observation $(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)$ as an uncontaminated datum in this case. In this criterion, a nonzero θ_i^* can be identified as a corrupted observation with high probability and the magnitude of θ_i^* is proportional to the contamination level. Thus, subsampling less contaminated observations need to select observations with smaller θ_i^*. Note that θ_i^* can be rewritten as

$$\theta_i^* = (\widetilde{y}_i - \widetilde{\boldsymbol{x}}_{\boldsymbol{i}}\boldsymbol{\beta}) + \widetilde{\boldsymbol{x}}_{\boldsymbol{i}}(\boldsymbol{\beta} - \boldsymbol{\beta^*}) - \xi_i, \tag{4.10}$$

that can be small only if $|\widetilde{y}_i - \widetilde{\boldsymbol{x}}_i\boldsymbol{\beta}|$ and $||\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}||$ both are small for a given estimator $\boldsymbol{\beta}$. Equivalently, this means that θ_i^* is small only if the gradient value of $\boldsymbol{\beta}$ is small. Thus, we conclude that "the low-gradient of loss" corresponds to the uncontaminated data in high probability and "the large-gradient of loss" corresponds to the corrupted data in high probability. This explains why "the low-gradient of loss" is suggested in use in this chapter (due to the pursuit of dismissing the effect of corruption).

A toy example to demonstrate such rationality of the using low-gradient is provided in Figures 4.1 and 4.2, where the proposed low-gradient subsampling is compared with

the uniform subsampling, the leverage subsampling and the large-gradient subsampling. Figures 4.1, 4.2 show clearly that the uniform subsampling, leverage subsampling, and large-gradient subsampling all perform poorly due to the mistreatment of the corrupted observations while the low-gradient subsampling performs well, supporting the explanations given in above.

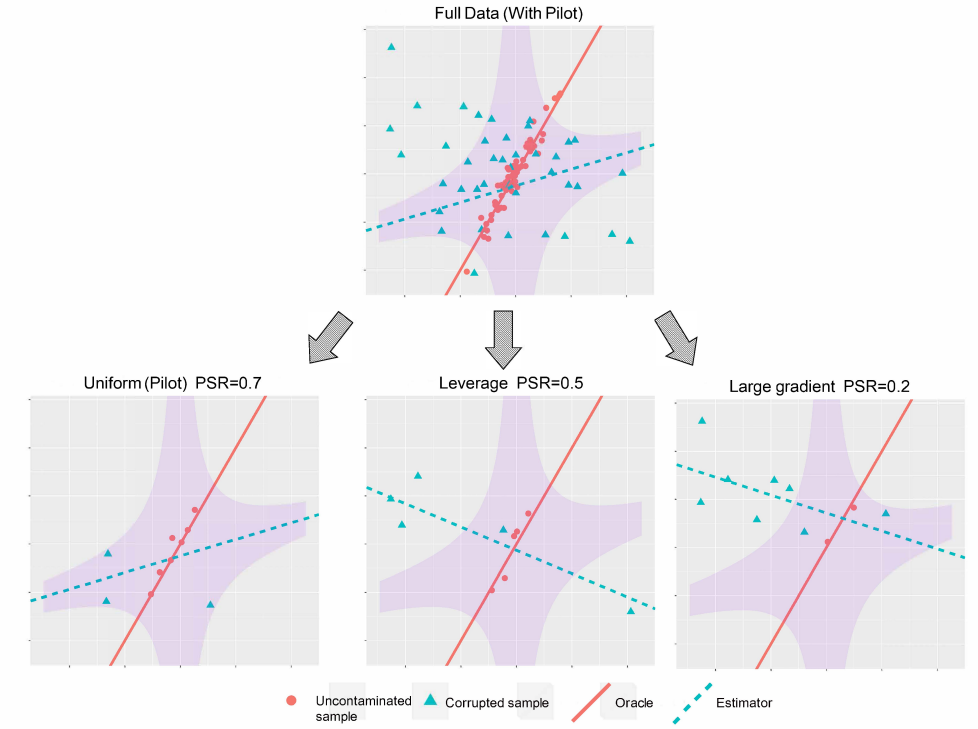


Figure 4.1: A toy example with $N = 100$, $p = 4$, $n = 10$, and $\epsilon = 0.4$. The subsampling results and corresponding estimators generated by using the uniform, the leverage, and the large-gradient subsampling are shown from left to right. All these three procedures have chosen large portion of highly corrupted data and small portion of the uncontaminated observations, corresponding to the low-gradient observations shown in the shaded purple region.

4.3 Theoretical Analysis

In this section, we provide a theoretical analysis on validity and effectiveness of the proposed low-gradient subsampling procedure. In particular, we derive an error bound of the sampled estimator from the oracle and show the statistical consistency of the procedure. We also

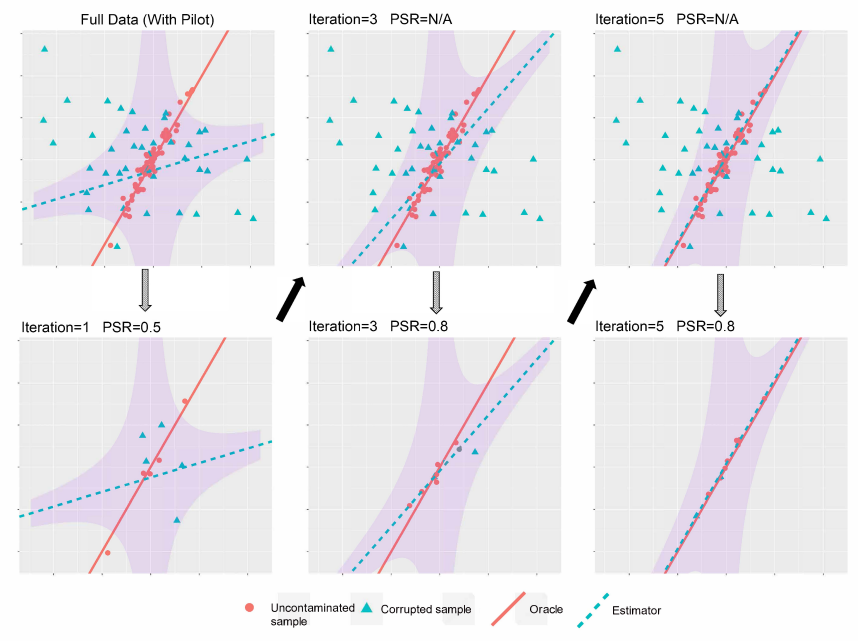


Figure 4.2: The resulted subsampling and estimators when the low-gradient subsampling is applied to the toy example. The results are shown when iteration steps t=1, 3 and 5(from left to right). It can be seen that the less corrupted observations, corresponding to the low-gradient values, are always sampled in high probability and as iteration step goes, more and more less corrupted observations are sampled, which leads to a very satisfying final estimator.

derive a lower bound on the maximum number of iteration steps necessary for the procedure to reach a given approximation accuracy.

The main results are stated in Theorems 1 and 2 below.

Recall that a random variable $\boldsymbol{x_i} \in \mathbb{R}^{1\times p}$ is said to be sub-Gaussian with variance proxy σ^2 if $E[\boldsymbol{x_i}] = 0$ and its moment generating function satisfies

$$E[\exp(s\boldsymbol{x_i})] \leq \exp(\frac{\sigma^2 s^2}{2}), \quad \forall s \in \mathbb{R}.$$

In the following, $\boldsymbol{x_i}, \boldsymbol{w_i}, \boldsymbol{o}, \boldsymbol{\xi}$ are the sub-Gaussian random variables with magnitudes $\frac{1}{N}\sigma_X^2$, $\frac{1}{N}\sigma_W^2, \frac{1}{N}\sigma_O^2$ and $\frac{1}{N}\sigma_\xi^2$, respectively, and all are independent of each other.

Theorem 4.1. *Let* $R = \max_{i=1,\ldots,N}\{||\boldsymbol{x_i}||^2\}$, $\Sigma_N = \frac{1}{N}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}$, $\sigma_N^2 = \frac{2}{N^2}\sum_{i=1}^N \frac{1}{\pi_i}||\widetilde{\boldsymbol{x}}_i||^4$ *and* T *is the number of update steps Algorithm 4.1 runs. If the subsampling size*

$$n > \frac{\sigma_N^2 \log p}{\delta^2[\min\{\frac{1}{2}\lambda_{\min}(\Sigma_N), \frac{1}{2}\lambda_{\min}(\Sigma_N)(\frac{1}{2}\lambda_{\min}(\Sigma_N) - 1)\} - \frac{R log p}{3N\delta}]}, \tag{4.11}$$

then the estimate $\widehat{\boldsymbol{\beta}}$ *for* $\boldsymbol{\beta^*}$ *generated by Algorithm 4.1 satisfies the following estimation in probability* $1-\delta$ *when* $T \to \infty$,

$$||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta^*}|| \lesssim C\sqrt{\frac{p\log p}{n}}. \tag{4.12}$$

where

$$C = \frac{(\sigma_X + a\epsilon\sigma_W)(\sigma_\xi + b\epsilon\sigma_O)}{a^2\sigma_W + 2a\sigma_W + 1} + \frac{a\epsilon\sigma_W(\sigma_X + a\epsilon\sigma_W)}{a^2\sigma_W + 2a\sigma_W + 1}||\beta^*||.$$

The proof of Theorem 4.1 will be given in Appendix.

Theorem 4.1 states that the proposed low-gradient subsampling is statistical consistent with an estimation precision $O(C\sqrt{\frac{p\log p}{n}})$, where C depends on the proportion of corruption ϵ, the magnitudes of corruption σ_W, σ_O, the magnitude of uncontaminated data σ_X and the corruption levels a, b. This precision approximately matches the optimal error bound $O(\sqrt{\frac{p\log p}{n}})$ known for the uncontaminated data case with a slightly larger constant C. Note that it is the first time to have such nearly state-of-art error bound for the sampled estimator in the corrupted massive data case.

Remark 4. *(i) The lower bound of subsampling size in (4.11) roughly scales with* $\frac{p^4 \log p}{N}$, *that is dominaned by* $1/N$ *in the "large-N-small-p" case and thus would not tend to infinity as* $N \to \infty$. *This shows that the subsampling size* n *is not needed to take too large in practice for achieving a satisfying approximation precision. In the "large-N-large-p" cases, a similar result can be derived by assuming certain sparsity on covariates.*

(ii). The error bound shown in (4.12) generalizes those previously established for the uncontam-inated data case [22, 45] (corresponding to $a = b = 0$ *and* $\epsilon = 0$ *in Theorem 4.1), the adversarial corrupted response data case [85](corresponding to* $a, \sigma_W = 0$, $b, \epsilon, \sigma_O \neq 0$*) and the corrupted covariates data case [84] (corresponding to* $b, \sigma_O = 0$, $a, \epsilon, \sigma_W \neq 0$*). We build the bound in the general corruption cases(*$a, b, \epsilon, \sigma_W, \sigma_O \neq 0$*). Except for some extra terms caused by additional assumptions, all derived bounds are of the order* $O(\sqrt{\frac{p \log p}{n}})$, *nearly optimal.*

(iii). It should be noted that the corruption patterns considered in Theorem 4.1 is rather general. In particular, the assumed data could have larger portion of contamination (larger ϵ*), allow higher level of corruption (larger* a, b*) and more complicated contamination forms(larger* σ_W, σ_O*). Theorem 4.1 supports that the proposed low-gradient subsampling can apply to those heavily corrupted data cases.*

Theorem 4.1 provides us a bound of approximation error under the assumption $T \to \infty$. Although it implies the feasibility of the proposed procedure for sufficient large iteration steps, it is computationally unaffordable in general, especially for the massive data case. In prospect of practical use, it is expected to have an error bound estimation for the sampled estimator at any iteration step, which then can be served as clarifying the minimum number of iteration steps necessary for the procedure to achieve a given approximation tolerance. Theorem 4.2 below provides such an estimation.

Theorem 4.2. *Let* $R = \max_{i=1,\ldots,N}\{||\boldsymbol{x_i}||^2\}$, $\Sigma_N = \frac{1}{N}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}$, $\sigma_N^2 = \frac{2}{N^2}\sum_{i=1}^N \frac{1}{\pi_i}||\widetilde{\boldsymbol{x}}_i||^4$ *and* T *is the number of update steps Algorithm 4.1 runs. If subsample size*

$$n > \frac{\sigma_N^2 \log p}{\delta^2[\min\{\frac{1}{2}\lambda_{\min}(\Sigma_N), \frac{1}{2}\lambda_{\min}(\Sigma_N)(\frac{1}{2}\lambda_{\min}(\Sigma_N) - 1)\} - \frac{R log p}{3N\delta}]},$$

and

$$T \gtrsim \log \frac{||\boldsymbol{\beta^{(0)}} - \boldsymbol{\beta^*}||}{\varepsilon} \doteq T_0, \tag{4.13}$$

then the T-step estimator $\boldsymbol{\beta^{(T)}}$ *for* $\boldsymbol{\beta^*}$ *satisfies the following estimation in probability* $1 - \delta$

$$||\widehat{\boldsymbol{\beta}}^{(\boldsymbol{T})} - \boldsymbol{\beta^*}|| \lesssim C\sqrt{\frac{p \log p}{n}} + \varepsilon.$$

where C is the same as in Theorem 1.

We present the proof of Theorem 4.2 in Appendix, also. Theorem 2 shows that T_0 in inequality (4.13) provides a lower bound of iteration steps for Algorithm 4.1 to obtain a sampled estimator meeting the approximation tolerance ε. It is seen from (4.13) that T_0 is controlled uniquely by the approximation precision of pilot $\boldsymbol{\beta}^{(\mathbf{0})}$, which reveals the importance of choice of the initial pilot. In particular, whenever the initial pilot has attained the precision $O(\sqrt{p\log p}n^{-\frac{1}{2}})$ and $\varepsilon = O(\sqrt{\frac{p\log p}{n}})$, one step of iteration is enough to achieve such ε-accurate estimator by (4.13). There are many one-step subsampling procedures that can yield the pilot approximately with such high precision, say, the subsampling procedures developed in [22,45,84,85] for the considered particular cases. Nevertheless, to our knowledge, there is no procedure available for generating such precise pilot for the general corruption case we consider in this work. Accordingly, we suggest the use of the uniform subsampling to define the initial pilot for simplicity.

4.4 Numerical Studies

In this section, we assess performance of the proposed low-gradient subsampling procedure (LGS) compared with some other tightly related procedures, including the uniform subsampling (UNIF), the gradient-based Markov subsampling [45] (GMS) in the linear regression case, the optimal subsampling method motivated from the A-optimality criterion [101] (OSMAC) in the logistic regression case and the maximum likelihood estimator (MLE) with full data. Among those competing procedures, the uniform subsampling is a well-known and very commonly applied one-step procedure, GMS is one of few low-gradient based subsampling procedures, OSMAC is a lately well-developed iterative subsampling procedure, and MLE is a traditionally useful estimator normally taken as a benchmark for comparison. Several large-gradient based subsampling methods [101, 117] have also be tested and it is found that most of them do not apply for the considered corrupted data cases and therefore, no comparison is made in this section.

The performance of the respective procedures is evaluated on a set of numerical simulations and real data applications. While the deviation of estimator $\widehat{\boldsymbol{\beta}}$ with maximum likelihood estimator $\boldsymbol{\beta}_{MLE}$ is often taken as the performance measure, we take the deviation(EE) of estimator $\widehat{\boldsymbol{\beta}}$ with the oracle $\boldsymbol{\beta}^*$, namely,

$$EE(\widehat{\beta}) = \frac{1}{M}\sum_{i=1}^{M}\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$$

as the measure, where M is the total number of trials. This is because that in the uncontaminated data case, $\boldsymbol{\beta}_{MLE}$ is very close to the oracle $\boldsymbol{\beta}^*$ but, in the corrupted data case, they are significant different. To make comparison fair, we enforce every procedure using a totally same number of samples R to get its estimator, and therefore, for one-step procedures, R is the subsampling size while for iterative procedures, $n = R/T$ is the subsampling size at each update step. Simulations are repeatedly conducted 500 times and the average performance over 500 times is taken as the final result. All numerical simulations are conducted on a personal computer using `R` v3.6.0 [88] with an Intel i7 processor with 12 logical processors and 16G of RAM. The real data application is conducted by using `R` v3.6.0 on Compute Canada's Cedar cluster with 32 logical processors and 200G of RAM.

4.4.1 Simulations

We present two simulations for comparison with artificially designed corruption datasets, one is for the linear regression and the other is for the logistic regression. Beside estimation error (EE), two other criteria, the positive screening rate (PSR) and the computational time (Times), are also evaluated and reported.

4.4.1.1 The Linear Regression Problem

In this simulation, the dataset $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^N$ is generated by the model

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{\xi},$$

where $\boldsymbol{X} \sim N_p(0, 3I_p)$, $\boldsymbol{\xi} \sim N(0, 5)$, $\boldsymbol{\beta^*}$ is fixed oracle. With reference to the corruption model (4.2), we consider the case of contamination level $a = b = 1$, the proportion of corruption $\epsilon = 0.2, 0.3, 0.4$ and the four types of corruption mode O1-O4 below. The contamination is characterized by W, O in (4.2) with each component independently and respectively generated according to

O1 (*mixed design*): $W_{ij} \sim 0.5N(-10, 3) + 0.5N(10, 3)$, $O_i \sim N(-10, 5)$.

O2 (*Negative contamination with centered design*): $W_{ij} \sim N(0, 10)$, $O_i \sim N(-10, 5)$.

O3 (*Background Noise*): $W_{ij} \sim N(0, 10)$, $O_i \sim N(0, 5)$.

O4 (*Sub-Gaussian Noise*): $W_{ij} \sim \text{Unif}(-10, 10)$, $O_i \sim \text{Unif}(0, 10)$.

In the simulation, $N = 10,000$, $p = 100$, $c = 0.01$, $d = 1$, $n = 500$ and $\boldsymbol{\beta^*} = (\beta_j^*)_{1\times p}$ with $\beta_j^* = 2$ or -2 in probability 0.5. The simulation results are shown in Tables 4.1-4.4.

Contamination Prop.		EE		Time (s)		PSR	
	R	$0.2N$	$0.4N$	$0.2N$	$0.4N$	$0.2N$	$0.4N$
$\epsilon = 0.2$	LGS	0.36 (0.07)	0.23 (0.08)	0.16	0.32	99.63%	99.76%
	UNIF	1.38 (0.04)	1.40 (0.03)	0.01	0.03	80.01%	80.00%
	GMS	0.85 (0.11)	0.91 (0.07)	0.33	0.59	94.00%	95.77%
	MLE	1.41		0.08		NA	
$\epsilon = 0.3$	LGS	0.48 (0.10)	0.45 (0.09)	0.16	0.30	99.12%	99.54%
	UNIF	1.54 (0.04)	1.56 (0.02)	0.01	0.03	70.01%	69.95%
	GMS	1.25 (0.05)	1.25 (0.07)	0.32	0.59	87.40%	87.89%
	MLE	2.04		0.07		NA	
$\epsilon = 0.4$	LGS	0.81 (0.10)	0.61 (0.08)	0.15	0.29	97.57%	98.44%
	UNIF	1.65 (0.02)	1.65 (0.01)	0.01	0.03	59.95%	60.10%
	GMS	1.45 (0.04)	1.47 (0.03)	0.30	0.61	80.04%	82.84%
	MLE	2.02		0.07		NA	

Table 4.1: Comparison results under O1 contamination mode.

Contamination Prop.		EE		Time (s)		PSR	
	R	$0.2N$	$0.4N$	$0.2N$	$0.4N$	$0.2N$	$0.4N$
$\epsilon = 0.2$	LGS	0.40 (0.07)	0.19 (0.05)	0.16	0.33	99.74%	99.77%
	UNIF	1.33 (0.04)	1.34 (0.02)	0.01	0.03	80.11%	80.15%
	GMS	0.82 (0.07)	0.82 (0.05)	0.32	0.60	93.81%	94.62%
	MLE	1.35		0.07		NA	
$\epsilon = 0.3$	LGS	0.59 (0.08)	0.40 (0.07)	0.16	0.31	98.98%	99.28%
	UNIF	1.55 (0.03)	1.55 (0.02)	0.01	0.03	70.40%	70.23%
	GMS	1.24 (0.05)	1.24 (0.03)	0.31	0.58	88.20%	88.97%
	MLE	2.07		0.08		NA	
$\epsilon = 0.4$	LGS	0.82 (0.08)	0.73 (0.08)	0.15	0.30	97.20%	97.78%
	UNIF	1.63 (0.02)	1.62 (0.01)	0.01	0.03	59.90%	59.94%
	GMS	1.41 (0.05)	1.43 (0.03)	0.31	0.60	81.43%	84.05%
	MLE	2.02		0.08		NA	

Table 4.2: Comparison results under O2 contamination mode.

Contamination Prop.		EE		Time (s)		PSR	
	R	$0.2N$	$0.4N$	$0.2N$	$0.4N$	$0.2N$	$0.4N$
$\epsilon = 0.2$	LGS	0.38 (0.08)	0.28 (0.06)	0.16	0.32	99.49%	99.65%
	UNIF	1.34 (0.04)	1.36 (0.02)	0.01	0.03	80.15%	80.07%
	GMS	0.81 (0.09)	0.83 (0.07)	0.31	0.63	93.75%	94.12%
	MLE	1.36		0.08		NA	
$\epsilon = 0.3$	LGS	0.56 (0.10)	0.45 (0.08)	0.17	0.30	99.08%	99.12%
	UNIF	1.52 (0.03)	1.52 (0.02)	0.01	0.03	69.74%	69.99%
	GMS	1.20 (0.05)	1.24 (0.05)	0.32	0.58	87.88%	88.37%
	MLE	2.05		0.07		NA	
$\epsilon = 0.4$	LGS	0.83 (0.10)	0.65 (0.09)	0.17	0.35	97.39%	97.96%
	UNIF	1.60 (0.02)	1.61 (0.01)	0.01	0.03	59.92%	60.17%
	GMS	1.41 (0.03)	1.44 (0.03)	0.33	0.67	81.14%	82.22%
	MLE	2.04		0.08		NA	

Table 4.3: Comparison results under O3 contamination mode.

Contamination Prop.		EE		Time (s)		PSR	
	R	$0.2N$	$0.4N$	$0.2N$	$0.4N$	$0.2N$	$0.4N$
$\epsilon = 0.2$	LGS	0.32 (0.09)	0.26 (0.06)	0.11	0.20	99.09%	99.18%
	UNIF	0.85 (0.04)	0.84 (0.02)	0.01	0.03	79.94%	79.93%
	GMS	0.44 (0.05)	0.44 (0.04)	0.20	0.39	93.59%	93.86%
	MLE	0.86		0.07		NA	
$\epsilon = 0.3$	LGS	0.45 (0.08)	0.37 (0.09)	0.16	0.30	97.86%	98.16%
	UNIF	1.04 (0.03)	1.05 (0.02)	0.01	0.03	70.19%	70.39%
	GMS	0.75 (0.05)	0.76 (0.07)	0.32	0.59	86.00%	87.46%
	MLE	2.33		0.07		NA	
$\epsilon = 0.4$	LGS	0.77 (0.12)	0.63 (0.11)	0.11	0.21	94.70%	95.78%
	UNIF	1.19 (0.03)	1.18 (0.02)	0.01	0.03	60.10%	60.08%
	GMS	0.96 (0.05)	0.96 (0.03)	0.20	0.40	79.29%	80.49%
	MLE	2.15		0.07		NA	

Table 4.4: Comparison results under O4 contamination mode.

EE		O1	O2	O3	O4
	LGS	0.36 (0.07)	0.40 (0.07)	0.38 (0.08)	0.32 (0.09)
$\epsilon = 0.2$	UNIF-Multi	1.42 (0.01)	1.40 (0.01)	1.36 (0.01)	0.89 (0.01)
	LGS	0.48 (0.10)	0.59 (0.08)	0.56 (0.10)	0.45 (0.08)
$\epsilon = 0.3$	UNIF-Multi	1.56 (0.01)	1.55 (0.01)	1.55 (0.01)	1.07 (0.01)
	LGS	0.81 (0.10)	0.82 (0.08)	0.83 (0.10)	0.77 (0.12)
$\epsilon = 0.4$	UNIF-Multi	1.65 (0.01)	1.63 (0.01)	1.65 (0.01)	1.22 (0.01)

Table 4.5: Comparison results of LGS and UNIF under same computation time(0.16s) for different contamination modes O1-O4.

Tables 4.1-4.4 demonstrate a consistent performance of all 4 procedures under different contamination modes. It can be seen from Tables 4.1-4.4 that the estimation error for LGS tend to decrease monotonically as the total subsampling size increases. The proposed procedure LGS outperforms the other 3 procedures UNIF, GMS and MLE in the sense of always achieving the lowest estimation error (EE) and the largest positive screening rate (PSR). It costs also less computation time than GMS, but it takes more time than UNIF and MLE. MLE and UNIF are the typical one-step estimation methods and the fast speed of implementation is their distinct advantage. Their precision can be obtained however is hardly improved, as shown in Tables 4.1-4.4 (EE). We have also re-implemented UNIF multiple rounds within the time constrains as listed in the Times LGS costs in the tables. We record the best over those multiple round results in Table 4.5. It is found that the best result can not still attain the same precision as LGS does. More precisely, as Table 4.5 indicates(for example, $\epsilon = 0.2$ under O1-O4), the best results of UNIF in the multiple rounds attain precision 1.42, 1.40, 1.36, 0.89 within the same time LGS costs, while LGS attain 0.36, 0.40, 0.38, 0.32. This shows that LGS is capable of attaining the highest precision estimation with a comparable or acceptable time for the considered corruption data cases.

Note that in the simulation, we have set $n = 500$, $R = 0.2N, 0.4N$ and thus LGS terminates at 4-th, 8-th update step. This shows that the LGS can generate a satisfying estimator with the lowest error in few steps, supporting the theoretical claims of Theorems 4.1 and 4.2.

