View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by University of Minnesota Digital Conservancy

Unconventional Regression for High-Dimensional Data Analysis

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Yuwen Gu

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Hui Zou, Adviser

June 2017

© Yuwen Gu 2017
ALL RIGHTS RESERVED

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Hui Zou, for his gracious support and encouragement over the past few years. His insights and vision have enlightened me in all aspects of statistical research and applications. I benefit so much from his sound knowledge in statistics, his wisdom and his professionalism. I feel extremely lucky to have Hui as my advisor.

My heartfelt gratitude also goes to the rest of my dissertation committee: Prof. Xiaoou Li, Prof. Wei Pan and Prof. Yuhong Yang. I thank them for their precious advice and insightful comments on my research. Meanwhile, I would like to give my sincere thanks to Prof. Galin Jones and Prof. Lan Wang for their support in my job search, to Yi Yang for all the insightful discussions, and to Boxiang Wang, Lan Liu, Bo Peng, Qi Yan, Gang Cheng, Qing Mai, Xin Zhang, Xiaoyi Zhu, Gongjun Xu, Zhihua Su and Jun Fan for their tremendous help in both research and life. I also thank all the other professors and graduate students in the School of Statistics for making my PhD life so enjoyable.

My special thanks are due to my parents, Jikang Gu and Fenlan Xu, for bringing me to the world in the first place and for supporting me mentally through every obstacle in my life.

Last but not least, this dissertation would not have been completed without my loving and supportive wife, Yanjia Yu.

DEDICATION

To my parents and wife.

ABSTRACT

Massive and complex data present new challenges that conventional sparse penalized mean regressions, such as the penalized least squares, cannot fully solve. For example, in high-dimensional data, non-constant variance, or heteroscedasticity, is commonly present but often receives little attention in penalized mean regressions. Heavy-tailedness is also frequently encountered in many high-dimensional scientific data. To resolve these issues, unconventional sparse regressions such as penalized quantile regression and penalized asymmetric least squares are the appropriate tools because they can infer the complete picture of the entire probability distribution.

Asymmetric least squares regression has wide applications in statistics, econometrics and finance. It is also an important tool in analyzing heteroscedasticity and is computationally friendlier than quantile regression. The existing work on asymmetric least squares only considers the traditional low dimension and large sample setting. We systematically study the Sparse Asymmetric LEast Squares (SALES) under high dimensionality and fully explore its theoretical and numerical properties. SALES may fail to tell which variables are important for the mean function and which variables are important for the scale/variance function, especially when there are variables that are important for both mean and scale. To that end, we further propose a COupled Sparse Asymmetric LEast Squares (COSALES) regression for calibrated heteroscedasticity analysis.

Penalized quantile regression has been shown to enjoy very good theoretical properties in the literature. However, the computational issue of penalized quantile regression has not yet been fully resolved in the literature. We introduce fast alternating direction method of multipliers (ADMM) algorithms for computing penalized quantile regression with the lasso, adaptive lasso, and folded concave penalties. The convergence properties of the proposed

algorithms are established and numerical experiments demonstrate their computational efficiency and accuracy.

To efficiently estimate coefficients in high-dimensional linear models without prior knowledge of the error distributions, sparse penalized composite quantile regression (CQR) provides protection against significant efficiency decay regardless of the error distribution. We consider both lasso and folded concave penalized CQR and establish their theoretical properties under ultrahigh dimensionality. A unified efficient numerical algorithm based on ADMM is also proposed to solve the penalized CQR. Numerical studies demonstrate the superior performance of penalized CQR over penalized least squares under many error distributions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Our decade has seen a surge of massive and complex data due to the advance of data acquisition technologies. As an integral part of the "big data" revolution, high-dimensional data are more and more frequently collected in a wide variety of fields, such as biomedical sciences, finance, climate studies, astronomy and neuroscience. These high-dimensional data pose many challenges both theoretically and numerically. This urges the development of new methodologies and tools to analyze high-dimensional data.

