Journal of Machine Learning Research 21 (2020) 1-43 Submitted 3/20; Revised 8/20; Published 8/20

Distributed High-dimensional Regression Under a Quantile Loss Function

Xi Chen xc13@stern.nyu.edu
Stern School of Business
New York University, New York, NY 10012, USA

Weidong Liu weidongl@sjtu.edu.cn
School of Mathematical Sciences and MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University, Shanghai, 200240, China

Xiaojun Mao maoxj@fudan.edu.cn
School of Data Science
Fudan University, Shanghai, 200433, China

Zhuoyi Yang zyang@stern.nyu.edu
Stern School of Business
New York University, New York, NY 10012, USA

Editor: Qiang Liu

Abstract

This paper studies distributed estimation and support recovery for high-dimensional linear regression model with heavy-tailed noise. To deal with heavy-tailed noise whose variance can be infinite, we adopt the quantile regression loss function instead of the commonly used squared loss. However, the non-smooth quantile loss poses new challenges to high-dimensional distributed estimation in both computation and theoretical development. To address the challenge, we transform the response variable and establish a new connection between quantile regression and ordinary linear regression. Then, we provide a distributed estimator that is both computationally and communicationally efficient, where only the gradient information is communicated at each iteration. Theoretically, we show that, after a constant number of iterations, the proposed estimator achieves a near-oracle convergence rate without any restriction on the number of machines. Moreover, we establish the theoretical guarantee for the support recovery. The simulation analysis is provided to demonstrate the effectiveness of our method.

Keywords: Distributed estimation, high-dimensional linear model, quantile loss, robust estimator, support recovery

1. Introduction

The development of internet technology has led to the generation of modern data that exhibits several challenges in statistical estimation:

1. The first challenge comes from the scalability of the data. In particular, modern large-scale data usually cannot be fit into memory or are collected in a distributed

Weidong Liu and Xiaojun Mao are the co-corresponding authors.

environment. For example, a personal computer usually has a limited memory size in GBs; while the data stored on a hard disk could have a size in TBs. In addition, sensor network data are naturally collected by many sensors. For these types of large-scale data, traditional methods, which load all the data into memory and run a certain optimization procedure (e.g., Lasso), are no longer applicable due to both storage and computation issues.

2. The second challenge comes from the dimensionality of data. High-dimensional data analysis has been an important research area in statistics over the past decade. A sparse model is commonly adopted in high-dimensional literature and support recovery is an important task for high-dimensional analysis (see, e.g., Zhao and Yu (2006); Wainwright (2009); Bühlmann and Van De Geer (2011); Tibshirani et al. (2015)). There are some recent work on statistical estimation for high-dimensional distributed data (see, e.g., Zhao et al. (2014), Lee et al. (2017), Battey et al. (2018)). However, these work usually adopt a de-biased approach, which leads to a dense estimated coefficient vector. Moreover, the *support recovery* problem in a distributed setting still largely remains open.

3. The third challenge comes from heavy-tailed noise, which is prevalent in practice (see, e.g., Hsu and Sabato (2016); Fan et al. (2017); Chen et al. (2018); Sun et al. (2020); Zhou et al. (2018)). When the finite variance assumption for the noise does not exist, most existing theories based on least squares or Huber loss in robust statistics will no longer be applicable.

The main purpose of the paper is to provide a new estimation approach for high-dimensional linear regression in a distributed environment and establish the theoretical results on both estimation and support recovery. More specifically, we consider the following linear model,

$$Y = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}^* + e, \tag{1}$$

where $\boldsymbol{X} = (1, X_1, \ldots, X_p)^{\mathrm{T}}$ is a $(p+1)$-dimensional vector, $\boldsymbol{\beta}^* = (\beta_0^*, \beta_1^*, \ldots, \beta_p^*)^{\mathrm{T}}$ is the true regression coefficient, with β_0^* being the intercept, and e is the noise. We only assume that e is independent of the covariate vector $(X_1, \ldots, X_p)^{\mathrm{T}}$ and the density function of e exists. It is worthwhile noting that the independence assumption has been adopted in estimating robust linear models when using a quantile loss function (see, e.g., Zou and Yuan (2008); Fan et al. (2014)). In Remark 7, we will briefly comment on how to extend our method to the case when the noise is not independent with covariates. Furthermore, we allow the dimension p to be much larger than the sample size n (e.g., $p = o(n^{\nu})$ for some $\nu > 0$). We assume that $\boldsymbol{\beta}^*$ is a sparse vector with s non-zero elements.

In this paper, we allow a very heavy-tailed noise e, whose variance can be infinite (e.g., Cauchy distribution). For such a heavy-tailed noise, the squared-loss based Lasso approach is no longer applicable. To address this challenge, we can assume without loss of generality that $\mathbb{P}(e \leq 0) = \tau$ for a specified quantile level $\tau \in (0,1)$ (otherwise, we can shift the first component to be $\beta_0^* - q_\tau$ so that this assumption holds, where q_τ is the τ-th quantile of e). Then, it is easy to see that

$$\boldsymbol{\beta}^* = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \mathbb{E}\rho_\tau(Y - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}),$$

where $\rho_\tau(x) = x(\tau - \mathbb{I}[x \leq 0])$ (see, e.g., Koenker (2005)) is known as the quantile regression (QR) loss function. Given n *i.i.d.* samples $(\boldsymbol{X}_i, Y_i)$ for $1 \leq i \leq n$, the high-dimensional QR estimator takes the following form,

$$\widehat{\boldsymbol{\beta}} = \mathop{\arg\min}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau(Y_i - \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda_n |\boldsymbol{\beta}|_1, \tag{2}$$

where $|\boldsymbol{\beta}|_1$ the ℓ_1-regularization of $\boldsymbol{\beta}$, and λ_n is the regularization parameter.

It is worthwhile noting that in robust statistical literature, the MOM (median of means) has been adopted to corrupted data in high-dimensional settings (Hsu and Sabato, 2014; Lugosi and Mendelson, 2016; Lecué et al., 2020; Lugosi et al., 2019; Lecué and Lerasle, 2019). However, the MOM is a multi-stage method that requires data splitting. Moreover, when true regression coefficients are sparse, support recovery guarantee is not available in existing MOM literature. Moreover, the quantile loss has been a useful approach to deal with heavy-tailed noise, see, e.g., Fan et al. (2014) for single quantile level and Zou and Yuan (2008) for multiple quantile levels. However, the existing literature does not address the challenging issue on efficient distributed implementation, which is the main focus of this paper.

Although the adoption of QR loss provides robustness to heavy-tailed noises, it also poses new challenges due to limited computation power and memory to store data especially when the sample size and dimension are both large. Therefore, distributed estimation procedure becomes increasingly important. The main purpose of the paper is to develop a new estimation approach for high-dimensional QR and establish the theoretical results on both *estimation* and *support recovery*. In fact, as we will survey in the next paragraph, the support recovery problem in a high-dimensional distributed setting still largely remains as an open problem.

In a distributed setting, let us assume n samples are stored in L local machines. In particular, we split the data index set $\{1, 2, \ldots, n\}$ into $\mathcal{H}_1, \ldots, \mathcal{H}_L$, where $\mathcal{H}_k$ denotes the set of indices on the k-th machine. For the ease of illustration, we assume that the data are evenly distributed (n/L is an integer) and each local machine has the sample size $|\mathcal{H}_k| = m = n/L$ (see Remark 6 at the end of Section 3 for the discussion on general data partitions). On each machine, one can construct a local estimator $\widehat{\boldsymbol{\beta}}_k$ by solving

$$\widehat{\boldsymbol{\beta}}_k = \mathop{\arg\min}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \frac{1}{m} \sum_{i \in \mathcal{H}_k} \rho_\tau(Y_i - \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda_m |\boldsymbol{\beta}|_1. \tag{3}$$

Then the final estimator of $\boldsymbol{\beta}^*$ can be naturally taken as the averaging estimator $\widehat{\boldsymbol{\beta}}_{avg} = \frac{1}{L}\sum_{k=1}^{L} \widehat{\boldsymbol{\beta}}_k$. This method is usually known as averaging divide-and-conquer approach (see, e.g., Li et al. (2013); Zhao et al. (2016); Fan et al. (2019b); Shi et al. (2018); Banerjee et al. (2019)). Although this method enjoys low communication cost (i.e., one-shot communication), the obtained estimator is usually no longer sparse. Instead of constructing the local estimator in its original form as in (3), there are a number of works that construct a de-biased estimator as the local estimator, and then take the average (see, e.g., Zhao et al. (2014); Lee et al. (2017); Battey et al. (2018)). The de-biased estimator has been popular in high-dimensional statistics (see, e.g., Belloni et al. (2013); Van de Geer et al. (2014);

Zhang and Zhang (2014); Javanmard and Montanari (2014) and references therein). Zhao et al. (2014) studied the averaging divide-and-conquer approach for high-dimensional QR based on de-biased estimator. There are several issues of the averaging de-biased estimator for high-dimensional distributed estimation. First, due to de-biasing, the local estimator on each machine is no longer sparse and thus the final averaging estimator cannot be used for support recovery. Second, the de-biased approach needs to estimate a $p \times p$ precision matrix $\mathbf{\Sigma}^{-1}$, which requires each machine to solve p optimization problems (see, e.g., Eq. (3.17) in Zhao et al. (2014)), while each optimization problem involves computing a variant of the CLIME estimator (Cai et al., 2011). In other words, instead of solving one p-dimensional optimization as in (3), the de-biased estimator requires to solve $(p+1)$ optimization problems. This would be computationally very expensive especially when p is large. Finally, the theoretical result of the averaging estimator requires that the number of machines L is not too large. For example, in high-dimensional QR, the theoretical development in Zhao et al. (2014) requires $L = o(n^{1/3}/(s \log^{5/3}(\max(p, n))))$, where s is the number of non-zero elements in $\boldsymbol{\beta}^*$. It would be an interesting theoretical question on how to remove such a constraint on L. In Wang et al. (2017), Jordan et al. (2019) and Fan et al. (2019a), they develop iterative methods with multiple rounds of aggregations (instead of one-shot averaging), which relax the condition on the number of machines. However, their methods and theory require the loss function to be second-order differentiable and thus cannot be applied to the *non-smooth* QR loss. We also note that Chen et al. (2019) studied distributed QR problem in a low dimensional setting, where $\boldsymbol{\beta}^*$ is dense and p grows much more slowly than n.

In this paper, we propose a new distributed estimator for estimating high-dimensional linear model with heavy-tailed noise. We first show that the estimation of regression coefficient $\boldsymbol{\beta}^*$ can be resorted to a penalized least squares optimization problem with a pseudo-response $\widetilde{Y}_i$ instead of Y_i. This leads to a pooled estimator, which essentially solves a Lasso problem with the squared loss based on $\widetilde{Y}_i$, without requiring any moment condition on the noise term. This pooled estimator is computationally much more efficient than solving high-dimensional QR (2) in a single machine setting.

Moreover, our result establishes an interesting connection between the QR estimation and the ordinary linear regression. This connection translates a non-smooth objective function to a smooth one, which greatly facilitates computation in a distributed setting. Given the transformed penalized least squares formulation, we further provide a communication efficient distributed algorithm, which runs iteratively and only communicates $(p+1)$-dimensional gradient information at each iteration (instead of the $(p+1) \times (p+1)$ matrix information). Our distributed algorithm is essentially an approximate Newton method (see, e.g., Shamir et al. (2014)), which uses gradient information to approximate Hessian information and thus allows efficient communication. In this paper, we provide a more intuitive derivation of the method simply based on the standard Lasso theory.

Then we establish the theoretical properties of the proposed distributed estimator. We first establish the convergence rate in ℓ_2-norm for one iteration (Theorem 2). Based on this result, we further characterize the convergence rate for multiple iterations. We show that, after a constant number of iterations, our method achieves a near-oracle rate of $\sqrt{s \log(\max(p, n))/n}$ (Theorem 3). This rate is identical to the rate of ℓ_1-regularized QR in a single machine setting (Belloni and Chernozhukov, 2011), and almost matches the oracle rate $\sqrt{s/n}$ (upto a logarithmic factor) where the true support is known. Furthermore, we provide the support recovery result of the distributed estimator. We first show that the estimated support is a subset of the true support with high probability (Theorem 4 and 5). Then we characterize the "beta-min" condition for the exact support recovery, and we show that the "beta-min" condition becomes weaker as the number of iterations increases (Theorem 5). Again, after a constant number of iterations, the lower bound in our "beta-min" condition matches the ideal case with all the samples on a single machine. To the best of our knowledge, this is the first support recovery result for high-dimensional robust distributed estimation.

1.1. Paper Organization and Notations

The rest of our paper is organized as follows. In Section 2 we define the estimator and provide our algorithm. In Section 3 we provide the theoretical guarantee for the convergence rate and support recovery for our estimator. Numerical experiments based on simulation are provided in Section 4 to illustrate the performance of the estimator. Section 5 gives some concluding remarks and future directions. The proofs of main theoretical results is relegated to the Appendix A.

For a vector $\boldsymbol{v} = (v_1, \ldots, v_n)^{\mathrm{T}}$, define $|\boldsymbol{v}|_1 = \sum_{i=1}^n |v_i|$ and $|\boldsymbol{v}|_2 = \sqrt{\sum_{i=1}^n v_i^2}$. For a matrix $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{p\times q}$, define $|\boldsymbol{A}|_\infty = \max_{1\le i\le p, 1\le j\le q} |a_{ij}|$, $\|\boldsymbol{A}\|_{L_1} = \max_{1\le j\le q} \sum_{i=1}^p |a_{ij}|$, $\|\boldsymbol{A}\|_{\mathrm{op}} = \max_{|v|_2=1} |\boldsymbol{A}v|_2$, and $\|\boldsymbol{A}\|_\infty = \max_{1\le i\le p} \sum_{j=1}^q |a_{ij}|$. For two sequences a_n and b_n we say $a_n \asymp b_n$ if and only if both $a_n = O(b_n)$ and $b_n = O(a_n)$ hold. For a matrix $\boldsymbol{A}$, define $\Lambda_{\max}(\boldsymbol{A})$ and $\Lambda_{\min}(\boldsymbol{A})$ to be the largest and smallest eigenvalues of $\boldsymbol{A}$ respectively. For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and two subsets of indices $S = \{s_1, \ldots, s_r\} \subseteq \{1, \ldots, m\}$ and $T = \{t_1, \ldots, t_q\} \subseteq \{1, \ldots, n\}$, we use $\boldsymbol{A}_{S\times T}$ to denote the r by q submatrix given by $(a_{s_i t_j})$. We use $C, c, c_0, c_1, \ldots$ to denote constants whose value may change from place to place, which do not depend on n, p, s and m.

2. Methodology

In this section, we introduce the proposed method. We start with a robust estimator with Lasso (REL), which establishes the connection between quantile regression (QR) and ordinary linear regression in a single machine setting. This proposed estimator will motivate the construction of our distributed estimator.

2.1. Robust Estimator with Lasso (REL)

Our method is inspired by the Newton-Raphson method. Consider the following stochastic optimization problem,

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min}\, \mathbb{E}[G(\boldsymbol{\beta}; \boldsymbol{X}, Y)], \tag{4}$$

where $G(\boldsymbol{\beta}; \boldsymbol{X}, Y)$ is the loss function. In $G(\boldsymbol{\beta}; \boldsymbol{X}, Y)$, $\boldsymbol{X}$ and Y are random covariates and response and $\boldsymbol{\beta}$ is the coefficient vector of interest. To solve this stochastic optimization problem, the population version of the Newton-Raphson iteration takes the following form

$$\widetilde{\boldsymbol{\beta}}_1 = \boldsymbol{\beta}_0 - \boldsymbol{H}(\boldsymbol{\beta}_0)^{-1}\mathbb{E}[g(\boldsymbol{\beta}_0; \boldsymbol{X}, Y)], \tag{5}$$

where $\boldsymbol{\beta}_0$ is an initial solution, $g(\boldsymbol{\beta}; \boldsymbol{X}, Y)$ is the subgradient of the loss function $G(\boldsymbol{\beta}; \boldsymbol{X}, Y)$ with respect to $\boldsymbol{\beta}$, and $\boldsymbol{H}(\boldsymbol{\beta}) := \partial \mathbb{E}[g(\boldsymbol{\beta}; \boldsymbol{X}, Y)]/\partial \boldsymbol{\beta}$ denotes the population Hessian matrix of $\mathbb{E}G(\boldsymbol{\beta}; \boldsymbol{X}, Y)$. In particular, let us consider the case where $G(\boldsymbol{\beta}; \boldsymbol{X}, Y)$ is the QR loss, i.e.,

$$G(\boldsymbol{\beta}; \boldsymbol{X}, Y) = \rho_\tau(Y - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}). \tag{6}$$

Given $G(\boldsymbol{\beta}; \boldsymbol{X}, Y)$ in (6), the subgradient and Hessian matrix take the form of $g(\boldsymbol{\beta}; \boldsymbol{X}, Y) = \boldsymbol{X}(\mathbb{I}[Y - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta} \le 0] - \tau)$ and $\boldsymbol{H}(\boldsymbol{\beta}) = \mathbb{E}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} f(\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}^*)))$, respectively. Here, $f(x)$ is the density function of the noise e. When the initial estimator $\boldsymbol{\beta}_0$ is close to the true parameter $\boldsymbol{\beta}^*$, $\boldsymbol{H}(\boldsymbol{\beta}_0)$ will be close to $\boldsymbol{H}(\boldsymbol{\beta}^*) = \boldsymbol{\Sigma} f(0)$, where $\boldsymbol{\Sigma} = \mathbb{E}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ is the population covariance matrix of the covariates $\boldsymbol{X}$. Using $\boldsymbol{H}(\boldsymbol{\beta}^*)$ in (5) motivates the following iteration,

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 - \boldsymbol{H}(\boldsymbol{\beta}^*)^{-1}\mathbb{E}[g(\boldsymbol{\beta}_0; \boldsymbol{X}, Y)] = \boldsymbol{\beta}_0 - \boldsymbol{\Sigma}^{-1} f^{-1}(0)\mathbb{E}[g(\boldsymbol{\beta}_0; \boldsymbol{X}, Y)]. \tag{7}$$

Further, under some regularity conditions, we have the following Taylor expansion of $\mathbb{E}[g(\boldsymbol{\beta}_0; \boldsymbol{X}, Y)]$ at $\boldsymbol{\beta}^*$,

$$\begin{aligned}\mathbb{E}[g(\boldsymbol{\beta}_0; \boldsymbol{X}, Y)] =& \boldsymbol{H}(\boldsymbol{\beta}^*)(\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*) + O(|\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*|_2^2)\\ =& \boldsymbol{\Sigma} f(0)(\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*) + O(|\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*|_2^2).\end{aligned}$$

Combine it with (7), and it is easy to see that

$$\begin{aligned}|\boldsymbol{\beta}_1 - \boldsymbol{\beta}^*|_2 =& |\boldsymbol{\beta}_0 - \boldsymbol{\Sigma}^{-1} f^{-1}(0)\left(\boldsymbol{\Sigma} f(0)(\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*) + O(|\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*|_2^2)\right) - \boldsymbol{\beta}^*|_2\\ =& O(|\boldsymbol{\beta}_0 - \boldsymbol{\beta}^*|_2^2).\end{aligned}$$

In summary, if we have a consistent estimator $\boldsymbol{\beta}_0$, we can refine it by the Newton-Raphson iteration in (7).