4.4.1.2 The Logistic Regression Problem

In this simulation, the dataset $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^N$ is generated according to the model

$$\pi_i = \frac{\exp(\boldsymbol{x_i}\boldsymbol{\beta^*})}{1+\exp(\boldsymbol{x_i}\boldsymbol{\beta^*})}, \quad y_i \sim B(\pi_i),$$

where $\boldsymbol{X} \sim N_p(0, \Sigma)$. Corresponding to the corruption model (4.2), we consider the case of proportion of corruption $\epsilon = 0.2, 0.3, 0.4$ and two types of corruption mode B1-B2 below. The contamination is characterized by $W, \widetilde{Y}$ in (4.2) with each component independently and respectively generated according to

B1(*Random Contamination*): $W_{ij} \sim N(0,1), \quad \widetilde{y}_i \sim B(0.5)$,

B2(*Orthogonal Signal Contamination*):

$$W_{ij} \sim N(0,1), \quad \widetilde{y}_i = \begin{cases} 1, & \pi_i < 0.5, \\ 0, & \pi_i \geq 0.5. \end{cases}$$

This B2 mode contains a heavy corruption case where the response variable might have an opposite change.

In this simulation, we fix $N = 10,000$, $p = 10, 50$, $\Sigma = I_p$, $c = 0.01$, $d = 1$, $n = 500$ and $\boldsymbol{\beta^*} = (\beta_j^*)_{1\times p}$ with $\beta_j^* = 1$ or -1 in probability 0.5. The simulation results are reported in Tables 4.6-4.9.

Contamination Prop.		Estimation Error		Time (seconds)		PSR	
	$R =$	$0.2N$	$0.4N$	$0.2N$	$0.4N$	$0.2N$	$0.4N$
$\epsilon = 0.2$	LGS	0.32 (0.02)	0.30 (0.15)	0.022	0.035	92.41%	92.26%
	UNIF	0.59 (0.03)	0.59 (0.02)	0.004	0.005	79.99%	80.01%
	OSMAC	0.59 (0.04)	0.59 (0.04)	0.006	0.009	87.00%	91.50%
	MLE	0.93		0.011		NA	
$\epsilon = 0.3$	LGS	0.32 (0.02)	0.32 (0.02)	0.022	0.033	87.48%	88.66%
	UNIF	0.71 (0.02)	0.70 (0.01)	0.004	0.005	69.87%	70.12%
	OSMAC	0.71 (0.02)	0.70 (0.02)	0.005	0.009	82.70%	85.05%
	MLE	1.00		0.011		NA	
$\epsilon = 0.4$	LGS	0.35 (0.05)	0.34 (0.05)	0.023	0.034	77.77%	79.87%
	UNIF	0.78 (0.02)	0.78 (0.01)	0.004	0.005	59.93%	60.00%
	OSMAC	0.78 (0.03)	0.78 (0.03)	0.005	0.009	77.63%	78.82%
	MLE	1.01		0.011		NA	

Table 4.6: Comparison results under B1 contamination mode with $p = 10$.

Tables 4.6-4.9 demonstrate a similar performance with the linear case. That is, LGS can yield the estimator with the highest precision and the largest PSR within a comparable

Contamination Prop.		Estimation Error		Time (seconds)		PSR	
	$R =$	$0.2N$	$0.4N$	$0.2N$	$0.4N$	$0.2N$	$0.4N$
$\epsilon = 0.2$	LGS	0.67 (0.01)	0.66 (0.01)	0.097	0.156	88.52%	89.48%
	UNIF	0.76 (0.02)	0.77 (0.02)	0.018	0.035	79.81%	79.81%
	OSMAC	0.77 (0.02)	0.77 (0.03)	0.027	0.048	86.74%	88.07%
	MLE	0.96		0.075		NA	
$\epsilon = 0.3$	LGS	0.65 (0.05)	0.66 (0.06)	0.095	0.160	88.24%	88.70%
	UNIF	0.84 (0.01)	0.84 (0.01)	0.016	0.030	70.27%	70.07%
	OSMAC	0.85 (0.01)	0.84 (0.02)	0.026	0.049	81.62%	85.68%
	MLE	1.00		0.070		NA	
$\epsilon = 0.4$	LGS	0.64 (0.09)	0.66 (0.06)	0.093	0.163	89.52%	88.68%
	UNIF	0.88 (0.01)	0.89 (0.01)	0.014	0.028	59.85%	59.83%
	OSMAC	0.89 (0.01)	0.89 (0.01)	0.026	0.046	77.17%	83.48%
	MLE	1.00		0.068		NA	

Table 4.7: Comparison results under B1 contamination mode with $p = 50$.

Contamination Prop.		Estimation Error		Time (seconds)		PSR	
	$R =$	$0.2N$	$0.4N$	$0.2N$	$0.4N$	$0.2N$	$0.4N$
$\epsilon = 0.2$	LGS	0.27 (0.02)	0.27 (0.02)	0.026	0.039	93.68%	93.79%
	UNIF	0.58 (0.03)	0.59 (0.02)	0.002	0.004	80.02%	80.01%
	OSMAC	0.59 (0.03)	0.59 (0.03)	0.005	0.008	87.00%	93.48%
	MLE	0.93		0.008		NA	
$\epsilon = 0.3$	LGS	0.32 (0.02)	0.31 (0.02)	0.025	0.037	82.92%	84.12%
	UNIF	0.70 (0.01)	0.70 (0.01)	0.003	0.004	70.02%	70.15%
	OSMAC	0.69 (0.03)	0.69 (0.03)	0.005	0.009	82.10%	84.01%
	MLE	1.00		0.008		NA	
$\epsilon = 0.4$	LGS	0.36 (0.05)	0.35 (0.05)	0.026	0.035	78.19%	80.43%
	UNIF	0.78 (0.02)	0.78 (0.01)	0.002	0.003	59.94%	60.03%
	OSMAC	0.78 (0.02)	0.78 (0.02)	0.005	0.006	77.66%	79.80%
	MLE	1.00		0.006		NA	

Table 4.8: Comparison results under B2 contamination mode with $p = 10$.

Contamination Prop.		Estimation Error		Time (seconds)		PSR	
	$R =$	$0.2N$	$0.4N$	$0.2N$	$0.4N$	$0.2N$	$0.4N$
$\epsilon = 0.2$	LGS	0.66 (0.03)	0.66 (0.01)	0.094	0.152	88.36%	87.72%
	UNIF	0.78 (0.01)	0.78 (0.01)	0.018	0.027	79.08%	79.86%
	OSMAC	0.77 (0.02)	0.76 (0.02)	0.025	0.043	86.28%	86.11%
	MLE	0.96		0.058		NA	
$\epsilon = 0.3$	LGS	0.69 (0.03)	0.67 (0.04)	0.090	0.159	92.16%	92.80%
	UNIF	0.84 (0.01)	0.84 (0.01)	0.017	0.027	70.16%	69.76%
	OSMAC	0.84 (0.01)	0.85 (0.01)	0.022	0.040	81.57%	84.96%
	MLE	1.00		0.050		NA	
$\epsilon = 0.4$	LGS	0.70 (0.04)	0.63 (0.06)	0.089	0.161	89.84%	91.80%
	UNIF	0.89 (0.01)	0.89 (0.01)	0.015	0.028	59.75%	60.34%
	OSMAC	0.88 (0.01)	0.89 (0.01)	0.021	0.036	77.15%	82.43%
	MLE	0.99		0.045		NA	

Table 4.9: Comparison results under B2 contamination mode with $p = 50$.

or slightly higher computation time. Note that OSMAC here is a typical large-gradient based subsampling method that has not yielded a high precision estimator(say, in the high corruption cases $\epsilon = 0.3, 0.4$.). This supports the observation that the large-gradient does not apply to the highly corrupted data case, nevertheless, the low-gradient subsampling works well.

The simulations conducted above uniformly illustrate that the proposed low-gradient subsampling does work well for highly contaminated massive data, and it can yield satisfying estimators with lower approximation error within finite steps. All the simulation results support the correctness of Theorems 4.1 and 4.2, and also support the feasibility and effectiveness of the proposed procedure.

4.4.2 Applications

In this subsection, we further access the proposed low-gradient subsampling by applying it to two application problems, the housing price prediction (HPP) problem and the supersymmetric particle classification (SPC) problem. The HPP problem is asked to predict the price of a house based on its properties like the lot size, living area and etc.. The SPC problem is to distinguish between a background process and a process where new supersymmetric particles are produced, utilizing the 18 kinematic features in the dataset. Statistically, the HPP is a regression problem and the SPC is a classification problem. Due

to the data collection process that unavoidably involves lie statement for privacy protection purpose and unavoidable effect of measurement environment in modern accelerators, the collected data are corrupted and unreliable. These two problems thus have been used as benchmark applications for robust regression and classification study.

In these applications, neither the oracle $\boldsymbol{\beta}^*$ nor the corruption pattern is known. To make it possible to compare the respective procedures, we, instead of using the estimation error $||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*||$, use the prediction error of estimator tested in a testing set to compare the performance. Since the datasets we are dealing with are the contaminated ones, directly taking observations from the dataset is inappropriate as the testing set, which in principle should be consisted purely of the uncontaminated observations. To dismiss this difficulty, we apply in each case a well-developed machine learning algorithm to approximately identify the uncontaminated data and constitute the testing set. The algorithms are all run 200 times and the average performance is taken as the final results to be reported.

4.4.2.1 The House Price Prediction Problem

The HPP problem we apply is the `SaratogaHouses` dataset [87] (N = 1728, p=15) that contains 15 covariates including the lot size, living area, percent of college-educated residents in the neighbourhood, number of bathrooms and etc.. Among the 15 covariates, 6 of them(type of heating system, type of sewer system, whether the house has central air and etc.) are discrete variables. To make it applicable for the gradient based procedures to work, that is developed only for continuous variables, we dismiss those 6 discrete covariates for application. For such HPP problem, the procedure that combines a S-estimation, a M-step, a D-estimation and another M-step (SMDM) is a well-known state-of-art algorithm for defining a robust regression estimator [62], included in package `robustbase` [78]. We apply SMDM to help us for selection of the testing set.

We write the considered dataset $\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}, \widetilde{y}_i)\}_{i=1}^N$. For implementation of the involved procedures, we need to define the training set $\widetilde{\mathcal{D}}_{train}$ and the testing set $\widetilde{\mathcal{D}}_{test}$, the former has to be selected from $\widetilde{\mathcal{D}}$ consisted of the contaminated observations while the later consisted of the uncontaminated data. With the help of SMDM, we firstly sort out all the contaminated observations from $\widetilde{\mathcal{D}}$[1] , denoted by $\widetilde{\mathcal{D}}_{cont}$, and then select a random subset $\widetilde{\mathcal{D}}_{sub}$ from $\widetilde{\mathcal{D}} \setminus \widetilde{\mathcal{D}}_{cont}$. The training set is defined as $\widetilde{\mathcal{D}}_{train} = \widetilde{\mathcal{D}}_{sub} \cup \widetilde{\mathcal{D}}_{cont}$, with variable size of $\widetilde{\mathcal{D}}_{sub}$, and the rest subset of $\widetilde{\mathcal{D}} \setminus \widetilde{\mathcal{D}}_{cont}$ being defined as the testing set $\widetilde{\mathcal{D}}_{test}$.

[1]Suppose SMDM defines its estimator $\widehat{\boldsymbol{\beta}}$ through minimizing the cost function $l(\cdot)$. $\psi(\cdot)$ is the linear-quadratic-quadratic (lqq) function which is proportional to the derivative of $l(\cdot)$. An observation x_i is identified as a corrupted one if $\psi(e_i)/e_i < 0.5$, where $e_i = y_i - x_i\widehat{\boldsymbol{\beta}}$.

The number of $\widetilde{\mathcal{D}}_{sub}$ controls the proportion of corruption $\epsilon = |\widetilde{\mathcal{D}}_{cont}|/|\widetilde{\mathcal{D}}_{sub} \cup \widetilde{\mathcal{D}}_{cont}|$ and therefore, varying the number of $\widetilde{\mathcal{D}}_{sub}$ provides a way to observe the effect of corruption level for the procedures in comparison.

We apply the low-gradient subsampling LGS to $\widetilde{\mathcal{D}}$ together with other 4 procedures, UNIF, GMS, MLE and SMDM on the full training set, in comparison. The total number of samples is fixed at $R = 0.1N$, the hyperparameters $\alpha = 0.1$, $c = 1, d = 0.01$, the contamination proportion ϵ varies from $0.091, 0.124, 0.166, 0.250$ to 0.334, and the other related tuning parameters are set as the references [45, 62]. Computation time is not reported as all procedures run near instantaneously.

The application results are listed in Table 4.10 and Figure 4.3. It can be seen that the prediction errors of all procedures monotonically decreases as the proportion of corruption ϵ decreases. They all have a similar prediction error when the data contains less corruption and however, LGS can achieve consistently greater predictive accuracy in all higher corruption cases as compared with UNIF, GMS, MLE and SMDM. As Table 4.10 shows, LGS has significantly improved prediction precision particularly in the heavily corrupted cases(say, ϵ=0.250, 0.334).

	Prediction Error (PE)				
$\lvert\mathcal{D}_{sub}\rvert =$	$0.5N$	$0.35N$	$0.25N$	$0.15N$	$0.1N$
ϵ	0.091	0.124	0.166	0.250	0.334
LGS	0.27 (0.02)	0.29 (0.04)	0.31 (0.06)	0.38 (0.09)	0.49 (0.14)
UNIF	0.29 (0.03)	0.33 (0.06)	0.38 (0.08)	0.51 (0.14)	0.68 (0.15)
GMS	0.27 (0.04)	0.29 (0.07)	0.32 (0.08)	0.40 (0.09)	0.54 (0.14)
MLE	0.27 (0.01)	0.30 (0.01)	0.38 (0.03)	0.46 (0.04)	0.62 (0.06)
SMDM	0.24 (0.01)	0.25 (0.01)	0.27 (0.01)	0.38 (0.03)	0.55 (0.07)

Table 4.10: Performance comparison of LGS as compared with UNIF, GMS, MLE and SMDM when applied to the house price prediction problem.

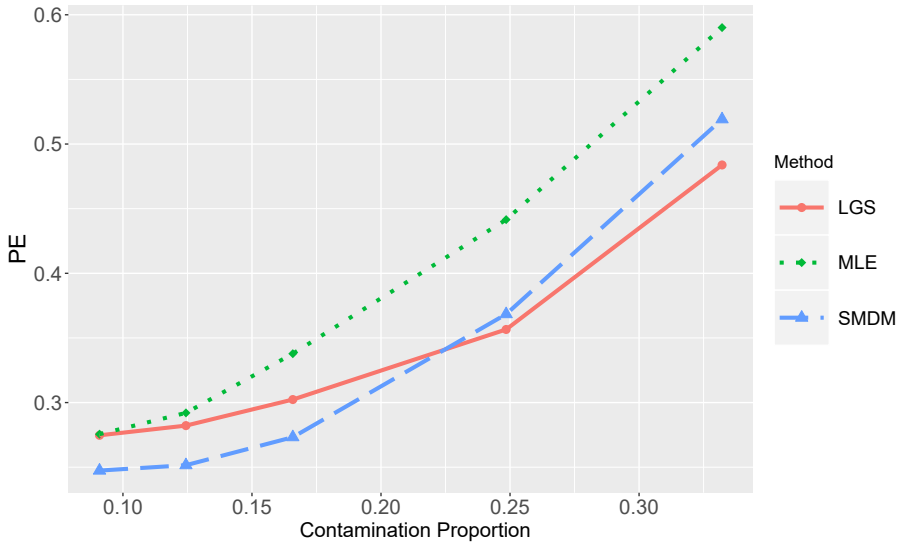


Figure 4.3: Predictive error of LGS, MLE and SMDM as corruption proportion varies.

4.4.2.2 The Supersymmetric Particle Classification Problem

We apply LGS together with UNIF and OSMAC, a well-developed classification algorithm [101], to the supersymmetric particle classification problem. The dataset collected for this problem is the `supersymmetric` dataset [4] where $N = 5,000,000$ and $p = 18$. The problem is to classify the supersymmetric particles according to 8 low level measured kinematic features and 10 high level computed kinematic features. The dataset is massive ($N = 5,000,000$) but lightly corrupted. To well observe the impact of corruption, we therefore add a variable contamination on the data according to the corruption model (4.2) with $a = 1$, ϵ ranging from 0.05 to 0.15 and

$$W_{ij} \sim U(\min(\boldsymbol{X}_{\cdot j}), \max(\boldsymbol{X}_{\cdot j})), \text{ for all } j = 1, 2, \ldots, 18, \quad y_i \sim B(0.5).$$

In this application, following [4] and [101], the first $4,500,000$ observations are used as a training set, and the last $500,000$ as the testing set. Comparisons are made in terms of classification precision for the obtained estimator $\widehat{\beta}$ to correctly classify the testing set. The comparison results are shown in Figure 4.4, where the total subsampling size $R = 2,000$ for all methods, and LGS is run for 5 iterations at $\alpha = 0.1$, $c = 1$ and $d = 0.01$.

We can see from Figure 4.4 that the classification precision of all the procedures decrease as the contamination level increases and all the methods are comparable in precision

when no noise is artificially added. When the data have corruption, however, LGS always outperforms at all noise levels.

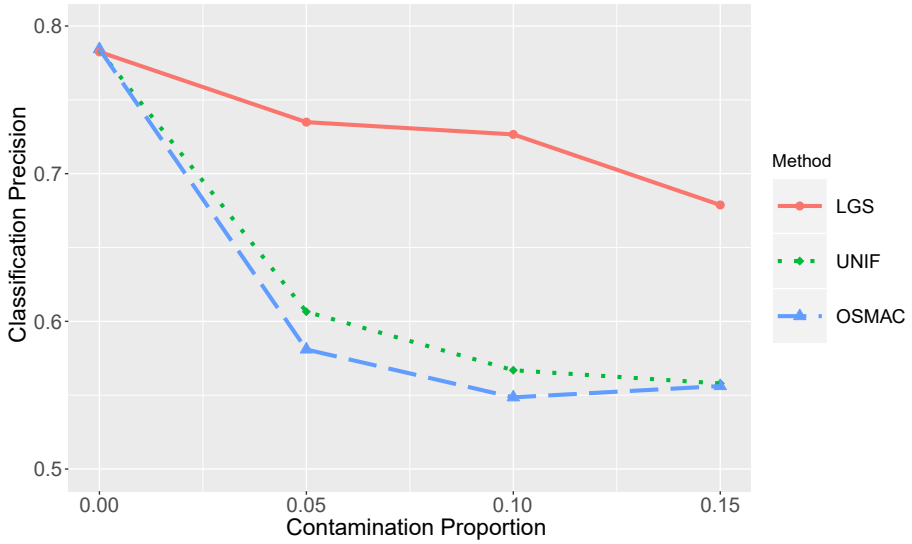


Figure 4.4: Classification error of LGS, UNIF and OSMAC as corruption proportion varies.

The applications in subsections 4.4.2.1 and 4.4.2.2 all show that the proposed LGS does can deal with heavily corrupted massive data as compared with other well-developed estimation procedures. It can yield more precise prediction estimator and can be practically applied to complicated massive data even contamination pattern unknown.

4.5 Conclusion

In this chapter, we have proposed a low-gradient subsampling procedure for dealing with corrupted massive data. Through uncovering the special role of low-gradient observations that corresponds to the less corrupted data, it is suggested the use of low-gradient subsampling in opposite to the traditional approach for prediction performance subsampling that use the large-gradient. In cooperation with a memorized regularization iteration, the proposed procedure recursively draws low-gradient observations and updates sampling probabilities according to the new prediction performance. It is shown that the procedure is statistically consistent with achieving a nearly optimal estimation precision $O(C\sqrt{\frac{p\log p}{n}})$, the same as that known for uncontaminated data case. It is also shown that procedure

can terminate in a finite step. The simulations and applications all support the validity of the established theories, and demonstrate its effectiveness and efficiency. In particular, the outperformance of the suggested procedure over other commonly used subsampling methods is demonstrated when the data contain heavy contamination or corruption.

There are several ways to improve the current research, say, establishing similar procedures and theories for more general nonlinear model and more complicated corruption pattern; reducing the computation complexity for gradient based subsampling(especially avoiding the use and computation of gradient matrix) and increasing the estimation precision by using more delicate iterative update scheme. All these problems are under our current research.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this dissertation, I developed new statistical methods for feature screening and subsampling with massive data, which can be distributively stored, heterogeneous, and highly corrupted.

Specifically, for feature screening with distributed data, I proposed a new distributed feature screening approach DHS, which is computationally convenient and is therefore suitable for huge data. The new algorithm enables the master machine to aggregate all of the results from the local machines and broadcast the aggregated result as guidance to each local machine. In this way, the new algorithm leverages much more information in the dataset through more communication between local machines to screen features. The proposed algorithm outperforms the existing marginal-information-based distributed methods by taking into consideration joint effects between features and rich communications between computers. We also established convergence of the algorithm and its sure screening property within a finite number of iterations. Empirical studies demonstrate that compared with existing screening methods, DHS has a remarkable capacity of retaining relevant features under all designs that we consider. In contrast to simple aggregating methods, DHS enables more communication among all machines to leverage more information contained in the whole dataset. These advantages render the DHS promising for feature screening in the large-N-large-p regime.

For the research of feature screening for ultrahigh dimensional heterogenous data, I proposed a novel method to screen irrelevant covariates for ultrahigh dimensional Gaussian mixture regression in Chapter 3. Specifically, the new method is built upon a sparsity-

restricted expectation-approximation-maximization procedure, which simultaneously detects varying sets of key features for multiple classes (mixture components). I also proved that sEAM procedure is non-decreased after every iteration, which extend the existing convergence theory in the general analysis of EM algorithm. The new method enjoys the all-class sure screening property and is computationally attractive in practice. Extensive experiments on synthetic and real datasets demonstrated that the new approach is superior to those of existing methods in accuracy, retaining capacity, positive selection rate and computational time. All these results showed that the new approach consistently outperforms existing methods.

For the research of analysis in corrupted massive data, I formulated a model for complete corrupted massive datasets where both covariates and response are contaminated. This model is more realistic with respect to real data which is subject to many different sources of measurement noise or heterogeneity in the datasets. To cope with the corrupted massive data, I proposed a low-gradient subsampling procedure. Through uncovering the special role of low-gradient observations that corresponds to the less corrupted data, it is suggested the use of low-gradient subsampling in opposite to the traditional approach for prediction performance subsampling that use the large-gradient. Specifically, I obtained a crude estimator based on a simple pilot selection and constructed subsets by selecting samples of small gradient to achieve a reliable improvement in each step. I showed that under mild conditions, the procedure is statistically consistent with achieving a nearly optimal estimation precision, the same as that known for uncontaminated data case. The simulations and applications all support the validity of the established theories, and demonstrated its effectiveness and efficiency. In particular, the outperformance of the suggested procedure over other commonly used subsampling methods is demonstrated when the data contain heavy contamination or corruption.

5.2 Future Research

In the remainder of this chapter, we present several possible directions for future research.

5.2.1 Large-N-large-p Data

Unbalanced data segments.

Currently, all analysis and theoretical results are based on equal size data segments. However, it isn't common in the real world, especially if the entire dataset is stored at different

locations and even the data segments are not I.I.D. To tackle this situation, I will consider a weight which proportion to data quality and sample size of sparse regression model or mixture of noise distributions.

Other model setups such as GLM.

I will extend the current distributed consensus sparse regression model to generalized linear model. The convergence and sure screening theories including finite-step theory will also be updated by rewriting the closed-form ADMM updating procedure.

More real-world large-N-large-p datasets.

So far, I considered several datasets generated synthetically and one real-world dataset, which is not enough for exploring the effectiveness and performance of the procedure. I will try to utilize DHS procedure in more real-world data in the application such as gene analysis of COVID-19 or shipment safety inspection of customs auto-scan.

5.2.2 Ultra-high Dimensional Heterogeneous Data

Estimating the number of components K

In the proposed method, I assumed that the number of classes K is pre-specified. The simulation study seems to indicate that the new method is relatively robust against the over-specification of K. Therefore, in practice, when K is unknown, one may consider a liberal model with an upper bound of K and carry out the proposed method with a sparsity restriction $\|\beta\| \leq s$ on the overall $\beta = (\beta_1, \ldots, \beta_K)$. With a proper s, the extended method may help to simultaneously screen out both irrelevant features and redundant classes. A careful analysis of this situation is a promising topic for future research.

Mixture of generalized regression models

The current work focuses on Gaussian mixture regressions. One may naturally extend the proposed method to mixture of generalized regression models.

More real data applications.

Due to technology limitations and resource constraints, the public data contain enormous amounts of heterogeneity that have not been fully exploited so far. It is essential and necessary to verify sEAM procedure on different applications. We will seek more suitable datasets and implement our procedure on them.

5.2.3 Corrupted Massive Data

A new adaptive probabilities criterion on observations.

Currently, I set probabilities according to gradient matrix, which is a preliminary exploration of sampling approach. Besides, the calculation of gradient matrix on the whole sample is time consuming. In the future, I will attempt to propose a new adaptive probabilities setting criterion to reduce computing time and improve performance of procedure.

Spare structure in LGS procedure.

Due to the computational burden, the dimension of datasets is usually low in our experiments. I will try to extend the proposed LGS procedure for high-dimensional corrupted massive datasets. Specifically, the LGS procedure will be extended by existing sparse regularization such LASSO, SCAD, $L_{1/2}$ or MCP.

Hyperparameters in LGS procedure.

LGS procedure contains many hyperparameters such as convex parameter α, regularization parameters c and d or even subsampling size n. All these hyperparameters make current LGS procedure difficult to set and run in practice. It would be better if a deep exploration of hyperparameters is implemented. I will trial statistical criterions such as AIC, BIC to gain a deep understanding on setting these hyperparameters.