It is well known that in the high-dimensional regime, traditional statistical analyses break down when the number of unknown parameters exceeds the number of observations ("large p small n") due to the "curse of dimensionality" (Donoho et al., 2000). Therefore, to solve high-dimensional problems, many methods have been proposed to reduce the dimensionality, usually by imposing some type of low-dimensional constraints on the model space. Exemplary approaches include the penalized regression with various sparsity constraints on the coefficients, matrix estimation with low-rank assumptions, covariance or precision matrix estimation with structure sparsity patterns, and so on. In all these examples, the regularization idea proves to be very successful. To that end, for the sparse penalized regression alone, various regularization techniques have been proposed to control the model

complexity and to achieve intended sparsity structures. Popular regularization methods include the lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), elastic net (Zou and Hastie, 2005), adaptive lasso (Zou, 2006), Dantzig selector (Candes and Tao, 2007), MCP (Zhang, 2010), and so on. Under such regularization, the penalized least squares regression has received tremendous attention in applications and has been widely adopted in practice to analyze high-dimensional data with a huge number of variables due to its nice theoretical guarantees and computational efficiency.

In the penalized least squares regression, the variance function is often assumed constant for theoretical convenience. However, in high-dimensional data, non-constant variance, or heteroscedasticity, is commonly present. Studies on expression quantitative trait locus (eQTL) confirmed the existence of heteroscedasticity in lots of high-dimensional data (Wang et al., 2012; Daye et al., 2012). Moreover, high-dimensional data subject to heavy-tailed errors are also commonly encountered in various scientific fields. For such data, conventional sparse penalized mean regressions, such as the penalized least squares, will encounter problems. To resolve these issues, unconventional sparse penalized regressions, such as the penalized quantile regression and penalized Huber regression, are the appropriate tools. In this dissertation, we systematically study a new type of unconventional sparse penalized regression, namely, the penalized asymmetric least squares regression, and its variant to deal with heteroscedasticity in high-dimensional data. We also discuss the computational issues of the penalized quantile regression. In terms of coefficient estimation in a linear model, we study the penalized composite quantile regression as a safe procedure for efficient coefficient estimation.

1.2 Dissertation Outline

The dissertation is mainly composed of three chapters (Chapters 2 – 4), which discuss three different types of unconventional regressions in the high-dimensional setting. The main

goal of this dissertation is to provide good methodologies and practical tools to deal with high-dimensional data that exhibit heteroscedasticity and heavy-tailedness.

In Chapter 2, we introduce the sparse asymmetric least squares and demonstrate its potential in dealing with heteroscedasticity in high-dimensional data. We establish the theoretical properties of both lasso and folded concave penalized asymmetric least squares. A unified numerical algorithm is proposed to solve penalized asymmetric least squares with various penalties. For more calibrated analysis of heteroscedasticity, we propose a coupled sparse asymmetric least squares regression and study both its theoretical and numerical properties.

In Chapter 3, we propose efficient alternating direction method of multipliers (ADMM) algorithms for numerically solving the penalized quantile regression with various penalties. We show the convergence properties of our proposed algorithms. Computational efficiency and accuracy of the proposed algorithms are demonstrated through extensive numerical studies.

In the context of estimating coefficients in a linear model, we study the penalized composite quantile regression in Chapter 4 under ultrahigh dimensionality. Composite quantile regression is known to be a safe alternative to quantile regression in providing high efficiency in coefficient estimation. We consider both lasso and folded concave penalized composite quantile regression and study their theoretical properties. Since the objective function is highly non-smooth, the penalized composite quantile regression poses great challenges in optimization. We propose a sparse coordinate descent ADMM algorithm for efficiently solving the penalized composite quantile regression. We demonstrate the superior finite sample performance of the penalized composite quantile regression over the penalized least squares regression under many error distributions in numerical studies.

Finally, we conclude the dissertation in Chapter 5, with a brief discussion of some potential future work.