Next, we show how to translate the Newton-Raphson iteration into a least squares optimization problem. First we rewrite the equation (7) to be

$$\begin{aligned}\boldsymbol{\beta}_1 &= \boldsymbol{\Sigma}^{-1}\Big(\boldsymbol{\Sigma}\boldsymbol{\beta}_0 - f^{-1}(0)\mathbb{E}[g(\boldsymbol{\beta}_0; \boldsymbol{X}, Y)]\Big)\\ &= \boldsymbol{\Sigma}^{-1}\mathbb{E}\Big[\boldsymbol{X}\Big\{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}_0 - f^{-1}(0)(\mathbb{I}[Y \le \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}_0] - \tau)\Big\}\Big].\end{aligned}$$

Let us define a new response variable $\widetilde{Y}$ as

$$\widetilde{Y} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}_0 - f^{-1}(0)(\mathbb{I}[Y \le \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}_0] - \tau).$$

Then $\boldsymbol{\beta}_1 = \boldsymbol{\Sigma}^{-1}\mathbb{E}(\boldsymbol{X}\widetilde{Y})$ is the best linear regression coefficient of $\widetilde{Y}$ on $\boldsymbol{X}$, i.e., $\boldsymbol{\beta}_1 = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \mathbb{E}(\widetilde{Y} - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta})^2$. To further encourage the sparsity of the estimator, it is natural to consider the following ℓ_1-regularized problem,

$$\boldsymbol{\beta}_{1,\lambda} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{2}\mathbb{E}(\widetilde{Y} - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta})^2 + \lambda|\boldsymbol{\beta}|_1, \tag{8}$$

where $\boldsymbol{\beta}_{1,\lambda}$ is sparse and can achieve a better convergence rate than $\boldsymbol{\beta}_0$. So far, we have shown that if we have a consistent estimator $\boldsymbol{\beta}_0$ of $\boldsymbol{\beta}^*$, then the estimation of the high-dimensional sparse $\boldsymbol{\beta}^*$ can be implemented by solving a penalized least squares optimization in (8) instead of the penalized QR optimization. It is well known that the latter optimization problem is computationally expensive when n is large since the QR loss is non-smooth. More importantly, the transformation from QR loss to least squares will greatly facilitate the development of the distributed estimator. In particular, our distributed estimator is derived from the Lasso theory, which is based on the squared loss (see Section 2.2).

Now, we are ready to define the empirical version of $\boldsymbol{\beta}_{1,\lambda}$ in a single machine setting. Let $\widehat{\boldsymbol{\beta}}_0$ be an initial estimator of $\boldsymbol{\beta}^*$ and $\widehat{f}(0)$ be an estimator of the density $f(0)$. We use $\widehat{\boldsymbol{\beta}}_0$ to denote the empirical version of the initial estimator, which is distinguished from the population version $\boldsymbol{\beta}_0$. Given n *i.i.d.* samples $(\boldsymbol{X}_i, Y_i)$ from (1), for each $1 \leq i \leq n$, we construct

$$\widetilde{Y}_i = \boldsymbol{X}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}_0 - \widehat{f}^{-1}(0)(\mathbb{I}[Y_i \leq \boldsymbol{X}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}_0] - \tau).$$

It is natural to estimate $\boldsymbol{\beta}^*$ by the empirical version of (8):

$$\widehat{\boldsymbol{\beta}}_{pool} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \Big\{ \frac{1}{2n} \sum_{i=1}^{n} (\widetilde{Y}_i - \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{\beta})^2 + \lambda_n |\boldsymbol{\beta}|_1 \Big\}. \tag{9}$$

We note that in a single machine setting, computing this pooled estimator essentially solves a Lasso problem, which is computationally much more efficient than solving an ℓ_1-regularized QR problem.

Finally, we choose $\widehat{f}(0)$ to be a kernel density estimator of $f(0)$:

$$\widehat{f}(0) = \frac{1}{nh} \sum_{i=1}^{n} K\Big(\frac{Y_i - \boldsymbol{X}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}_0}{h} \Big),$$

where $K(x)$ is a kernel function which satisfies the condition (C3) (see Section 3) and $h \to 0$ is the bandwidth. The selection of bandwidth will be discussed in our theoretical results (see Section 3).

In the next section, we will introduce a distributed robust estimator with Lasso which can estimate $\boldsymbol{\beta}^*$ with a near-oracle convergence rate.

2.2. Distributed Robust Estimator with Lasso

Given our new proposed estimator $\widehat{\boldsymbol{\beta}}_{pool}$, we can use the approximate Newton method to solve the distributed estimation problem. To illustrate this technique from the Lasso theory, we first consider a general convex quadratic optimization as follows,

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{2} \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{\beta} - \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{b} + \lambda_n |\boldsymbol{\beta}|_1, \tag{10}$$

where $\boldsymbol{A}$ is a non-negative definite matrix and $\boldsymbol{b}$ is a vector in $\mathbb{R}^{p+1}$. From standard Lasso theory (see Bühlmann and Van De Geer (2011)), we have the following proposition.

Proposition 1 *Assume the following conditions hold*

$$|\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{b}|_\infty \leq \lambda_n/2, \tag{11}$$

$$\min_{\delta:|\delta|_1\le c_1\sqrt{s}|\delta|_2} \frac{\delta^{\mathrm{T}}\boldsymbol{A}\delta}{|\delta|_2^2} \ge c_2, \quad c_1, c_2 > 0. \tag{12}$$

where s is the sparsity of $\boldsymbol{\beta}^*$, i.e., $s = \sum_{j=0}^{p} \mathbb{I}[\beta_j^* \neq 0]$. Then we have

$$|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2 \le c\sqrt{s}\lambda_n, \tag{13}$$

for some constant $c > 0$.

Note that the condition (12) is known as the compatibility condition, which is used to provide the ℓ_2-consistency of the Lasso estimator. For the purpose of completeness, we include a proof of Proposition 1 in the Appendix A. As one can see from (11), if we can choose a matrix $\boldsymbol{A}$ and a vector $\boldsymbol{b}$ such that λ_n is as small as possible, we can obtain a fast convergence rate of $\widehat{\boldsymbol{\beta}}$.

Now let us discuss how to use Proposition 1 to develop our distributed estimator. Suppose that n samples are stored in $L = n/m$ machines and each local machine has m samples. We first split the data index set $\{1, 2, \ldots, n\}$ into $\mathcal{H}_1, \ldots, \mathcal{H}_L$ with $|\mathcal{H}_k| = m$ and the k-th machine stores samples $\{(\boldsymbol{X}_i, Y_i):\ i \in \mathcal{H}_k\}$. Let us define

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{m}\sum_{i\in\mathcal{H}_k} \boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}}, \quad \widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{X}_i\boldsymbol{X}_i^{\mathrm{T}} = \frac{1}{L}\sum_{k=1}^{L}\widehat{\boldsymbol{\Sigma}}_k, \tag{14}$$

as the sample covariance matrix on the k-th machine and the sample covariance matrix of the entire dataset, respectively. It is worthwhile noting that our algorithm does not need to explicitly compute and communicate $\widehat{\boldsymbol{\Sigma}}_k$ (for $k \neq 1$) (see Algorithm 1 for more details).

In Proposition 1, we first choose $\boldsymbol{A} = \widehat{\boldsymbol{\Sigma}}_1$ to be the sample covariance matrix computed on the first machine. Our goal is to construct a vector $\boldsymbol{b}$ such that $|\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{b}|_\infty$ can be as small as possible. Note that

$$\begin{aligned}\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{b} =& \widehat{\boldsymbol{\Sigma}}_1\boldsymbol{\beta}^* - \boldsymbol{b}\\ =& \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^* + (\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}})\boldsymbol{\beta}^* - \boldsymbol{b}.\end{aligned} \tag{15}$$

It can be proved that $\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*$ is close to $\boldsymbol{z}_n := \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i\widetilde{Y}_i$ (see Proposition 11 in the Appendix A). We note that $\boldsymbol{z}_n$ can be computed effectively in a distributed setting since

$$\boldsymbol{z}_n = \frac{1}{L}\sum_{k=1}^{L}\boldsymbol{z}_{nk}, \quad \boldsymbol{z}_{nk} = \frac{1}{m}\sum_{i\in\mathcal{H}_k}\boldsymbol{X}_i\widetilde{Y}_i,$$

where $\boldsymbol{z}_{nk}$ can be computed on the k-th local machine. Therefore we can rewrite (15) as

$$\begin{aligned}|\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{b}|_\infty =& |\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^* - \boldsymbol{z}_n + \boldsymbol{z}_n + (\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}})\boldsymbol{\beta}^* - \boldsymbol{b}|_\infty\\ \le& |\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^* - \boldsymbol{z}_n|_\infty + |\boldsymbol{z}_n + (\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}})\boldsymbol{\beta}^* - \boldsymbol{b}|_\infty.\end{aligned}$$

Since $\boldsymbol{\beta}^*$ is unknown, in order to make the second term as small as possible, it is natural to set

$$\boldsymbol{b} = \boldsymbol{z}_n + (\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}})\widehat{\boldsymbol{\beta}}_0.$$

For $\boldsymbol{A} = \widehat{\boldsymbol{\Sigma}}_1$ and $\boldsymbol{b} = \boldsymbol{z}_n + (\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}})\widehat{\boldsymbol{\beta}}_0$, we can prove that (see Eq. (39) in the proof of Theorem 2 and 3)

$$|\widehat{\boldsymbol{\Sigma}}_1\boldsymbol{\beta}^* - \boldsymbol{b}|_\infty \leq \lambda_n/2,$$

for some specified λ_n (see Theorem 2). With $\boldsymbol{A}$ and $\boldsymbol{b}$ in place, the equation (10) leads to the following ℓ_1-regularized quadratic programming,

$$\widehat{\boldsymbol{\beta}}^{(1)} = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \frac{1}{2m}\sum_{i\in\mathcal{H}_1}(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\beta})^2 - \boldsymbol{\beta}^{\mathrm{T}}\Big\{\boldsymbol{z}_n + (\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}})\widehat{\boldsymbol{\beta}}_0\Big\} + \lambda_n|\boldsymbol{\beta}|_1. \tag{16}$$

Note that when $m = n$, we have $\widehat{\boldsymbol{\beta}}^{(1)} = \widehat{\boldsymbol{\beta}}_{pool}$. In other words, when the data is pooled on a single machine, the proposed distributed estimator automatically reduces to $\widehat{\boldsymbol{\beta}}_{pool}$ in (9). We also note that $\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\beta}}_0$ in the vector $\boldsymbol{b}$ can be computed effectively in a distributed manner. In particular, each local machine computes and communicates a $(p+1)$-dimensional vector $\widehat{\boldsymbol{\Sigma}}_k\widehat{\boldsymbol{\beta}}_0 = \frac{1}{m}\sum_{i\in\mathcal{H}_k}\boldsymbol{X}_i(\boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_0)$ to the first machine. Then the first machine computes $\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\beta}}_0$ by

$$\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{\beta}}_0 = \frac{1}{L}\sum_{k=1}^{L}\widehat{\boldsymbol{\Sigma}}_k\widehat{\boldsymbol{\beta}}_0.$$

Our algorithm only communicates $\boldsymbol{z}_{nk} = \frac{1}{m}\sum_{i\in\mathcal{H}_k}\boldsymbol{X}_i\widetilde{Y}_i$ and $\widehat{\boldsymbol{\Sigma}}_k\widehat{\boldsymbol{\beta}}_0$ to the first machine at each iteration. Therefore, the per-iteration communication complexity is only $O(p)$ and there is no need to communicate the $(p+1)\times(p+1)$ sample covariance matrix $\widehat{\boldsymbol{\Sigma}}_k$.

Given (16) as the estimator from the first iteration, it is easy to construct an iterative estimator. In particular, let $\widehat{\boldsymbol{\beta}}^{(t-1)}$ be the distributed REL in the $(t-1)$-th iteration. Define

$$\widehat{f^{(t)}}(0) = \frac{1}{nh_t}\sum_{i=1}^{n} K\left(\frac{Y_i - \boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{(t-1)}}{h_t}\right),$$

as the density estimator in the t-th iteration where $h_t \to 0$ is the bandwidth for the t-th iteration. The bandwidth h_t shrinks as t grows, whose rate will be specified in Theorem 3. Let us define

$$\widetilde{Y}_i^{(t)} = \boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{(t-1)} - (\widehat{f^{(t)}}(0))^{-1}\left(\mathbb{I}\left[Y_i \leq \boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{(t-1)}\right] - \tau\right), \tag{17}$$

and

$$\boldsymbol{z}_n^{(t)} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i\widetilde{Y}_i^{(t)}.$$

As in (16), our distributed estimator $\widehat{\boldsymbol{\beta}}^{(t)}$ is the solution of the following ℓ_1-regularized quadratic programming problem:

$$\widehat{\boldsymbol{\beta}}^{(t)} = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \frac{1}{2m}\sum_{i\in\mathcal{H}_1}(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\beta})^2 - \boldsymbol{\beta}^{\mathrm{T}}\left\{\boldsymbol{z}_n^{(t)} + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}^{(t-1)}\right\} + \lambda_{n,t}\,|\boldsymbol{\beta}|_1\,. \tag{18}$$

It is worthwhile noting that the convex optimization problem (18) has been extensively studied in the optimization literature and several efficient optimization methods have been

Algorithm 1 Distributed high-dimensional QR estimator

```
Input: Data on local machines $\{\boldsymbol{X}_i, Y_i : i \in \mathcal{H}_k\}$ for $k = 1, \ldots, L$, the number of iterations $t$, quantile level $\tau$, kernel function $K$, a sequence of bandwidths $h_g$ for $g = 1, \ldots, t$ and the regularization parameters $\lambda_0$, $\lambda_{n,g}$ for $g = 1, \ldots, t$.
Compute the initial estimator $\widehat{\boldsymbol{\beta}}^{(0)} = \widehat{\boldsymbol{\beta}}_0$ based on $\{\boldsymbol{X}_i, Y_i : i \in \mathcal{H}_1\}$:
    $\widehat{\boldsymbol{\beta}}_0 = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{m} \sum_{i \in \mathcal{H}_1} \rho_\tau(Y_i - \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda_0 |\boldsymbol{\beta}|_1.$   (19)
for $g = 1, 2 \ldots, t$ do
    Transmit $\widehat{\boldsymbol{\beta}}^{(g-1)}$ to all local machines.
    for $k = 1, \ldots, L$ do
        The $k$-th machine computes $\widehat{f}^{(g,k)}(0) := \frac{1}{m} \sum_{i \in \mathcal{H}_k} K\left(\frac{Y_i - \boldsymbol{X}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(g-1)}}{h_g}\right)$ and sends it back to the first machine.
    end for
    The first machine computes $\widehat{f}^{(g)}(0)$ based on
        $\widehat{f}^{(g)}(0) = \frac{1}{L} \sum_{k=1}^{L} \widehat{f}^{(g,k)}(0).$
    Transmit $\widehat{f}^{(g)}(0)$ to all local machines.
    for $k = 1, \ldots, L$ do
        The $k$-th machine computes $\widehat{\boldsymbol{\Sigma}}_k \widehat{\boldsymbol{\beta}}^{(g-1)}$ and $\boldsymbol{z}_{nk} = \frac{1}{m} \sum_{i \in \mathcal{H}_k} \boldsymbol{X}_i \widetilde{Y}_i^{(g)}$ based on (17) and sends them back to the first machine.
    end for
    Compute the estimator $\widehat{\boldsymbol{\beta}}^{(g)}$ on the first machine based on (18).
end for
Output: The final estimator $\widehat{\boldsymbol{\beta}}^{(t)}$.
```

developed, e.g., FISTA (Beck and Teboulle, 2009), active set method (Solntsev et al., 2015), and PSSgb (Projected Scaled Subgradient, Gafni-Bertsekas variant, (Schmidt, 2010)). In our experiments, we adopt the PSSgb optimization method for solving (18). We present the entire distributed estimation procedure in Algorithm 1.