5.3 Final Remarks

When the aim of a dissertation is to further the frontiers of a research discipline, it is arguable as to whether or not a dissertation can be completed at all. There is always one more question to answer, one more topic to discuss, or one more example to give. Thus there is always one more thing to write or to append to the dissertation at any given stage. However, although the dissertation cannot be completed, it must be concluded, which is where I should stop now. Although I have covered serval applications in writing this dissertation, there are still many areas of research to explore. I am sure that what I have learnt from writing this dissertation will serve me well in my future endeavors.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Bruce Alberts. *Molecular biology of the cell*. WW Norton & Company, 2017.

[3] Sivaraman Balakrishnan, Martin J Wainwright, and Bin Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120, 2017.

[4] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physis with deep learning. *Nature Communications*, 5(4308), 2014. PMID: 30078922.

[5] Jeffrey D Banfield and Adrian E Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

[6] Emre Barut, Jianqing Fan, and Anneleen Verhasselt. Conditional sure independence screening. *Journal of the American Statistical Association*, 111(515):1266–1277, 2016.

[7] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[8] Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. *arXiv preprint arXiv:1506.02428*, 2015.

[9] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[10] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[11] Russell A Boyles. On the convergence of the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 45(1):47–50, 1983.

[12] Sébastien Bubeck. Theory of convex optimization for machine learning. *arXiv preprint arXiv:1405.4980*, 15, 2014.

[13] Peter Bühlmann, Markus Kalisch, and Lukas Meier. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1:255–278, 2014.

[14] Lorenzo Camponovo, Olivier Scaillet, and Fabio Trojani. Robust subsampling. *Journal of Econometrics*, 167(1):197–210, 2012.

[15] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The annals of Statistics*, 35(6):2313–2351, 2007.

[16] Jiahua Chen and Zehua Chen. Extended bic for small-n-large-p sparse glm. *Statistica Sinica*, pages 555–574, 2012.

[17] Jiahua Chen, Xianming Tan, and Runchu Zhang. Inference for normal mixtures in mean and variance. *Statistica Sinica*, pages 443–465, 2008.

[18] Laming Chen and Yuantao Gu. The convergence guarantees of a non-convex approach for sparse recovery. *IEEE Transactions on Signal Processing*, 62(15):3754–3767, 2014.

[19] Mengjie Chen, Chao Gao, and Zhao Ren. A general decision theory for hubers ϵ -contamination model. *Electronic Journal of Statistics*, 10(2):3752–3774, 2016.

[20] Xiaolin Chen, Xiaojing Chen, and Hong Wang. Robust feature screening for ultra-high dimensional right censored data via distance correlation. *Computational Statistics & Data Analysis*, 119:118–138, 2018.

[21] Xueying Chen and Min-ge Xie. A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, pages 1655–1684, 2014.

[22] Yudong Chen and Constantine Caramanis. Orthogonal matching pursuit with noisy and missing data: Low and high dimensional results. *arXiv preprint arXiv:1206.0823*, 2012.

[23] Yudong Chen and Constantine Caramanis. Noisy and missing data regression: Distribution-oblivious support recovery. In *International Conference on Machine Learning*, pages 383–391. PMLR, 2013.

[24] Yudong Chen, Constantine Caramanis, and Shie Mannor. Robust sparse regression under adversarial corruption. In *International Conference on Machine Learning*, pages 774–782. PMLR, 2013.

[25] Qianshun Cheng, HaiYing Wang, and Min Yang. Information-based optimal subdata selection for big data logistic regression. *Journal of Statistical Planning and Inference*, 209:112–122, 2020.

[26] Hengjian Cui, Runze Li, and Wei Zhong. Model-free feature screening for ultrahigh dimensional discriminant analysis. *Journal of the American Statistical Association*, 110(510):630–641, 2015.

[27] Arnak Dalalyan and Yin Chen. Fused sparsity and robust estimation for linear models with unknown variance. *Advances in Neural Information Processing Systems*, 25, 2012.

[28] Arnak S Dalalyan and Philip Thompson. Outlier-robust estimation of a sparse linear model using l_1-penalized huber's m-estimator. *arXiv preprint arXiv:1904.06288*, 2019.

[29] Michał Dereziński, Manfred K Warmuth, and Daniel Hsu. Leveraged volume sampling for linear regression. *arXiv preprint arXiv:1802.06749*, 2018.

[30] Emilie Devijver. Finite mixture regression: a sparse variable selection by model selection for clustering. *Electronic journal of statistics*, 9(2):2642–2674, 2015.

[31] Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

[32] Petros Drineas, Michael W Mahoney, Shan Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische mathematik*, 117(2):219–249, 2011.

[33] Rui Duan, Mary Regina Boland, Zixuan Liu, Yue Liu, Howard H Chang, Hua Xu, Haitao Chu, Christopher H Schmid, Christopher B Forrest, John H Holmes, et al. Learning from electronic health records across multiple sites: A communication-efficient and privacy-preserving distributed algorithm. *Journal of the American Medical Informatics Association*, 27(3):376–385, 2020.

[34] Yonina C Eldar and Gitta Kutyniok. *Compressed sensing: theory and applications*. Cambridge university press, 2012.

[35] Jianqing Fan, Yang Feng, and Rui Song. Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association*, 106(494):544–557, 2011.

[36] Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National science review*, 1(2):293–314, 2014.

[37] Jianqing Fan, Runze Li, Cun-Hui Zhang, and Hui Zou. *Statistical foundations of data science*. Chapman and Hall/CRC, 2020.

[38] Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008.

[39] Jianqing Fan, Richard Samworth, and Yichao Wu. Ultrahigh dimensional feature selection: beyond the linear model. *The Journal of Machine Learning Research*, 10:2013–2038, 2009.

[40] Jianqing Fan and Rui Song. Sure independence screening in generalized linear models with np-dimensionality. *The Annals of Statistics*, 38(6):3567–3604, 2010.

[41] William Fithian and Trevor Hastie. Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of statistics*, 42(5):1693, 2014.

[42] Charles Fox. *Data science for transport: a self-study guide with computer exercises*. Springer, 2018.

[43] Jerome H Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. springer open, 2017.

[44] Roland Glowinski. *Numerical methods for nonlinear variational problems*. Springer series in computational physics. Springer- Verlag, 1984.

[45] Tieliang Gong, Quanhan Xi, and Chen Xu. Robust gradient-based markov subsampling. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 4004–4011. AAAI Press, 2020.

[46] Anders Gorst-Rasmussen and Thomas Scheike. Independent screening for single-index hazard rate models with ultrahigh dimensional features. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(2):217–245, 2013.

[47] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 196. John Wiley & Sons, 2011.

[48] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC, New York, 2015.

[49] BS He, Hai Yang, and SL Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and applications*, 106(2):337–356, 2000.

[50] Martin Hilbert and Priscila López. The worlds technological capacity to store, communicate, and compute information. *science*, 332(6025):60–65, 2011.

[51] Nhat Ho, Chiao-Yu Yang, and Michael I Jordan. Convergence rates for gaussian mixtures of experts. *Journal of Machine Learning Research*, 23(323):1–81, 2022.

[52] Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.

[53] Cheng Huang and Xiaoming Huo. A distributed one-step estimator. *Mathematical Programming*, 174(1):41–76, 2019.

[54] Danyang Huang, Runze Li, and Hansheng Wang. Feature screening for ultrahigh dimensional categorical data with applications. *Journal of Business & Economic Statistics*, 32(2):237–244, 2014.

[55] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[56] Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

[57] Bo Jiang, Shiqian Ma, and Shuzhong Zhang. Alternating direction method of multipliers for real and complex polynomial optimization models. *Optimization*, 63(6):883–898, 2014.

[58] Michael I Jordan, Jason D Lee, and Yun Yang. Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 2018.

[59] Jian Kang, Hyokyoung G Hong, and Yi Li. Partition-based ultrahigh-dimensional variable screening. *Biometrika*, 104(4):785–800, 2017.

[60] Abbas Khalili and Jiahua Chen. Variable selection in finite mixture of regression models. *Journal of the american Statistical association*, 102(479):1025–1038, 2007.

[61] Abbas Khalili, Jiahua Chen, and Shili Lin. Feature selection in finite mixture of sparse normal linear models in high-dimensional feature space. *Biostatistics*, 12(1):156–172, 2011.

[62] Manuel Koller and Werner A. Stahel. Sharpening wald-type inference in robust regression for small samples. *Computational Statistics & Data Analysis*, 55(8):2504C2515, 2011.

[63] Jeongyeol Kwon and Constantine Caramanis. Em converges for a mixture of many linear regressions. In *International Conference on Artificial Intelligence and Statistics*, pages 1727–1736. PMLR, 2020.

[64] Andrea E Lamont, Jeroen K Vermunt, and M Lee Van Horn. Regression mixture models: Does modeling the covariance between independent variables and latent classes improve the results? *Multivariate behavioral research*, 51(1):35–52, 2016.

[65] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*, volume 23. Springer Science & Business Media, 1991.

[66] Guoyin Li and Ting Kei Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.

[67] Runze Li, Dennis KJ Lin, and Bing Li. Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry*, 29(5):399–409, 2013.

[68] Runze Li, Wei Zhong, and Liping Zhu. Feature screening via distance correlation learning. *Journal of the American Statistical Association*, 107(499):1129–1139, 2012.

[69] Xiaodong Li. Compressed sensing and matrix completion with constant proportion of corruptions. *Constructive Approximation*, 37(1):73–99, 2013.

[70] Xingxiang Li, Runze Li, Zhiming Xia, and Chen Xu. Distributed feature screening via componentwise debiasing. *Journal of machine learning research*, 21, 2020.

[71] Yifeng Li, Chih-Yu Chen, and Wyeth W Wasserman. Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016.

[72] Bo Liu, Ying Wei, Yu Zhang, and Qiang Yang. Deep neural networks for high dimension, low sample size data. In *IJCAI*, pages 2287–2293, 2017.

[73] Liu Liu, Yanyao Shen, Tianyang Li, and Constantine Caramanis. High dimensional robust sparse regression. In *International Conference on Artificial Intelligence and Statistics*, pages 411–421. PMLR, 2020.

[74] Qinghua Liu, Xinyue Shen, and Yuantao Gu. Linearized ADMM for nonconvex nonsmooth optimization with convergence analysis. *IEEE Access*, 7:76131–76144, 2019.

[75] Mike Loukides. *What is data science?* " O'Reilly Media, Inc.", 2011.

[76] Ping Ma, Xinlian Zhang, Xin Xing, Jingyi Ma, and Michael Mahoney. Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1026–1035. PMLR, 2020.

[77] Lester Mackey, Ameet Talwalkar, and Michael I Jordan. Divide-and-conquer matrix factorization. *Advances in neural information processing systems*, 24, 2011.

[78] Martin Maechler, Peter Rousseeuw, Christophe Croux, Valentin Todorov, Andreas Ruckstuhl, Matias Salibian-Barrera, Tobias Verbeke, Manuel Koller, Eduardo L. T. Conceicao, and Maria Anna di Palma. *robustbase: Basic Robust Statistics*, 2020. R package version 0.93-6.

[79] Sindri Magnússon, Pradeep Chathuranga Weeraddana, Michael G Rabbat, and Carlo Fischione. On the convergence of alternating direction lagrangian methods for nonconvex structured optimization problems. *IEEE Transactions on Control of Network Systems*, 3(3):296–309, 2015.

[80] Michael W Mahoney et al. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

[81] Ryan McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 456–464, 2010.

[82] Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[83] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6:355–378, 2019.

[84] Brian McWilliams, Gabriel Krummenacher, Mario Lucic, and Joachim M Buhmann. Fast and robust least squares estimation in corrupted linear models. In *Advances in Neural Information Processing Systems*, pages 415–423, 2014.

[85] Nam H Nguyen and Trac D Tran. Robust lasso with missing and grossly corrupted observations. *IEEE transactions on information theory*, 59(4):2036–2058, 2012.

[86] Nam H Nguyen and Trac D Tran. Exact recoverability from dense corrupted observations via $l-1$-minimization. *IEEE transactions on information theory*, 59(4):2017–2035, 2013.

[87] Randall Pruim, Daniel Kaplan, and Nicholas Horton. *mosaicData: Project MOSAIC Data Sets*, 2020. R package version 0.18.0.

[88] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[89] Randall Reese, Xiaotian Dai, and Guifang Fu. Strong sure screening of ultra-high dimensional categorical data. *arXiv preprint arXiv:1801.03539*, 2018.

[90] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

[91] Alessandro Rudi, Daniele Calandriello, Luigi Carratino, and Lorenzo Rosasco. On fast leverage score sampling and optimal learning. *arXiv preprint arXiv:1810.13258*, 2018.

[92] Xinyue Shen, Laming Chen, Yuantao Gu, and Hing-Cheung So. Square-root lasso with nonconvex regularization: An admm approach. *IEEE Signal Processing Letters*, 23(7):934–938, 2016.

[93] Nicolas Städler, Peter Bühlmann, and Sara Van De Geer. l1-penalization for mixture regression models. *Test*, 19(2):209–256, 2010.

[94] Ye Tian and Yang Feng. Rase: A variable screening framework via random subspace ensembles. *Journal of the American Statistical Association*, pages 1–12, 2021.

[95] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[96] Joel A Tropp. User-friendly tools for random matrices: An introduction. Technical report, California Inst of Tech Pasadena Div of Engineering and Applied Science, 2012.

[97] Paul Tseng. An analysis of the em algorithm and entropy-like proximal point methods. *Mathematics of Operations Research*, 29(1):27–44, 2004.

[98] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531, 1975.

[99] HaiYing Wang. More efficient estimation for logistic regression with optimal subsamples. *Journal of machine learning research*, 20, 2019.

[100] HaiYing Wang, Min Yang, and John Stufken. Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association*, 114(525):393–405, 2019.

[101] HaiYing Wang, Rong Zhu, and Ping Ma. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844, 2018. PMID: 30078922.

[102] Hansheng Wang. Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*, 104(488):1512–1524, 2009.

[103] Jialei Wang, Mladen Kolar, Nathan Srebro, and Tong Zhang. Efficient distributed learning with sparsity. In *International Conference on Machine Learning*, pages 3636–3645. PMLR, 2017.

[104] Xiangyu Wang and Chenlei Leng. High dimensional ordinary least squares projection for screening variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(3):589–611, 2016.

[105] Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

[106] Zhaoran Wang, Quanquan Gu, Yang Ning, and Han Liu. High dimensional em algorithm: Statistical optimization and asymptotic normality. *Advances in neural information processing systems*, 28:2512, 2015.

[107] Yuanshan Wu and Guosheng Yin. Conditional quantile screening in ultrahigh-dimensional heterogeneous data. *Biometrika*, 102(1):65–76, 2015.

[108] Chen Xu and Jiahua Chen. The sparse mle for ultrahigh-dimensional feature screening. *Journal of the American Statistical Association*, 109(507):1257–1269, 2014.

[109] Chen Xu, Yongquan Zhang, Runze Li, and Xindong Wu. On the feasibility of distributed kernel regression for big data. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):3041–3052, 2016.

[110] Yangyang Xu, Wotao Yin, Zaiwen Wen, and Yin Zhang. An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China*, 7(2):365–384, 2012.

[111] Xinyang Yi and Constantine Caramanis. Regularized em algorithms: A unified framework and statistical guarantees. *Advances in Neural Information Processing Systems*, 28, 2015.

[112] Jian Zhang. Screening and clustering of sparse regressions with finite non-gaussian mixtures. *Biometrics*, 73(2):540–550, 2017.

[113] Xuchao Zhang, Shuo Lei, Liang Zhao, Arnold P Boedihardjo, and Chang-Tien Lu. Robust regression via heuristic corruption thresholding and its adaptive estimation variation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(3):1–22, 2019.

[114] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

[115] Qian Zhao, Deyu Meng, and Zongben Xu. A recursive divide-and-conquer approach for sparse principal component analysis. *arXiv preprint arXiv:1211.7219*, 2012.

[116] Li-Ping Zhu, Lexin Li, Runze Li, and Li-Xing Zhu. Model-free feature screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*, 106(496):1464–1475, 2011.

[117] Rong Zhu. Gradient-based sampling: An adaptive importance sampling for least-squares. *Advances in neural information processing systems*, 29:406–414, 2016.

Appendix A

Supplementary Information for Chapter 2

The Appendix contains the proofs of the lemma and theorems in the Chapter 2.

A.1 Proof of Theorem 2.1

To prove Theorem 2.1, we need to introduce some notation first. Let $\boldsymbol{X} \in \mathbb{R}^{N\times p}$ and $\boldsymbol{Y} \in \mathbb{R}^N$ are the overall feature matrix and the overall response vector. Recall that, for $l = 1, \ldots, m$, $\boldsymbol{X}_l \in \mathbb{R}^{n\times p}$ and $\boldsymbol{Y}_l \in \mathbb{R}^n$ are the observed feature matrix and the response vector in the lth segment, respectively. Denote $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^\intercal, \boldsymbol{\alpha}_2^\intercal, \ldots, \boldsymbol{\alpha}_m^\intercal)^\intercal$ as the master auxiliary vector of m sets of $\boldsymbol{\alpha}_l$, $l = 1, \ldots, m$. Similarly, we use $\boldsymbol{\rho} = (\boldsymbol{\rho}_1^\intercal, \boldsymbol{\rho}_2^\intercal, \ldots, \boldsymbol{\rho}_m^\intercal)^\intercal$ to represent the master of all Lagrange multiplier vectors $\boldsymbol{\rho}_l$'s. Let $h_l(\boldsymbol{\alpha}_l) = \frac{1}{2}\|\boldsymbol{Y}_l - \boldsymbol{X}_l\boldsymbol{\alpha}_l\|_2^2$, $h(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{l=1}^m \|\boldsymbol{Y}_l - \boldsymbol{X}_l\boldsymbol{\alpha}_l\|_2^2$ and $\lambda = \max_{l=1,\ldots,m} \lambda_{\max}(\boldsymbol{X}_l^\intercal \boldsymbol{X}_l)$. Let $\overline{\boldsymbol{\beta}}^{(t+1)}$ denote the β-update without the hard thresholding function $H_k(\cdot)$, i.e., $\overline{\boldsymbol{\beta}}^{(t+1)} = \arg\min_{\boldsymbol{\beta}} \sum_{l=1}^m \langle \boldsymbol{\rho}_l^{(t)}, \boldsymbol{\alpha}_l^{(t)} - \boldsymbol{\beta}\rangle + \frac{\eta}{2}\sum_{l=1}^m \|\boldsymbol{\alpha}_l^{(t)} - \boldsymbol{\beta}\|^2$. Before proving Theorem 2.1, we need to show the following lemma first.

Lemma A.1. *In the DHS procedure in Algorithm 2.1, for any $1 \le l \le m$ and every t, the following results hold:*

(a). $\boldsymbol{\rho}_l^{(t)} = -\nabla h_l(\boldsymbol{\alpha}_l^{(t)})$.

(b). $\|\nabla h_l(\boldsymbol{\alpha}_l^{(t+1)}) - \nabla h_l(\boldsymbol{\alpha}_l^{(t)})\|_2^2 \le \lambda^2 \|\boldsymbol{\alpha}_l^{(t+1)} - \boldsymbol{\alpha}_l^{(t)}\|_2^2$.

(c). $\frac{\eta}{2}\|\boldsymbol{\alpha}^{(t)}-\overline{\boldsymbol{\beta}}^{(t+1)}\|_2^2-\frac{\eta}{2}\|\boldsymbol{\alpha}^{(t)}-\boldsymbol{\beta}^{(t+1)}\|_2^2=\sum_{l=1}^m\langle\boldsymbol{\rho}_l^{(t)},\boldsymbol{\beta}^{(t+1)}-\bar{\boldsymbol{\beta}}^{(t+1)}\rangle-\frac{m\eta}{2}\|\boldsymbol{\beta}^{(t+1)}-\bar{\boldsymbol{\beta}}^{(t+1)}\|_2^2.$

Proof of Lemma A.1. Part (a) follows directly from the optimality of $\boldsymbol{\alpha}_l^{(t)}$. For any $l = 1,\ldots,m$, we have

$$0=\nabla h_l(\boldsymbol{\alpha}_l^{(t)})+\boldsymbol{\rho}_l^{(t-1)}+\eta(\boldsymbol{\alpha}_l^{(t)}-\boldsymbol{\beta}^{(t)})=\nabla h_l(\boldsymbol{\alpha}_l^{(t)})+\boldsymbol{\rho}_l^{(t)},$$

and therefore,

$$\boldsymbol{\rho}_l^{(t)}=-\nabla h_l(\boldsymbol{\alpha}_l^{(t)}).$$

For part (b), by the definition of $h_l(\boldsymbol{\alpha}_l)$, it follows that

$$\nabla h_l(\boldsymbol{\alpha}_l)=-\boldsymbol{X}_l^\top(\boldsymbol{Y}_l-\boldsymbol{X}_l\boldsymbol{\alpha}_l). \tag{A.1}$$

Therefore,

$$\|\nabla h_l(\boldsymbol{\alpha}_l^{(t+1)})-\nabla h_l(\boldsymbol{\alpha}_l^{(t)})\|_2^2=||\boldsymbol{X}_l^\top\boldsymbol{X}_l(\boldsymbol{\alpha}_l^{(t+1)}-\boldsymbol{\alpha}_l^{(t)})||_2^2\le\lambda^2||\boldsymbol{\alpha}_l^{(t)}-\boldsymbol{\alpha}_l^{(t+1)}||_2^2$$

The last inequality follows from the fact that $\|\boldsymbol{B}\boldsymbol{z}\|^2\le\lambda_{\max}(\boldsymbol{B}^\intercal\boldsymbol{B})\|\boldsymbol{z}\|^2$ for any $\boldsymbol{z}$.

For part (c), since

$$\overline{\boldsymbol{\beta}}^{(t+1)}=\arg\min_{\boldsymbol{\beta}}\sum_{l=1}^m\langle\boldsymbol{\rho}_l^{(t)},\boldsymbol{\alpha}_l^{(t)}-\boldsymbol{\beta}\rangle+\frac{\eta}{2}\sum_{l=1}^m\|\boldsymbol{\alpha}_l^{(t)}-\boldsymbol{\beta}\|^2,$$

we have

$$\sum_{l=1}^m\boldsymbol{\rho}_l^{(t)}+\eta\sum_{l=1}^m\left(\boldsymbol{\alpha}_l^{(t)}-\overline{\boldsymbol{\beta}}^{(t+1)}\right)=0.$$

Thus, it follows that

$$\sum_{l=1}^m\boldsymbol{\alpha}_l^{(t)}=\frac{1}{\eta}\sum_{l=1}^m\boldsymbol{\rho}_l^{(t)}+m\overline{\boldsymbol{\beta}}^{(t+1)}.$$

Therefore, the left-hand side of part (c) equals to

$$\begin{aligned}\text{LHS}&=\frac{\eta}{2}\sum_{l=1}^m\langle\boldsymbol{\beta}^{(t+1)}-\overline{\boldsymbol{\beta}}^{(t+1)},2\boldsymbol{\alpha}_l^{(t)}-\overline{\boldsymbol{\beta}}^{(t+1)}-\boldsymbol{\beta}^{(t+1)}\rangle\\&=\frac{\eta}{2}\langle\boldsymbol{\beta}^{(t+1)}-\overline{\boldsymbol{\beta}}^{(t+1)},\frac{2}{\eta}\sum_{l=1}^m\boldsymbol{\rho}_l^{(t)}+m(\overline{\boldsymbol{\beta}}^{(t+1)}-\boldsymbol{\beta}^{(t+1)})\rangle\\&=\sum_{l=1}^m\langle\boldsymbol{\rho}_l^{(t)},\boldsymbol{\beta}^{(t+1)}-\bar{\boldsymbol{\beta}}^{(t+1)}\rangle-\frac{m\eta}{2}\|\boldsymbol{\beta}^{(t+1)}-\bar{\boldsymbol{\beta}}^{(t+1)}\|_2^2.\end{aligned} \tag{A.2}$$

□

Proof of Theorem 2.1. Note that the augmented Lagrangian function $L(\alpha, \beta, \rho)$ is strong convex with respect to each α_l and $\overline{\boldsymbol{\beta}}$ with the uniform convexity coefficient η/m. This entails that

$$L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t)}, \overline{\boldsymbol{\beta}}^{(t+1)}, \boldsymbol{\rho}^{(t)}) \geq \frac{\eta}{m}\|\overline{\boldsymbol{\beta}}^{(t+1)} - \boldsymbol{\beta}^{(t)}\|_2^2, \tag{A.3}$$

$$L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) \geq \frac{\eta}{m}\|\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}\|_2^2. \tag{A.4}$$

By the definition of $\overline{\boldsymbol{\beta}}^{(t+1)}$, $L(\alpha, \beta, \rho)$ and part (c) of Lemma A.1, one obtains

$$\begin{aligned}
&L_\eta(\boldsymbol{\alpha}^{(t)}, \overline{\boldsymbol{\beta}}^{(t+1)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) \\
&= \sum_{l=1}^m \langle \boldsymbol{\rho}_l^{(t)}, \boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)} \rangle + \frac{\eta}{2}\|\boldsymbol{\alpha}^{(t)} - \overline{\boldsymbol{\beta}}^{(t+1)}\|_2^2 - \frac{\eta}{2}\|\boldsymbol{\alpha}^{(t)} - \boldsymbol{\beta}^{(t+1)}\|_2^2 \\
&= \sum_{l=1}^m 2\langle \boldsymbol{\rho}_l^{(t)} - \boldsymbol{\rho}_l^{(t+1)}, \boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)} \rangle + \sum_{l=1}^m 2\langle \boldsymbol{\rho}_l^{(t+1)}, \boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)} \rangle - \frac{m\eta}{2}\|\boldsymbol{\beta}^{(t+1)} - \bar{\boldsymbol{\beta}}^{(t)}\|_2^2 \\
&= \sum_{l=1}^m 2\langle \nabla h_l(\boldsymbol{\alpha}_l^{(t+1)}) - \nabla h_l(\boldsymbol{\alpha}_l^{(t)}), \boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)} \rangle + \sum_{l=1}^m 2\langle \boldsymbol{\rho}_l^{(t+1)}, \boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)} \rangle \\
&\quad - \frac{m\eta}{2}\|\boldsymbol{\beta}^{(t+1)} - \bar{\boldsymbol{\beta}}^{(t)}\|_2^2 \\
&\geq -\sum_{l=1}^m \left(\|\nabla h_l(\boldsymbol{\alpha}_l^{(t+1)}) - \nabla h_l(\boldsymbol{\alpha}_l^{(t)})\|_2^2 + \|\boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)}\|_2^2 \right) + \sum_{l=1}^m 2\langle \boldsymbol{\rho}_l^{(t+1)}, \boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)} \rangle \\
&\quad - \frac{m\eta}{2}\|\boldsymbol{\beta}^{(t+1)} - \bar{\boldsymbol{\beta}}^{(t)}\|_2^2 \\
&\geq -\lambda^2\|\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}\|_2^2 - m\|\boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)}\|_2^2 + \sum_{l=1}^m 2\langle \boldsymbol{\rho}_l^{(t+1)}, \boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)} \rangle \\
&\quad - \frac{m\eta}{2}\|\boldsymbol{\beta}^{(t+1)} - \bar{\boldsymbol{\beta}}^{(t)}\|_2^2,
\end{aligned} \tag{A.5}$$

where the second inequality holds from part (b) of Lemma A.1.