For the choice of the initial estimator $\widehat{\boldsymbol{\beta}}_0$, we propose to solve the high-dimensional QR problem using the data on the first machine, i.e.,

$$\widehat{\boldsymbol{\beta}}_0 = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{m} \sum_{i \in \mathcal{H}_1} \rho_\tau(Y_i - \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda_0 |\boldsymbol{\beta}|_1. \tag{20}$$

Note that although this paper uses the (20) as the initial estimator, one can adopt any estimator as $\widehat{\boldsymbol{\beta}}_0$ as long as it satisfies the condition (C6) (see Section 3).

We assume the quantile level τ is pre-specified in Algorithm 1. Our paper mainly focuses on the algorithm for distributed estimation under a general τ and develop the related theoretical results. Different choices of τ correspond to different loss functions we want to use and different parameters we are interested in. The choice of τ to fit the model is a separate topic which clearly depends on the practical problem and the parameters we are interested in. For example, without the covariate $\boldsymbol{X}$ (for briefness), β_0^* is the τ-quantile of Y and the choice of τ depends on what quantile of Y we are interested in. In extreme climate studies, people would like to choose τ as some large values (0.9 and 0.99) or small values (0.1 and 0.01) to evaluate the extreme climate performance. In economic domain, to learn the problem associated with median salary, we can simply set $\tau = 0.5$.

3. Theoretical Results

In this section we provide the theoretical results for our distributed method. We define

$$S = \{0 \le i \le p : \beta_i^* \ne 0\},$$

as the support of $\boldsymbol{\beta}^*$ and $s = |S|$. We assume the following regular conditions.

(C1) The density function of the noise $f(\cdot)$ is bounded and Lipschitz continuous (i.e., $|f(x) - f(y)| \le C_L|x - y|$ for any $x, y \in \mathbb{R}$ and some constant $C_L > 0$). Moreover, we assume $f(0) > c > 0$ for some constant c.

(C2) Suppose that $\boldsymbol{\Sigma} = \mathbb{E}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ satisfies

$$\left\| \boldsymbol{\Sigma}_{S^c \times S} \boldsymbol{\Sigma}_{S \times S}^{-1} \right\|_\infty \le 1 - \alpha, \tag{21}$$

for some $0 < \alpha < 1$. Also assume that $c_0^{-1} \le \Lambda_{\min}(\boldsymbol{\Sigma}) \le \Lambda_{\max}(\boldsymbol{\Sigma}) \le c_0$ for some constant $c_0 > 0$.

(C3) Assume that the kernel function $K(\cdot)$ is integrable with $\int_{-\infty}^{\infty} K(u) \mathrm{d}u = 1$. Moreover, assume that $K(\cdot)$ satisfies $K(u) = 0$ if $|u| \ge 1$. Further, assume $K(\cdot)$ is differentiable and its derivative $K'(\cdot)$ is bounded.

(C4) We assume that the covariate $\boldsymbol{X}$ satisfies the sub-Gaussian condition for some $t > 0$ and $C > 0$,

$$\sup_{|\boldsymbol{\theta}|_2 = 1} \mathbb{E} \exp(t(\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{X})^2) \le C.$$

(C5) The dimension p satisfies $p = O(n^\nu)$ for some $\nu > 0$. The local sample size m on each machine satisfies $m \ge n^c$ for some $0 < c < 1$, and the sparsity level s satisfies $s = O(m^r)$ for some $0 < r < 1/3$.

(C6) The initial estimator $\widehat{\boldsymbol{\beta}}_0$ satisfies $|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*|_2 = O_{\mathbb{P}}(\sqrt{s(\log n)/m})$. Furthermore, assume that $\mathbb{P}(\mathrm{supp}(\widehat{\boldsymbol{\beta}}_0) \subseteq S) \to 1$.

Condition (C1) is a regular condition on the smoothness of the density function $f(\cdot)$. Condition (C2) is the standard irrepresentable condition, which is commonly adopted to establish support recovery in high-dimensional statistics literature (see, e.g., Zhao and Yu (2006); Wainwright (2009); Bühlmann and Van De Geer (2011); Tibshirani et al. (2015)). Condition (C3) is a standard condition on the kernel function $K(\cdot)$ (see an example of $K(\cdot)$ in Section 4). Condition (C4) is a regular condition on the distribution of $\boldsymbol{X}$ while Condition (C5) is on dimension p, local sample size m and sparsity level s. The conditions $m \ge n^c$ for some $0 < c < 1$ and $s = O(m^r)$ make sure that our algorithm achieves the near-oracle convergence rate only using a finite number of iterations (see Eq. (26) below).

Condition (C6) is a condition on the convergence rate and support recovery of the initial estimator. Note that in Algorithm 1, the initial estimator $\widehat{\boldsymbol{\beta}}_0$ is proposed as the solution to the high-dimensional QR problem using data on the first machine. It can be shown that $\widehat{\boldsymbol{\beta}}_0$ in (20) fulfills condition (C6) under conditions (C1), (C2), (C4), (C5) and some regularity conditions (Fan et al., 2014). In addition, we also show that the condition (C6) is satisfied for the proposed estimator for the t-th iteration $\widehat{\beta}^{(t)}$, which serves as the initial estimator for the $(t+1)$-th iteration, in Theorems 2–5. We also note that by $p = O(n^{\nu})$ in (C5), we have that $\log(\max(n,p)) = C_1 \log(n)$ for some constant $C_1 > 0$. Therefore, we will use $\log(n)$ in our convergence rates (instead of $\log(\max(n,p))$) for notational simplicity.

Let $\{a_n\}$ be the convergence rate of the initial estimator, i.e., $|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*|_2 = O_{\mathbb{P}}(a_n)$. By condition (C6) we can assume that $a_n = \sqrt{s(\log n)/m}$. We first provide the convergence rate for $\widehat{\boldsymbol{\beta}}^{(1)}$ after one iteration.

Theorem 2 *Let $|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*|_2 = O_{\mathbb{P}}(a_n)$ and choose the bandwidth $h \asymp a_n$, take*

$$\lambda_n = C_0 \left(\sqrt{\frac{\log n}{n}} + a_n \sqrt{\frac{s \log n}{m}} \right),$$

with C_0 being a sufficiently large constant. Under (C1)-(C6), we have

$$\left| \widehat{\boldsymbol{\beta}}^{(1)} - \boldsymbol{\beta}^* \right|_2 = O_{\mathbb{P}} \left(\sqrt{\frac{s \log n}{n}} + a_n \sqrt{\frac{s^2 \log n}{m}} \right). \tag{22}$$

With the choice of the bandwidth h shrinking at the same rate as a_n, conclusion (22) shows that one iteration enables a refinement of the estimator with its rate improved from a_n to $\max\{\sqrt{s(\log n)/n}, a_n\sqrt{s^2(\log n)/m}\}$ where $\sqrt{s^2(\log n)/m} = o(1)$ by condition (C5). By recursive applications of Theorem 2, we provide the convergence rate for the multi-iteration estimator $\widehat{\boldsymbol{\beta}}^{(t)}$. The next theorem shows that an iterative refinement of the initial estimator will improve the estimation accuracy and achieve a near-oracle rate after a constant number of iterations.

In particular, let us define

$$a_{n,g} = \sqrt{\frac{s \log n}{n}} + s^{(2g+1)/2} \left(\frac{\log n}{m} \right)^{(g+1)/2}, \quad 0 \le g \le t. \tag{23}$$

From Theorem 3 below, we can see that $a_{n,g}$ is the convergence rate of the estimator $\widehat{\boldsymbol{\beta}}^{(g)}$ after g iterations.

Theorem 3 *Assume that the initial estimator $\widehat{\boldsymbol{\beta}}_0$ satisfies $|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*|_2 = O_{\mathbb{P}}(\sqrt{s(\log n)/m})$. Let $h_g \asymp a_{n,g-1}$ for $1 \le g \le t$, and take*

$$\lambda_{n,g} = C_0 \left(\sqrt{\frac{\log n}{n}} + a_{n,g-1} \sqrt{\frac{s \log n}{m}} \right), \tag{24}$$

with C_0 being a sufficiently large constant. Under (C1)-(C6), we have

$$\left| \widehat{\boldsymbol{\beta}}^{(t)} - \boldsymbol{\beta}^* \right|_2 = O_{\mathbb{P}} \left(\sqrt{\frac{s \log n}{n}} + s^{(2t+1)/2} \left(\frac{\log n}{m} \right)^{(t+1)/2} \right). \tag{25}$$

It can be shown that when the iteration number t is sufficiently large, i.e.,

$$t \geq \frac{\log(n/m)}{\log(c_0 m/(s^2 \log n))}, \quad \text{for some } c_0 > 0, \tag{26}$$

the second term in (25) is dominated by the first term, and the convergence rate in (25) becomes $|\widehat{\boldsymbol{\beta}}^{(t)} - \boldsymbol{\beta}^*|_2 = O_{\mathbb{P}}(\sqrt{s(\log n)/n})$. We note that this rate matches the convergence rate of the ℓ_1-regularized QR estimator in a single machine setup (see Belloni and Chernozhukov (2011)). Moreover, it nearly matches the oracle convergence rate $\sqrt{s/n}$ (upto a logarithmic factor) when the support of $\boldsymbol{\beta}^*$ is known. We also note that the conditions $m \geq n^c$ and $s = o(m^{1/3})$ in (C5) ensure that the right hand side of (26) is bounded by a constant, which implies that a constant number of iterations would guarantee a near-oracle rate of $\widehat{\boldsymbol{\beta}}^{(t)}$.

The following theorems provide results on support recovery of the proposed estimators $\widehat{\boldsymbol{\beta}}^{(1)}$ and $\widehat{\boldsymbol{\beta}}^{(t)}$. Recall $S = \{j : \beta_j^* \neq 0\}$ is the support of $\boldsymbol{\beta}^*$. Let $\widehat{\boldsymbol{\beta}}^{(1)} = (\widehat{\beta}_0^{(1)}, \widehat{\beta}_1^{(1)}, \ldots, \widehat{\beta}_p^{(1)})^{\mathrm{T}}$ and

$$\widehat{S}^{(1)} = \left\{ j : \widehat{\beta}_j^{(1)} \neq 0 \right\}.$$

Theorem 4 *Assume that the conditions in Theorem 2 hold.*

(i) We have $\widehat{S}^{(1)} \subseteq S$ with probability tending to one.

(ii) In addition, suppose that for a sufficiently large constant $C > 0$,

$$\min_{j \in S} |\beta_j^*| \geq C \|\boldsymbol{\Sigma}_{S \times S}^{-1}\|_\infty \left(\sqrt{\frac{\log n}{n}} + a_n \sqrt{\frac{s \log n}{m}} \right). \tag{27}$$

Then we have $\widehat{S}^{(1)} = S$ with probability tending to one.

Based on Theorem 4, we can further obtain the support recovery result for $\widehat{\boldsymbol{\beta}}^{(t)}$, which requires a weaker condition on $\min_{j \in S} \left|\beta_j^*\right|$. Denote $\widehat{\boldsymbol{\beta}}^{(t)} = (\widehat{\beta}_0^{(t)}, \widehat{\beta}_1^{(t)}, \ldots, \widehat{\beta}_p^{(t)})^{\mathrm{T}}$ and

$$\widehat{S}^{(t)} = \left\{ j : \widehat{\beta}_j^{(t)} \neq 0 \right\}.$$

Theorem 5 *Assume the conditions in Theorem 3 hold.*

(i) We have $\widehat{S}^{(t)} \subseteq S$ with probability tending to one.

(ii) In addition, suppose that for a sufficiently large constant $C > 0$,

$$\min_{j \in S} |\beta_j^*| \geq C \|\boldsymbol{\Sigma}_{S \times S}^{-1}\|_\infty \left(\sqrt{\frac{\log n}{n}} + s^t \left(\frac{\log n}{m}\right)^{(t+1)/2} \right). \tag{28}$$

Then we have $\widehat{S}^{(t)} = S$ with probability tending to one.

Note that the "beta-min" condition gets weaker as t increases. When t satisfies (26), the condition (28) will reduce to $\min_{j\in S}\left|\beta_j^*\right| \geq C\|\boldsymbol{\Sigma}_{S\times S}^{-1}\|_\infty\sqrt{\frac{\log n}{n}}$, which matches the rate of the lower bound for the "beta-min" condition in Lasso in a single machine setting (see Wainwright (2009)).

Furthermore, we state the results in both Theorem 4 and 5 by a high-probability statement "with probability tending to one". The convergence rate actually can be represented as $1-q_n$, where $q_n = O(1-\mathbb{P}(\mathrm{supp}(\widehat{\boldsymbol{\beta}}_0)\subseteq S)) + O(n^{-\gamma})$ is a small quantity goes to 0 when both n and p go to ∞. More specifically, the convergence rate depends on the convergence rate $\mathbb{P}(\mathrm{supp}(\widehat{\boldsymbol{\beta}}_0)\subseteq S)\to 1$ for the initial estimator $\widehat{\boldsymbol{\beta}}_0$. Below we further provide two remarks on our method.

Remark 6 *It is worthwhile noting that we assume the data is evenly split only for the ease of discussions. In fact, the local sample size m in our theoretical results is the sample size on the first machine in Algorithm 1 (a.k.a. the central machine in distributed computing). As long as the sample size m on the first machine is specified, our method does not depend on the partition of the entire dataset.*

Remark 7 *We note that the proposed estimator can be generalized to the case when the noise e and the covariates $\boldsymbol{X}$ are not independent. More specifically, without the independence assumption, we assume $\mathbb{P}(e\leq 0|\boldsymbol{X}) = \tau$ for some specified $\tau\in(0,1)$. The Hessian matrix becomes $\boldsymbol{H}(\boldsymbol{\beta}^*) = \mathbb{E}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}f(0|\boldsymbol{X}))$. Although $\boldsymbol{H}(\boldsymbol{\beta}^*)$ no longer takes the form of $\boldsymbol{\Sigma}f(0)$ when the noise depends on covariates, it can be approximate by*

$$\boldsymbol{D}_h(\boldsymbol{\beta}_0) = \mathbb{E}\left(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\frac{1}{h}K\left(\frac{Y-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}_0}{h}\right)\right),$$

for a positive kernel function $K(\cdot)$ (i.e., $K(x)>0$ for all x). Let $\widehat{\boldsymbol{\beta}}_0$ be an initial estimator of $\boldsymbol{\beta}^*$. Given n i.i.d. *samples $(\boldsymbol{X}_i, Y_i)$ from (1), for each $1\leq i\leq n$, we construct the following quantities:*

$$\gamma_{i,h} = \sqrt{\frac{1}{h}K\left(\frac{Y_i-\boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_0}{h}\right)}, \quad \widetilde{\boldsymbol{X}}_{i,h} = \gamma_{i,h}\boldsymbol{X}_i, \quad \widehat{\boldsymbol{D}}_h = \frac{1}{n}\sum_{i=1}^n \widetilde{\boldsymbol{X}}_{i,h}\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}},$$

$$\widetilde{Y}_{i,h} = \widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_0 - \frac{\mathbb{I}[Y_i\leq \boldsymbol{X}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_0]-\tau}{\gamma_{i,h}}.$$

Then, we can construct the pooled estimator (i.e., the counterpart of (9)) by solving the following Lasso problem with both transformed input $\widetilde{\boldsymbol{X}}_{i,h}$ and response $\widetilde{Y}_{i,h}$:

$$\widehat{\boldsymbol{\beta}} = \mathop{\arg\min}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}\Big\{\frac{1}{2n}\sum_{i=1}^n(\widetilde{Y}_{i,h}-\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}\boldsymbol{\beta})^2 + \lambda_n|\boldsymbol{\beta}|_1\Big\}. \tag{29}$$

Using a similar distributed approach described in Section 2.2, the pooled estimator in Eq. (29) can be extended into a distributed estimator.

Although the extension to the dependent case seems relatively straightforward, the nonparametric estimation of the conditional density $f(0|\boldsymbol{X})$ has the issue of "curse of dimensionality", especially when $\boldsymbol{X}$ is high-dimensional. Without any strong assumption on $f(0|\boldsymbol{X})$, it requires a huge number of local samples to construct an accurate estimator $\widehat{\boldsymbol{D}}_{1,h} = \frac{1}{m}\sum_{i\in\mathcal{H}_1}\widetilde{\boldsymbol{X}}_{i,h}\widetilde{\boldsymbol{X}}_{i,h}^{\mathrm{T}}$ in the distributed implementation. We leave more investigation of the dependent noise case to future work.

4. Simulation Study

In this section, we report the simulation studies to illustrate the performance of our distributed REL.

4.1. Simulation Setup

We consider the following linear model

$$Y_i = \boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\beta}^* + e_i, \quad i = 1, 2, \ldots, n,$$

where $\boldsymbol{X}_i^{\mathrm{T}} = (1, X_{i,1}, \ldots, X_{i,p})$ is a $(p+1)$-dimensional covariate vector and $(X_{i,1}, \ldots, X_{i,p})$s are drawn *i.i.d.* from a multivariate normal distribution $N(0, \boldsymbol{\Sigma})$. The covariance matrix $\boldsymbol{\Sigma}$ is constructed by $\boldsymbol{\Sigma}_{ij} = 0.5^{|i-j|}$ for $1 \le i, j \le p$. We fix the dimension $p = 500$ and choose the loss function to be the QR loss with quantile level $\tau = 0.3$. Note that other choices of τ lead to similar results in the experiment. We provide additional experimental results for $\tau = 0.5$ in the appendix. Let s be the sparsity level and the true coefficient is set to

$$\boldsymbol{\beta}^* = (\frac{10}{s}, \frac{20}{s}, \frac{30}{s}, \ldots, \frac{10(s-1)}{s}, 10, 0, 0 \ldots, 0).$$

We consider the following three noise distributions:

1. Normal: the noise $e_i \sim \mathrm{N}(0,1)$.

2. Cauchy: the noise $e_i \sim \mathrm{Cauchy}(0,1)$.

3. Exponential: the noise $e_i \sim \exp(1)$.

We note that the variance of the Cauchy distribution is infinite. The initial estimator is computed by directly solving the ℓ_1-regularized QR optimization using only the data on the first machine (see Eq. (19)). At each iteration, the constant C_0 in the regularization parameter $\lambda_{n,g}$ in (24) is chosen by validation. In particular, we choose C_0 to minimize the quantile loss on an independently generated validation dataset with the sample size n. Moreover, we could also apply cross-validation or an information criterion such as BIC to choose λ_n.