Since $\boldsymbol{\beta}^{(t+1)}$ minimizes the augmented Lagrangian function, we have

$$\begin{aligned}
&\sum_{l=1}^m \{h_l(\overline{\boldsymbol{\beta}}^{(t+1)}) + \langle \boldsymbol{\rho}_l^{(t+1)}, \overline{\boldsymbol{\beta}}^{(t+1)} - \boldsymbol{\beta}^{(t+1)} \rangle + \frac{\eta}{2}\|\overline{\boldsymbol{\beta}}^{(t+1)} - \boldsymbol{\beta}^{(t+1)}\|_2^2\} = L_\eta(\overline{\boldsymbol{\beta}}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t+1)}) \\
&\qquad \leq L_\eta(\overline{\boldsymbol{\beta}}^{(t+1)}, \overline{\boldsymbol{\beta}}^{(t+1)}, \boldsymbol{\rho}^{(t+1)}) = \sum_{l=1}^m h_l(\overline{\boldsymbol{\beta}}^{(t+1)}).
\end{aligned} \tag{A.6}$$

Inequality (A.6) implies

$$\sum_{l=1}^{m}\langle \boldsymbol{\rho}_l^{(t+1)}, \boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)}\rangle \geq \frac{m\eta}{2}\|\boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)}\|_2^2. \tag{A.7}$$

Combining (A.5), (A.7) and by the optimality property of $\overline{\boldsymbol{\beta}}^{(t+1)}$, we obtain

$$\begin{aligned} &L_\eta(\boldsymbol{\alpha}^{(t)}, \overline{\boldsymbol{\beta}}^{(t+1)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) \\ \geq\ & -\lambda^2\|\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}\|_2^2 + \frac{m(\eta-2)}{2}\|\boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)}\|_2^2. \end{aligned} \tag{A.8}$$

By Lemma A.1, a direct calculation from the $\boldsymbol{\rho}_l$-update of (2.3) leads to

$$\begin{aligned} &L_\eta(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t+1)}) \\ =\ & -\frac{1}{\eta}\|\boldsymbol{\rho}^{(t+1)} - \boldsymbol{\rho}^{(t)}\|_2^2 = -\frac{1}{\eta}\sum_{l=1}^{m}\|\nabla h_l(\boldsymbol{\alpha}_l^{(t)}) - \nabla h_l(\boldsymbol{\alpha}_l^{(t+1)})\|_2^2 \\ \geq\ & -\frac{\lambda^2}{\eta}\|\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}\|_2^2. \end{aligned} \tag{A.9}$$

Combining (A.3), (A.4), (A.8) and (A.9), one obtains

$$\begin{aligned} &L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t+1)}) \\ =\ & L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t)}, \overline{\boldsymbol{\beta}}^{(t+1)}, \boldsymbol{\rho}^{(t)}) + L_\eta(\boldsymbol{\alpha}^{(t)}, \overline{\boldsymbol{\beta}}^{(t+1)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) \\ &+ L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) \\ &+ L_\eta(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t)}) - L_\eta(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\rho}^{(t+1)}) \\ \geq\ & \frac{\eta}{m}\|\overline{\boldsymbol{\beta}}^{(t+1)} - \boldsymbol{\beta}^{(t)}\|_2^2 + \left(\frac{\eta}{m} - \lambda^2 - \frac{\lambda^2}{\eta}\right)\|\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}\|_2^2 + \frac{m(\eta-2)}{2}\|\boldsymbol{\beta}^{(t+1)} - \overline{\boldsymbol{\beta}}^{(t+1)}\|_2^2. \end{aligned}$$

As the step parameter η satisfies $\eta > \max\{2, (1/2)m\lambda^2 + \sqrt{(1/4)m^2\lambda^4 + m\lambda^2}\}$ one has

$$\frac{\eta}{m} > 0, \quad \frac{\eta}{m} - \lambda^2 - \frac{\lambda^2}{\eta} \geq 0, \quad \text{and} \quad \frac{m(\eta-2)}{2} \geq 0.$$

Therefore, $L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)})$ is monotonically decreasing. Also, $L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)})$ is lower bounded because $(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)})$ belongs to the feasible set. Together with the fact that it is monotonically decreasing, we conclude that $L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)})$ is convergent.

By the convergence of $L_\eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\rho}^{(t)})$ and the non-expansive property of the hard thresholding function, $\|\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}\|_2$, $\|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}\|_2$ and $\|\boldsymbol{\rho}^{(t+1)} - \boldsymbol{\rho}^{(t)}\|_2$ converge to zero as $t \to \infty$. □

Before presenting the detailed proof of Theorem 2.2, we first show one technical lemma as follows.

Lemma A.2. *Let $\{Y_i\}_{i=1,\ldots,n}$ be independent random variables from the exponential family of distributions with natural parameters $\theta_i \in \Theta$. Let μ_i and σ_i^2 denote the mean and variance of Y_i, respectively. Let $\{\nu_i\}_{i=1,\ldots,n}$ be real numbers satisfying*

$$\sum_{i=1}^{n} \nu_i^2 \sigma_i^2 = 1, \quad \max_{1\le i\le n} \nu_i^2 = O(n^{-1}).$$

Then, there exists a positive sequence $d_n = o(n^{1/2})$ such that

$$P\Big(\sum_{i=1}^{n} \nu_i (Y_i - \mu_i) > d_n\Big) \le \exp\big(-d_n^2/3\big)$$

for sufficiently large n.

Lemma A.2 establishes an important property of the exponential family, which will be used in Theorem 2.2. The proof of Lemma A.2 can be found in [16] and is thus omitted.

A.2 Proof of Theorem 2.2

We now prove Theorem 2.2.

Proof of Theorem 2.2. We prove this theorem by showing that at the t-th iteration, the estimator $\widehat{s}_t$ asymptotically falls into the collection of overfitted models that contain s^* as a submodel. This is guaranteed by the fact that the minimum of the augmented Lagrangian function based on an overfitted model is asymptotically less than that of any underfitted model with at least one feature in s^* excluded.

For convenience, let $L_\eta(\boldsymbol{\beta}_{s'}) = L_\eta(\boldsymbol{\alpha}, \boldsymbol{\beta}_{s'}, \boldsymbol{\rho})$. Let $\boldsymbol{\beta}_s$ be the unrestricted $\boldsymbol{\beta}$- update estimator of $\boldsymbol{\beta}$ based on model s. The theorem is proved if $P\{\widehat{s} \in S_+^k\} \to 1$. Thus, it suffices to show that

$$P\left(\min_{s\in S_-^k} L_\eta(\boldsymbol{\beta}_s) \le \max_{s\in S_+^k} L_\eta(\boldsymbol{\beta}_s)\right) \to 0, \quad \text{as} \quad N \to \infty.$$

For any $s \in S_-^k$, define $s' = s \cup s^* \in S_+^{2k}$. Take one $\boldsymbol{\beta}_{s'}$ that is close to $\boldsymbol{\beta}_{s'}^*$ such that $\|\boldsymbol{\beta}_{s'} - \boldsymbol{\beta}_{s'}^*\| = \omega_1 N^{-\tau_1}$, wehre ω_1 and $\tau_1 > 0$ are defined in Assumption 2. Let $S_N(\boldsymbol{\beta})$ and

$H_N(\boldsymbol{\beta})$ be the gradient and the Hessian of the augmented Lagrangian function $L_\eta(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho})$, respectively. By Taylor's expansion, we have

$$\begin{aligned} L_\eta(\boldsymbol{\beta}^*_{s'}) - L_\eta(\boldsymbol{\beta}_{s'}) &= (\boldsymbol{\beta}^*_{s'} - \boldsymbol{\beta}_{s'}) S_N(\boldsymbol{\beta}^*_{s'}) - \frac{1}{2}(\boldsymbol{\beta}_{s'} - \boldsymbol{\beta}^*_{s'}) H_N(\widetilde{\boldsymbol{\beta}}^*_{s'})(\boldsymbol{\beta}_{s'} - \boldsymbol{\beta}^*_{s'}) \\ &\leq \omega_1 N^{-\tau_1} \|S_N(\boldsymbol{\beta}^*_{s'})\|_2 - \frac{\eta m}{2} \|\boldsymbol{\beta}_{s'} - \boldsymbol{\beta}^*_{s'}\|_2^2 \\ &\leq \omega_1 N^{-\tau_1} \|S_N(\boldsymbol{\beta}^*_{s'})\|_2 - \frac{c_1 \omega_1^2}{2} N^{1-2\tau_1}, \end{aligned}$$

where $\widetilde{\boldsymbol{\beta}}^*_{s'}$ is an intermediate value between $\boldsymbol{\beta}_{s'}$ and $\boldsymbol{\beta}^*_{s'}$. Thus, it follows

$$\begin{aligned} P\big(L_\eta(\boldsymbol{\beta}^*_{s'}) - L_\eta(\boldsymbol{\beta}_{s'}) \geq 0\big) &\leq P\big(\omega_1 N^{-\tau_1} \|S_N(\boldsymbol{\beta}^*_{s'})\|_2 - \frac{c_1 \omega_1^2}{2} N^{1-2\tau_1} \geq 0\big) \\ &= P\big(\|S_N(\boldsymbol{\beta}^*_{s'})\|_2 \geq \frac{c_1 \omega_1}{2} N^{1-\tau_1}\big). \end{aligned}$$

Since $L_\eta(\boldsymbol{\beta}_{s'})$ is convex in $\boldsymbol{\beta}_{s'}$, the above result holds for any $\boldsymbol{\beta}_{s'}$ satisfying $\|\boldsymbol{\beta}_{s'} - \boldsymbol{\beta}^*_{s'}\| \geq \omega_1 N^{-\tau_1}$. Now consider t-iteration with $\boldsymbol{\alpha}_l^{(t)}$, $\boldsymbol{\beta}^{(t)}$ and $\boldsymbol{\rho}_l^{(t)}$. For any $s \in S_-^k$, let $\breve{\boldsymbol{\beta}}^{(t)}_{s'}$ be augmented $\boldsymbol{\beta}_s^{(t)}$ with zero entries corresponding to the elements in $s' \setminus s^*$. By Assumption 1, $\|\breve{\boldsymbol{\beta}}^{(t)}_{s'} - \boldsymbol{\beta}^*_{s'}\|_2 \geq \|\boldsymbol{\beta}^*_{s'}\|_2 \geq \omega_1 N^{-\tau_1}$. Therefore, it suffices to show that

$$P\big(\|S_N^{(t)}(\boldsymbol{\beta}^*_{s'})\|_2 \geq c_1 \omega_1 / 2 \cdot N^{1-\tau_1}\big) = o(1). \tag{A.10}$$

By the definition of $L_\eta(\boldsymbol{\alpha}, \boldsymbol{\beta}^*_{s'}, \boldsymbol{\rho})$, one has

$$S_N^{(t)}(\boldsymbol{\beta}^*_{s'}) = -\sum_{l=1}^m \boldsymbol{\rho}_l^{(t)} - \eta \sum_{l=1}^m (\boldsymbol{\alpha}_l^{(t)} - \boldsymbol{\beta}^*_{s'}).$$

Additionally, by the $\boldsymbol{\alpha}_l$-update and $\boldsymbol{\rho}_l$-update in (2.3), one has

$$\boldsymbol{X}_l^\top \boldsymbol{X}_l \boldsymbol{\alpha}_l^{(t+1)} = \boldsymbol{X}_l^\top \boldsymbol{Y}_l + \eta \boldsymbol{\beta}_l^{(t+1)} - \eta \boldsymbol{\alpha}_l^{(t+1)} - \boldsymbol{\rho}_l^{(t)} = \boldsymbol{X}_l^\top \boldsymbol{Y}_l - \boldsymbol{\rho}_l^{(t+1)}.$$

It is equivalent to

$$\boldsymbol{\rho}_l^{(t+1)} = \boldsymbol{X}_l^\top \boldsymbol{Y}_l - \boldsymbol{X}_l^\top \boldsymbol{X}_l \boldsymbol{\alpha}_l^{(t+1)}. \tag{A.11}$$

Hence,

$$
\begin{aligned}
\|S_N^{(t)}(\boldsymbol{\beta}_{s'}^*)\|_2 &= \big\| \sum_{l=1}^{m} \boldsymbol{\rho}_l^{(t)} + \eta(\boldsymbol{\alpha}_l^{(t)} - \boldsymbol{\beta}_{s'}^*) \big\|_2 \\
&= \big\| \sum_{l=1}^{m} (\boldsymbol{X}_l^\top \boldsymbol{Y}_l - \boldsymbol{X}_l^\top \boldsymbol{X}_l \boldsymbol{\alpha}_l^{(t)} + \eta \boldsymbol{\alpha}_l^{(t)} - \eta \boldsymbol{\beta}_{s'}^*) \big\|_2 \\
&= \| \sum_{l=1}^{m} \boldsymbol{X}_l^\top (\boldsymbol{Y}_l - \boldsymbol{X}_l \boldsymbol{\beta}_{s'}^*) - \sum_{l=1}^{m} (\eta I - \boldsymbol{X}_l^\top \boldsymbol{X}_l)(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t)}) \|_2 \\
&\leq \|\boldsymbol{X}^\top (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_{s'}^*)\|_2 + \sum_{l=1}^{m} \|(\eta I - \boldsymbol{X}_l^\top \boldsymbol{X}_l)(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t)})\|_2 \\
&:= (i) + (ii).
\end{aligned} \tag{A.12}
$$

For term (i) in (A.12), we have

$$
\begin{aligned}
& P\left(\|\boldsymbol{X}^\top (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_{s'}^*)\|_2 \geq \frac{c_1 \omega_1}{6} N^{1-\tau_1} \right) \\
\leq & \sum_{j \in s'} P\left(\left\{ \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}_{s'}^*) x_{ij} \right\}^2 \geq k^{-1} c_1^2 \omega_1^2 / 36 \cdot N^{2-2\tau_1} \right) \\
= & \sum_{j \in s'} P\left(\left\{ \sum_{i=1}^{n} (y_i - E(y_i|\boldsymbol{x}_i)) x_{ij} \right\}^2 \geq k^{-1} c_1^2 \omega_1^2 / 36 \cdot N^{2-2\tau_1} \right).
\end{aligned} \tag{A.13}
$$

Let $\nu_i = x_{ij} / \sqrt{\sum_{i=1}^{N} x_{ij}^2}$. Thus $\sum_{i=1}^{N} \nu_i^2 = 1$. By Assumptions 2 and 3, one has $\max_i \{\nu_i^2\} \leq N^{-1}$, $\sum_{i=1}^{N} x_{ij}^2 \leq c_2^2 N$ and $k \leq \omega_2 N^{\tau_2}$. Therefore, the conditions in Lemma A.2 are all satisfied. By Lemma A.2,

$$
\begin{aligned}
& P\left(\sum_{i=1}^{n} \{y_i - E(y_i|\boldsymbol{x}_i)\} x_{ij} \geq \frac{c_1 \omega_1}{6} k^{-\frac{1}{2}} N^{1-\tau_1} \right) \\
\leq & P\left(\sum_{i=1}^{N} \nu_i \{y_i - E(y_i|\boldsymbol{x}_i)\} \geq \frac{c_1 \omega_1}{6 c_2} N^{\frac{1}{2}(1-2\tau_1-\tau_2)} \right) \\
\leq & \exp\left(-\frac{c_1^2 \omega_1^2}{108 c_2^2} N^{1-2\tau_1-\tau_2} \right).
\end{aligned} \tag{A.14}
$$

Using the same strategy, we obtain that

$$P\left(\sum_{i=1}^{n}\{y_i - E(y_i|\boldsymbol{x}_i)\}x_{ij} \leq -\frac{c_1\omega_1}{6}k^{-\frac{1}{2}}N^{1-\tau_1}\right) \leq \exp\left(-\frac{c_1^2\omega_1^2}{108c_2^2}N^{1-2\tau_1-\tau_2}\right). \tag{A.15}$$

Combining the inequalities (A.14) and (A.15), we have

$$\begin{aligned} & P\left(\|\boldsymbol{X}^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}_{s'}^*)\|_2 \geq \frac{c_1\omega_1}{6}N^{1-\tau_1}\right) \\ \leq\ & 4k\exp\big(-\frac{c_1^2\omega_1^2}{108c_2^2}N^{1-2\tau_1-\tau_2}\big) \\ \leq\ & 4\omega_2\exp\big(\tau_2\log N - \frac{c_1^2\omega_1^2}{108c_2^2}N^{1-2\tau_1-\tau_2}\big) = o(1). \end{aligned} \tag{A.16}$$

For term (ii) in (A.12), we have

$$\sum_{l=1}^{m}\|(\eta I - \boldsymbol{X}_l^\top\boldsymbol{X}_l)(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t)})\|_2 \leq (\eta - c_1 n)\sum_{l=1}^{m}\|\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t)}\|_2$$

and

$$\begin{aligned} & \|\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t)}\|_2 \\ =\ & \|\boldsymbol{\beta}_{s'}^* - (\boldsymbol{X}_l^\top\boldsymbol{X}_l + \eta I)^{-1}(\eta\boldsymbol{\beta}^{(t)} + \boldsymbol{X}_l^\top\boldsymbol{X}_l\boldsymbol{\alpha}_l^{(t-1)})\|_2 \\ =\ & \|(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t-1)}) - (\boldsymbol{X}_l^\top\boldsymbol{X}_l + \eta I)^{-1}\eta(\boldsymbol{\beta}^{(t)} - \boldsymbol{\alpha}_l^{(t-1)})\|_2 \\ \leq\ & \|(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t-1)}) - (\boldsymbol{X}_l^\top\boldsymbol{X}_l + \eta I)^{-1}\eta(\overline{\boldsymbol{\beta}}^{(t)} - \boldsymbol{\alpha}_l^{(t-1)})\|_2 \\ =\ & \|(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t-1)}) - (\boldsymbol{X}_l^\top\boldsymbol{X}_l + \eta I)^{-1}\boldsymbol{X}_l^\top(\boldsymbol{Y}_l - \boldsymbol{X}_l\boldsymbol{\alpha}_l^{(t-1)})\|_2 \\ =\ & \|(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t-1)}) - (\boldsymbol{X}_l^\top\boldsymbol{X}_l + \eta I)^{-1}[\boldsymbol{X}_l^\top\boldsymbol{X}_l(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t-1)}) + \boldsymbol{X}_l^\top(\boldsymbol{Y}_l - \boldsymbol{X}_l\boldsymbol{\beta}_{s'}^*)]\|_2 \\ =\ & \|(\boldsymbol{X}_l^\top\boldsymbol{X}_l + \eta I)^{-1}\eta(\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t-1)}) - (\boldsymbol{X}_l^\top\boldsymbol{X}_l + \eta I)^{-1}\boldsymbol{X}_l^\top(\boldsymbol{Y}_l - \boldsymbol{X}_l\boldsymbol{\beta}_{s'}^*)\|_2 \\ \leq\ & \frac{\eta}{\eta + c_1 n}\|\boldsymbol{\beta}_{s'}^* - \boldsymbol{\alpha}_l^{(t-1)}\|_2 + \frac{1}{\eta + c_1 n}\|\boldsymbol{X}_l^\top(\boldsymbol{Y}_l - \boldsymbol{X}_l\boldsymbol{\beta}_{s'}^*)\|_2 \\ \leq\ & \ldots \leq \left(\frac{\eta}{\eta + c_1 n}\right)^t\|\boldsymbol{\beta}_{s'}^* - \boldsymbol{\beta}^{(0)}\|_2 + \sum_{j=1}^{t}\frac{\eta^{j-1}}{(\eta + c_1 n)^j}\|\boldsymbol{X}_l^\top(\boldsymbol{Y}_l - \boldsymbol{X}_l\boldsymbol{\beta}_{s'}^*)\|_2. \end{aligned} \tag{A.17}$$

The first term of last inequality of (A.17) approaches to 0 as $t \to \infty$. Following the same strategy in (A.13)-(A.16) of term (i), the second term of last inequality of (A.17) is sufficiently small with probability approaching to 1 as $n \to \infty$.

Combining the above inequalities and letting $t \to \infty$, we get

$$P(L_\eta(\boldsymbol{\beta}^*_{s'}) - L_\eta(\boldsymbol{\beta}_{s'}) \geq 0) \leq 4\omega_2 \exp\big(\tau_2 \log N - \frac{c_1^2\omega_1^2}{108c_2^2} N^{1-2\tau_1-\tau_2}\big).$$

Consequently, by Bonferroni inequality and condition $\tau_1 + \tau_2 \leq (1-a)/2$, we have

$$\begin{aligned} P\{\min_{s\in S_-^k} L_\eta(\boldsymbol{\beta}_{s'}) \leq L_\eta(\boldsymbol{\beta}^*_{s'})\} &\leq \sum_{s\in S_-^k} P\{L_\eta(\boldsymbol{\beta}_{s'}) \leq L_\eta(\boldsymbol{\beta}^*_{s'})\} \\ &\leq p^k P\{L_\eta(\boldsymbol{\beta}_{s'}) \leq L_\eta(\boldsymbol{\beta}^*_{s'})\} \\ &\leq 4\omega_2 \exp\big(\tau_2 \log N + \omega_2 N^{a+\tau_2} - c_1^2\omega_1^2/108c_2^2 \cdot N^{1-2\tau_1-\tau_2}\big) \\ &= o(1). \end{aligned} \tag{A.18}$$

Consequently,

$$P\{\min_{s\in S_-^k} L_\eta(\boldsymbol{\beta}_s) \leq \max_{s\in S_+^k} L_\eta(\boldsymbol{\beta}_s)\} \leq P\{\min_{s\in S_-^k} L_\eta(\breve{\boldsymbol{\beta}}_{s'}) \leq L_\eta(\boldsymbol{\beta}^*_{s'})\} \to 0 \quad \text{as} \quad N \to \infty.$$

The theorem is therefore proved. □

A.3 Proof of Theorem 2.3

Next we prove Theorem 2.3.

Proof of Theorem 2.3. To access the performance of the finite-iteration DHS procedure, following the similar strategy of Theorem 2.2, it is sufficient to show that the first term of (A.17) is small enough for a large iteration T. Namely, for $m\eta \geq c_1 N$, we need to choose appropriate initialization $\boldsymbol{\beta}^{(0)}$ and T such that

$$\left(\frac{\eta}{\eta + c_1 n}\right)^T \|\boldsymbol{\beta}^*_{s'} - \boldsymbol{\beta}^{(0)}\|_2 \leq \frac{\omega_1}{6} N^{1-\tau_1}.$$

From the initialization of $\boldsymbol{\beta}^{(0)}$ in Theorem 2.3, we have

$$\begin{aligned} \left(\frac{\eta}{\eta + c_1 n}\right)^T \|\boldsymbol{\beta}^*_{s'} - \boldsymbol{\beta}^{(0)}\|_2 &\leq \left(\frac{\eta}{\eta + c_1 n}\right)^T \frac{\omega_1}{6} \log p \cdot m^{1-\tau_1} n^{-\tau_1} \\ &\leq \left(\frac{\eta}{\eta + c_1 n}\right)^T \frac{\omega_1}{6} m N^{1-\tau_1}. \end{aligned} \tag{A.19}$$

The theorem then follows from the requirement that

$$\left(\frac{\eta}{\eta + c_1 n}\right)^T \frac{\omega_1}{6} m N^{1-\tau_1} \leq \frac{\omega_1}{6} N^{1-\tau_1}. \tag{A.20}$$

Without lose of generality, we take logarithmic function here with a natural basis e. Then, the inequality (A.20) amounts to

$$\begin{aligned} (\text{A.20}) &\Leftrightarrow \left(\frac{\eta}{\eta + c_1 n}\right)^T \leq \frac{1}{m} \\ &\Leftrightarrow T \log\left(\frac{\eta}{\eta + c_1 n}\right) \leq -\log m \\ &\Leftrightarrow T \geq \frac{\log m}{\log(1 + c_1 n/\eta)} \end{aligned} \tag{A.21}$$

By the assumption $m\eta \geq c_1 N$, the above inequality holds if

$$T \geq 1/\log 2 \cdot \log m = T_0.$$

Theorem 2.3 is verified. □

Now we prove Corollary 2.1.