For the choice of the kernel function $K(\cdot)$, we use a biweight kernel function

$$K(x) = \begin{cases} 0, & \text{if} \quad x \le -1, \\ -\frac{315}{64}x^6 + \frac{735}{64}x^4 - \frac{525}{64}x^2 + \frac{105}{64}, & \text{if} \quad -1 \le x \le 1, \\ 0, & \text{if} \quad x \ge 1. \end{cases}$$

It is easy to verify that $K(\cdot)$ satisfies the condition (C3). We also note that other choices of $K(\cdot)$ provide similar results.

From Theorem 2 and 3 in Section 3, the bandwidth is set to $h_g = c a_{n,g-1}$ for some constant $c > 0$, where $a_{n,g-1}$ is defined in (23). In our simulation study, we choose $h_g = \sqrt{\frac{s \log n}{n}} + s^{-1/2}\left(c_0 \frac{s^2 \log n}{m}\right)^{(g+1)/2}$ (i.e., set the constant $c = 1$) for convenience. Note that the constant c_0 is used to ensure that $\frac{s^2 \log n}{m} < 1$, and we set $c_0 = 0.1$ in the following experiments. In fact, our algorithm is quite robust with respect to the choice of the bandwidth (see the sensitivity analysis in Section 4.5). All the results reported in this section are average of 100 independent runs of simulations.

We compare the performance of the proposed distributed REL (dist REL for short) with other two approaches:

1. Averaging divide-and-conquer (Avg-DC) which computes the ℓ_1-regularized QR (see Eq. (3)) on each local machine and combines the local estimators by taking the average.

2. Robust estimator with Lasso (REL) on a single machine with pooled data (see Eq. (9)), which is denoted by pooled REL.

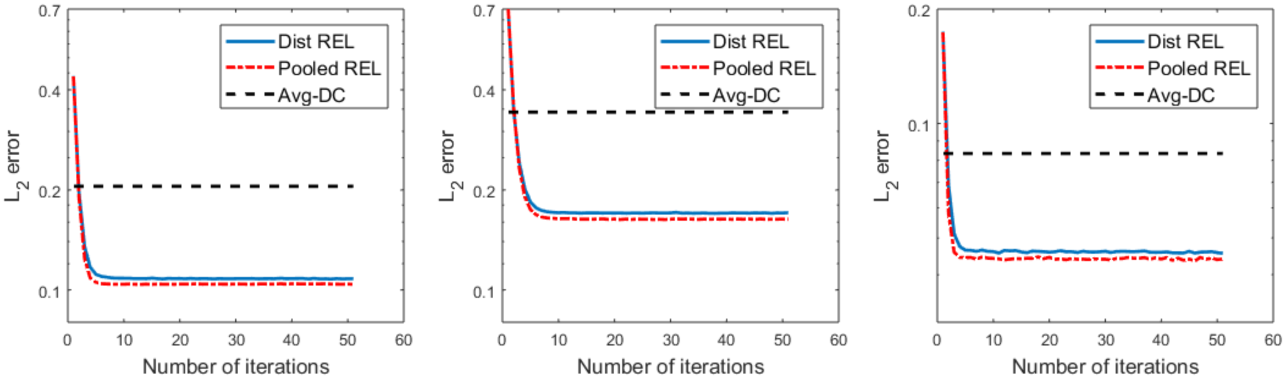
Note that the ℓ_1-regularized QR estimator in (2) and the de-biased averaging divide-and-conquer estimator (see Zhao et al. (2014)) are not included in most comparisons because they are computationally very expensive to be implemented in our setting, with large n and p. Moreover, the de-biased estimator generates a dense estimated coefficient due to the de-biasing procedure. In the experiment on computation efficiency, we compare the running time of our method to the ℓ_1-regularized QR estimator. The result shows that our method achieves a similar performance as the ℓ_1-regularized QR estimator and it is computationally much more efficient.

4.2. Effect of the Number of Iterations

We first show the performance of our distribute REL by varying the number of iterations. We fix the sample size $n = 10000$, local sample size $m = 500$, the sparsity level $s = 20$ and dimension $p = 500$. We plot the ℓ_2-error from the true QR coefficients versus the number of iterations. Since the Avg-DC only requires one-shot communication, we use a horizontal line to show its performance. The results are shown in Figure 1. From the result, both pooled REL and distributed REL outperform the Avg-DC algorithm and become stable after a few iterations. Therefore, for the rest of the experiments in this section, we use 50 as the number of iterations in the algorithm. Moreover, the distributed REL almost matches the performance of pooled REL for all three noises.

4.3. Effect of the QR Loss Under Heavy-Tailed Noise

We study the effect of the QR loss in the presence of heavy-tailed noise. We compare with the standard Lasso estimator in a single machine setting with pooled data. We vary the sample size n and compute the F_1-score and the ℓ_2-error for the distributed REL, Pooled



(a) Normal noise (b) Cauchy noise (c) Exponential noise

Figure 1: The ℓ_2-error from the true QR coefficient versus the number of iterations. The sample size n is fixed to $n = 10000$ and the local sample size m is 500.

Table 1: The F_1-score and ℓ_2-error of the distributed REL, pooled REL, Avg-DC, and Lasso estimator under different sample size n. Noises are generated from normal distribution. The local sample size is fixed to $m = 500$.

n	Dist REL		Pooled REL		Avg-DC		Lasso	
	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error
2500	0.90	0.189	0.83	0.183	0.23	0.255	1.00	0.161
5000	0.95	0.138	0.91	0.132	0.14	0.221	1.00	0.113
10000	0.97	0.102	0.93	0.097	0.10	0.203	1.00	0.079
15000	0.98	0.085	0.96	0.083	0.09	0.196	1.00	0.065
20000	0.99	0.073	0.96	0.069	0.08	0.192	1.00	0.056
25000	0.99	0.067	0.97	0.050	0.08	0.196	1.00	0.050

REL, Avg-DC, and the Lasso estimator. The F_1-score is defined as

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2}\right)^{-1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

which is commonly used as an evaluation of support recovery (note that F_1-score=1 implies perfect support recovery). In Table 1, 2 and 3, we report the results for all three types of noises. As expected, when the noise is normal, the Lasso estimator has smaller ℓ_2-error and better support recovery. However, when the noise has a slightly heavier tail (e.g., exponential noise), both the distributed REL and pooled REL outperform the Lasso estimator in ℓ_2-error. In the case of heavy-tailed noise (e.g., Cauchy noise), the Lasso approach completely fails with very large ℓ_2-errors while the distributed REL is much better in both ℓ_2-error and support recovery. It is clear that the Lasso estimator is not robust to heavy-tailed noises, and therefore we omit the Lasso estimator in the rest of the simulation studies.

Table 2: The F_1-score and ℓ_2-error of the distributed REL, pooled REL, Avg-DC, and Lasso estimator under different sample size n. Noises are generated from Cauchy distribution. The local sample size is fixed to $m = 500$.

n	Dist REL		Pooled REL		Avg-DC		Lasso	
	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error
2500	0.84	0.320	0.75	0.312	0.25	0.436	0.25	151.4
5000	0.92	0.229	0.85	0.221	0.16	0.380	0.26	138.8
10000	0.96	0.168	0.89	0.160	0.11	0.349	0.27	128.3
15000	0.98	0.139	0.92	0.132	0.09	0.338	0.25	132.1
20000	0.97	0.118	0.93	0.113	0.08	0.329	0.26	121.0
25000	0.98	0.107	0.94	0.101	0.08	0.330	0.23	120.8

Table 3: The F_1-score and ℓ_2-error of the distributed REL, pooled REL, Avg-DC, and Lasso estimator under different sample size n. Noises are generated from exponential distribution. The local sample size is fixed to $m = 500$.

n	Dist REL		Pooled REL		Avg-DC		Lasso	
	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error
2500	0.96	0.093	0.91	0.089	0.25	0.115	1.00	0.102
5000	0.98	0.069	0.92	0.066	0.15	0.101	1.00	0.094
10000	0.99	0.051	0.96	0.048	0.10	0.092	1.00	0.069
15000	0.99	0.043	0.97	0.040	0.09	0.089	1.00	0.054
20000	1.00	0.037	0.98	0.034	0.08	0.086	1.00	0.048
25000	0.99	0.033	0.98	0.031	0.08	0.087	1.00	0.043

Table 4: The ℓ_2-error, precision, and recall of the three estimators under different combinations of the sample size n and local sample size m. Noises are generated from normal distribution.

m		200			500			1000		
n		5000	10000	20000	5000	10000	20000	5000	10000	20000
Pooled REL	Precision	0.79	0.85	0.92	0.79	0.89	0.93	0.78	0.85	0.92
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.136	0.098	0.071	0.138	0.101	0.073	0.135	0.100	0.072
Dist REL	Precision	0.98	0.99	1.00	0.91	0.95	0.98	0.83	0.89	0.95
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.154	0.111	0.081	0.142	0.105	0.076	0.137	0.102	0.074
Avg DC	Precision	0.05	0.04	0.04	0.08	0.06	0.05	0.13	0.08	0.06
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.348	0.328	0.314	0.225	0.205	0.199	0.180	0.156	0.145
CSL	Precision	0.86	0.85	0.88	0.08	1.00	1.00	1.00	1.00	1.00
	Recall	0.95	0.93	0.94	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.480	0.455	0.452	0.218	0.201	0.190	0.154	0.141	0.098

Another interesting phenomena revealed in Tables 1-3 is that, in terms of the F_1-score, the distributed REL is slightly better than pooled REL. This is indeed affected by the selection of regularization parameter λ_n. According to our Theorem 2, we set λ_n for the first round on the order of $s \log n/m$, where m is the local sample size and n the total sample size. For the pooled estimator where $m = n$, this term becomes $s \log n/n$, which becomes smaller. Therefore, our distributed estimator has already eliminated many features for the first round due to a larger regularization parameter, which leads to a slightly better precision. It is noted that this also happens in the following experiments.

4.4. Effect of Sample Size and Local Sample Size

In this section, we investigate how the performance of the distributed REL changes with the total sample size n and the local sample size m. We also compare our estimator with the Communication-efficient Surrogate Likelihood (CSL) estimator proposed in Jordan et al. (2019). The original method in Jordan et al. (2019) requires second-order differentiable loss functions, which is not directly applicable to quantile loss function. Thus, we adopt a smoothing technique to smooth the QR loss function as in Horowitz (1998); Chen et al. (2019). We fix sparsity level $s = 20$, $p = 500$, and vary the sample size $n \in \{5000, 10000, 20000\}$ and the local sample size $m \in \{200, 500, 1000\}$. The precision, recall of the support recovery and the ℓ_2-error are reported for each estimator. The results are shown in Table 4, 5 and 6.

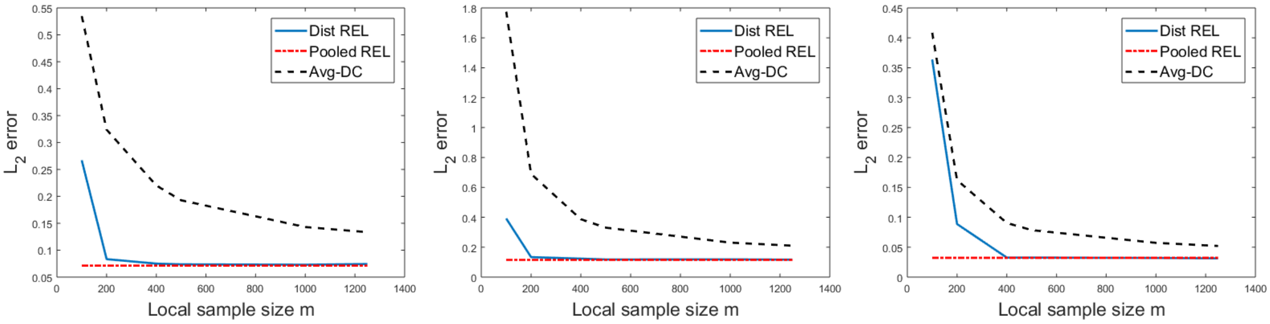
From the results, we observe that both distributed REL and pooled REL outperform the Avg-DC algorithm and CSL estimator in all settings. The ℓ_2-error of the distributed REL improves as the local sample size m grows and it becomes close to pooled REL when m is large. This is expected since the pooled REL is a special case of distributed REL with $m = n$. We also observe that the precision and recall of the distributed REL are both close to 1, which indicates good support recovery. In particular, the recall of our

Table 5: The ℓ_2-error, precision, and recall of the three estimators under different combinations of the sample size n and local sample size m. Noises are generated from Cauchy distribution.

m		200			500			1000		
n		5000	10000	20000	5000	10000	20000	5000	10000	20000
Pooled REL	Precision	0.72	0.84	0.89	0.75	0.82	0.88	0.70	0.81	0.87
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.220	0.159	0.118	0.221	0.161	0.116	0.221	0.156	0.114
Dist REL	Precision	0.98	0.99	1.00	0.86	0.91	0.95	0.76	0.87	0.92
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.251	0.181	0.134	0.230	0.169	0.122	0.223	0.158	0.117
Avg DC	Precision	0.05	0.04	0.04	0.08	0.06	0.04	0.14	0.08	0.06
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.704	0.671	0.667	0.375	0.355	0.332	0.291	0.245	0.235
CSL	Precision	0.09	0.12	0.12	0.28	0.35	0.48	0.64	0.77	0.89
	Recall	0.91	0.93	0.90	0.97	0.97	0.98	0.98	0.98	0.99
	ℓ_2-error	0.834	0.790	0.728	0.324	0.327	0.312	0.255	0.195	0.171

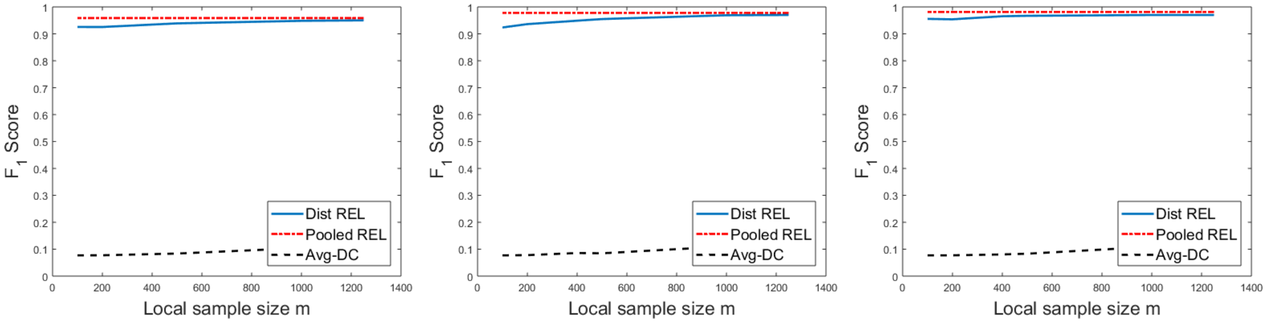
Table 6: The ℓ_2-error, precision, and recall of the three estimators under different combinations of the sample size n and local sample size m. Noises are generated from exponential distribution.

m		200			500			1000		
n		5000	10000	20000	5000	10000	20000	5000	10000	20000
Pooled REL	Precision	0.90	0.98	0.98	0.88	0.94	0.96	0.86	0.93	0.96
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.060	0.045	0.031	0.059	0.042	0.032	0.059	0.042	0.030
Dist REL	Precision	1.00	1.00	1.00	0.95	0.98	0.99	0.91	0.95	0.98
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.076	0.061	0.043	0.062	0.044	0.034	0.060	0.042	0.031
Avg DC	Precision	0.05	0.04	0.04	0.07	0.06	0.04	0.15	0.09	0.05
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.168	0.162	0.154	0.090	0.084	0.079	0.072	0.062	0.054
CSL	Precision	0.86	0.85	0.88	0.08	1.00	1.00	1.00	1.00	1.00
	Recall	0.95	0.93	0.94	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.480	0.455	0.452	0.218	0.201	0.190	0.154	0.141	0.098



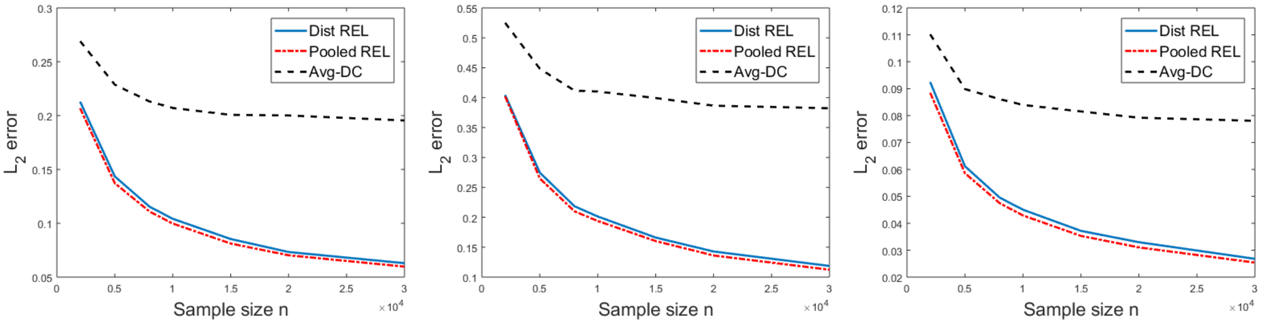
(a) Normal noise (b) Cauchy noise (c) Exponential noise

Figure 2: The ℓ_2-error from the true QR coefficient versus the local sample size m, with the total sample size fixed to $n = 20000$.



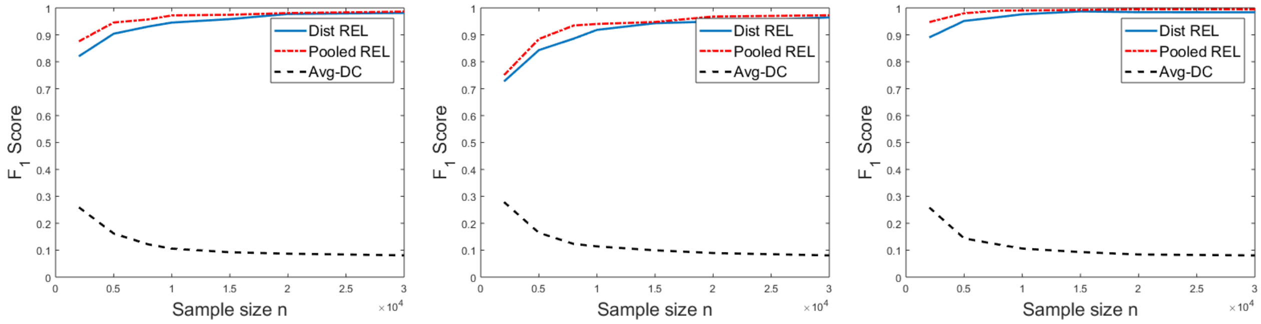
(a) Normal noise (b) Cauchy noise (c) Exponential noise

Figure 3: The F_1-score versus the local sample size m, with the total sample size fixed to $n = 20000$.



(a) Normal noise (b) Cauchy noise (c) Exponential noise

Figure 4: The ℓ_2-error from the true QR coefficient versus the sample size n, with the local sample size fixed to $m = 500$.



(a) Normal noise (b) Cauchy noise (c) Exponential noise

Figure 5: The F_1-score versus the sample size n, with the local sample size fixed to $m = 500$.

distributed REL is always 1, implying that all the relevant variables are selected. The precision of our method is close to 1, which indicates that only a very small number of irrelevant variables are selected. On the other hand, the precision of Avg-DC is very small because the averaging procedure results in a dense estimator, especially when m is small. In addition, the performance of CSL estimator heavily depends on m. For example, for Cauchy error distribution in Table 5, a smaller m leads to a relatively poor performance.

For better visualization, with the sample size $n = 20000$ fixed, we vary the local sample size m and plot the ℓ_2-error and F_1-score of pooled REL, distributed REL and Avg-DC estimator. The results are presented in Figure 2 and 3. Similarly, in Figure 4 and 5, we fix the local sample size $m = 500$ and vary the total sample size n.

From Figure 2 we can see that the ℓ_2-error of distributed REL is close to that of pooled REL when m is not too small, and both of them outperform the Avg-DC estimator. From Figure 4 we observe that the ℓ_2-error of distributed REL is close to that of pooled REL and both errors decrease as the sample size n becomes large. However, the ℓ_2-error of the Avg-DC estimator stays large and fails to converge as the sample size n increases. From

Table 7: The F_1-score and ℓ_2-error of the distributed REL, pooled REL, and Avg-DC under different sample size n and choices of bandwidth constant c. Local sample size $m = 500$. Noises are generated from Cauchy distribution.

n	c	Dist REL		Pooled REL		Avg-DC	
		F_1-score	ℓ_2-error	F_1-score	ℓ_2-error	F_1-score	ℓ_2-error
5000	0.5	0.99	0.249	0.96	0.236	0.17	0.377
10000	0.5	1.00	0.183	0.99	0.171	0.12	0.356
20000	0.5	0.99	0.130	0.99	0.123	0.09	0.348
5000	1	0.99	0.253	0.96	0.241	0.16	0.373
10000	1	0.99	0.179	0.98	0.170	0.11	0.345
20000	1	1.00	0.125	0.98	0.117	0.09	0.328
5000	2	0.99	0.259	0.97	0.245	0.16	0.38
10000	2	1.00	0.188	0.98	0.177	0.11	0.347
20000	2	1.00	0.131	0.99	0.124	0.09	0.332
5000	5	0.99	0.255	0.97	0.239	0.16	0.378
10000	5	1.00	0.185	0.98	0.173	0.11	0.349
20000	5	1.00	0.138	0.98	0.124	0.09	0.339
5000	10	1.00	0.270	0.99	0.252	0.16	0.382
10000	10	1.00	0.194	0.99	0.180	0.1	0.346
20000	10	1.00	0.136	0.98	0.121	0.09	0.331

Figure 3 and 5 we can see that the F_1-score of both distributed REL and pooled REL are close to 1, while the Avg-DC approach clearly fails in support recovery in high-dimensional settings.

4.5. Sensitivity Analysis for the Bandwidth

In this section, we study the sensitivity of the scaling constant in the bandwidth of the proposed REL. Recall that the bandwidth is $h = ca_{n,g}$ where $a_{n,g}$ is defined in (23) with $c > 0$ being the scaling constant. We vary the sample size n and the constant c from 0.5 to 10 and compute the F_1-score and the ℓ_2-error of the distributed REL, pooled REL, and the Avg-DC estimator. Due to space limitations, we report the Cauchy noise case as an example. For other noises, the performance is even less sensitive. The results are shown in Table 7.

From Table 7, we observe that both distributed REL and pooled REL exhibit good performance under all choices of bandwidth constant. Therefore even under a suboptimal choice of bandwidth constant, the distributed REL still achieves small ℓ_2-error and good support recovery.

4.6. Effect of the Sparsity

In this section we investigate how the performance of the distributed REL algorithm changes with the sparsity level of the true coefficient $\boldsymbol{\beta}^*$. We fix the sample size $n = 10000$ and the local sample size $m = 500$, and we set the constant c_0 in h_g to be 0.01. Recall that the true

Table 8: The ℓ_2-error, precision, and recall of the three estimators with different sparsity level s. Noises are generated from normal distribution. The local sample size is fixed to $m = 500$.

Sparsity s		5	10	20	30	50	100
Pooled REL	Precision	0.98	0.96	0.86	0.82	0.73	0.66
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.063	0.080	0.096	0.117	0.141	0.191
	Relative ℓ_2-error($\times 10^{-2}$)	0.426	0.408	0.360	0.361	0.341	0.329
Dist REL	Precision	1.00	0.98	0.94	0.93	0.91	0.88
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.065	0.082	0.101	0.123	0.150	0.202
	Relative ℓ_2-error($\times 10^{-2}$)	0.441	0.418	0.379	0.379	0.363	0.347
Avg DC	Precision	0.02	0.03	0.06	0.08	0.11	0.20
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.147	0.175	0.204	0.243	0.280	0.368
	Relative ℓ_2-error($\times 10^{-2}$)	0.988	0.890	0.760	0.751	0.675	0.633

coefficient is set to be

$$\boldsymbol{\beta}^* = (\frac{10}{s}, \frac{20}{s}, \frac{30}{s}, \ldots, \frac{10(s-1)}{s}, 10, 0, 0 \ldots, 0).$$

We vary the sparsity level s in $\{5, 10, 20, 30, 50, 100\}$ and report the precision, recall and ℓ_2-error. Since the ℓ_2-norm of the true coefficient $\boldsymbol{\beta}^*$ changes with the sparsity level s, we also report the relative ℓ_2-error which is defined by $|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2 / |\boldsymbol{\beta}^*|_2$. The results are shown in Table 8, 9 and 10.

From the result, we can observe that the ℓ_2-errors of all three estimators become larger as the sparsity level s increases and the distributed REL algorithm performs much better than the Avg-DC algorithm. Moreover, the performance of the distributed REL is very close to the performance of the pooled REL.

4.7. Computation Time Comparison

We further study the computation efficiency of our proposed estimator. We fix the local sample size m, dimension p, and vary the sample size n. In Table 11, we report the F_1-score, ℓ_2-error, and the computation time of distributed REL, pooled REL, Avg-DC, and the ℓ_1-regularized QR estimator. To solve the ℓ_1-regularized QR estimator, we formulate it into a standard linear programming problem (LP) and solve it by Gurobi (Gurobi Optimization, 2020), which is the state-of-the-art LP solver. We implement the three distributed algorithms (distributed REL, pooled REL and Avg-DC) in a fully synchronized distributed setting.

From Table 11 we can see that the distributed REL is much faster than the ℓ_1-regularized QR estimator. In fact, for larger sample size (i.e., $n > 20000$), we cannot implement the ℓ_1-regularized QR method due to memory and computation time issues. We also note that the computation time of the pooled REL is similar to the distributed version. This is because for the comparison propose, simulated datasets can still be fully stored in memory, and thus

Table 9: The ℓ_2-error, precision, and recall of the three estimators with different sparsity level s. Noises are generated from Cauchy distribution. The local sample size is fixed to $m = 500$.

Sparsity s		5	10	20	30	50	100
Pool QR	Precision	0.95	0.91	0.79	0.73	0.66	0.64
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.103	0.129	0.156	0.186	0.230	0.318
	Relative ℓ_2-error($\times 10^{-2}$)	0.696	0.656	0.581	0.574	0.555	0.547
Dist QR	Precision	0.97	0.95	0.91	0.87	0.86	0.84
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.105	0.132	0.163	0.194	0.239	0.330
	Relative ℓ_2-error($\times 10^{-2}$)	0.709	0.674	0.608	0.598	0.578	0.567
Avg DC	Precision	0.02	0.04	0.06	0.07	0.11	0.20
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.264	0.319	0.347	0.419	0.542	0.885
	Relative ℓ_2-error($\times 10^{-2}$)	1.779	1.628	1.295	1.293	1.308	1.252

Table 10: The ℓ_2-error, precision, and recall of the three estimators with different sparsity level s. Noises are generated from exponential distribution. The local sample size is fixed to $m = 500$.

Sparsity s		5	10	20	30	50	100
Pooled REL	Precision	0.97	0.97	0.95	0.92	0.87	0.79
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.026	0.034	0.043	0.049	0.062	0.080
	Relative ℓ_2-error($\times 10^{-2}$)	0.178	0.171	0.160	0.151	0.149	0.138
Dist REL	Precision	0.99	0.99	0.98	0.98	0.98	0.99
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.027	0.035	0.045	0.052	0.066	0.092
	Relative ℓ_2-error($\times 10^{-2}$)	0.185	0.180	0.169	0.161	0.160	0.158
Avg DC	Precision	0.02	0.04	0.05	0.07	0.11	0.20
	Recall	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.054	0.065	0.083	0.099	0.113	0.151
	Relative ℓ_2-error($\times 10^{-2}$)	0.365	0.329	0.311	0.305	0.273	0.260

Table 11: The F_1-score, ℓ_2-error, and computation time of the distributed REL, pooled REL, Avg-DC, and ℓ_1-regularized QR estimator under different sample size n. Noises are generated from Cauchy distribution. The local sample size is fixed to $m = 500$.

n	Dist REL			Pooled REL		
	F_1-score	ℓ_2-error	Time	F_1-score	ℓ_2-error	Time
5000	0.95	0.137	0.40	0.90	0.132	0.44
10000	0.97	0.099	0.42	0.92	0.095	0.45
15000	0.98	0.083	0.42	0.95	0.080	0.47
20000	0.99	0.074	0.44	0.96	0.071	0.48
n	Avg-DC			ℓ_1-QR		
	F_1-score	ℓ_2-error	Time	F_1-score	ℓ_2-error	Time
5000	0.15	0.223	2.82	0.95	0.132	159.6
10000	0.10	0.202	3.08	0.97	0.091	576.1
15000	0.09	0.198	3.07	0.98	0.077	1223.1
20000	0.08	0.192	3.15	0.99	0.068	2059.3

pooled REL takes the advantage of solving the entire optimization problem in memory. For large-scale datasets that cannot be stored in memory, the pool REL is no longer applicable.

5. Conclusions and Future Directions

In this paper, we address the problem of distributed estimation for high-dimensional linear model with the presence of heavy-tailed noise. The proposed method achieves the same convergence rate as the ideal case with pooled data. Furthermore, we establish the support recovery guarantee of the proposed method. One key insight from this work is that a non-smooth loss can be transformed into a smooth one by constructing a new response. Our method is essentially an iterative refinement approach in a distributed environment, which is superior to the averaging divide-and-conquer scheme.

One important future direction is to further investigate the inference problem. We note that Zhao et al. (2014) first provide the inference result based on averaging de-biased QR local estimators. As we mentioned, this approach might suffer from heavy computational cost and requires a condition on the number of machines. It would be interesting to develop computationally efficient inference approaches without any restriction on the number of machines. Moreover, the idea of transforming to ℓ_1-regularized least-squares problem and the iterative distributed implementation can be generalized other high-dimensional problems, e.g., ℓ_1-regularized Huber regression in robust statistics. Our algorithm can also be generalized to handle other sparsity-inducing penalties, such as SCAD or MCP (Fan and Li, 2001; Zhang, 2010). Deriving the corresponding theoretical results for other sparsity-inducing penalties would be another interesting future direction.

Acknowledgments

Xi Chen and Zhuoyi Yang are supported by NSF via IIS-1845444. Weidong Liu is supported by National Program on Key Basic Research Project (973 Program, 2018AAA0100704), NSFC Grant No. 11825104 and 11690013, Youth Talent Support Program, and a grant from Australian Research Council. Xiaojun Mao is supported by Shanghai Sailing Program 19YF1402800.

Appendix A. Proof of Results

In this section, we provide the proofs of our main results and some technical lemmas.

A.1. Proof of Proposition 1

Proposition 1 1 *Assume the following conditions hold*

$$|\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{b}|_\infty \leq \lambda_n/2,$$

$$\min_{\delta:|\delta|_1 \leq c_1\sqrt{s}|\delta|_2} \frac{\delta^{\mathrm{T}}\boldsymbol{A}\delta}{|\delta|_2^2} \geq c_2, \quad c_1, c_2 > 0.$$

where s is the sparsity of $\boldsymbol{\beta}^*$, i.e., $s = \sum_{j=0}^{p} \mathbb{I}[\beta_j^* \neq 0]$. Then we have

$$|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2 \leq c\sqrt{s}\lambda_n,$$

for some constant $c > 0$.

Proof We first show that $|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_1 \leq 4\sqrt{s}|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2$. Let S be the support of $\boldsymbol{\beta}$. By the definition of $\widehat{\boldsymbol{\beta}}$, we have

$$\begin{aligned}\frac{1}{2}\widehat{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{A}\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{b} - (\frac{1}{2}\boldsymbol{\beta}^{*\mathrm{T}}\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{\beta}^{*\mathrm{T}}\boldsymbol{b}) \leq & \lambda_n(|\boldsymbol{\beta}^*|_1 - |\widehat{\boldsymbol{\beta}}|_1) \\ = & \lambda_n(|\boldsymbol{\beta}_S^*|_1 - |\widehat{\boldsymbol{\beta}}_S|_1 - |\widehat{\boldsymbol{\beta}}_{S^C}|_1) \\ \leq & \lambda_n|(\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}})_S|_1 - \lambda_n|(\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}})_{S^C}|_1.\end{aligned}$$

Since $\boldsymbol{A}$ is non-negative definite, we have

$$\begin{aligned}\frac{1}{2}\widehat{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{A}\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{b} - (\frac{1}{2}\boldsymbol{\beta}^{*\mathrm{T}}\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{\beta}^{*\mathrm{T}}\boldsymbol{b}) \geq & (\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{b})(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \\ \geq & -|\boldsymbol{A}\boldsymbol{\beta}^* - \boldsymbol{b}|_\infty|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_1 \\ \geq & -\lambda_n|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_1/2.\end{aligned}$$

Combine the two inequalities and we get $|(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)_{S^C}|_1 \leq 3|(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)_S|_1$ and this implies $|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_1 \leq 4|(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)_S|_1 \leq 4\sqrt{s}|(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)_S|_2 \leq 4\sqrt{s}|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2$.