Proof of Corollary 2.1. Following the similar steps of Theorem 2.3, if the initialization $\boldsymbol{\beta}^{(0)}$ further satisfies

$$\|\boldsymbol{\beta}^*_{s'} - \boldsymbol{\beta}^{(0)}\|_2 \leq \frac{\omega_1}{6} \log p N^{-\tau_1}.$$

We have

$$\begin{aligned} \left(\frac{\eta}{\eta + c_1 n}\right)^T \|\boldsymbol{\beta}^*_{s'} - \boldsymbol{\beta}^{(0)}\|_2 &\leq \left(\frac{\eta}{\eta + c_1 n}\right)^T \frac{\omega_1}{6} \log p N^{-\tau_1} \\ &\leq \left(\frac{\eta}{\eta + c_1 n}\right)^T \frac{\omega_1}{6} N^{1-\tau_1}. \end{aligned} \tag{A.22}$$

The theorem thus follows from the requirement that

$$\left(\frac{\eta}{\eta + c_1 n}\right)^T \leq 1. \tag{A.23}$$

It is easy to see that the above requirement (A.23) holds for any $T \geq 1$. Hence Corollary 2.1 is proved. □

Appendix B

Supplementary Information for Chapter 3

In this appendix, we provide the proofs of theorems in the Chapter 3.

B.1 Proof of Theorem 3.1

Proof of Theorem 3.1: For the convenience of presentation, let

$$\Delta(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik}^{(t)} \log \Big(\frac{\pi_k \phi(y_i; x_i^{\mathrm{T}} \beta_k, \sigma_k^2)}{\pi_k^{(t)} \phi(y_i; x_i^{\mathrm{T}} \beta_k^{(t)}, \sigma_k^{2(t)})} \Big),$$

where $r_{ik}^{(t)} = E(z_{ik} \mid y_i, \theta^{(t)}) \in [0, 1]$ is defined in the Expectation step of the proposed procedure and $\theta^{(t)} = (\pi^{(t)}, \beta^{(t)}, \sigma^{2(t)})$ is the parameter update at the tth iteration.

By the definition of adjusted likelihood (3.10), we have

$$\begin{aligned}\tilde{\ell}_n(\theta^{(t+1)}) - \tilde{\ell}_n(\theta^{(t)}) &= \frac{1}{n}\sum_{i=1}^{n}\Big[\log\Big\{\sum_{k=1}^{K}\pi_k^{(t+1)}\phi(y_i; x_i^{\mathrm{T}}\beta_k^{(t+1)}, \sigma_k^{2(t+1)})\Big\} \\ &\quad - \log\Big\{\sum_{k=1}^{K}\pi_k^{(t)}\phi(y_i; x_i^{\mathrm{T}}\beta_k^{(t)}, \sigma_k^{2(t)})\Big\}\Big] + \sum_{k=1}^{K}\big\{\psi(\sigma_k^{2(t)}) - \psi(\sigma_k^{2(t+1)})\big\} \\ &= \frac{1}{n}\sum_{i=1}^{n}\Big[\log\Big\{\sum_{k=1}^{K} r_{ik}^{(t)}\pi_k^{(t+1)}\phi(y_i; x_i^{\mathrm{T}}\beta_k^{(t+1)}, \sigma_k^{2(t+1)}) r_{ik}^{-1(t)}\Big\} \\ &\quad - \log\Big\{\sum_{k=1}^{K}\pi_k^{(t)}\phi(y_i; x_i^{\mathrm{T}}\beta_k^{(t)}, \sigma_k^{2(t)})\Big\}\Big] + \sum_{k=1}^{K}\big\{\psi(\sigma_k^{2(t)}) - \psi(\sigma_k^{2(t+1)})\big\}.\end{aligned} \tag{B.1}$$

Since $\sum_{k=1}^{K} r_{ik}^{(t)} = 1$, Jensen's inequality implies that

$$\log\Big\{\sum_{k=1}^{K} r_{ik}^{(t)}\pi_k^{(t+1)}\phi(y_i; x_i^{\mathrm{T}}\beta_k^{(t+1)}, \sigma_k^{2(t+1)}) r_{ik}^{-1(t)}\Big\} \geq \sum_{k=1}^{K} r_{ik}^{(t)} \log\Big\{\pi_k^{(t+1)}\phi(y_i; x_i^{\mathrm{T}}\beta_k^{(t+1)}, \sigma_k^{2(t+1)}) r_{ik}^{-1(t)}\Big\},$$

and the second term of (B.1) can be expressed as

$$\log\Big\{\sum_{k=1}^{K}\pi_k^{(t)}\phi(y_i; x_i^{\mathrm{T}}\beta_k^{(t)}, \sigma_k^{2(t)})\Big\} = \sum_{k=1}^{K} r_{ik}^{(t)} \log\Big\{\sum_{k=1}^{K}\pi_k^{(t)}\phi(y_i; x_i^{\mathrm{T}}\beta_k^{(t)}, \sigma_k^{2(t)})\Big\}.$$

Thus, we further have

$$\tilde{\ell}_n(\theta^{(t+1)}) - \tilde{\ell}_n(\theta^{(t)}) \geq \Delta(\pi_k^{(t+1)}, \beta_k^{(t+1)}, \sigma_k^{2(t+1)}; \theta^{(t)}) + \textstyle\sum_{k=1}^{K}\Big\{\psi(\sigma_k^{2(t)}) - \psi(\sigma_k^{2(t+1)})\Big\}. \tag{B.2}$$

To prove the theorem, it suffice to show that the right-hand side of (B.2) is non-negative for $t \geq 1$, to which we now turn.

Recall that

$$\beta_k^{(t+1)} = \arg\max_{\beta_k} h(\beta_k; \theta^{(t)}), \text{ subject to } \|\beta_k\|_0 \leq s_k,$$

where

$$h(\beta_k; \theta^{(t)}) = g(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) + (\beta_k - \beta_k^{(t)})^{\mathrm{T}} g'(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) - \frac{u}{2}\|\beta_k - \beta_k^{(t)}\|_2^2$$

with $g(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)})$ defined in the Expectation step.

When $t \geq 1$, we have

$$\begin{aligned}
g(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) &= h(\beta_k^{(t)}; \theta^{(t)}) \\
&\leq h(\beta_k^{(t+1)}; \theta^{(t)}) \\
&= g(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) + (\beta_k^{(t+1)} - \beta_k^{(t)})^{\mathrm{T}} g'(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) - \frac{u}{2}\|\beta_k^{(t+1)} - \beta_k^{(t)}\|_2^2 \\
&= g(\beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}) - \frac{u}{2}\|\beta_k^{(t+1)} - \beta_k^{(t)}\|^2 + \frac{1}{n}\sum_{i=1}^{n} r_{ik}^{(t)} \Big\{ \frac{\sigma_k^{-2(t)}}{2}(y_i - x_i^{\mathrm{T}}\beta_k^{(t+1)})^2 \\
&\quad - \frac{\sigma_k^{-2(t)}}{2}(y_i - x_i^{\mathrm{T}}\beta_k^{(t)})^2 + \sigma_k^{-2(t)} x_i (y_i - x_i^{\mathrm{T}}\beta_k^{(t)})(\beta_k^{(t+1)} - \beta_k^{(t)}) \Big\} \\
&= g(\beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}) - \frac{u}{2}\|\beta_k^{(t+1)} - \beta_k^{(t)}\|^2 + \frac{1}{n}\sum_{i=1}^{n} r_{ik}^{(t)} \big\{ b(\beta_k^{(t+1)}) - b(\beta_k^{(t)}) \\
&\quad - b'(\beta_k^{(t)})(\beta_k^{(t+1)} - \beta_k^{(t)}) \big\}, \qquad \text{(B.3)}
\end{aligned}$$

where $b(\beta) = \sigma_k^{-2(t)}(y_i - x_i^{\mathrm{T}}\beta)^2/2, b'(\beta) = \partial b(\beta)/\partial\beta = -\sigma_k^{-2(t)} x_i (y_i - x_i^T \beta)$. Since

$$b(\beta_k^{(t+1)}) - b(\beta_k^{(t)}) - (\beta_k^{(t+1)} - \beta_k^{(t)})^{\mathrm{T}} b'(\beta_k^{(t)}) = \frac{\sigma_k^{-2(t)}}{2}(\beta_k^{(t+1)} - \beta_k^{(t)})^{\mathrm{T}} x_i x_i^{\mathrm{T}} (\beta_k^{(t+1)} - \beta_k^{(t)}),$$

inequality (B.3) further implies that

$$\begin{aligned}
g(\beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) &\leq g(\beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}) + \frac{1}{n}\sum_{i=1}^{n} r_{ik}^{(t)} \Big\{ \frac{1}{2}\sigma_k^{-2(t)} (\beta_k^{(t+1)} - \beta_k^{(t)})^{\mathrm{T}} x_i x_i^{\mathrm{T}} \\
&\qquad (\beta_k^{(t+1)} - \beta_k^{(t)}) \Big\} - \frac{u}{2}\|\beta_k^{(t+1)} - \beta_k^{(t)}\|^2 \\
&\leq g(\beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}) + \frac{1}{2}(\sigma_k^{-2(t)}\rho - u)\|\beta_k^{(t+1)} - \beta_k^{(t)}\|^2 \\
&\leq g(\beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}), \qquad \text{(B.4)}
\end{aligned}$$

where the last inequality holds as we require $u \geq \sigma_k^{-2(t)}\rho$.

$\Delta(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)})$ can be expressed as

$$\Delta(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)}) = \sum_{k=1}^{K} g(\beta_k, \sigma_k^2; \theta^{(t)}) + \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} r_{ik}^{(t)} \log(\pi_k) - Q(\theta^{(t)}; \theta^{(t)}).$$

Inequality (B.4) further implies that

$$\Delta(\pi_k, \beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}) \geq \Delta(\pi_k, \beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) \qquad \text{(B.5)}$$

for $t \geq 1$.

Also, with $\Delta(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)})$, we can re-express $\sigma_k^{2(t+1)}$ and $\pi_k^{(t+1)}$ by

$$\begin{aligned}
\pi_k^{(t+1)} &= \arg\max_{\pi_k} \Delta(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)}), \\
\sigma_k^{2(t+1)} &= \arg\max_{\sigma_k^2} \Delta(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)}) - \psi(\sigma_k^2).
\end{aligned}$$

Therefore, by definition, we have

$$\Delta(\pi_k, \beta_k, \sigma_k^{2(t+1)}; \theta^{(t)}) - \sum_{k=1}^{K} \psi(\sigma_k^{2(t+1)}) \geq \Delta(\pi_k, \beta_k, \sigma_k^{2(t)}; \theta^{(t)}) - \sum_{k=1}^{K} \psi(\sigma_k^{2(t)}), \tag{B.6}$$

and

$$\Delta(\pi_k^{(t+1)}, \beta_k, \sigma_k^2; \theta^{(t)}) \geq \Delta(\pi_k^{(t)}, \beta_k, \sigma_k^2; \theta^{(t)}). \tag{B.7}$$

Inequalities (B.5)-(B.7) imply that

$$\begin{aligned}
\Delta(\pi_k^{(t+1)}, \beta_k^{(t+1)}, \sigma_k^{2(t+1)}; \theta^{(t)}) - \sum_{k=1}^{K} \psi(\sigma_k^{2(t+1)}) &\geq \Delta(\pi_k^{(t+1)}, \beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}) - \sum_{k=1}^{K} \psi(\sigma_k^{2(t)}) \\
&\geq \Delta(\pi_k^{(t)}, \beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}) - \sum_{k=1}^{K} \psi(\sigma_k^{2(t)}) \\
&\geq \Delta(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) - \sum_{k=1}^{K} \psi(\sigma_k^{2(t)}),
\end{aligned}$$

where $\Delta(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}) = 0$. This indicates that the right-hand side of (B.2) is non-negative when $t \geq 1$. The theorem is therefore proved. □

B.2 Proof of Theorem 3.2

Proof of Theorem 3.2: We prove the theorem in the following four steps. 1) First, we show that, with a given dataset (Y, X), $\theta^{(t)}$ is confined in a bounded region for any initial $\theta^{(0)}$. Thus, $\theta^{(t)}$ has at least one cluster point $\tilde{\theta}$. 2) Using the boundedness of $\theta^{(t)}$, we then show that $\|\theta^{(t+1)} - \theta^{(t)}\|_2^2 \to 0$ as $t \to \infty$. 3) Next, we show that any cluster point of $\theta^{(t)}$ is a local maximum of $\tilde{\ell}_n(\theta)$ subject to $\|\beta_k\|_0 \leq s_k$. 4) Finally, we show that $\tilde{\theta}$ is the unique cluster point, which implies $\theta^{(t)} \to \tilde{\theta}$ as $t \to \infty$.

We first prove the boundedness of $\theta^{(t)}$. By (3.6), $\beta_k^{(t)}$ has a sparse structure; its non-zero entries are indicated by $\mathcal{M}_k$ with $\tau(\mathcal{M}_k) \le s_k$. With a given dataset, there are at most finite number of sparse structures. On any of these sparse structures, we require the corresponding $X_{\mathcal{M}_k}^{\mathrm{T}} X_{\mathcal{M}_k}$ to be positive-definite and thus $\tilde{\ell}_n(\theta)$ is strictly concave in β_k with respect to $\mathcal{M}_k$. By Theorem 3.1, we have $\tilde{\ell}_n(\theta^{(t+1)}) \ge \tilde{\ell}_n(\theta^{(t)})$, which implies that $\beta_k^{(t)}$ is confined in a bounded region. Since $\beta_k^{(t)}$ is bounded, equation (3.8) further implies that $\sigma_k^{2(t)}$ and $\pi_k^{(t)}$ are both bounded. Therefore, $\theta^{(t)} = (\pi_1^{(t)}, \ldots, \pi_K^{(t)}, \beta_1^{(t)}, \ldots, \beta_K^{(t)}, \sigma_1^{2(t)}, \ldots, \sigma_K^{2(t)})$ is confined in a bounded region.

With the boundedness of $\theta^{(t)}$, we then show that the difference between $\theta^{(t+1)}$ and $\theta^{(t)}$ goes to zero as $t \to \infty$. For the ease of presentation, let $\tilde{\ell}_n(\pi_k, \beta_k, \sigma_k^2)$ stand for $\tilde{\ell}_n(\theta)$ and

$$\tilde{\Delta}(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)}) = \Delta(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)}) - \sum_{k=1}^{K} \psi(\sigma_k^2).$$

Using the same technique in obtaining (B.2), we have

$$\begin{aligned}
&\tilde{\ell}_n\big(\pi_k, \beta_k, \sigma_k^{2(t+1)}\big) - \tilde{\ell}_n\big(\pi_k, \beta_k, \sigma_k^{2(t)}\big) \\
&\ge \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik}^{(t)} \Big\{ \log\Big(\frac{\pi_k \phi(y_i; x^{\mathrm{T}} \beta_k, \sigma_k^{2(t+1)})}{\pi_k \phi(y_i; x^{\mathrm{T}} \beta_k, \sigma_k^{2(t)})} \Big) \Big\} + \sum_{k=1}^{K} \big\{ \psi(\sigma_k^{2(t)}) - \psi(\sigma_k^{2(t+1)}) \big\} \\
&= \tilde{\Delta}\big(\pi_k, \beta_k, \sigma_k^{2(t+1)}; \theta^{(t)}\big) - \tilde{\Delta}\big(\pi_k, \beta_k, \sigma_k^{2(t)}; \theta^{(t)}\big) \ge 0,
\end{aligned} \tag{B.8}$$

where the last inequality is implied by (B.6). Similarly, by (B.7) and (B.5), we have

$$\tilde{\ell}_n\big(\pi_k^{(t+1)}, \beta_k, \sigma_k^2\big) - \tilde{\ell}_n\big(\pi_k^{(t)}, \beta_k, \sigma_k^2\big) \ge \Delta\big(\pi_k^{t+1}, \beta_k, \sigma_k^2; \theta^{(t)}\big) - \Delta\big(\pi_k^{(t)}, \beta_k, \sigma_k^2; \theta^{(t)}\big) \ge 0, \tag{B.9}$$

and

$$\tilde{\ell}_n\big(\pi_k, \beta_k^{(t+1)}, \sigma_k^{2(t)}\big) - \tilde{\ell}_n\big(\pi_k, \beta_k^{(t)}, \sigma_k^{2(t)}\big) \ge \Delta\big(\pi_k, \beta_k^{(t+1)}, \sigma_k^{2(t)}; \theta^{(t)}\big) - \Delta\big(\pi_k, \beta_k^{(t)}, \sigma_k^{2(t)}; \theta^{(t)}\big) \ge 0. \tag{B.10}$$

Inequalities (B.8)-(B.10) together imply that

$$\tilde{\ell}_n\big(\pi_k^{(t+1)}, \beta_k^{(t+1)}, \sigma_k^{2(t+1)}\big) \ge \tilde{\ell}_n\big(\pi_k^{(t+1)}, \beta_k^{(t+1)}, \sigma_k^{2(t)}\big) \ge \tilde{\ell}_n\big(\pi_k^{(t)}, \beta_k^{(t+1)}, \sigma_k^{2(t)}\big) \ge \tilde{\ell}_n\big(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t)}\big). \tag{B.11}$$

By Taylor's expansion, there exists a $\check{\sigma}_k^2$ between $\sigma_k^{2(t)}$ and $\sigma_k^{2(t+1)}$ such that

$$\begin{aligned}
\tilde{\Delta}\big(\pi_k, \beta_k, \sigma_k^{2(t)}; \theta^{(t)}\big) = \tilde{\Delta}\big(\pi_k, \beta_k, \sigma_k^{2(t+1)}; \theta^{(t)}\big) &+ \big(\sigma_k^{2(t+1)} - \sigma_k^{2(t)}\big)^{\mathrm{T}} \tilde{\Delta}'\big(\pi_k, \beta_k, \sigma_k^{2(t+1)}; \theta^{(t)}\big) \\
&+ \frac{1}{2} \big(\sigma_k^{2(t+1)} - \sigma_k^{2(t)}\big)^{\mathrm{T}} \tilde{\Delta}''\big(\pi_k, \beta_k, \check{\sigma}_k^2; \theta^{(t)}\big) \big(\sigma_k^{2(t+1)} - \sigma_k^{2(t)}\big),
\end{aligned}$$

where $\tilde{\Delta}'(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)})$ and $\tilde{\Delta}''(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)})$ denote the first and second partial derivative of $\tilde{\Delta}(\pi_k, \beta_k, \sigma_k^2; \theta^{(t)})$ with respect to σ_k^2. By the definition of $\sigma_k^{2(t+1)}$, one has $\tilde{\Delta}'(\pi_k, \beta_k, \sigma_k^{2(t+1)}; \theta^{(t)}) = 0$. Thus, we have

$$\begin{aligned}\tilde{\Delta}\big(\pi_k, \beta_k, \sigma_k^{2(t+1)}; \theta^{(t)}\big) - \tilde{\Delta}\big(\pi_k, \beta_k, \sigma_k^{2(t)}; \theta^{(t)}\big) &= -\frac{\tilde{\Delta}''\big(\pi_k, \beta_k, \check{\sigma}_k^2; \theta^{(t)}\big)}{2} \|\sigma_k^{2(t+1)} - \sigma_k^{2(t)}\|_2^2 \\ &\geq \frac{1}{n} \sum_{k=1}^{K} \nu_n^2 \check{\sigma}_k^{-6} \|\sigma_k^{2(t+1)} - \sigma_k^{2(t)}\|_2^2.\end{aligned}$$

This together with (B.11) and (B.8) implies that

$$\begin{aligned}\tilde{\ell}_n\big(\pi_k^{(t+1)}, \beta_k^{(t+1)}, \sigma_k^{2(t+1)}\big) - \tilde{\ell}_n\big(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t)}\big) &\geq \tilde{\ell}_n\big(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t+1)}\big) - \tilde{\ell}_n\big(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t)}\big) \\ &\geq \tilde{\Delta}\big(\pi_k, \beta_k, \sigma_k^{2(t+1)}; \theta^{(t)}\big) - \tilde{\Delta}\big(\pi_k, \beta_k, \sigma_k^{2(t)}; \theta^{(t)}\big) \\ &\geq \frac{1}{n} \sum_{k=1}^{K} \nu_n^2 \check{\sigma}_k^{-6} \|\sigma_k^{2(t+1)} - \sigma_k^{2(t)}\|_2^2. \qquad \text{(B.12)}\end{aligned}$$

By (B.11), $\tilde{\ell}_n(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t)})$ is increasing with respect to t and is bounded above due to the boundedness of $\theta^{(t)}$. Thus, the left-hand side of (B.12) goes to zero as $t \to \infty$. By (3.8), $\sigma_k^{2(t)}$ is bounded away from zero and thus $\check{\sigma}_k^2$ is also bounded away from zero. Inequality (B.12) thus implies that $\|\sigma_k^{2(t+1)} - \sigma_k^{2(t)}\|_2^2 \to 0$ as $t \to \infty$.

Using (B.4) and (B.9), one can similarly show that

$$\tilde{\ell}_n\big(\pi_k^{(t+1)}, \beta_k^{(t+1)}, \sigma_k^{2(t+1)}\big) - \tilde{\ell}_n\big(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t)}\big) \geq \frac{1}{2} \sum_{k=1}^{K} \big(u - \sigma_k^{-2(t)} \rho\big) \|\beta_k^{(t+1)} - \beta_k^{(t)}\|_2^2 \qquad \text{(B.13)}$$

and

$$\tilde{\ell}_n\big(\pi_k^{(t+1)}, \beta_k^{(t+1)}, \sigma_k^{2(t+1)}\big) - \tilde{\ell}_n\big(\pi_k^{(t)}, \beta_k^{(t)}, \sigma_k^{2(t)}\big) \geq \sum_{k=1}^{K} \zeta \pi_k^{(t)} \|\pi_k^{(t+1)} - \pi_k^{(t)}\|_2^2, \qquad \text{(B.14)}$$

where $\zeta > 0$ is a lower bound of $\sum_{i=1}^{n} \phi(y_i, x_i^{\mathrm{T}} \beta_k^{(t)}, \sigma_k^{2(t)})/2 \sum_{k=1}^{K} \phi(y_i, x_i^{\mathrm{T}} \beta_k^{(t)}, \sigma_k^{2(t)})$. The above two inequalities imply that $\|\beta_k^{(t+1)} - \beta_k^{(t)}\|_2^2 \to 0$ and $\|\pi^{(t+1)} - \pi^{(t)}\|_2^2 \to 0$ as $t \to \infty$. These results together with $\|\sigma_k^{2(t+1)} - \sigma_k^{2(t)}\|_2^2 \to 0$ imply that

$$\|\theta^{(t+1)} - \theta^{(t)}\|_2^2 \to 0 \text{ as } t \to \infty. \qquad \text{(B.15)}$$

Since $\theta^{(t)}$ is bounded, it has at least one cluster point $\tilde{\theta} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_K, \tilde{\beta}_1, \ldots, \tilde{\beta}_K, \tilde{\sigma}_1^2, \ldots, \tilde{\sigma}_K^2)$. Let $\theta^{(t_m)}$ be a subsequence of $\theta^{(t)}$ such that $\theta^{(t_m)} \to \tilde{\theta}$ as $m \to \infty$. By (B.15), we also have $\theta^{(t_m+1)} \to \tilde{\theta}$ as $m \to \infty$. In the next step, we show that $\tilde{\theta}$ is a local maximum of $\tilde{\ell}_n(\theta)$ subject to $\|\beta_k\|_0 \leq s_k$.

Recall that

$$\pi_k^{(t_m+1)} = \arg\max_{\pi_k} Q\big(\theta; \theta^{(t_m)}\big),$$

and

$$\sigma_k^{2(t_m+1)} = \arg\max_{\sigma_k^2} Q\big(\theta; \theta^{(t_m)}\big) - \sum_{k=1}^{K} \psi\big(\sigma_k^2\big).$$

With $m \to \infty$, we immediately obtain

$$(\tilde{\pi}_k, \tilde{\sigma}_k^2) = \arg\max_{\pi_k, \sigma_k^2} Q\big(\theta; \tilde{\theta}\big) - \sum_{k=1}^{K} \psi\big(\sigma_k^2\big). \tag{B.16}$$

We now look into $\tilde{\beta}_k$. Since $\|\beta_k^{(t)}\|_0 \leq s_k$, we naturally have $\|\tilde{\beta}_k\|_0 \leq s_k$. By (B.6),

$$\beta_k^{(t_m+1)} = \arg\max_{\beta_k} h\big(\beta_k; \theta^{(t_m)}\big), \text{ subject to } \|\beta_k\|_0 \leq s_k.$$

With $m \to \infty$, we have

$$\tilde{\beta}_k = \arg\max_{\beta_k} h\big(\beta_k; \tilde{\theta}\big), \text{ subject to } \|\beta_k\|_0 \leq s_k. \tag{B.17}$$

We investigate $\tilde{\beta}_k$ in the following two cases.

Case 1 : $\|\tilde{\beta}_k\|_0 < s_k$. Equations (3.9) and (B.17) imply that

$$\tilde{\beta}_k = H_{s_k}\big(\tilde{\beta}_k + \frac{1}{u} g'(\tilde{\beta}_k, \tilde{\sigma}_k^2; \tilde{\theta})\big).$$

Since $\|\tilde{\beta}_k\|_0 < s_k$, we have

$$\tilde{\beta}_k = \tilde{\gamma}_k = \tilde{\beta}_k + \frac{1}{u} g'\big(\tilde{\beta}_k, \tilde{\sigma}_k^2; \tilde{\theta}\big),$$

which implies $g'(\tilde{\beta}_k, \tilde{\sigma}_k^2; \tilde{\theta}) = 0$.