By the definition of $\widehat{\boldsymbol{\beta}}$ and the first order condition, we have $|\boldsymbol{A}\widehat{\boldsymbol{\beta}} - \boldsymbol{b}|_\infty \leq \lambda_n$. Combine this with (11) and we have $|\boldsymbol{A}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)|_\infty \leq 2\lambda_n$. Together with the condition (12) we have

$$|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2 \leq c(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)^{\mathrm{T}}\boldsymbol{A}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \leq 2c\lambda_n|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_1 \leq 8c\lambda_n\sqrt{s}|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2.$$

■

A.2. Proof of Some Technical Lemmas

In this section, we introduce some technical lemmas which will be used in our main proof. Let

$$U_n = \sup_{\left|\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right|_2 \le a_n} \Big| \frac{1}{n} \sum_{i=1}^{n} \left[\boldsymbol{X}_i \mathbb{I}\left[e_i \le \boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)\right] - \boldsymbol{X}_i F\left(\boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)\right)\right] \\ - \frac{1}{n} \sum_{i=1}^{n} \left[\boldsymbol{X}_i \mathbb{I}\left[e_i \le 0\right] - \boldsymbol{X}_i F\left(0\right)\right] \Big|_{\infty}. \qquad (30)$$

Lemma 8 *For any $\gamma > 0$, there exists a constant $c > 0$ such that*

$$\mathbb{P}\left(U_n \ge c\sqrt{\frac{s a_n \log n}{n}} \right) = O\left(n^{-\gamma}\right).$$

Proof [Proof of Lemma 8] Let

$$\begin{aligned} \boldsymbol{C}_{nj}(\boldsymbol{\beta}) &= \frac{1}{n} \sum_{k=1}^{n} \left[X_{kj} \mathbb{I}\left[e_k \le \boldsymbol{X}_{k,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)\right] - X_{kj} F\left(\boldsymbol{X}_{k,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)\right)\right] \\ &\quad - \frac{1}{n} \sum_{k=1}^{n} \left[X_{kj} \mathbb{I}\left[e_k \le 0\right] - X_{kj} F\left(0\right)\right]. \end{aligned}$$

For notation briefness, we denote $\boldsymbol{\beta}_S^* = (\beta_1^*, \ldots, \beta_s^*)^{\mathrm{T}}$. For every i, we divide the interval $[\beta_i^* - a_n, \beta_i^* + a_n]$ into n^M small subintervals and each has length $2a_n/n^M$, where M is a large positive constant. Therefore, there exists a set of points in $\mathbb{R}^{p+1}$, $\{\boldsymbol{\beta}_k, 1 \le k \le q_n\}$ with $q_n \le n^{Ms}$, such that for any $\boldsymbol{\beta}$ in the ball $|\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*|_2 \le a_n$, we have $|\boldsymbol{\beta}_S - \boldsymbol{\beta}_{k,S}|_2 \le 2\sqrt{s} a_n/n^M$ for some $1 \le k \le q_n$ and $|\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*|_2 \le a_n$. We can see that

$$\left| F\left(\boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right)\right) - F\left(\boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)\right)\right| \le C\sqrt{s} a_n n^{-M} \left|\boldsymbol{X}_{i,S}\right|_2,$$

and

$$\begin{aligned} &\left| \mathbb{I}\left[e_i \le \boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right)\right] - \mathbb{I}\left[e_i \le \boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)\right] \right| \\ \le& \mathbb{I}\left[\boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right) - 2\left|\boldsymbol{X}_{i,S}\right|_2 \sqrt{s} a_n n^{-M} \le e_i \le \boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right) + 2\left|\boldsymbol{X}_{i,S}\right|_2 \sqrt{s} a_n n^{-M} \right] \\ =:& \boldsymbol{G}_{i,k}. \end{aligned}$$

Denote the right hand of the above equation by $\boldsymbol{G}_{i,k}$ and let $\mathbb{E}_*(\cdot)$ be the conditional expectation given $\{\boldsymbol{X}_i, 1 \le i \le n\}$. Then we have

$$\begin{aligned} \mathbb{E}_*\left(\boldsymbol{G}_{i,k}\right) =& F\left(\boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right) + 2\left|\boldsymbol{X}_{i,S}\right|_2 \sqrt{s} a_n n^{-M}\right) \\ &- F\left(\boldsymbol{X}_{i,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right) - 2\left|\boldsymbol{X}_{i,S}\right|_2 \sqrt{s} a_n n^{-M}\right). \end{aligned}$$

It is straightforward to conclude that $|\mathbb{E}(|X_{ij}|\boldsymbol{G}_{i,k})| \le C\sqrt{s} a_n n^{-M} \mathbb{E}|X_{ij}||\boldsymbol{X}_{i,S}|_2 \le C s a_n n^{-M}$ and $\mathbb{E}(X_{ij}^2 \boldsymbol{G}_{i,k}^2) \le C s a_n n^{-M}$. By the exponential inequality, we can obtain that for any large γ, there exists a constant c such that

$$\sup_k \mathbb{P}\Big(\frac{1}{n} \Big| \sum_{i=1}^{n} (|X_{ij}|\boldsymbol{G}_{ik} - \mathbb{E}|X_{ij}|\boldsymbol{G}_{ik}) \Big| \ge c\sqrt{\frac{s a_n \log n}{n}} \Big) \le C n^{-\gamma s}.$$

Note that

$$\begin{aligned}\sup_{\left|\boldsymbol{\beta}_S-\boldsymbol{\beta}_S^*\right|_2\le a_n}\left|C_{n,j}\left(\boldsymbol{\beta}\right)\right|-\sup_k\left|C_{n,j}\left(\boldsymbol{\beta}_k\right)\right| \quad&\le\quad C\sqrt{s}a_nn^{-M}\frac{1}{n}\sum_{i=1}^n|\boldsymbol{X}_{i,S}|_2\\&\quad+\frac{1}{n}\Big|\sum_{i=1}^n(|X_{ij}|\boldsymbol{G}_{ik}-\mathbb{E}|X_{ij}|\boldsymbol{G}_{ik})\Big|\\&\quad+\frac{1}{n}\Big|\sum_{i=1}^n\mathbb{E}(|X_{ij}|\boldsymbol{G}_{ik})\Big|.\end{aligned}$$

Therefore

$$\sup_j\Big[\sup_{\left|\boldsymbol{\beta}_S-\boldsymbol{\beta}_S^*\right|_2\le a_n}\left|C_{n,j}\left(\boldsymbol{\beta}\right)\right|-\sup_k\left|C_{n,j}\left(\boldsymbol{\beta}_k\right)\right|\Big]=O_{\mathbb{P}}\left(\sqrt{\frac{sa_n\log n}{n}}\right). \tag{31}$$

It is enough to show that $\sup_j\sup_k|C_{n,j}(\boldsymbol{\beta}_k)|$ satisfies the bound in the lemma. Since the density function of e_k is bounded, we have

$$\mathbb{E}(C_{n,j}(\boldsymbol{\beta}_k))^2\le Cn^{-1}\left|\boldsymbol{\beta}_{k,S}-\boldsymbol{\beta}_S^*\right|_2\le Cn^{-1}a_n.$$

By the exponential inequality (Lemma 1 in Cai and Liu (2011)) and the fact that $\sqrt{s\log n}=o(\sqrt{na_n})$, we have

$$\sup_j\sup_k\mathbb{P}\left(\left|\boldsymbol{C}_{n,j}\left(\boldsymbol{\beta}_k\right)\right|\ge C\sqrt{\frac{sa_n\log n}{n}}\right)=O\left(n^{-\gamma s}\right).$$

We complete the proof of the lemma. ■

Lemma 9 *Assume that (C1)-(C6) hold. Let* $|\widehat{\boldsymbol{\beta}}_0-\boldsymbol{\beta}^*|_2=O_{\mathbb{P}}(a_n)$ *and* $\mathbb{P}(supp(\widehat{\boldsymbol{\beta}}_0)\subseteq S)\to 1$*. Let* $h\ge cs(\log n)/n$ *for some* $c>0$ *and* $h=O(a_n)$*. We have*

$$\left|\widehat{f}\left(0\right)-f\left(0\right)\right|=O_{\mathbb{P}}\left(\sqrt{\frac{s\log n}{nh}}+a_n\right).$$

Proof [Proof of Lemma 9] Denote $\widehat{S}$=supp($\widehat{\boldsymbol{\beta}}_0$) and let

$$D_{n,h}\left(\boldsymbol{\beta}\right)=\frac{1}{nh}\sum_{i=1}^nK\left(\frac{Y_i-\boldsymbol{X}_{i,S}^{\mathrm{T}}\boldsymbol{\beta}_S}{h}\right).$$

We have $|\boldsymbol{\beta}_S^*-\widehat{\boldsymbol{\beta}}_{0,S}|_2=O_{\mathbb{P}}(a_n)$. To prove the proposition, without loss of generality, we can assume that $|\boldsymbol{\beta}_S^*-\widehat{\boldsymbol{\beta}}_{0,S}|_2\le a_n$ and $\widehat{S}\subseteq S$. It follows that $\widehat{f}(0)=D_{n,h}(\widehat{\boldsymbol{\beta}}_0)$ and

$$\left|\widehat{f}\left(0\right)-f\left(0\right)\right|\le\sup_{|\boldsymbol{\beta}_S-\boldsymbol{\beta}_S^*|_2\le a_n}\left|D_{n,h}\left(\boldsymbol{\beta}\right)-f\left(0\right)\right|.$$

Recall the definition of $\{\boldsymbol{\beta}_k, 1 \leq k \leq q_n\}$ in the proof of Lemma 8. We have

$$\left|\frac{1}{h}K\left(\frac{Y_i - \boldsymbol{X}_{i,S}^{\mathrm{T}}\boldsymbol{\beta}_S}{h}\right) - \frac{1}{h}K\left(\frac{Y_i - \boldsymbol{X}_{i,S}^{\mathrm{T}}\boldsymbol{\beta}_{k,S}}{h}\right)\right| \leq Ch^{-2}\left|\boldsymbol{X}_{i,S}^{\mathrm{T}}\left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_{k,S}\right)\right|.$$

This yields that

$$\sup_{|\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*|_2 \leq a_n} \left|D_{n,h}\left(\boldsymbol{\beta}\right) - f\left(0\right)\right| - \sup_{1 \leq k \leq q_n} \left|D_{n,h}\left(\boldsymbol{\beta}_k\right) - f\left(0\right)\right| \leq \frac{C\sqrt{s}a_n}{n^{M+1}h^2}\sum_{i=1}^{n}\left|\boldsymbol{X}_{i,S}\right|_2.$$

Since $\max_{i,j}\mathbb{E}|X_{i,j}|^2 < \infty$ (due to the sub-Gaussian condition (C4)), for any $\gamma > 0$, by letting M large enough, we have

$$\sup_{|\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*|_2 \leq a_n} \left|D_{n,h}\left(\boldsymbol{\beta}\right) - f\left(0\right)\right| - \sup_{1 \leq k \leq n^{Ms}} \left|D_{n,h}\left(\boldsymbol{\beta}_k\right) - f\left(0\right)\right| = O_{\mathbb{P}}\left(n^{-\gamma}\right). \tag{32}$$

It is enough to show that $\sup_k |D_{n,h}(\boldsymbol{\beta}_k) - \mathbb{E}D_{n,h}(\boldsymbol{\beta}_k)|$ and $\sup_k |\mathbb{E}D_{n,h}(\boldsymbol{\beta}_k) - f(0)|$ satisfy the bound in the proposition. Let $\mathbb{E}_*(\cdot)$ denote the conditional expectation given $\{\boldsymbol{X}_k\}$. We have

$$\begin{aligned}\mathbb{E}_*\left\{\frac{1}{h}K\left(\frac{e_i - \boldsymbol{X}_{i,S}^{\mathrm{T}}\left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)}{h}\right)\right\} =& \int_{-\infty}^{\infty} K\left(x\right) f\left\{hx + \boldsymbol{X}_{i,S}^{\mathrm{T}}\left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)\right\} dx \\ =& f\left(0\right) + O\left(h + \left|\boldsymbol{X}_{i,S}^{\mathrm{T}}\left(\boldsymbol{\beta}_S - \boldsymbol{\beta}_S^*\right)\right|\right).\end{aligned}$$

Since $\sup_{|\boldsymbol{\alpha}|_2=1}\mathbb{E}|\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{X}| \leq C$, we have

$$\left|\mathbb{E}D_{n,h}\left(\boldsymbol{\beta}_k\right) - f\left(0\right)\right| \leq C\left(h + \left|\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right|_2\right) = O(h + a_n).$$

It remains to bound $\sup_k |D_{n,h}(\boldsymbol{\beta}_k) - \mathbb{E}D_{n,h}(\boldsymbol{\beta}_k)|$. Put

$$\xi_{i,k} = K\left(\frac{e_i - \boldsymbol{X}_{i,S}^{\mathrm{T}}\left(\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right)}{h}\right).$$

We have

$$\mathbb{E}_*\xi_{i,k}^2 = h\int_{-\infty}^{\infty}\left\{K\left(x\right)\right\}^2 f\left\{hx + \boldsymbol{X}_{i,S}^{\mathrm{T}}\left(\boldsymbol{\beta}_{k,S} - \boldsymbol{\beta}_S^*\right)\right\} dx \leq Ch.$$

Since $K(x)$ is bounded, we have, by the exponential inequality (Lemma 1 in Cai and Liu (2011)) and the fact that $s\log n = O(nh)$, for any $\gamma > 0$, there exists a constant $C > 0$ such that

$$\sup_k \mathbb{P}\left(\left|\sum_{i=1}^{n}(\xi_{i,k} - \mathbb{E}\xi_{i,k})\right| \geq C\sqrt{nhs\log n}\right) = O\left(n^{-\gamma s}\right).$$

By letting $\gamma > M$, we can obtain that

$$\left|\sup_k |D_{n,h}(\boldsymbol{\beta}_k) - \mathbb{E}D_{n,h}(\boldsymbol{\beta}_k)|\right| = O_{\mathbb{P}}\left(\sqrt{\frac{s\log n}{nh}}\right).$$

This completes the proof. ■

Lemma 10 *We have*

$$\max_{1\le j\le p}\left\| n^{-1}\sum_{k=1}^{n}|X_{kj}|\boldsymbol{X}_{k,S}\boldsymbol{X}_{k,S}^{\mathrm{T}}\right\|_{\mathrm{op}} = O_{\mathbb{P}}(1).$$

Proof [Proof of Lemma 10]

For a unit ball B in $\mathbb{R}^s$, we have the fact that there exist q_s balls with centers $\boldsymbol{x}_1,\dots,\boldsymbol{x}_{q_s}$ and radius z (i.e., $B_i=\{\boldsymbol{x}\in\mathbb{R}^s : |\boldsymbol{x}-\boldsymbol{x}_i|\le z\}$, $1\le i\le q_s$) such that $B\subseteq\cup_{i=1}^{q_s}B_i$ and q_s satisfies $q_s\le(1+2/z)^s$. So for any $|\boldsymbol{x}|_2=1$ in the unit sphere, there exists some $\boldsymbol{x}_i$ such that $|\boldsymbol{x}-\boldsymbol{x}_i|_2\le z$ and so this $\boldsymbol{x}_i$ satisfies $1-z\le|\boldsymbol{x}_i|_2\le 1+z$. Therefore, there exists a subset $K\subset\{1,2,\dots,q_s\}$ such that $\{\boldsymbol{x}:|\boldsymbol{x}|_2=1\}\subseteq\cup_{i\in K}B_i$ and $1-z\le|\boldsymbol{x}_i|_2\le 1+z$ for $i\in K$. We have $d_s:=|K|\le q_s\le(1+2/z)^s$.

For any $s\times s$ symmetric matrix $\boldsymbol{A}$, we have

$$|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}|-|\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y}|\le|(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{x}+\boldsymbol{y})|.$$

So $\|\boldsymbol{A}\|_{\mathrm{op}}=\sup_{|\boldsymbol{x}|_2=1}|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}|\le\max_{i\in K}|\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}_i|+z(2+z)\|\boldsymbol{A}\|_{\mathrm{op}}$. Now take $z=1/4$, we have $\|\boldsymbol{A}\|_{\mathrm{op}}\le 3\max_{i\in K}|\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}_i|$ and $d_s\le 9^s$. It is enough to prove that

$$\max_{1\le j\le p}\max_{i\in K}\frac{1}{n}\sum_{k=1}^{n}|X_{kj}|(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_{k,S})^2=O_{\mathbb{P}}(1).$$

Define $\widehat{X}_{kj}=X_{kj}\mathbb{I}[|X_{kj}|\le\log n]$. By the sub-Gaussian condition on $\boldsymbol{X}$, it is enough to show that

$$\max_{1\le j\le p}\max_{i\in K}\frac{1}{n}\sum_{k=1}^{n}|\widehat{X}_{kj}|(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_{k,S})^2=O_{\mathbb{P}}(1).$$

Set

$$Y_{kij}=|\widehat{X}_{kj}|(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_{k,S})^2\mathbb{I}[|\widehat{X}_{kj}|(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_{k,S})^2\le(s+1)(\log n)^3].$$

Note that

$$np9^s\max_{k,j}\max_{i\in K}\mathbb{P}\left(|\widehat{X}_{kj}|(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_{k,S})^2\ge(s+1)(\log n)^3\right)=o(1).$$

It suffices to prove that $\max_{1\le j\le p}\max_{i\in K}\frac{1}{n}\sum_{k=1}^{n}Y_{kij}=O_p(1)$. It is easy to see that $\mathbb{E}Y_{kij}\le\mathbb{E}|X_{kj}|(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_{k,S})^2\le C(\mathbb{E}X_{kj}^2)^{1/2}\sup_{|\boldsymbol{x}|_2=1}(\mathbb{E}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{X}_{k,S})^4)^{1/2}=O(1)$ and similarly, $\mathbb{E}Y_{kij}^2=O(1)$, uniformly in k,i,j. By Bernstein's inequality,

$$\mathbb{P}\left(\Big|\frac{1}{n}\sum_{k=1}^{n}(Y_{kij}-\mathbb{E}Y_{kij})\Big|\ge 1\right)\le e^{-c_1 n}+e^{-c_2\frac{n}{(s+1)(\log n)^3}},$$

for some positive constants c_1 and c_2 uniformly in i,j. Since $s=O(m^r)$ for some $0<r<1/3$, we have

$$np9^s\left(e^{-c_1 n}+e^{-c_2\frac{n}{(s+1)(\log n)^3}}\right)=o(1).$$

This proves $\max_{1\le j\le p}\max_{i\in K}\frac{1}{n}\sum_{k=1}^{n}Y_{kij}=O_p(1)$. ■

A.3. Proof of Theorem 2 and Theorem 3

We first state a proposition for the proof of our main theorems.