Case 2 : $\|\tilde{\beta}_k\|_0 = s_k$. By definition of $h(\beta_k; \tilde{\theta})$, we have

$$h\big(\beta_k; \tilde{\theta}\big) = g\big(\tilde{\beta}_k, \tilde{\sigma}_k^2; \tilde{\theta}\big) + \big(\beta_k - \tilde{\beta}_k\big)^{\mathrm{T}} g'\big(\tilde{\beta}_k, \tilde{\sigma}_k^2; \tilde{\theta}\big) - \frac{u}{2}\|\beta_k - \tilde{\beta}_k\|_2^2.$$

(B.17) implies that

$$\frac{\partial h\big(\beta_k;\tilde{\theta}\big)}{\partial \beta_{kj}}|_{\beta_k=\tilde{\beta}_k} = \frac{\partial g\big(\beta_k,\tilde{\sigma}_k^2;\tilde{\theta}\big)}{\partial \beta_{kj}}|_{\beta_k=\tilde{\beta}_k} = 0$$

for any j such that $\tilde{\beta}_{kj} \neq 0$.

Combining the results from both cases, we know $\tilde{\beta}_k$ is the maximum of $g(\beta_k, \tilde{\sigma}_k^2; \tilde{\theta})$ with respect to its sparsity structure. This indicates that $\tilde{\beta}_k$ is a local maximum of $Q(\theta;\tilde{\theta})$ subject to $\|\beta_k\|_0 \leq s_k$.

The role of $\tilde{\beta}_k$ together with (B.16) suggests that $\tilde{\theta}$ is a local maximum of $Q(\theta;\tilde{\theta}) - \sum_{k=1}^K \psi(\sigma_k^2)$ under the sparsity constraint. Thus, we have

$$\frac{\partial\big\{Q\big(\theta;\tilde{\theta}\big) - \sum_{k=1}^K \psi\big(\sigma_k^2\big)\big\}}{\partial\theta}|_{\theta=\tilde{\theta}} = 0 \tag{B.18}$$

at $\tilde{\pi}_k$, $\tilde{\sigma}_k^2$ and the non-zero part of $\tilde{\beta}_k$. $\ell_n(\theta)$ can be expressed as

$$\tilde{\ell}_n\big(\theta\big) = Q\big(\theta;\theta^{(t)}\big) - \sum_{k=1}^K \psi\big(\sigma_k^2\big) + G\big(\theta;\theta^{(t)}\big), \tag{B.19}$$

where

$$G\big(\theta;\theta^{(t)}\big) = \frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K r_{ik}^{(t)} \log\Big(\frac{\pi_k\phi(y_i;x_i^{\mathrm{T}}\beta_k,\sigma_k^2)}{\sum_{k=1}^K \pi_k\phi(y_i;x_i^{\mathrm{T}}\beta_k,\sigma_k^2)}\Big).$$

By Jensen's inequality, we have

$$\begin{aligned} G\big(\theta;\theta^{(t)}\big) - G\big(\theta^{(t)};\theta^{(t)}\big) &= \frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K r_{ik}^{(t)} \log\Big(\frac{\pi_k\phi(y_i;x_i^{\mathrm{T}}\beta_k,\sigma_k^2)}{r_{ik}^{(t)}\sum_{k=1}^K \pi_k\phi(y_i;x_i^{\mathrm{T}}\beta_k,\sigma_k^2)}\Big) \\ &\leq \frac{1}{n}\sum_{i=1}^n \log\Big(\sum_{k=1}^K r_{ik}^{(t)}\frac{\pi_k\phi(y_i;x_i^{\mathrm{T}}\beta_k,\sigma_k^2)}{r_{ik}^{(t)}\sum_{k=1}^K \pi_k\phi(y_i;x_i^{\mathrm{T}}\beta_k,\sigma_k^2)}\Big) = 0, \end{aligned}$$

which implies that $\theta^{(t)} = \arg\max_\theta G(\theta;\theta^{(t)})$. Therefore, we have

$$\frac{\partial G\big(\theta;\tilde{\theta}\big)}{\partial\theta}|_{\theta=\tilde{\theta}} = 0. \tag{B.20}$$

Taking derivative on both sides of (B.19), we then obtain

$$\frac{\partial\tilde{\ell}_n\big(\theta\big)}{\partial\theta}|_{\theta=\tilde{\theta}} = \frac{\partial\big\{Q\big(\theta;\tilde{\theta}\big) - \sum_{k=1}^K \psi\big(\sigma_k^2\big)\big\}}{\partial\theta}|_{\theta=\tilde{\theta}} + \frac{\partial G\big(\theta;\tilde{\theta}\big)}{\partial\theta}|_{\theta=\tilde{\theta}} = 0$$

at $\tilde{\pi}_k$, $\tilde{\sigma}_k^2$ and the non-zero part of $\tilde{\beta}_k$. This implies that $\tilde{\theta}$ is a local maximum of $\tilde{\ell}_n(\theta)$ subject to $\|\beta_k\|_0 \leq s_k$.

We finalize the proof of theorem by showing the uniqueness of $\tilde{\theta}$. With a given dataset, there are at most finite number of sparse structures of β_k. The concavity of $\tilde{\ell}_n(\theta)$ with respect to $\mathcal{M}_k$ further implies that $\tilde{\ell}_n(\theta)$ has at most finite number of local maxima subject to $\|\beta_k\|_0 \leq s_k$. Suppose that $\theta^{(t)}$ has two distinct cluster points, say $\tilde{\theta}_1$ and $\tilde{\theta}_2$ with $\|\tilde{\theta}_1 - \tilde{\theta}_2\|_2 = \epsilon > 0$. Following the previous arguments, both $\tilde{\theta}_1$ and $\tilde{\theta}_2$ are the local maxima of $\tilde{\ell}_n(\theta)$ under the sparsity constraint. Since $\theta^{(t)}$ is bounded and $\|\theta^{(t+1)} - \theta^{(t)}\|_2 \to 0$ as $t \to \infty$, there must be infinitely many $\theta^{(t)}$ which are at least $\epsilon/3$ distance from $\tilde{\theta}_1$ and $\tilde{\theta}_2$. Those $\theta^{(t)}$ would have another cluster point $\tilde{\theta}_3$, which is also a local maximum of $\tilde{\ell}_n(\theta)$. Following this strategy, one can then find infinitely many local maxima of $\tilde{\ell}_n(\theta)$. This contradicts the argument on the finite number of local maxima. Therefore, $\tilde{\theta}$ is the unique cluster point of a bounded sequence $\theta^{(t)}$. This implies that

$$\theta^{(t)} \to \tilde{\theta}, \text{ as } t \to \infty.$$

The proof is therefore complete. □

B.3 Preliminaries for the Proof of Theorem 3.3

To facilitate the proof of Theorem 3.3, we first state a few technical lemmas as below. Our Lemmas B.1 - B.3 correspond to Lemmas 5.9, 5.14, 5.46 of [34], respectively. Readers may refer to Chapter 5 of [34] for their proofs. Here, we use $\|v\|_{\Psi_2}$ and $\|v\|_{\Psi_1}$ to denote two types of the Orlicz norm of a random variable v; their expressions are respectively given by

$$\|v\|_{\Psi_2} = \sup_{q \geq 1} q^{-1/2} (E|v|^q)^{1/q} \text{ and } \|v\|_{\Psi_1} = \sup_{q \geq 1} q^{-1} (E|v|^q)^{1/q}.$$

Lemma B.1. *Consider a finite number of independent centered sub-Gaussian random variables v_i. Then, $\sum_{i=1}^n v_i$ is also a centered sub-Gaussian random variable. Moreover,*

$$\Big\| \sum_{i=1}^n v_i \Big\|_{\Psi_2}^2 \leq c \sum_{i=1}^n \|v_i\|_{\Psi_2}^2,$$

where $c > 0$ is an absolute constant.

Lemma B.2. *Let v_1 and v_2 be two sub-Gaussian random variables, $v_1 v_2$ is a sub-exponential random variable with*

$$\|v_1 v_2\|_{\Psi_1} \leq c \max\big\{ \|v_1\|_{\Psi_2}^2, \|v_2\|_{\Psi_2}^2 \big\},$$

where $c > 0$ is an absolute constant.

Lemma B.3. *Let $v_1, \ldots, v_n$ be the n independent random variables valued in some Banach space and $\mathcal{F}$ be a function class defined on this Banach space. For any increasing convex function $\phi(\cdot)$ we have*

$$E\Big[\phi\Big\{\sup_{f\in\mathcal{F}}|\sum_{i=1}^{n} f(v_i) - Ev|\Big\}\Big] \leq E\Big[\phi\Big\{\sup_{f\in\mathcal{F}} 2|\sum_{i=1}^{n}\xi_i f(v_i)|\Big\}\Big],$$

where $\xi_1, \ldots, \xi_n$ are independent and identically distributed Rademacher random variables that are independent of $v_1, \ldots, v_n$.

We prove Theorem 3.3 by showing that, with a good $\theta^{(0)}$ and a sufficient number of iterations t, $\beta_k^{(t)}$ falls within a neighborhood of β_k^*, which ensures the sure screening of $\widehat{\mathcal{M}}_k^{(t)}$. To this end, let

$$\gamma_k = \arg\max_{\gamma} h(\gamma; \theta) \tag{B.21}$$

with $h(\gamma;\theta)$ defined in (3.5). Let $\mathcal{Y}$ and $\mathcal{X}$ denote the support of response y and feature x, respectively. We define the population version of function $g(\beta_k, \sigma_k^2; \theta)$ by

$$g_p(\beta_k, \sigma_k^2; \theta) = \iint_{\mathcal{Y}\times\mathcal{X}} r_k \log\big(\phi(y, x^{\mathrm{T}}\beta_k, \sigma_k^2)\big) f(y \mid x, \theta^*)\varrho(x)dydx,$$

where $\varrho(x)$ is the density of x and $r_k = \pi_k\phi(y, x^{\mathrm{T}}\beta_k, \sigma_k^2)/\sum_{k=1}^{K}\pi_k\phi(y, x^{\mathrm{T}}\beta_k, \sigma_k^2)$. Let

$$\bar{\gamma}_k = \arg\max_{\gamma} h_p(\gamma; \theta)$$

with

$$h_p(\gamma;\theta) = g_p(\beta_k, \sigma_k^2; \theta) + (\gamma - \beta_k)^{\mathrm{T}} g_p'(\beta_k, \sigma_k^2; \theta) - \frac{u}{2}\|\gamma - \beta_k\|_2^2,$$

where $g_p'(\beta_k, \sigma_k^2; \theta) = E(r_k\sigma_k^{-2}x(y - x^{\mathrm{T}}\beta_k))$ is the partial derivative of g_p with respect to β_k. Let$\|.\|_\infty$ denote the infinity norm of a vector or matrix. As a preliminary result to the proof of Theorem 3.3, we first derive a probability bound of $\|\gamma_k - \bar{\gamma}_k\|_2$ in the following proposition.

Proposition 1. *Suppose that Assumptions 1 and 4 are satisfied and γ_k in (B.21) is obtained based on a $\theta = (\pi_1, \ldots, \pi_K, \beta_1, \ldots, \beta_K, \sigma_1^2, \ldots, \sigma_K^2)$ with $\|\beta_k - \beta_k^*\|_2 \leq w\|\beta_k^*\|_2$ for some $w \in (0,1)$. Let $\vartheta_k = \max\{1, \max_k(\|\beta_k^*\|_1 + \sigma_k^*)^2\}$. Then, there exists constants $C_4, C_5 > 0$ such that*

$$\|\gamma_k - \bar{\gamma}_k\|_\infty \leq C_5 u^{-1}\sigma_k^{-2}\big(\vartheta_k + \|\beta_k\|_0^{\frac{1}{2}}\|\beta_k^*\|_2\big)\Big(\frac{\log p - \log\big(\frac{\delta}{4}\big)}{n}\Big)^{\frac{1}{2}}$$

with probability at least $1 - \delta$ for any $\delta \in (4\exp(-C_4 n + \log p), 1)$.

Proof. By definition, γ_k can be expressed by

$$\gamma_k = \beta_k + u^{-1} \sum_{i=1}^{n} r_{ik} \sigma_k^{-2} (x_i y_i - x_i x_i^{\mathrm{T}} \beta_k)/n$$

with $r_{ik} = \pi_k \phi(y_i, x_i^{\mathrm{T}} \beta_k, \sigma_k^2) / \sum_{k=1}^{K} \pi_k \phi(y_i, x_i^{\mathrm{T}} \beta_k, \sigma_k^2)$ and

$$\bar{\gamma}_k = \beta_k + u^{-1} g_p'(\beta_k, \sigma_k^2; \theta) = \beta_k + u^{-1} E(r_k \sigma_k^{-2} x(y - x^{\mathrm{T}} \beta_k)).$$

Thus, we have

$$\begin{aligned}
&\big\|\gamma_k - \bar{\gamma}_k\big\|_\infty \\
&= \Big\| u^{-1} \frac{1}{n} \sum_{i=1}^{n} r_{ik} \sigma_k^{-2} (x_i y_i - x_i x_i^{\mathrm{T}} \beta_k) - u^{-1} E(r_k \sigma_k^{-2} x(y - x^{\mathrm{T}} \beta_k)) \Big\|_\infty \\
&\le u^{-1} \Big\| \frac{1}{n} \sum_{i=1}^{n} r_{ik} \sigma_k^{-2} x_i y_i - E(r_k \sigma_k^{-2} xy) \Big\|_\infty + u^{-1} \Big\| \frac{1}{n} \sum_{i=1}^{n} r_{ik} \sigma_k^{-2} x_i x_i^{\mathrm{T}} \beta_k - E(r_k \sigma_k^{-2} x x^{\mathrm{T}} \beta_k) \Big\|_\infty .
\end{aligned} \tag{B.22}$$

We first work on the first term of (B.22). For any $\lambda > 0$

$$\begin{aligned}
& E\big[\exp\big\{ \lambda \big\| \frac{1}{n} \sum_{i=1}^{n} r_{ik} \sigma_k^{-2} x_i y_i - E(r_k \sigma_k^{-2} xy) \big\|_\infty \big\} \big] && \text{(B.23)} \\
&= E\big[\max_{j \in \{1, \ldots, p\}} \exp\big\{ \lambda \mid \frac{1}{n} \sum_{i=1}^{n} r_{ik} \sigma_k^{-2} x_{ij} y_i - E(r_k \sigma_k^{-2} x_j y) \mid \big\} \big] \\
&\le \sum_{j=1}^{p} E\big[\exp\big\{ \lambda \mid \frac{1}{n} \sum_{i=1}^{n} r_{ik} \sigma_k^{-2} x_{ij} y_i - E(r_k \sigma_k^{-2} x_j y) \mid \big\} \big] \\
&\le \sum_{j=1}^{p} E\big[\exp\big\{ 2\lambda \mid \frac{1}{n} \sum_{i=1}^{n} \xi_i r_{ik} \sigma_k^{-2} x_{ij} y_i \mid \big\} \big], && \text{(B.24)}
\end{aligned}$$

where the last inequality is implied by Lemma B.3 with $\xi_1, \ldots, \xi_n$ being the independent Rademacher random variables.

Since $r_{ik} \in [0, 1]$, the Ledoux-Talagrand contraction for Rademacher processes [65] further implies that

$$E\Big[\exp\Big\{ 2\lambda \mid \frac{1}{n} \sum_{i=1}^{n} \xi_i r_{ik} \sigma_k^{-2} x_{ij} y_i \mid \Big\} \Big] \le E\Big[\exp\Big\{ 4\lambda \mid \frac{1}{n} \sum_{i=1}^{n} \xi_i \sigma_k^{-2} x_{ij} y_i \mid \Big\} \Big]. \tag{B.25}$$

By model (3.1), y_i can be expressed as

$$y_i = \sum_{k=1}^{K} z_{ik}(x_i^{\mathrm{T}}\beta_k^* + \varepsilon_k),$$

where z_{ik} is a binary indicator with $z_{ik} = 1$ if the ith observation belongs to class k and $z_{ik} = 0$ otherwise, and $\varepsilon_k \sim \mathcal{N}(0, \sigma_k^{2*})$ is a random term independent of z_{ik}. Since ε_k is a zero mean normal random variable with variance σ_k^{2*}, by Example 5.8 in [34], we have $\|\varepsilon_k\|_{\Psi_2} \leq c_1\sigma_k^*$ for some constant $c_1 > 0$. Also, by Assumption 3, $| x_{ij} | \leq \omega_4$ for some $\omega_4 > 0$. Thus, we have

$$\|x_i^{\mathrm{T}}\beta_k^* + \varepsilon_k\|_{\Psi_2} \leq \|x_i^{\mathrm{T}}\beta_k^*\|_{\Psi_2} + \|\varepsilon_k\|_{\Psi_2} \leq \omega_4\|\beta_k^*\|_1 + c_1\sigma_k^*.$$

This implies that

$$\|y_i\|_{\Psi_2} = \big\| \sum_{k=1}^{K} z_{ik}(x_i^{\mathrm{T}}\beta_k^* + \varepsilon_k)\big\|_{\Psi_2} \leq \big\| \max_k \{ \mid x_i^{\mathrm{T}}\beta_k^* + \varepsilon_k \mid \}\big\|_{\Psi_2} \leq \max_k \{\omega_4\|\beta_k^*\|_1 + c_1\sigma_k^*\}.$$

Since ξ_i is a Rademacher random variable independent of x_{ij} and y_i, by Lemma B.2, we further have

$$\|\xi_i x_{ij} y_i\|_{\Psi_1} = \|x_{ij} y_i\|_{\Psi_1} \leq \max\{\omega_4^2, \max_k(\omega_4\|\beta_k^*\|_1 + c_1\sigma_k^*)^2\} \leq c_2\vartheta_k,$$

where $c_2 = \max\{\omega_4^2, c_1^2\}$ and $\vartheta_k = \max\{1, \max_k(\|\beta_k^*\|_1 + \sigma_k^*)^2\}$. Therefore, the right-hand side of (B.25) is bounded by

$$\begin{aligned}
&E\big[\exp\{4\lambda|\frac{1}{n}\sum_{i=1}^{n}\xi_i\sigma_k^{-2}x_{ij}y_i|\}\big] \\
&\leq E\big[\max\{\exp\big(4\lambda\frac{1}{n}\sum_{i=1}^{n}\xi_i\sigma_k^{-2}x_{ij}y_i\big), \exp\big(-4\lambda\frac{1}{n}\sum_{i=1}^{n}\xi_i\sigma_k^{-2}x_{ij}y_i\big)\}\big] \\
&\leq E\big[\exp\big(4\lambda\frac{1}{n}\sum_{i=1}^{n}\xi_i\sigma_k^{-4}x_{ij}y_i\big)\big] + E\big[\exp\big(-4\lambda\frac{1}{n}\sum_{i=1}^{n}\xi_i\sigma_k^{-2}x_{ij}y_i\big)\big]. \qquad \text{(B.26)}
\end{aligned}$$

Note that $E(\xi_i x_{ij} y_i) = E(\xi_i)E(x_{ij}y_i) = 0$. By Lemma 5.15 of [34], when $\lambda \leq c_3\sigma_k^2 n/\vartheta_k$ for some $c_3 > 0$, the first term of (B.26) is bounded by

$$E\big[\exp\big(4\lambda n^{-1}\xi_i\sigma_k^{-2}x_{ij}y_i\big)\big] \leq \exp\big(c_4\lambda^2\sigma_k^{-4}n^{-2}\vartheta_k^2\big), \qquad \text{(B.27)}$$

where $c_4 > 0$ is a constant. Following the same arguments, the second term of (B.26) is also bounded by (B.27). This indicates that (B.26) is bounded by

$$2\exp\left[c_4\lambda^2\sigma_k^{-4}n^{-1}\vartheta_k^2\right]. \tag{B.28}$$

By (B.25) and (B.28), we obtained an upper bound of (B.23) as below.

$$E\Big[\exp\Big\{\lambda\Big\|\frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_iy_i - E(r_k\sigma_k^{-2}xy)\Big\|_\infty\Big\}\Big] \le 2p\exp\left[c_4\lambda^2\sigma_k^{-4}n^{-1}\vartheta_k^2\right]. \tag{B.29}$$

With (B.29), we derive a probability bound for the first term of (B.22) as follows. Let δ_1' be an arbitrary positive value. By Markov inequality, we have

$$\begin{aligned}
&\mathrm{pr}\Big[\Big\|\frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_iy_i - E(r_k\sigma_k^{-2}xy)\Big\|_\infty > \delta_1'\Big] \\
&= \mathrm{pr}\Big[\exp\Big\{\lambda\Big\|\frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_iy_i - E(r_k\sigma_k^{-2}xy)\Big\|_\infty\Big\} \ge \exp(\lambda\delta_1')\Big] \\
&\le \frac{E\Big[\exp\Big\{\lambda\big\|\frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_iy_i - E(r_k\sigma_k^{-2}xy)\big\|_\infty\Big\}\Big]}{\exp(\lambda\delta_1')} \\
&\le \frac{2p\exp\left[c_4\lambda^2\sigma_k^{-4}n^{-1}\vartheta_k^2\right]}{\exp(\lambda\delta_1')} \\
&= 2\exp\left[c_4\lambda^2\sigma_k^{-4}n^{-1}\vartheta_k^2 - \lambda\delta_1' + \log p\right],
\end{aligned} \tag{B.30}$$

where $\lambda \le c_3\sigma_k^2 n/\vartheta_k$. Especially, (B.30) holds for $\lambda = n\sigma_k^4\delta_1'/2c_4\vartheta_k^2$ with $\delta_1' \le 2c_3c_4\vartheta_k\sigma_k^{-2}$, we have

$$\mathrm{pr}\Big[\Big\|\frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_iy_i - E(r_k\sigma_k^{-2}xy)\Big\|_\infty > \delta_1'\Big] \le 2\exp\Big(\frac{-n\sigma_k^4\delta_1'^2}{4c_4\vartheta_k^2} + \log p\Big). \tag{B.31}$$

Denote the right-hand side of (B.31) by $\delta_1/2$; we can then express δ_1' in terms of δ_1. Therefore, (B.31) implies that

$$u^{-1}\Big\|\frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_iy_i - E(r_k\sigma_k^{-2}xy)\Big\|_\infty > 2c_4^{\frac{1}{2}}u^{-1}\sigma_k^{-2}\vartheta_k\Big(\frac{\log p - \log\left(\frac{\delta_1}{4}\right)}{n}\Big)^{\frac{1}{2}} \tag{B.32}$$

with probability at most $\delta_1/2$ for $\delta_1 \in (4\exp(-nc_3^2c_4 + \log p), \min\{2, 4p\})$. We have thus obtained a probability bound on the first term of (B.22).

We next work on the second term of (B.22). Let $A = \frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_i x_i^{\mathrm{T}} - E(r_k\sigma_k^{-2}xx^{\mathrm{T}})$ be a $p \times p$ random matrix. We use

$$a_{jl} = \frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_{ij}x_{il} - E(r_k\sigma_k^{-2}x_j x_l^{\mathrm{T}})$$

to denote the (j,l)th element of A for $j,l \in \{1,\ldots,p\}$. Then, we have

$$\big\|\frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_i x_i^{\mathrm{T}}\beta_k - E(r_k\sigma_k^{-2}xx^{\mathrm{T}}\beta_k)\big\|_\infty = \|A\beta_k\|_\infty \le a_{max}\|\beta_k\|_1, \tag{B.33}$$

where $a_{max} = \max_{\{j,l\}} \mid a_{jl} \mid$. Let δ_2' be an arbitrary positive value. By Proposition 5.16 of [34] (Bernstein-type inequality), we have

$$\mathrm{pr}\big(\mid a_{jl} \mid > \delta_2'\big) \le 2\exp\big(-c_5\omega_4^{-4}\sigma_k^4 n\delta_2'^2\big)$$

for some $c_5 > 0$. Consequently, by Bonferroni inequality, we have

$$\mathrm{pr}\big(a_{max} > \delta_2'\big) \le \sum_{j=1}^{p}\sum_{l=1}^{p}\mathrm{pr}\big(\mid a_{jl} \mid \ge \delta_2'\big) \le 2p^2\exp\big(-c_5\omega_4^{-4}\sigma_k^4 n\delta_2'^2\big). \tag{B.34}$$

Denote the right-hand side of (B.34) by $\delta_2/2$; we can express δ_2' in terms of δ_2. We then have

$$a_{max} > c_5^{\frac{1}{2}}2^{\frac{1}{2}}\sigma_k^{-2}\omega_4^2\Big(\frac{\log p - \log\frac{\delta_2}{4}}{n}\Big)^{\frac{1}{2}} \tag{B.35}$$

with probability at most $\delta_2/2$ for $\delta_2 \in (0,2)$. By condition of the proposition, $\|\beta_k\|_1$ is bounded by

$$\|\beta_k\|_1 \le \|\beta_k\|_0^{\frac{1}{2}}\|\beta_k\|_2 \le \|\beta_k\|_0^{\frac{1}{2}}(\|\beta_k^*\|_2 + \|\beta_k - \beta_k^*\|_2) \le (1+w)\|\beta_k\|_0^{\frac{1}{2}}\|\beta_k^*\|_2.$$

This inequality together with (B.33) and (B.35) indicates that

$$u^{-1}\big\|\frac{1}{n}\sum_{i=1}^{n} r_{ik}\sigma_k^{-2}x_i x_i^{\mathrm{T}}\beta_k - E(r_k\sigma_k^{-2}xx^{\mathrm{T}}\beta_k)\big\|_\infty > c_5^{\frac{1}{2}}2^{\frac{1}{2}}(1+w)u^{-1}\sigma_k^{-2}\omega_4^2\|\beta_k\|_0^{\frac{1}{2}}\|\beta_k^*\|_2\Big(\frac{\log p - \log\frac{\delta_2}{4}}{n}\Big)^{\frac{1}{2}} \tag{B.36}$$

with probability at most $\delta_2/2$. We have thus obtained a probability bound on the second term of (B.22).

Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be the events indicated in (B.32) and (B.36), respectively. We then have

$$\begin{aligned}\mathrm{pr}(\mathcal{E}_1^c \cap \mathcal{E}_2^c) &= 1 - \mathrm{pr}(\mathcal{E}_1 \cup \mathcal{E}_2) \\ &\geq 1 - \mathrm{pr}(\mathcal{E}_1) - \mathrm{pr}(\mathcal{E}_2) \\ &\geq 1 - \delta_1/2 - \delta_2/2 \\ &\geq 1 - \delta\end{aligned}$$

with $\delta = \max\{\delta_1, \delta_2\} \in (4\exp(-C_4 n + \log p), 1)$ and $C_4 = c_3^2 c_4$.

When $\mathcal{E}_1^c$ and $\mathcal{E}_2^c$ both occur, inequality (B.22) together with (B.32) and (B.36) implies that

$$\|\gamma_k - \bar{\gamma}_k\|_\infty \leq C_5 u^{-1} \sigma_k^{-2} \big(\vartheta_k + \|\beta_k\|_0^{\frac{1}{2}} \|\beta_k^*\|_2\big) \Big(\frac{\log p - \log\big(\frac{\delta}{4}\big)}{n}\Big)^{\frac{1}{2}},$$

where $C_5 = \max\{2c_4^{1/2}, (2c_5)^{1/2}(1+w)\omega_4^2\}$. This completes the proof of Proposition 1. □

Proposition 1 indicates that, when β_k is not far away from β_k^*, the distance between γ_k and $\bar{\gamma}_k$ is well controlled as $n \to \infty$. This result is helpful in obtaining an error bound for $\|\beta_k^{(t)} - \beta_k^*\|_2$, which is a core part in the proof of Theorem 3.3.

B.4 Proof of Theorem 3.3

Proof of Theorem 3.3: Note that

$$\begin{aligned}\mathrm{pr}\big(\mathcal{M}_k^* \subset \widehat{\mathcal{M}}_k^{(t)}\big) &\geq \mathrm{pr}\big(\min_{j \in \mathcal{M}_k^*} \mid \beta_{kj}^{(t)} \mid > 0\big) \\ &\geq \mathrm{pr}\big(\min_{j \in \mathcal{M}_k^*} \mid \beta_{kj}^* \mid - \max_{j \in \mathcal{M}_k^*} (\mid \beta_{kj}^{(t)} - \beta_{kj}^* \mid) > 0\big) \\ &\geq \mathrm{pr}\big(\max_{j \in \mathcal{M}_k^*} \mid \beta_{kj}^{(t)} - \beta_{kj}^* \mid < \omega_1 n^{-\kappa_1}\big), \qquad \text{(B.37)}\end{aligned}$$

where the last inequality is based on Assumption 2. Since

$$\max_{j \in \mathcal{M}_k^*} \mid \beta_{kj}^{(t)} - \beta_{kj}^* \mid \leq \big\|\beta_k^{(t)} - \beta_k^*\big\|_\infty \leq \big\|\beta_k^{(t)} - \beta_k^*\big\|_2,$$

it suffices to show that

$$\big\|\beta_k^{(t)} - \beta_k^*\big\|_2 < \omega_1 n^{-\kappa_1} \qquad \text{(B.38)}$$

with an overwhelming probability. Recall that

$$\beta_k^{(t+1)} = \arg\max_{\beta_k} h(\beta_k; \theta^{(t)}), \quad \text{subject to } \|\beta_k\|_0 \leq s_k \qquad \text{(B.39)}$$

and

$$\widehat{\mathcal{M}}_k^{(t)} = \{j : \beta_{kj}^{(t)} \neq 0, 1 \leq j \leq p\}.$$

Let

$$\gamma_k^{(t)} = \arg\max_{\gamma_k} h(\gamma_k; \theta^{(t)}).$$

Since $h(\beta_k; \theta^{(t)})$ is a simple quadratic function in β_k, solving (B.39) amounts to choosing the largest s_k components in $\gamma_k^{(t)}$. That is, $\beta_k^{(t+1)} = H_{s_k}(\gamma_k^{(t)})$.

For an arbitrary p-dimensional vector $\gamma = (\gamma_1, \ldots, \gamma_p)$, we use

$$H(\gamma, \widehat{\mathcal{M}}_k^{(t)}) = \begin{cases} \gamma_j, \ j \in \widehat{\mathcal{M}}_k^{(t)} \\ 0, \ \text{otherwise} \end{cases}$$

to denote a truncating operator on γ with respect to $\widehat{\mathcal{M}}_k^{(t)}$.

In order to show (B.38), we derive an upper bound of $\|\beta_k^{(t)} - \beta_k^*\|_2$, which consists of two types of error terms. Specifically, let $\bar{\gamma}_k^{(t)} = \arg\max_{\gamma_k} h_p(\gamma_k; \theta^{(t)})$ with h_p defined in Proposition 1. Later, we will show that, for a given iteration number T and all $t \in \{1, \ldots, T\}$,

$$\left\|\beta_k^{(t)} - \beta_k^*\right\|_2 \leq C_6 s_k \big(\vartheta_k + \|\beta_k^*\|_2\big) \Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} + \eta^{\frac{t}{2}} \left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \tag{B.40}$$

with probability at least $1 - \delta$, where $C_6 > 0$, $\eta \in (0, 1)$ and $\delta \in (4\exp(-C_4 n + \log p), 1)$.

By Assumption 1, when n is large enough, $\log p \leq d_0 n^a$ for some non-negative constant d_0. By setting $\delta = 4s_k^* \exp(-C_4 n^a)$, we have

$$\begin{aligned} C_6 s_k \big(\vartheta_k + \|\beta_k^*\|_2\big) \Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} &> C_6 s_k^* \|\beta_k^*\|_2 \Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} \\ &\geq C_6 d_1 \omega_1 (s_k^*)^{\frac{1}{2}} (\log p)^{\frac{1}{2}} n^{\frac{-2\kappa_1 - 1}{2}}, \end{aligned} \tag{B.41}$$

where $d_1 = (C_4/d_0)^{1/2}$. Since we require $\|\beta_k^{(0)} - \beta_k^*\|_2 \leq \frac{\omega_1}{8} (\log p)^{\frac{1}{2}} n^{-\kappa_1}$ and $T \geq C_3 \log\Big(C_2^{-1} (n/s_k^*)^{\frac{1}{2}}\Big)$, the second term of (B.40) is bounded by (B.41) with $C_2 = d_1^{-1} C_6^{-1}$ and $C_3 = 2\log^{-1}(\eta^{-1})$. Hence, when n is large, (B.40) is dominated by its first term.

By Assumption 2, we have

$$\vartheta_k + \|\beta_k^*\|_2 \leq (1 + \omega_3^{-1})^2 \max_k \left\{\|\beta_k^*\|_1^2\right\} + \|\beta_k^*\|_2 \leq c_6 \max_k s_k^{2*} n^{2\kappa_2},$$

where $c_6 = \omega_2^2(1+\omega_3^{-1})^2 + \omega_2$. This together with Assumption 3 implies that

$$C_6 s_k\big(\vartheta_k + \|\beta_k^*\|_2\big)\Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} \le C_6 c_6 d_2 \max_k s_k^{2*} n^{2\kappa_2+(a-1)/2} \le C_6\omega_7^3 c_6 d_2 n^{3\kappa_3+2\kappa_2+(a-1)/2} \tag{B.42}$$

with $d_2 = (d_0 + C_4)^{1/2}$. As $\kappa_1 + 2\kappa_2 + 3\kappa_3 < (1-a)/2$, (B.42) further implies that, when n is large enough,

$$\big\|\beta_k^{(T)} - \beta_k^*\big\|_2 < \omega_1 n^{-\kappa_1}$$

with probability at least $1 - 4s_k^* \exp(-C_4 n^a)$. Property (B.38) is therefore justified and the sure screening of $\widehat{\mathcal{M}}_k^{(T)}$ is ensured.

With the discussion above, we complete the proof by showing (B.40) via mathematical induction. To this end, we first verify (B.40) when $t = 1$. By Triangle inequality, we have

$$\big\|\beta_k^{(1)} - \beta_k^*\big\|_2 \le \big\|H_{s_k}(\gamma_k^{(0)}) - \beta_k^*\big\|_2 \le \big\|H_{s_k}(\gamma_k^{(0)}) - H(\bar\gamma_k^{(0)}, \widehat{\mathcal{M}}_k^{(1)})\big\|_2 + \big\|H(\bar\gamma_k^{(0)}, \widehat{\mathcal{M}}_k^{(1)}) - \beta_k^*\big\|_2. \tag{B.43}$$

For the first term of the right-hand side of (B.43), we have

$$\big\|H_{s_k}(\gamma_k^{(0)}) - H(\bar\gamma_k^{(0)}, \widehat{\mathcal{M}}_k^{(1)})\|_2 = s_k^{\frac{1}{2}}\big\|H_{s_k}(\gamma_k^{(0)}) - H(\bar\gamma_k^{(0)}, \widehat{\mathcal{M}}_k^{(1)})\big\|_\infty \le s_k^{\frac{1}{2}}\big\|\gamma_k^{(0)} - \bar\gamma_k^{(0)}\big\|_\infty. \tag{B.44}$$

We now work on the second term of the right-hand side of(B.43). To ease the presentation, denote $\lambda_{min}(A)$ and $\lambda_{max}(A)$ the smallest and largest eigenvalues of an arbitrary positive definite matrix A, respectively. Let $\lambda_1 = \lambda_{min}(\Sigma)$ and $\lambda_2 = \lambda_{max}(\Sigma)$. By definition, we have

$$\begin{aligned}
&\big\|\bar\gamma_k^{(0)} - \beta_k^*\big\|_2 \\
&= \big\|\beta_k^{(0)} + u^{-1} g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^{(0)}) - \beta_k^*\big\|_2 \\
&\le \big\|\beta_k^{(0)} + u^{-1} g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^*) - \beta_k^*\big\|_2 + u^{-1}\big\|g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^{(0)}) - g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^*)\big\|_2,
\end{aligned} \tag{B.45}$$

and

$$\begin{aligned}
g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^*) &= E\big\{r_k^* \sigma_k^{-2(0)} x(y - x^{\mathrm{T}}\beta_k^{(0)})\big\} \\
&= E\big\{r_k^* \sigma_k^{-2(0)} x x^{\mathrm{T}}(\beta_k^* - \beta_k^{(0)})\big\} + E\big\{r_k^* \sigma_k^{-2(0)} x x^{\mathrm{T}} \varepsilon_k\big\} \\
&= \sigma_k^{-2(0)} E(r_k^* x x^{\mathrm{T}})(\beta_k^* - \beta_k^{(0)}),
\end{aligned} \tag{B.46}$$

Also, since $r_k^* \in (0,1]$, we have $\lambda_{max}\{E(r_k^* xx^{\mathrm{T}})\} \leq \lambda_2$. Following similar strategy of A.2 of [63], we have $\lambda_{min}\{E(r_k^* xx^{\mathrm{T}})\} \geq \lambda_1 - d\lambda_2$ with $d \in (0, \lambda_1/\lambda_2)$. Thus, $g_p(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^*)$ is $\sigma_k^{-2(0)}(\lambda_1 - d\lambda_2)$-strongly concave and $\sigma_k^{-2(0)}\lambda_2$-smooth. More details of condition of strongly concave and smooth function can be found in Condition 2 and 3 of [3] or Condition 3.2 of [106]. By Theorem 3.7 of [12], which is a classical result that ensures linear convergence of gradient ascent when applied to a smooth and strongly concave function, we have

$$
\begin{aligned}
&\left\|\beta_k^{(0)} + u^{-1} g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^*) - \beta_k^*\right\|_2^2 \\
&\leq \left(1 - 2u^{-1}\sigma_k^{-2(0)} \frac{\lambda_1\lambda_2 - d\lambda_2^2}{\lambda_1 + (1-d)\lambda_2}\right)\left\|\beta_k^{(0)} - \beta_k^*\right\|_2^2 + \left(u^{-2} - \frac{2u^{-1}\sigma_k^{2(0)}}{\lambda_1 + (1-d)\lambda_2}\right)\sigma_k^{-4(0)}\lambda_2^2\left\|\beta_k^{(0)} - \beta_k^*\right\|_2^2 \\
&= \left(1 - u^{-1}\sigma_k^{-2(0)}\lambda_2\right)^2\left\|\beta_k^{(0)} - \beta_k^*\right\|_2^2. \qquad \text{(B.47)}
\end{aligned}
$$

(B.47) implies that the first term of (B.45) is upper bounded by

$$
\left\|\beta_k^{(0)} + u^{-1} g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^*) - \beta_k^*\right\|_2 \leq \mid 1 - u^{-1}\sigma_k^{-2(0)}\lambda_2 \mid \left\|\beta_k^{(0)} - \beta_k^*\right\|_2. \qquad \text{(B.48)}
$$

By (B.46), the second term of (B.45) can be rewritten as

$$
u^{-1}\left\|g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^{(0)}) - g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^*)\right\|_2 = u^{-1}\sigma_k^{-2(0)}\left\|E\{ \mid r_k^{(0)} - r_k^* \mid xx^{\mathrm{T}}\}(\beta_k^* - \beta_k^{(0)})\right\|_2. \qquad \text{(B.49)}
$$

Remark A.1 of [63] further indicates that

$$
\mid r_k^{(0)} - r_k^* \mid \leq \alpha, \qquad \text{(B.50)}
$$

where $\alpha \in (0, 1/4)$. (B.49) together with (B.50) implies that

$$
u^{-1}\left\|g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^{(0)}) - g_p'(\beta_k^{(0)}, \sigma_k^{2(0)}; \theta^*)\right\|_2 \leq u^{-1}\sigma_k^{-2(0)}\alpha\lambda_2\left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \qquad \text{(B.51)}
$$

Since we require $u\sigma_k^{2(0)} > \lambda_2$, we have $1 - u^{-1}\sigma_k^{-2(0)}\lambda_2 > 0$. Plugging (B.48) and (B.51) into (B.45), we further have

$$
\left\|\bar{\gamma}_k^{(0)} - \beta_k^*\right\|_2 \leq \eta'\left\|\beta_k^{(0)} - \beta_k^*\right\|_2,
$$

where $\eta' = 1 - (1-\alpha)u^{-1}\sigma_k^{-2(0)}\lambda_2$. Straightforward computations give the claim that $\eta' \in (0,1)$. To further simplify the statement, we require the step size $u \leq C_1^{-1}\sigma_k^{-2(0)}\omega_5$ with $C_1^{-1} \in (\alpha, 1)$. This requirement of u together with Assumption 3 indicates that $\eta' \leq \eta = 1 - C_1 + \alpha \leq 1$ and

$$
\left\|\bar{\gamma}_k^{(0)} - \beta_k^*\right\|_2 \leq \eta'\left\|\beta_k^{(0)} - \beta_k^*\right\|_2, \qquad \text{(B.52)}
$$

By the requirement of $\beta_k^{(0)}$ in Theorem 3.3 and Assumption 1, we have

$$\left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \leq \frac{\omega_1}{8}(\log p)^{\frac{1}{2}} n^{-\kappa_1} \leq \frac{1}{8} p^{\frac{1}{2}} \min_{j \in \mathcal{M}_k^*} \mid \beta_{kj}^* \mid \leq \frac{1}{8}\left\|\beta_k^*\right\|_2. \tag{B.53}$$

(B.53) shows that $\beta_k^{(0)}$ falls in a neighborhood of β_k^*. Thus, (B.52) together with (B.53) implies that

$$\left\|\bar{\gamma}_k^{(0)} - \beta_k^*\right\|_2 \leq \frac{\eta}{8}\left\|\beta_k^*\right\|_2. \tag{B.54}$$

Clearly, $\bar{\gamma}_k^{(0)}$ falls into a small neighborhood of β_k^*, so that Lemma 5.1 of [106] becomes applicable. Thus, we have

$$\left\|H_{s_k}(\bar{\gamma}_k^{(0)}, \widehat{\mathcal{M}}_k^{(1)}) - \beta_k^*\right\|_2 \leq c_6 s_k^{*\frac{1}{2}} \left\|\gamma_k^{(0)} - \bar{\gamma}_k^{(0)}\right\|_\infty + \left\{1 + 4(s_k^*/s_k)^{\frac{1}{2}}\right\}^{\frac{1}{2}} \left\|\bar{\gamma}_k^{(0)} - \beta_k^*\right\|_2. \tag{B.55}$$

Plugging (B.44), (B.52) and (B.55) into (B.43) further implies that

$$\begin{aligned}
\left\|\beta_k^{(1)} - \beta_k^*\right\|_2 &\leq (s_k^{\frac{1}{2}} + c_6 s_k^{*\frac{1}{2}}) \left\|\gamma_k^{(0)} - \bar{\gamma}_k^{(0)}\right\|_\infty + \left\{1 + 4(s_k^*/s_k)^{\frac{1}{2}}\right\}^{\frac{1}{2}} \left\|\bar{\gamma}_k^{(0)} - \beta_k^*\right\|_2 \\
&\leq c_7 s_k^{\frac{1}{2}} \left\|\gamma_k^{(0)} - \bar{\gamma}_k^{(0)}\right\|_\infty + \left\{1 + 4(s_k^*/s_k)^{\frac{1}{2}}\right\}^{\frac{1}{2}} \eta \left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \\
&\leq c_7 s_k^{\frac{1}{2}} \left\|\gamma_k^{(0)} - \bar{\gamma}_k^{(0)}\right\|_\infty + \eta^{\frac{1}{2}} \left\|\beta_k^{(0)} - \beta_k^*\right\|_2
\end{aligned}$$

with $c_7 = c_6 + 1$, where the second inequality holds from the assumption that $s_k \geq s_k^*$ and last inequality is from the assumption of s_k that

$$s_k \geq \frac{16 s_k^*}{(\eta^{-1} - 1)^2}.$$

By the requirement of u in Theorem 3.3 and (B.53), Proposition 1 implies that

$$\left\|\beta_k^{(1)} - \beta_k^*\right\|_2 \leq c_7 C_5 s_k^{\frac{1}{2}} u^{-1} \sigma_k^{-2(0)} \left(\vartheta_k + s_k^{\frac{1}{2}} \|\beta_k^*\|_2\right) \left(\frac{\log p - \log(\delta/4)}{n}\right)^{\frac{1}{2}} + \eta^{\frac{1}{2}} \left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \tag{B.56}$$

$$\leq C_6 s_k \left(\vartheta_k + \|\beta_k^*\|_2\right) \left(\frac{\log p - \log(\delta/4)}{n}\right)^{\frac{1}{2}} + \eta^{\frac{1}{2}} \left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \tag{B.57}$$

with probability at least $1 - \delta/T$, where $\delta/T \in (4\exp(-C_4 n + \log p), 1)$, $C_6 = c_7 \omega_6^{-1} C_5/(1 - \eta^{1/2}) > 0$ and the last inequality is from the fact that $1 - \eta^{1/2} < 1$. Since $T \geq 1$, this verifies (B.40) when $t = 1$. Also, since we require $\mid \sigma_k^{2(0)} - \sigma_k^{2*} \mid \leq \omega_8 \sigma_k^{2*}$. Following the similar decomposition strategy of B.3 in [63], we have

$$\mid \sigma_k^{2(1)} - \sigma_k^{2*} \mid \leq \omega_9 \sigma_k^{2*}. \tag{B.58}$$

We next show that (B.40) holds at the tth iteration with the assumption that it is valid at the $(t-1)$th iteration for $1 < t \leq T$. By (B.40), we have

$$\begin{aligned}\big\|\beta_k^{(t-1)} - \beta_k^*\big\|_2 &\leq C_6 s_k\big(\vartheta_k + \|\beta_k^*\|_2\big)\Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} + \eta^{\frac{t-1}{2}}\big\|\beta_k^{(0)} - \beta_k^*\big\|_2 \\ &\leq \frac{1}{8}(1-\eta^{\frac{1}{2}})\big\|\beta_k^*\big\|_2 + \frac{1}{8}\eta^{\frac{1}{2}}\big\|\beta_k^*\big\|_2 = \frac{1}{8}\big\|\beta_k^*\big\|_2, \end{aligned} \tag{B.59}$$

where the last inequality is from (B.42) and $\eta^{(t-1)/2} \leq \eta^{1/2}$. (B.59) implies that $\beta_k^{(t-1)}$ also falls into the neighborhood of β_k^*. Similar to (B.58), we have $\mid \sigma_k^{2(t-1)} - \sigma_k^{2*} \mid \leq \omega_8 \sigma_k^{2*}$. Following the similar strategy of (B.45)-(B.52), we then have

$$\big\|\bar{\gamma}_k^{(t-1)} - \beta_k^*\big\|_2 \leq \eta\big\|\beta_k^{(t-1)} - \beta_k^*\big\|_2 \leq \frac{\eta}{8}\big\|\beta_k^*\big\|_2.$$

Similar to (B.56), by taking union bound we have

$$\begin{aligned}\big\|\beta_k^{(t)} - \beta_k^*\big\|_2 &\leq c_7\omega_6^{-1}C_5 s_k\big(\vartheta_k + \|\beta_k^*\|_2\big)\Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} + \eta^{\frac{1}{2}}\big\|\beta_k^{(t-1)} - \beta_k^*\big\|_2 \\ &\leq \Big(1 + \frac{\eta^{\frac{1}{2}}}{1-\eta^{\frac{1}{2}}}\Big)c_7\omega_6^{-1}C_5 s_k\big(\vartheta_k + \|\beta_k^*\|_2\big)\big(\frac{\log p - \log(\delta/4)}{n}\big)^{\frac{1}{2}} + \eta^{\frac{t}{2}}\big\|\beta_k^{(0)} - \beta_k^*\big\|_2 \\ &= C_6 s_k\big(\vartheta_k + \|\beta_k^*\|_2\big)\Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} + \eta^{\frac{t}{2}}\big\|\beta_k^{(0)} - \beta_k^*\big\|_2 \end{aligned} \tag{B.60}$$

with probability at least $1 - \delta/T$, where $\delta/T \in (4\exp(-C_4 n + \log p), 1)$. This verifies (B.40) at the tth iteration for $1 < t \leq T$. (B.57) and (B.60) together implies (B.40) holds for any $t \in \{1, \ldots, T\}$. This completes the proof of Theorem 3.3. □

B.5 Proof of Corollary 3.1

Proof of Corollary 3.1: To prove the corollary, it suffice to show that

$$\mathrm{pr}\big(\big\|\beta_k^{(t)} - \beta_k^*\big\|_2 < \omega_1 n^{-\kappa_1}\big) \to 1, \text{ as } n \to \infty \tag{B.61}$$

for any $t \geq 1$. We use similar strategy in Theorem 3.3 to prove (B.61). To use the tricks in Theorem 3.3, we firstly need to check conditions (B.53) and (B.54). By the requirement of $\beta_k^{(0)}$ and $\kappa_1 + 2\kappa_2 + 3\kappa_3 < (1-a)/2$, we have

$$\big\|\beta_k^{(0)} - \beta_k^*\big\|_2 \leq \frac{\omega_1}{8}\Big(\frac{\log p}{n}\Big)^{\frac{1}{2}} < \frac{\omega_1}{8}n^{-(\kappa_1+2\kappa_2+3\kappa_3)} \leq \frac{\omega_1}{8}n^{-\kappa_1} \leq \frac{1}{8}s_k^{-\frac{1}{2}*}\big\|\beta_k^*\big\|_2.$$

Also, following similar arguments of (B.45)–(B.52), we have

$$\left\|\bar{\gamma}_k^{(0)} - \beta_k^*\right\|_2 \leq \eta \left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \leq \frac{\eta}{8} s_k^{-\frac{1}{2}*} \left\|\beta_k^*\right\|_2.$$

By mathematical induction, we then have, for any $t \geq 1$,

$$\left\|\beta_k^{(t)} - \beta_k^*\right\|_2 \leq C_6 s_k \big(\vartheta_k + \|\beta_k^*\|_2\big) \Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} + \eta^{\frac{t}{2}} \left\|\beta_k^{(0)} - \beta_k^*\right\|_2 \tag{B.62}$$

with probability $1 - \delta$. By setting $\delta = 4s_k^* \exp(-C_4 n^a)$, the first term of (B.62) is upper bounded by

$$C_6 s_k \big(\vartheta_k + \|\beta_k^*\|_2\big) \Big(\frac{\log p - \log(\delta/4)}{n}\Big)^{\frac{1}{2}} \leq C_6 \omega_7^3 c_6 d_2 n^{3\kappa_3 + 2\kappa_2 + (a-1)/2} < C_6 \omega_7^3 c_6 d_2 n^{-\kappa_1}. \tag{B.63}$$

As we require $\|\beta_k^{(0)} - \beta_k^*\|_2 < \omega_1 n^{-\kappa_1}/8$, (B.63) implies that, when n is sufficiently large and for any $T \geq 1$,

$$\left\|\beta_k^{(T)} - \beta_k^*\right\|_2 \leq \omega_1 n^{-\kappa_1}$$

with probability at least $1 - 4s_k^* \exp(-C_4 n^a)$. (B.61) is therefore justified and the sure screening of $\widehat{\mathcal{M}}_k^{(T)}$ is ensured. This completes the proof of corollary. □

Appendix C

Supplementary Information for Chapter 4

In this appendix, we provide the proofs of Theorems 4.1 and 4.2.

C.1 The proof of Theorem 4.1

We begin with statement of two lemmas that will be used in the proofs.