Proposition 11 *Assume that (C1)-(C6) hold. Let $|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*|_2 = O_{\mathbb{P}}(a_n)$ and $h \asymp a_n$. We have*

$$\left|\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right|_\infty = O_{\mathbb{P}}\left(\sqrt{\frac{\log n}{n}} + a_n^2\right).$$

Proof [Proof of Proposition 11] Recall the definition of U_n in (30). For the initial estimator, we have $\widehat{\boldsymbol{\beta}}_{0,S^c} = 0$ with high probability. Due to the fact that $\boldsymbol{\beta}^*_{S^c} = \mathbf{0}$ and $\boldsymbol{\beta}_{0,S^c} = \mathbf{0}$, by $|\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}}_0|_2 = O_{\mathbb{P}}(a_n)$, we have

$$\begin{aligned}
&\left|\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right|_\infty \\
= &\left|-\frac{\widehat{f}^{-1}(0)}{n}\sum_{k=1}^n \boldsymbol{X}_k\left(\mathbb{I}\left[Y_k \le \boldsymbol{X}_k^{\mathrm{T}}\widehat{\boldsymbol{\beta}}_0\right] - \tau\right) + \widehat{\boldsymbol{\Sigma}}\left(\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*\right)\right|_\infty \\
\le &\left|\frac{\widehat{f}^{-1}(0)}{n}\sum_{k=1}^n \boldsymbol{X}_k\left\{F\left(\boldsymbol{X}_{k,S}^{\mathrm{T}}\left(\boldsymbol{\beta}_S - \widehat{\boldsymbol{\beta}}_{0,S}\right)\right) - F(0)\right\} + \frac{1}{n}\sum_{k=1}^n \boldsymbol{X}_k\boldsymbol{X}_{k,S}^{\mathrm{T}}\left(\widehat{\boldsymbol{\beta}}_{0,S} - \boldsymbol{\beta}^*_S\right)\right|_\infty \\
&+\left|\widehat{f}^{-1}(0)\right|\left|\frac{1}{n}\sum_{k=1}^n\left[\boldsymbol{X}_k\mathbb{I}\left[e_k \le 0\right] - \boldsymbol{X}_kF(0)\right]\right|_\infty + \left|\widehat{f}^{-1}(0)\right|U_n.
\end{aligned}$$

For the last term, by Lemma 8, we have $|\widehat{f}^{-1}(0)|U_n = O_{\mathbb{P}}(\sqrt{sa_n(\log n)/n})$. For the second term of the right hand side, we have

$$\left|\widehat{f}^{-1}(0)\right|\left|\frac{1}{n}\sum_{k=1}^n\left[\boldsymbol{X}_k\mathbb{I}\left[e_k \le 0\right] - \boldsymbol{X}_kF(0)\right]\right|_\infty = O_{\mathbb{P}}\Big(\sqrt{\frac{\log p}{n}}\Big).$$

Denote the first term of the right hand side to be $\boldsymbol{H}$. For the first component of $\boldsymbol{H}$, by second order Taylor expansion, under (C1) we have

$$\begin{aligned}
&\frac{\widehat{f}^{-1}(0)}{n}\sum_{k=1}^n X_{kj}\left\{F\left(\boldsymbol{X}_{k,S}^{\mathrm{T}}\left(\boldsymbol{\beta}^*_S - \widehat{\boldsymbol{\beta}}_{0,S}\right)\right) - F(0)\right\} \\
=&\frac{\widehat{f}^{-1}(0)f(0)}{n}\sum_{k=1}^n X_{kj}\boldsymbol{X}_{k,S}^{\mathrm{T}}\left(\boldsymbol{\beta}^*_S - \widehat{\boldsymbol{\beta}}_{0,S}\right) + \frac{C\widehat{f}^{-1}(0)}{n}\sum_{k=1}^n |X_{kj}|\left\{\boldsymbol{X}_{k,S}^{\mathrm{T}}\left(\boldsymbol{\beta}^*_S - \widehat{\boldsymbol{\beta}}_{0,S}\right)\right\}^2.
\end{aligned}$$

It is standard to show that

$$\mathbb{P}\left(|\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}|_\infty \le C\sqrt{\frac{\log n}{n}}\right) \to 1.$$

Since $\Lambda_{\max}(\boldsymbol{\Sigma}) \le c_0$, we have

$$\left|\frac{1}{n}\sum_{k=1}^n \boldsymbol{X}_k\boldsymbol{X}_{k,S}^{\mathrm{T}}\left(\boldsymbol{\beta}^*_S - \widehat{\boldsymbol{\beta}}_{0,S}\right)\right|_\infty \le O_{\mathbb{P}}\left(\sqrt{\frac{s\log n}{n}}a_n\right) + \left|\boldsymbol{\Sigma}\left(\boldsymbol{\beta}^*_S - \widehat{\boldsymbol{\beta}}_{0,S}\right)\right|_\infty$$

$$= \quad O_{\mathbb{P}}(a_n).$$

Denote $(1, |X_{k1}|, \ldots, |X_{kp}|)^{\mathrm{T}}$ by $|\boldsymbol{X}_k|$. Then by Lemma 9 and 10, we have

$$\begin{aligned}|\boldsymbol{H}|_\infty \leq & \left|\widehat{f}^{-1}(0) f(0) - 1\right| \left|\frac{1}{n}\sum_{k=1}^{n} \boldsymbol{X}_k \boldsymbol{X}_{k,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_S^* - \widehat{\boldsymbol{\beta}}_{0,S}\right)\right|_\infty \\ & + C\widehat{f}^{-1}(0) \left|\frac{1}{n}\sum_{k=1}^{n} |\boldsymbol{X}_k| \left\{\boldsymbol{X}_{k,S}^{\mathrm{T}} \left(\boldsymbol{\beta}_S^* - \widehat{\boldsymbol{\beta}}_{0,S}\right)\right\}^2\right|_\infty \\ = & O_{\mathbb{P}}\left(\left(\sqrt{\frac{s\log n}{nh}} + a_n\right) a_n\right) + O_{\mathbb{P}}(a_n^2).\end{aligned}$$

So we can easily have

$$\left|\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right|_\infty = O_{\mathbb{P}}\left(\sqrt{\frac{\log p}{n}} + \sqrt{\frac{s a_n \log n}{n}} + a_n\sqrt{\frac{s\log n}{nh}} + a_n^2\right).$$

Since $h \asymp a_n$ and $s a_n = o(1)$, we prove the proposition. ■

Proof [Proof of Theorem 2 and Theorem 3]

First, we show the results for Theorem 2. Define $\widetilde{\boldsymbol{\beta}}$ to be the solution of the following optimization problem:

$$\widetilde{\boldsymbol{\beta}} = \underset{\boldsymbol{\theta}\in\mathbb{R}^{p+1}, \boldsymbol{\theta}_{S^c}=0}{\arg\min} \frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\widehat{\boldsymbol{\Sigma}}_1\boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{T}}\left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\} + \lambda_n |\boldsymbol{\theta}|_1,$$

where $\boldsymbol{\theta}_{S^c}$ denotes the subset vector with the coordinates of $\boldsymbol{\theta}$ in S^c. Then there exist sub-gradients $\widetilde{\boldsymbol{Z}}$ with $|\widetilde{\boldsymbol{Z}}|_\infty \leq 1$ such that

$$\widehat{\boldsymbol{\Sigma}}_{1,S\times S}\widetilde{\boldsymbol{\beta}}_S - \left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\}_S + \lambda_n \widetilde{\boldsymbol{Z}}_S = 0. \tag{33}$$

It is enough to show that there exist sub-gradients $\boldsymbol{Z}$ that satisfy

$$\widehat{\boldsymbol{\Sigma}}_1\widehat{\boldsymbol{\beta}} - \left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\} + \lambda_n \boldsymbol{Z} = 0, \tag{34}$$

$|\boldsymbol{Z}_S|_\infty \leq 1$ and $|\boldsymbol{Z}_{S^c}|_\infty < 1$, i.e., $|Z_i|$ are strictly less than one for $i \in S^c$. To construct such $\boldsymbol{Z}$, we let $\boldsymbol{Z}_S = \widetilde{\boldsymbol{Z}}_S$ and

$$\boldsymbol{Z}_{S^c} = \quad -\lambda_n^{-1}\left\{\left(\widehat{\boldsymbol{\Sigma}}_1\widetilde{\boldsymbol{\beta}}\right)_{S^c} - \left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\}_{S^c}\right\}.$$

Lemma 12 *Under the conditions in Theorem 2, we have, with probability tending to one,*

$$|Z_i| \leq v$$

uniformly for $i \in S^c$, for some $0 < v < 1$.

Proof [Proof of Lemma 12] Recall that

$$\widehat{\boldsymbol{\Sigma}}_{1,S\times S}\widetilde{\boldsymbol{\beta}}_S - \left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\}_S = -\lambda_n \widetilde{\boldsymbol{Z}}_S. \tag{35}$$

Write (35) as

$$\begin{aligned}-\lambda_n\widetilde{\boldsymbol{Z}}_S =& \boldsymbol{\Sigma}_{S\times S}\left(\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S\right) + \left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \boldsymbol{\Sigma}_{S\times S}\right)\left(\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S\right) + \widehat{\boldsymbol{\Sigma}}_{1,S\times S}\boldsymbol{\beta}^*_S \\ &- \left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\}_S.\end{aligned}$$

This implies that

$$\begin{aligned}\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S &= \boldsymbol{\Sigma}^{-1}_{S\times S}\Big\{ - \lambda_n\widetilde{\boldsymbol{Z}}_S - \left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \boldsymbol{\Sigma}_{S\times S}\right)\left(\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S\right) \\ &\qquad - \widehat{\boldsymbol{\Sigma}}_{1,S\times S}\boldsymbol{\beta}_S + \left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\}_S\Big\} \\ &= \boldsymbol{\Sigma}^{-1}_{S\times S}\Big\{ - \lambda_n\widetilde{\boldsymbol{Z}}_S - \left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \boldsymbol{\Sigma}_{S\times S}\right)\left(\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S\right) \\ &\qquad - \left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \widehat{\boldsymbol{\Sigma}}_{S\times S}\right)\left(\boldsymbol{\beta}^*_S - \widehat{\boldsymbol{\beta}}_{0,S}\right) + \left(\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right)_S\Big\}.\end{aligned}$$

By (38), we have with probability tending to one,

$$\begin{aligned}\left|\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S\right|_2 &\le C\sqrt{s}\lambda_n + C\sqrt{\frac{s\log n}{m}}\left|\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S\right|_2 \\ &\quad + C\sqrt{s}\left(\sqrt{\frac{\log n}{m}} + \sqrt{\frac{\log n}{n}}\right)\left|\boldsymbol{\beta}^*_S - \widehat{\boldsymbol{\beta}}_{0,S}\right|_2 + C\sqrt{s}\left|\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right|_\infty.\end{aligned}$$

By the choice of λ_n, Proposition 11 and $a_n = O(\sqrt{s(\log n)/m})$,

$$\left|\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S\right|_2 \le C\sqrt{s}\lambda_n, \tag{36}$$

with probability tending to one.

Due to the definition of $\boldsymbol{Z}_{S^c}$, we have that

$$\begin{aligned}&\boldsymbol{Z}_{S^c} \\ =& -\lambda_n^{-1}\left\{\left(\widehat{\boldsymbol{\Sigma}}_1\widetilde{\boldsymbol{\beta}}\right)_{S^c} - \left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\}_{S^c}\right\} \\ =& -\lambda_n^{-1}\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}\widehat{\boldsymbol{\Sigma}}^{-1}_{1,S\times S}\left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\}_S + \widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}\widehat{\boldsymbol{\Sigma}}^{-1}_{1,S\times S}\widetilde{\boldsymbol{Z}}_S \\ &+ \lambda_n^{-1}\left\{\boldsymbol{z}_n + \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right\}_{S^c} \\ =& -\lambda_n^{-1}\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}\widehat{\boldsymbol{\Sigma}}^{-1}_{1,S\times S}\left[\left\{\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right\}_S + \left(\widehat{\boldsymbol{\Sigma}}_{S\times\{1,\ldots,p+1\}} - \widehat{\boldsymbol{\Sigma}}_{1,S\times\{1,\ldots,p+1\}}\right)\left(\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}}_0\right)\right] \\ &+ \widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}\widehat{\boldsymbol{\Sigma}}^{-1}_{1,S\times S}\widetilde{\boldsymbol{Z}}_S + \lambda_n^{-1}\left\{\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right\}_{S^c} \\ &+ \lambda_n^{-1}\left(\widehat{\boldsymbol{\Sigma}}_{S^c\times\{1,\ldots,p+1\}} - \widehat{\boldsymbol{\Sigma}}_{1,S^c\times\{1,\ldots,p+1\}}\right)\left(\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}}_0\right).\end{aligned} \tag{37}$$

Note that

$$\begin{aligned}
&\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}\\
=&\left(\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}-\boldsymbol{\Sigma}_{S^c\times S}\right)\left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}-\boldsymbol{\Sigma}_{S\times S}^{-1}\right)+\boldsymbol{\Sigma}_{S^c\times S}\left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}-\boldsymbol{\Sigma}_{S\times S}^{-1}\right)\\
&+\left(\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}-\boldsymbol{\Sigma}_{S^c\times S}\right)\boldsymbol{\Sigma}_{S\times S}^{-1}+\boldsymbol{\Sigma}_{S^c\times S}\boldsymbol{\Sigma}_{S\times S}^{-1}.
\end{aligned}$$

By the proof of Lemma 10, we can easily get

$$\left\|\widehat{\boldsymbol{\Sigma}}_{1,S\times S}-\boldsymbol{\Sigma}_{S\times S}\right\|_{\text{op}}=O_{\mathbb{P}}\Big(\sqrt{\frac{s+\log n}{m}}\Big).$$

This yields that

$$\left\|\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}-\boldsymbol{\Sigma}_{S\times S}^{-1}\right\|_{\text{op}}=O_{\mathbb{P}}\Big(\sqrt{\frac{s+\log n}{m}}\Big).$$

Then

$$\begin{aligned}
&\left\|\left(\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}-\boldsymbol{\Sigma}_{S^c\times S}\right)\left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}-\boldsymbol{\Sigma}_{S\times S}^{-1}\right)\right\|_{\infty}\\
\le&s^{3/2}\left|\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}-\boldsymbol{\Sigma}_{S^c\times S}\right|_{\infty}\left\|\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}-\boldsymbol{\Sigma}_{S\times S}^{-1}\right\|_{\text{op}}\\
=&O_{\mathbb{P}}\left(s^2(\log n)/m\right).
\end{aligned}$$

Similarly,

$$\begin{aligned}
\left\|\boldsymbol{\Sigma}_{S^c\times S}\left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}-\boldsymbol{\Sigma}_{S\times S}^{-1}\right)\right\|_{\infty}\le&s\|\boldsymbol{\Sigma}\|_{\text{op}}\left\|\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}-\boldsymbol{\Sigma}_{S\times S}^{-1}\right\|_{\text{op}}\\
=&O_{\mathbb{P}}\left(s\sqrt{\frac{s+\log n}{m}}\right),
\end{aligned}$$

and

$$\begin{aligned}
\left\|\left(\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}-\boldsymbol{\Sigma}_{S^c\times S}\right)\boldsymbol{\Sigma}_{S\times S}^{-1}\right\|_{\infty}\le&s^{3/2}\left|\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}-\boldsymbol{\Sigma}_{S^c\times S}\right|_{\infty}\left\|\boldsymbol{\Sigma}_{S\times S}^{-1}\right\|_{\text{op}}\\
=&O_{\mathbb{P}}\left(\sqrt{\frac{s^3\log n}{m}}\right).
\end{aligned}$$

So we have $\|\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}\|_\infty\le o_{\mathbb{P}}(1)+\|\boldsymbol{\Sigma}_{S^c\times S}\boldsymbol{\Sigma}_{S\times S}^{-1}\|_\infty$. Since C_0 in λ_n is sufficiently large, we can see that $\lambda_n^{-1}|\boldsymbol{z}_n-\widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*|_\infty$ is small enough.