Lemma C.1. *Suppose* $\boldsymbol{X} \in \mathbb{R}^{n\times k}$ *and* $\boldsymbol{V} \in \mathbb{R}^{n\times m}$ *are two zero-mean sub-Gaussian matrices with parameters* $\frac{1}{n}\sigma_X^2$ *and* $\frac{1}{n}\sigma_V^2$*. Then for any fixed vector* $\boldsymbol{v}$ *and n=1 we have with high probability*

$$||(\boldsymbol{V}^T\boldsymbol{X} - E[\boldsymbol{V}^T\boldsymbol{X}])\boldsymbol{v}||_\infty \leq \sigma_X\sigma_V||\boldsymbol{v}||\log p.$$

Lemma C.2. *Consider a finite sequence* $\{\boldsymbol{X_k}\}$ *of independent, random, Hermitian matrices with dimension p. Assume that*

$$E\boldsymbol{X_k} = 0 \quad \textit{and} \quad \lambda_{\max}(\boldsymbol{X_k}) \leq R.$$

Define the random matrix

$$\boldsymbol{Y} = \sum_k \boldsymbol{X_k}.$$

Then

$$\sigma^2 = \sigma^2(\boldsymbol{Y}) = ||E(\boldsymbol{Y}^2)||,$$

and

$$E\lambda_{\max}(\boldsymbol{Y}) \leq \sqrt{2\sigma^2 \log p} + \frac{1}{3} R \log d.$$

Furthermore, for all $t \geq 0$,

$$P(\lambda_{\max}(\boldsymbol{Y}) \geq t) \leq p \exp\left(\frac{-t^2/2}{\sigma^2 + Rt/3}\right).$$

The proof of Theorem 4.1: We prove Theorem 1 by four steps.

Step 1. Establishing a relation between $||\boldsymbol{\beta^{(t)}} - \boldsymbol{\beta^*}||$ and $||\boldsymbol{\beta^{(t-1)}} - \boldsymbol{\beta^*}||$.

Recall that by Algorithm 4.1, the update of LGS at t-th iteration satisfies

$$(\frac{1}{n}\widetilde{\boldsymbol{X}}_{D_t}^T \widetilde{\boldsymbol{X}}_{D_t} + diag(\nu^{(t)})\boldsymbol{I})\boldsymbol{\beta^{(t)}} = (\frac{1}{n}\widetilde{\boldsymbol{X}}_{D_t}^T \widetilde{\boldsymbol{Y}}_{D_t} + diag(\nu^{(t)})\boldsymbol{\beta^{(t-1)}}). \tag{C.1}$$

Let $\Sigma_n^{(t)} = \frac{1}{n}\widetilde{\boldsymbol{X}}_{D_t}^T \widetilde{\boldsymbol{X}}_{D_t}$, $\boldsymbol{D_\nu^{(t)}} = \text{diag}(\nu^{(t)})$ and $\boldsymbol{b_n^{(t)}} = \frac{1}{n}\widetilde{\boldsymbol{X}}_{D_t}^T \widetilde{\boldsymbol{Y}}_{D_t}$. The update of $\boldsymbol{\beta^{(t)}}$ can then be rewritten as

$$\boldsymbol{\beta^{(t)}} = (\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1}(\boldsymbol{b_n^{(t)}} + \boldsymbol{D_\nu^{(t)}}\boldsymbol{\beta^{(t-1)}}). \tag{C.2}$$

From equation (C.2), we have

$$\begin{aligned} ||\boldsymbol{\beta^{(t)}} - \boldsymbol{\beta^*}|| &= ||(\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1}(\boldsymbol{b_n^{(t)}} + \boldsymbol{D_\nu^{(t)}}\boldsymbol{\beta^{(t-1)}}) - (\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1}(\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})\boldsymbol{\beta^*}|| \\ &\leq \lambda_{\max}(\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1}||(\boldsymbol{b_n^{(t)}} - \Sigma_n^{(t)}\boldsymbol{\beta^*}) + \boldsymbol{D_\nu^{(t)}}(\boldsymbol{\beta^{(t-1)}} - \boldsymbol{\beta^*})||. \end{aligned} \tag{C.3}$$

To estimate $||\boldsymbol{\beta^{(t)}} - \boldsymbol{\beta^*}||$, we first establish a relation between $\lambda_{\max}(\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1}$ and $\lambda_{\max}(\Sigma_N + \boldsymbol{D_\nu^{(t)}})^{-1}$ below. Note that

$$\begin{aligned} \lambda_{\max}(\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1} - \lambda_{\max}(\Sigma_N + \boldsymbol{D_\nu^{(t)}})^{-1} &\leq \lambda_{\max}((\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1} - (\Sigma_N + \boldsymbol{D_\nu^{(t)}})^{-1}) \\ &\leq \lambda_{\max}(\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1}\lambda_{\max}(\Sigma_N + \boldsymbol{D_\nu^{(t)}})^{-1}\lambda_{\max}(\Sigma_n^{(t)} - \Sigma_N), \end{aligned}$$

we have

$$\lambda_{\max}(\Sigma_n^{(t)} + \boldsymbol{D_\nu^{(t)}})^{-1}[1 - \lambda_{\max}(\Sigma_N + \boldsymbol{D_\nu^{(t)}})^{-1}\lambda_{\max}(\Sigma_n^{(t)} - \Sigma_N)] \leq \lambda_{\max}(\Sigma_N + \boldsymbol{D_\nu^{(t)}})^{-1}. \tag{C.4}$$

If

$$\frac{\lambda_{\max}(\Sigma_n^{(t)} - \Sigma_N)}{\lambda_{\min}(\Sigma_N + \boldsymbol{D_\nu^{(t)}})} \leq \frac{1}{2}, \tag{C.5}$$

we then can obtain

$$\begin{aligned}\lambda_{\max}(\Sigma_n^{(t)}+\boldsymbol{D_\nu^{(t)}})^{-1} &\leq \frac{\lambda_{\max}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})^{-1}}{[1-\lambda_{\max}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})^{-1}\lambda_{\max}(\Sigma_n^{(t)}-\Sigma_N)]} = \frac{1}{\frac{1}{\lambda_{\max}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})^{-1}}-\lambda_{\max}(\Sigma_n^{(t)}-\Sigma_N)} \\ &= \frac{1}{\lambda_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})-\lambda_{\max}(\Sigma_n^{(t)}-\Sigma_N)} = \frac{1}{\lambda_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}\frac{1}{1-\frac{\lambda_{\max}(\Sigma_n^{(t)}-\Sigma_N)}{\lambda_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}} \\ &\leq \frac{1}{\lambda_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}[1+2\frac{\lambda_{\max}(\Sigma_n^{(t)}-\Sigma_N)}{\lambda_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}] \\ &\leq \frac{1}{\lambda_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}+2\frac{\lambda_{\max}(\Sigma_n^{(t)}-\Sigma_N)}{\lambda^2_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}.\end{aligned} \tag{C.6}$$

This first inequality holds by inequality (C.4) and the second inequality holds by combining the fact that $\frac{1}{1-x} < 1+2x$ for any $0 < x < \frac{1}{2}$ and inequality (C.5). The inequalities (C.3) and (C.6) thus imply that

$$||\boldsymbol{\beta^{(t)}}-\boldsymbol{\beta^*}|| \leq [\frac{1}{\lambda_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}+2\frac{\lambda_{\max}(\Sigma_n^{(t)}-\Sigma_N)}{\lambda^2_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}]||(\boldsymbol{b_n^{(t)}}-\Sigma_n^{(t)}\boldsymbol{\beta^*})+\boldsymbol{D_\nu^{(t)}}(\boldsymbol{\beta^{(t-1)}}-\boldsymbol{\beta^*})||.$$

That is,

$$||\boldsymbol{\beta^{(t)}}-\boldsymbol{\beta^*}|| \leq \Delta^{(t)}||(\boldsymbol{b_n^{(t)}}-\Sigma_n^{(t)}\boldsymbol{\beta^*})+\boldsymbol{D_\nu^{(t)}}(\boldsymbol{\beta^{(t-1)}}-\boldsymbol{\beta^*})||, \tag{C.7}$$

where

$$\Delta^{(t)} = \frac{1}{\lambda_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}+2\frac{\lambda_{\max}(\Sigma_n^{(t)}-\Sigma_N)}{\lambda^2_{\min}(\Sigma_N+\boldsymbol{D_\nu^{(t)}})}. \tag{C.8}$$

The inequality (C.7) shows the discrepancy between two steps of update. Apparently, such discrepancy depends on two factors, one is the similarity of Hessian matrices between sampled subset $\widetilde{\mathcal{D}}_t$ and the whole dataset $\widetilde{\mathcal{D}}$, and the other is the prediction deviation of $\boldsymbol{\beta^*}$ at current sampled subset $\widetilde{\mathcal{D}}_t$. We will show a sufficient condition of (A.5) in Step 4.

Step 2. Concluding an upper bound of $||\boldsymbol{b_n}-\Sigma_n\boldsymbol{\beta^*}||$.

With reference to (4.2), we have $||\boldsymbol{UW}|| \leq \epsilon||\boldsymbol{W}||$,$||\boldsymbol{UO}|| \leq \epsilon||\boldsymbol{O}||$. For simplicity of presentation, we write $\boldsymbol{UW}$ as $\overline{\boldsymbol{W}}$, $\boldsymbol{UO}$ as $\overline{\boldsymbol{O}}$ and assume the subset $\widetilde{\mathcal{D}}_s$ is generated from

a given $\boldsymbol{\beta}$. Note that

$$\begin{aligned}||\boldsymbol{b_n} - \boldsymbol{\Sigma_n}\boldsymbol{\beta^*}|| &= ||(\boldsymbol{S}\widetilde{\boldsymbol{X}})^T(\boldsymbol{S}\widetilde{\boldsymbol{Y}}) - (\boldsymbol{S}\widetilde{\boldsymbol{X}})^T(\boldsymbol{S}\widetilde{\boldsymbol{X}})\boldsymbol{\beta^*}|| \\ &= ||[\boldsymbol{S}(\boldsymbol{X} + a\overline{\boldsymbol{W}})]^T\boldsymbol{S}(\boldsymbol{X}\boldsymbol{\beta^*} + \boldsymbol{\xi} + b\overline{\boldsymbol{O}}) - [\boldsymbol{S}(\boldsymbol{X} + a\overline{\boldsymbol{W}})]^T\boldsymbol{S}(\boldsymbol{X} + a\overline{\boldsymbol{W}})\boldsymbol{\beta^*}|| \\ &\leq ||(\boldsymbol{S}\boldsymbol{X})^T\boldsymbol{S}\boldsymbol{\xi}|| + ||a(\boldsymbol{S}\overline{\boldsymbol{W}})^T\boldsymbol{S}\boldsymbol{\xi}|| + ||b(\boldsymbol{S}\boldsymbol{X})^T\boldsymbol{S}\overline{\boldsymbol{O}}|| + ||ab(\boldsymbol{S}\overline{\boldsymbol{W}})^T\boldsymbol{S}\overline{\boldsymbol{O}}|| \\ &\quad + ||a(\boldsymbol{S}\boldsymbol{X})^T\boldsymbol{S}\overline{\boldsymbol{W}}\boldsymbol{\beta^*}|| + ||a^2(\boldsymbol{S}\overline{\boldsymbol{W}})^T\boldsymbol{S}\overline{\boldsymbol{W}}\boldsymbol{\beta^*}||,\end{aligned} \tag{C.9}$$

where $\boldsymbol{S} = \sqrt{\frac{1}{n}\boldsymbol{P}\boldsymbol{D}}$, $\boldsymbol{P}$ is the $n \times N$ subsampling matrix, $\boldsymbol{D} = \text{diag}\{d_i\}$ and $d_i \propto ||\boldsymbol{g_i}||^{-1}$. From definition (4.8) of $\boldsymbol{g_i}$, we have

$$\begin{aligned}||\boldsymbol{g_i}|| &= ||\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}{}^T(\widetilde{y}_i - \widetilde{\boldsymbol{x}}_{\boldsymbol{i}}\boldsymbol{\beta})|| = ||(\boldsymbol{x_i} + a\overline{\boldsymbol{W}}_{\boldsymbol{i}})^T[(\boldsymbol{x_i}\boldsymbol{\beta^*} + \xi_i) + b\overline{\boldsymbol{O}}_{\boldsymbol{i}} - (\boldsymbol{x_i} + a\overline{\boldsymbol{W}}_{\boldsymbol{i}})\boldsymbol{\beta}]|| \\ &= ||(\boldsymbol{x_i} + a\overline{\boldsymbol{W}}_{\boldsymbol{i}})^T[(\boldsymbol{x_i} + a\overline{\boldsymbol{W}}_{\boldsymbol{i}})(\boldsymbol{\beta^*} - \boldsymbol{\beta}) + \xi_i + b\overline{\boldsymbol{O}}_{\boldsymbol{i}} - a\overline{\boldsymbol{W}}_{\boldsymbol{i}}\boldsymbol{\beta^*}]|| \\ &\leq ||[\boldsymbol{x_i}{}^T\boldsymbol{x_i} + 2a\boldsymbol{x_i}{}^T\overline{W}_i + a^2\overline{\boldsymbol{W}}_{\boldsymbol{i}}{}^T\overline{\boldsymbol{W}}_{\boldsymbol{i}}](\boldsymbol{\beta^*} - \boldsymbol{\beta})|| + ||\boldsymbol{x_i}\xi_i|| + ||a\overline{\boldsymbol{W}}_{\boldsymbol{i}}{}^T\xi_i|| \\ &\quad + ||b\boldsymbol{x_i}{}^T\overline{\boldsymbol{O}}_{\boldsymbol{i}}|| + ||ab\overline{\boldsymbol{W}}_{\boldsymbol{i}}{}^T\overline{\boldsymbol{O}}_{\boldsymbol{i}}|| + ||a\boldsymbol{x_i}{}^T\overline{\boldsymbol{W}}_{\boldsymbol{i}}\boldsymbol{\beta^*}|| + ||a^2\overline{\boldsymbol{W}}_{\boldsymbol{i}}{}^T\overline{\boldsymbol{W}}_{\boldsymbol{i}}\boldsymbol{\beta^*}||.\end{aligned}$$

To show that the influence of corrupted points is decreased under LGS procedure, we multiply all observations from the whole dataset by

$$\sigma_x^2||\boldsymbol{\beta^*} - \boldsymbol{\beta}||\sqrt{p}\text{log}p.$$

This is equivalent to scaling the diagonal matrix D to make the diagonal entries d_i proportional to the following term

$$\frac{\sigma_x^2||\boldsymbol{\beta^*} - \boldsymbol{\beta}||}{(\sigma_x^2 + 2a\sigma_x\sigma_w + a^2\sigma_w^2)||\boldsymbol{\beta^*} - \boldsymbol{\beta}|| + (\sigma_x + a\sigma_w)(\sigma_\xi + b\sigma_o) + (a\sigma_x\sigma_w + a^2\sigma_w^2)||\boldsymbol{\beta^*}||} \triangleq q_i. \tag{C.10}$$

Combining with Lemma C.1, the following inequalities hold with high probability,

$$
\begin{aligned}
&||(\boldsymbol{SX})^T\boldsymbol{S\xi}|| \lesssim \sigma_X\sigma_\xi\sqrt{\frac{p\log p}{n}}q_i^2,\\
&||a(\boldsymbol{S\overline{W}})^T\boldsymbol{S\xi}|| \lesssim a\epsilon\sigma_W\sigma_\xi\sqrt{\frac{p\log p}{n}}q_i^2,\\
&||b(\boldsymbol{SX})^T\boldsymbol{S\overline{O}}|| \lesssim b\epsilon\sigma_X\sigma_O\sqrt{\frac{p\log p}{n}}q_i^2,\\
&||ab(\boldsymbol{S\overline{W}})^T\boldsymbol{S\overline{O}}|| \lesssim ab\epsilon^2\sigma_W\sigma_O\sqrt{\frac{p\log p}{n}}q_i^2,\\
&||a(\boldsymbol{SX})^T\boldsymbol{S\overline{W}}\boldsymbol{\beta^*}|| \lesssim a\epsilon\sigma_X\sigma_W||\beta^*||\sqrt{\frac{p\log p}{n}}q_i^2,\\
&||a^2(\boldsymbol{S\overline{W}})^T\boldsymbol{S\overline{W}}\boldsymbol{\beta^*}|| \lesssim a^2\epsilon^2\sigma_W^2||\beta^*||\sqrt{\frac{p\log p}{n}}q_i^2.
\end{aligned}
\tag{C.11}
$$

We observe that the inequalities (C.11) all contain term q_i. To clarify the effect of using low-gradient, we consider without loss of generality the heavily corrupted data case that the previous estimator is totally misleading(say, $||\boldsymbol{\beta^*}-\boldsymbol{\beta}|| \to \infty$) and set $\sigma_X = 1$. Consequently, by (C.10), we have

$$
\lim_{\substack{||\boldsymbol{\beta^*}-\boldsymbol{\beta}||\to\infty\\ \sigma_x=1}} q_i^2 = \frac{1}{a^2\sigma_w + 2a\sigma_w + 1}.
$$

Applying (C.11) to (C.9), we obtain

$$
||\boldsymbol{b_n} - \boldsymbol{\Sigma_n}\boldsymbol{\beta^*}|| \lesssim C\sqrt{\frac{p\log p}{n}},
\tag{C.12}
$$

where

$$
C = \frac{(\sigma_X + a\epsilon\sigma_W)(\sigma_\xi + b\epsilon\sigma_O)}{a^2\sigma_W + 2a\sigma_W + 1} + \frac{a\epsilon\sigma_W(\sigma_X + a\epsilon\sigma_W)}{a^2\sigma_W + 2a\sigma_W + 1}||\beta^*||.
$$

Step 3. Showing that $\Delta^{(t)}$ in (C.7) is monotonically decreasing.

By definition of $\Delta^{(t)}$ in (C.7) and $\boldsymbol{D_\nu^{(t)}} = \boldsymbol{I}$, we have

$$
\begin{aligned}
\Delta^{(t)} - \Delta^{(t-1)} &= 2\frac{\lambda_{\max}(\Sigma_n^{(t)} - \Sigma_N) - \lambda_{\max}(\Sigma_n^{(t-1)} - \Sigma_N)}{\lambda_{\min}^2(\Sigma_N + \boldsymbol{I})}\\
&\le 2\frac{\lambda_{\max}(\Sigma_n^{(t)} - \Sigma_n^{(t-1)})}{\lambda_{\min}^2(\Sigma_N + \boldsymbol{I})}.
\end{aligned}
$$

Recall that $\Sigma_n^{(t)} = \frac{1}{n}\widetilde{\boldsymbol{X}}_{D_t}^T\widetilde{\boldsymbol{X}}_{D_t} = \frac{1}{n}(\boldsymbol{S}\widetilde{\boldsymbol{X}})^T(\boldsymbol{S}\widetilde{\boldsymbol{X}})$, we have $(\Sigma_n^{(t-1)} - \Sigma_n^{(t)}) = (\sigma_{ij})_{p\times p}$, where $\sigma_{ij} = \sum_{s\in\mathcal{D}_{t-1}}\widetilde{x}_{si}\widetilde{x}_{sj} - \sum_{s\in\mathcal{D}_t}\widetilde{x}_{si}\widetilde{x}_{sj} \geq 0$. This latter inequality clearly holds by (4.10). Thus, $(\Sigma_n^{(t-1)} - \Sigma_n^{(t)})$ is semi-positive and $\lambda_{\max}(\Sigma_n^{(t)} - \Sigma_n^{(t-1)}) \leq 0$, which implies the monotone decreasing of $\Delta^{(t)}$.

Step 4. Completing the proof of Theorem 4.1.

For a fix iteration step T, the following inequality can be obtained from (C.7), Step 3 and by mathematical induction,

$$||\boldsymbol{\beta^{(T)}} - \boldsymbol{\beta^*}|| \leq \sum_{t=1}^{T}\Delta^t||\boldsymbol{b_n} - \Sigma_n\boldsymbol{\beta^*}|| + \Delta^T||\boldsymbol{\beta^{(0)}} - \boldsymbol{\beta^*}||, \tag{C.13}$$

Letting $T \to \infty$, we get the following inequality from (C.13),

$$\begin{aligned} ||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta^*}|| &\leq \sum_{t=1}^{\infty}\Delta^t||\boldsymbol{b_n} - \Sigma_n\boldsymbol{\beta^*}|| \\ &= \frac{\Delta}{1-\Delta}||\boldsymbol{b_n} - \Sigma_n\boldsymbol{\beta^*}|| \leq ||\boldsymbol{b_n} - \Sigma_n\boldsymbol{\beta^*}||, \end{aligned} \tag{C.14}$$

This second inequality holds under the condition $\Delta \leq \frac{1}{2}$. That is, (C.14) is established if $\Delta \leq \frac{1}{2}$. This condition however can be deduced from assumption (4.11) in Theorem 4.1. In fact, applying Lemma C.2(also see Th6.1.1 from Tropp [96]), we have

$$\lambda_{\max}(\Sigma_n - \Sigma_N) = \lambda_{\max}(\frac{1}{n}\widetilde{\boldsymbol{X}}_D^T\widetilde{\boldsymbol{X}}_D - \frac{1}{N}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}) = \lambda_{\max}(\frac{1}{N}\sum_{i=1}^{N}(\frac{\tau_i}{p_i} - 1)\widetilde{\boldsymbol{x}_i}^T\widetilde{\boldsymbol{x}_i}) \triangleq \lambda_{\max}(\boldsymbol{A_i}),$$

where $\tau_i \sim B(1, p_i)$, we have $E[\frac{1}{N}(\frac{\tau_i}{p_i} - 1)\widetilde{\boldsymbol{x}_i}^T\widetilde{\boldsymbol{x}_i}] = 0$ and

$$\lambda_{\max}(\frac{1}{N}(\frac{\tau_i}{p_i} - 1)\widetilde{\boldsymbol{x}_i}^T\widetilde{\boldsymbol{x}_i}) \leq \lambda_{\max}(\frac{1}{N}\widetilde{\boldsymbol{x}_i}^T\widetilde{\boldsymbol{x}_i}) = \frac{R}{N}, \text{ where } R = \max_{i=1,\ldots,N}\{||\widetilde{\boldsymbol{x}_i}||^2\}.$$

Thus, the variance of $\boldsymbol{A_i}$ is

$$\begin{aligned} E(\boldsymbol{A_i}^2) &= E[\frac{1}{N^2}\sum_{i=1}^{N}(\frac{\tau_i^2}{p_i^2} - 2\frac{\tau_i}{p_i} + 1)||\widetilde{\boldsymbol{x}_i}||^2] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}(\frac{1}{p_i} - 1)||\widetilde{\boldsymbol{x}_i}||^2 \\ &\leq \frac{1}{N^2 n}\sum_{i=1}^{N}\frac{1}{\pi_i}||\widetilde{\boldsymbol{x}_i}||^4. \end{aligned}$$

By Lemma C.2,

$$E[\lambda_{\max}(\Sigma_n - \Sigma_N)] = E[\lambda_{\max}(\boldsymbol{A_i})] \leq \sqrt{2\frac{1}{N^2 n}\sum_{i=1}^{N}\frac{1}{\pi_i}||\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}||^4 \log p} + \frac{R}{3N}\log p = \sigma_N n^{-\frac{1}{2}}\sqrt{\log p} + \frac{R}{3N}\log p,$$

where $\sigma_N^2 = 2\frac{1}{N^2}\sum_{i=1}^{N}\frac{1}{\pi_i}||\widetilde{\boldsymbol{x}}_{\boldsymbol{i}}||^4$. For any given δ, from Markov's inequality we have that

$$P\{\lambda_{\max}(\Sigma_n - \Sigma_N) > \sigma_N n^{-\frac{1}{2}}\sqrt{\log p}\delta^{-1} + \frac{R}{3N\delta}\log p\} \leq \delta. \tag{C.15}$$

Thus, condition (C.5) holds if

$$n \geq \frac{\sigma_N^2 \log p}{\delta^2(\frac{1}{2}\lambda_{\min}(\Sigma_N + \boldsymbol{I}) - \frac{R\log p}{3N\delta})^2}. \tag{C.16}$$

Applying (C.15) to (C.8), we have $\Delta \leq \frac{1}{2}$ if

$$n \geq \frac{\sigma_N^2 \log p}{\delta^2(\frac{1}{2}\lambda_{\min}(\Sigma_N + \boldsymbol{I})(\frac{1}{2}\lambda_{\min}(\Sigma_N + \boldsymbol{I}) - 1) - \frac{R\log p}{3N\delta})^2}. \tag{C.17}$$

Combining (C.7), (C.12), (C.14), (C.16) and (C.17) gives

$$||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta^*}|| \lesssim C\sqrt{\frac{p\log p}{n}},$$

where

$$C = \frac{(\sigma_X + a\epsilon\sigma_W)(\sigma_\xi + b\epsilon\sigma_O)}{a^2\sigma_W + 2a\sigma_W + 1} + \frac{a\epsilon\sigma_W(\sigma_X + a\epsilon\sigma_W)}{a^2\sigma_W + 2a\sigma_W + 1}||\beta^*||,$$

that complete the proof of Theorem 4.1.

C.2 The proof of Theorem 4.2

The proof of Theorem 4.2: A completely similar argument with Steps 1-3 of the proof of Theorem 4.1 shows

$$||\boldsymbol{\beta^{(T)}} - \boldsymbol{\beta^*}|| \leq \sum_{t=1}^{T}\Delta^t||\boldsymbol{b_n} - \Sigma_n\boldsymbol{\beta^*}|| + \Delta^T||\boldsymbol{\beta^{(0)}} - \boldsymbol{\beta^*}||.$$

For any given tolerance ε, when $T \gtrsim \log \frac{||\boldsymbol{\beta^{(0)}} - \boldsymbol{\beta^*}||}{\varepsilon}$, we have

$$\Delta^T ||\boldsymbol{\beta^{(0)}} - \boldsymbol{\beta^*}|| \lesssim e^{-(1-\Delta)T} ||\boldsymbol{\beta^{(0)}} - \boldsymbol{\beta^*}|| \lesssim \varepsilon. \tag{C.18}$$

Applying (C.12) and (C.18) to (C.13), we obtain

$$||\widehat{\boldsymbol{\beta}}^{(\boldsymbol{T})} - \boldsymbol{\beta^*}|| \leq ||\boldsymbol{b_n} - \Sigma_n \boldsymbol{\beta^*}|| + \Delta^T ||\boldsymbol{\beta^{(0)}} - \boldsymbol{\beta^*}|| \lesssim C\sqrt{\frac{p \log p}{n}} + \varepsilon.$$

The theorem is therefore proved.