Since $\|\boldsymbol{\Sigma}_{S^c\times S}\boldsymbol{\Sigma}_{S\times S}^{-1}\|_\infty\le 1-\alpha$ and $|\widetilde{\boldsymbol{Z}}_S|_\infty\le 1$, we have $|\widehat{\boldsymbol{\Sigma}}_{1,S^c\times S}\widehat{\boldsymbol{\Sigma}}_{1,S\times S}^{-1}\widetilde{\boldsymbol{Z}}_S|_\infty\le 1-\alpha/2$ with probability tending to one. Note that $\mathbb{P}(\operatorname{supp}(\widehat{\boldsymbol{\beta}}_0)\subseteq S)\to 1$, we have

$$\begin{aligned}
&\lambda_n^{-1}\left|\left(\widehat{\boldsymbol{\Sigma}}_{1,S\times\{1,\ldots,p+1\}}-\widehat{\boldsymbol{\Sigma}}_{S\times\{1,\ldots,p+1\}}\right)\left(\boldsymbol{\beta}^*-\widehat{\boldsymbol{\beta}}_0\right)\right|_\infty\\
=&O_{\mathbb{P}}(1)\lambda_n^{-1}\sqrt{s(\log n)/m}\left|\boldsymbol{\beta}^*-\widehat{\boldsymbol{\beta}}_0\right|_2\\
=&O_{\mathbb{P}}\Big(\lambda_n^{-1}a_n\sqrt{s(\log n)/m}\Big)\\
=&O_{\mathbb{P}}(1/C_0),
\end{aligned}$$

and

$$
\begin{aligned}
&\left|\left(\widehat{\boldsymbol{\Sigma}}_{1,S^c\times\{1,\ldots,p+1\}} - \widehat{\boldsymbol{\Sigma}}_{S^c\times\{1,\ldots,p+1\}}\right)\left(\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}}_0\right)\right|_\infty \\
=&O_{\mathbb{P}}(1)\lambda_n^{-1}\sqrt{s(\log n)/m}\left|\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}}_0\right|_2 \\
=&O_{\mathbb{P}}(1/C_0).
\end{aligned}
$$

The above arguments, together with (37), imply uniformly for $j \in S^c$ and some $v < 1$,

$$
|Z_j| \le v < 1.
$$

■

By Lemma 12, uniformly for $i \in S^c$ and some $v < 1$,

$$
|Z_i| \le v < 1
$$

with probability tending to one. By this primal-dual witness construction, we have $\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}$ with probability tending to one. Thus

$$
\mathbb{P}\left(|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_1 \le \sqrt{s}|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2\right) \to 1.
$$

It is easy to see that

$$
\left|\widehat{\boldsymbol{\Sigma}}_1\widehat{\boldsymbol{\beta}} - \boldsymbol{z}_n - \left(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}}\right)\widehat{\boldsymbol{\beta}}_0\right|_\infty \le \lambda_n,
$$

due to equation (33) and $|\boldsymbol{Z}|_\infty \le 1$. It is standard to show that for some $C > 0$,

$$
\mathbb{P}\left(|\widehat{\boldsymbol{\Sigma}}_1 - \boldsymbol{\Sigma}|_\infty \le C\sqrt{\frac{\log n}{m}}\right) \to 1. \tag{38}
$$

Note that $\mathbb{P}(\mathrm{supp}(\widehat{\boldsymbol{\beta}}_0) \subseteq S) \to 1$. By Proposition 11,

$$
\begin{aligned}
|\widehat{\boldsymbol{\Sigma}}_1\boldsymbol{\beta}^* - \boldsymbol{z}_n - (\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}})\widehat{\boldsymbol{\beta}}_0|_\infty &\le |\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*|_\infty + |(\widehat{\boldsymbol{\Sigma}}_1 - \widehat{\boldsymbol{\Sigma}})(\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*)|_\infty \\
&= O_{\mathbb{P}}\left(\sqrt{\frac{\log n}{n}} + a_n^2 + \sqrt{\frac{s\log n}{m}}a_n\right) \\
&= \frac{O_{\mathbb{P}}(1)}{C_0}\lambda_n.
\end{aligned} \tag{39}
$$

Therefore, by letting C_0 in λ_n being sufficiently large, we have $|\widehat{\boldsymbol{\Sigma}}_1(\boldsymbol{\beta}^* - \widehat{\boldsymbol{\beta}})|_\infty \le 2\lambda_n$ with probability tending to one. By the following condition

$$
\min_{\delta:|\delta|_1\le c_1\sqrt{s}|\delta|_2} \frac{\delta^{\mathrm{T}}\widehat{\boldsymbol{\Sigma}}_1\delta}{|\delta|_2^2} \ge c_2, \qquad c_1, c_2 > 0, \tag{40}
$$

we can further have

$$
\left|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right|_2^2 \le C\left(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right)^{\mathrm{T}}\widehat{\boldsymbol{\Sigma}}_1\left(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right) \le C\lambda_n\left|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right|_1 \le C\lambda_n\sqrt{s}\left|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right|_2.
$$

This proves that $|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2 \leq C\lambda_n\sqrt{s}$.

To prove Theorem 2, it is enough to show that $\widehat{\boldsymbol{\Sigma}}_1$ satisfies condition (40). We have, with probability tending to one,

$$\begin{aligned}\delta^{\mathrm{T}}\widehat{\boldsymbol{\Sigma}}_1\delta \geq& |\delta|_2^2\,\lambda_{\min}(\boldsymbol{\Sigma}) - |\delta|_1^2\left|\widehat{\boldsymbol{\Sigma}}_1 - \boldsymbol{\Sigma}\right|_\infty\\ \geq& |\delta|_2^2\,\lambda_{\min}(\boldsymbol{\Sigma}) - |\delta|_2^2\,s\left|\widehat{\boldsymbol{\Sigma}}_1 - \boldsymbol{\Sigma}\right|_\infty\\ \geq& c\,|\delta|_2^2\,,\end{aligned}$$

for some $c > 0$ as $s = o((m/\log n)^{1/2})$. This completes the proof of Theorem 2.

For $t = 1$, note that we assume $|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}^*|_2 = O_{\mathbb{P}}(\sqrt{s(\log n)/m})$. Then let $a_n = \sqrt{s(\log n)/m}$ in Theorem 2 and it is easy to see Theorem 3 holds for $t = 1$. Now suppose Theorem 3 holds for $t = k-1$ with some $k \geq 2$. Then for $t = k$ with initial estimator being $\widehat{\boldsymbol{\beta}}^{(k-1)}$, we have $a_{n,k-1} = \sqrt{\frac{s\log n}{n}} + s^{(2k-1)/2}\left(\frac{\log n}{m}\right)^{k/2}$. Hence by Theorem 2 again and the condition on s,

$$\begin{aligned}\left|\widehat{\boldsymbol{\beta}}^{(k)} - \boldsymbol{\beta}^*\right|_2 &= O_{\mathbb{P}}\left(\sqrt{\frac{s\log n}{n}} + a_{n,k-1}\sqrt{\frac{s^2\log n}{m}}\right)\\ &= O_{\mathbb{P}}\left(\sqrt{\frac{s\log n}{n}} + s^{(2k+1)/2}\left(\frac{\log n}{m}\right)^{(k+1)/2}\right).\end{aligned}$$

This implies that Theorem 3 holds for $t = k$. Then it completes the proof of Theorem 3. ■

A.4. Proof of Theorem 4 and Theorem 5

Proof [Proof of Theorem 4 and Theorem 5] Theorem 4 (i) and 5 (i) follow directly from the proof of Theorem 2. As for Theorem 4 (ii), note that $\mathbb{P}(\widehat{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}) \to 1$. Recall

$$\begin{aligned}\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^* \;=\; \boldsymbol{\Sigma}_{S\times S}^{-1}\Big\{ &-\lambda_n\widetilde{\boldsymbol{Z}}_S - \left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \boldsymbol{\Sigma}_{S\times S}\right)\left(\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^*\right)\\ &- \left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \widehat{\boldsymbol{\Sigma}}_{S\times S}\right)\left(\boldsymbol{\beta}_S^* - \widehat{\boldsymbol{\beta}}_{0,S}\right) + \left(\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right)_S\Big\}.\end{aligned}$$

By Equation (36), we obtain that, with probability tending to one,

$$\begin{aligned}\left|\boldsymbol{\Sigma}_{S\times S}^{-1}\left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \boldsymbol{\Sigma}_{S\times S}\right)\left(\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^*\right)\right|_\infty &\leq C\|\boldsymbol{\Sigma}_{S\times S}^{-1}\|_\infty\|\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \boldsymbol{\Sigma}_{S\times S}\|_{\mathrm{op}}\left|\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^*\right|_2\\ &\leq C\|\boldsymbol{\Sigma}_{S\times S}^{-1}\|_\infty\sqrt{s(s+\log n)/m}\left|\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^*\right|_\infty,\end{aligned}$$

$$\left|\boldsymbol{\Sigma}_{S\times S}^{-1}\left(\widehat{\boldsymbol{\Sigma}}_{1,S\times S} - \widehat{\boldsymbol{\Sigma}}_{S\times S}\right)\left(\boldsymbol{\beta}_S^* - \widehat{\boldsymbol{\beta}}_{0,S}\right)\right|_\infty \leq C\|\boldsymbol{\Sigma}_{S\times S}^{-1}\|_\infty a_n\sqrt{(s+\log n)/m},$$

and

$$\left|\boldsymbol{\Sigma}_{S\times S}^{-1}\left(\boldsymbol{z}_n - \widehat{\boldsymbol{\Sigma}}\boldsymbol{\beta}^*\right)_S\right|_\infty = O_{\mathbb{P}}\left(\|\boldsymbol{\Sigma}_{S\times S}^{-1}\|_\infty\sqrt{\frac{\log n}{n}} + \|\boldsymbol{\Sigma}_{S\times S}^{-1}\|_\infty a_n^2\right).$$

Table 12: The ℓ_2-error, precision, and recall of the three estimators under different combinations of the sample size n and local sample size m. Noises are generated from normal distribution and quantile level $\tau = 0.5$.

m		200			500			1000		
n		5000	10000	20000	5000	10000	20000	5000	10000	20000
Pooled REL	Precision	0.83	0.91	0.94	0.81	0.87	0.94	0.82	0.86	0.93
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.133	0.097	0.071	0.131	0.094	0.069	0.130	0.098	0.070
Dist REL	Precision	0.98	0.99	1.00	0.91	0.95	0.98	0.86	0.90	0.96
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.149	0.112	0.088	0.137	0.098	0.073	0.132	0.100	0.072
Avg DC	Precision	0.05	0.04	0.04	0.07	0.05	0.04	0.14	0.08	0.05
	Recall	0.99	1.00	1.00	0.98	0.99	1.00	0.97	0.99	0.99
	ℓ_2-error	0.341	0.324	0.313	0.219	0.202	0.192	0.174	0.156	0.139

With Lemma 11 and the choice of λ_n, we obtain that

$$\left|\widetilde{\boldsymbol{\beta}}_S - \boldsymbol{\beta}^*_S\right|_\infty \le C\|\boldsymbol{\Sigma}^{-1}_{S\times S}\|_\infty \left(\sqrt{\frac{\log n}{n}} + a_n\sqrt{\frac{s\log n}{m}}\right).$$

Then Theorem 4 (ii) follows from the above and together with the lower bound condition on $\min_{j\in S}|\beta^*_j|$.

Theorem 5 (ii) follows from the similar proof of Theorem 4 (ii) by replacing the initial estimator as $\widehat{\boldsymbol{\beta}}^{(t-1)}$ and the lower bound condition on $\min_{j\in S}|\beta^*_j|$. ■

Appendix B. Additional Experiments

In this section we provide some additional experiment results using quantile level $\tau = 0.5$. The results are reported in Tables 12, 13 and 14. The observations are similar to the case of $\tau = 0.3$ in Section 4.4.

References

Moulinath Banerjee, Cecile Durot, Bodhisattva Sen, et al. Divide and conquer in non-standard problems and the super-efficiency phenomenon. *Ann. Statist.*, 47(2):720–757, 2019.

Heather Battey, Jianqing Fan, Han Liu, Junwei Lu, and Ziwei Zhu. Distributed testing and estimation under sparse high dimensional models. *Ann. Statist.*, 46(3):1352, 2018.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.*, 2(1):183–202, 2009.

Alexandre Belloni and Victor Chernozhukov. ℓ1-penalized quantile regression in high-dimensional sparse models. *Ann. Statist.*, 39(1):82–130, 2011.

Table 13: The ℓ_2-error, precision, and recall of the three estimators under different combinations of the sample size n and local sample size m. Noises are generated from Cauchy distribution and quantile level $\tau = 0.5$.

m		200			500			1000		
n		5000	10000	20000	5000	10000	20000	5000	10000	20000
Pooled REL	Precision	0.82	0.88	0.94	0.85	0.91	0.95	0.83	0.89	0.93
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.118	0.083	0.063	0.120	0.087	0.063	0.119	0.085	0.062
Dist REL	Precision	0.99	0.99	1.00	0.93	0.96	0.99	0.85	0.92	0.96
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.135	0.098	0.072	0.124	0.091	0.066	0.121	0.087	0.064
Avg DC	Precision	0.05	0.04	0.04	0.09	0.06	0.04	0.14	0.08	0.06
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.303	0.302	0.280	0.196	0.176	0.166	0.159	0.133	0.122

Table 14: The ℓ_2-error, precision, and recall of the three estimators under different combinations of the sample size n and local sample size m. Noises are generated from exponential distribution and quantile level $\tau = 0.5$.

m		200			500			1000		
n		5000	10000	20000	5000	10000	20000	5000	10000	20000
Pooled REL	Precision	0.83	0.87	0.98	0.82	0.93	0.95	0.83	0.92	0.95
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.071	0.053	0.035	0.068	0.050	0.038	0.065	0.047	0.034
Dist REL	Precision	0.89	0.97	0.99	0.86	0.95	0.98	0.85	0.94	0.99
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.086	0.063	0.040	0.073	0.051	0.039	0.072	0.049	0.035
Avg DC	Precision	0.08	0.07	0.06	0.07	0.08	0.05	0.09	0.06	0.07
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	ℓ_2-error	0.188	0.188	0.179	0.100	0.098	0.095	0.085	0.073	0.063

Alexandre Belloni, Victor Chernozhukov, et al. Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):521–547, 2013.

Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

Tony Cai and Weidong Liu. Adaptive thresholding for sparse covariance matrix estimation. *J. Am. Statist. Ass.*, 106(494):672–684, 2011.

Tony Cai, Weidong Liu, and Xi Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *J. Am. Statist. Ass.*, 106(494):594–607, 2011.

Mengjie Chen, Chao Gao, Zhao Ren, et al. Robust covariance and scatter matrix estimation under Huber's contamination model. *Ann. Statist.*, 46(5):1932–1960, 2018.

Xi Chen, Weidong Liu, and Yichen Zhang. Quantile regression under memory constraint. *Ann. Statist.*, 47(6):3244–3273, 12 2019.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Statist. Ass.*, 96(456):1348–1360, 2001.

Jianqing Fan, Yingying Fan, and Emre Barut. Adaptive robust variable selection. *Ann. Statist.*, 42(1):324–351, 2014.

Jianqing Fan, Quefeng Li, and Yuyan Wang. Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions. *J. R. Statist. Soc. B*, 79(1):247–265, 2017.

Jianqing Fan, Yongyi Guo, and Kaizheng Wang. Communication-efficient accurate statistical estimation. *arXiv preprint arXiv:1906.04870*, 2019a.

Jianqing Fan, Dong Wang, Kaizheng Wang, and Ziwei Zhu. Distributed estimation of principal eigenspaces. *Ann. Statist.*, 47(6):3009, 2019b.

LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020. URL `http://www.gurobi.com`.

Joel L Horowitz. Bootstrap methods for median regression models. *Econometrica*, 66(6):1327–1351, 1998.

Daniel Hsu and Sivan Sabato. Heavy-tailed regression with a generalized median-of-means. In *Proceedings of the International Conference on Machine Learning*, 2014.

Daniel Hsu and Sivan Sabato. Loss minimization and parameter estimation with heavy tails. *The Journal of Machine Learning Research*, 17(1):543–582, 2016.

Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

Michael I Jordan, Jason D Lee, and Yun Yang. Communication-efficient distributed statistical inference. *J. Am. Statist. Ass.*, 114(526):668–681, 2019.

Roger Koenker. *Quantile regression.* Cambridge university press, 2005.

Guillaume Lecué and Matthieu Lerasle. Learning from MOMs principles: Le cams approach. *Stochastic Processes and Their Applications*, 129(11):4385–4410, 2019.

Guillaume Lecué, Matthieu Lerasle, et al. Robust machine learning by median-of-means: theory and practice. *Ann. Statist.*, 48(2):906–931, 2020.

Jason D Lee, Qiang Liu, Yuekai Sun, and Jonathan E Taylor. Communication-efficient sparse regression. *The Journal of Machine Learning Research*, 18(5):1–30, 2017.

Runze Li, Dennis KJ Lin, and Bing Li. Statistical inference in massive data sets. *Appl. Stoch. Model Bus.*, 29(5):399–409, 2013.

Gabor Lugosi and Shahar Mendelson. Risk minimization by median-of-means tournaments. *arXiv preprint arXiv:1608.00757*, 2016.

Gabor Lugosi, Shahar Mendelson, et al. Regularization, sparse recovery, and median-of-means tournaments. *Bernoulli*, 25(3):2075–2106, 2019.

Mark Schmidt. Graphical model structure learning with ℓ_1-regularization. *PhD thesis, University of British Columbia*, 2010.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the International Conference on Machine Learning*, 2014.

Chengchun Shi, Wenbin Lu, and Rui Song. A massive data framework for m-estimators with cubic-rate. *J. Am. Statist. Ass.*, 113(524):1698–1709, 2018.

Stefan Solntsev, Jorge Nocedal, and Richard H Byrd. An algorithm for quadratic ℓ_1-regularized optimization with a flexible active-set strategy. *Optim. Method. Softw.*, 30 (6):1213–1237, 2015.

Qiang Sun, Wen-Xin Zhou, and Jianqing Fan. Adaptive Huber regression. *J. Am. Statist. Ass.*, 115(529):254–265, 2020.

Robert Tibshirani, Martin Wainwright, and Trevor Hastie. *Statistical learning with sparsity: the Lasso and generalizations.* Chapman and Hall/CRC, 2015.

Sara Van de Geer, Peter Bühlmann, Ya'acov Ritov, Ruben Dezeure, et al. On asymptotically optimal confidence regions and tests for high-dimensional models. *Ann. Statist.*, 42(3): 1166–1202, 2014.

Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Trans. Inf. Theory*, 55(5): 2183–2202, 2009.

Jialei Wang, Mladen Kolar, Nathan Srebro, and Tong Zhang. Efficient distributed learning with sparsity. In *Proceedings of the International Conference on Machine Learning*, 2017.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.*, 38(2):894–942, 2010.

Cun-Hui Zhang and Stephanie S Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *J. R. Statist. Soc. B*, 76(1):217–242, 2014.

Peng Zhao and Bin Yu. On model selection consistency of Lasso. *The Journal of Machine Learning Research*, 7(Nov):2541–2563, 2006.

Tianqi Zhao, Mladen Kolar, and Han Liu. A general framework for robust testing and confidence regions in high-dimensional quantile regression. *arXiv preprint arXiv:1412.8724*, 2014.

Tianqi Zhao, Guang Cheng, Han Liu, et al. A partially linear framework for massive heterogeneous data. *Ann. Statist.*, 44(4):1400–1437, 2016.

Wen-Xin Zhou, Koushiki Bose, Jianqing Fan, and Han Liu. A new perspective on robust m-estimation: Finite sample theory and applications to dependence-adjusted multiple testing. *Ann. Statist.*, 46(5):1904, 2018.

Hui Zou and Ming Yuan. Composite quantile regression and the oracle model selection theory. *Ann. Statist.*, 32(3):1108–1126, 2008.