Chapter 2

High-Dimensional Generalizations of Asymmetric Least Squares Regression and Their Applications

Asymmetric least squares regression is an important method that has wide applications in statistics, econometrics and finance. The existing work on asymmetric least squares only considers the traditional low dimension and large sample setting. In this chapter, we systematically study the Sparse Asymmetric LEast Squares (SALES) regression under high dimensions where the penalty functions include the lasso and nonconvex penalties. We develop a unified efficient algorithm for fitting SALES and establish its theoretical properties. As an important application, SALES is used to detect heteroscedasticity in high-dimensional data. Another method for detecting heteroscedasticity is the sparse quantile regression. However, both SALES and the sparse quantile regression may fail to tell which variables are important for the conditional mean and which variables are important for the conditional scale/variance, especially when there are variables that are important for both the mean and the scale. To that end, we further propose a COupled Sparse Asymmetric LEast Squares (COSALES) regression which can be efficiently solved by an algorithm similar to that for solving SALES. We establish theoretical properties of COSALES. In particular, COSALES using the SCAD penalty or MCP is shown to consistently identify the two important subsets for the mean and scale simultaneously, even when the two subsets overlap. We demonstrate

the empirical performance of SALES and COSALES by simulated and real data.

2.1 Introduction

High-dimensional data have received tremendous attention in the last decade due to the advance of data collection technology. Sparse estimation, which uses penalization or regularization techniques to perform variable selection and estimation simultaneously, has become a mainstream approach for analyzing high-dimensional data. Popular penalized estimators include the L_1-type selectors such as the lasso (Tibshirani, 1996) and Dantzig (Candes and Tao, 2007) selectors and the nonconvex penalized estimators such as the SCAD (Fan and Li, 2001) and MCP (Zhang, 2010) estimators. Some embrace the L_1-regularization for its computational efficiency, while others prefer to use the nonconvex penalization due to its oracle (Fan and Li, 2001) property.

The current literature on sparse estimation often assumes homoscedasticity. For example, the existing theory for the sparse linear regression model is based on the classical linear model assumption in which the mean function is linear and the errors are i.i.d. with zero mean and constant variance. The heteroscedasticity issue is often overlooked for theoretical convenience. However, heteroscedasticity often exists due to heterogeneity in measurement units or accumulation of outlying observations from numerous sources of inputs. This is particularly relevant with high-dimensional data. For example, in genomics experiments, tens of thousands of genes are often analyzed simultaneously by microarrays and occasional outlying measurements appearing in numerous experimental and data-preprocessing steps can accumulate to form heteroscedasticity in the data obtained therein. These data sets are often of high dimension since only a small number of subjects are available for the study. Several studies on expression quantitative trait loci (eQTLs) (Wang et al., 2012; Daye et al., 2012) confirmed the presence of heteroscedasticity in these high-dimensional data and it was shown that genetic variants have effects on both the mean and the scale (i.e.,

standard deviation) of gene expression levels. In such scenarios, it is important to incorporate heteroscedasticity to make inference from the limited amount of data. To our knowledge, most existing work on high-dimensional data analysis fails to address the heteroscedasticity issue.

The sparse quantile regression was proposed in Wang et al. (2012) to detect heteroscedasticity in high-dimensional data. Quantile regression (Koenker and Bassett, 1978) is appropriate under heteroscedasticity, because it uses an asymmetric absolute value loss. The key word is "asymmetric," not the absolute value loss. The absolute value loss is computationally more challenging than the squared error loss. Computational efficiency is always one of the primary considerations in high-dimensional data analysis. This motivates us to study the asymmetric least squares (ALS) regression under high dimensionality. The ALS regression has been studied in Efron (1991). It is also known as the expectile regression in econometrics and finance. See Newey and Powell (1987); Taylor (2008); Kuan et al. (2009); Xie et al. (2014). The key idea in ALS is to assign different squared error loss to the positive and negative residuals, respectively. By doing so, one can infer a more complete description of the conditional distribution than ordinary least squares (OLS). Thus, ALS and quantile regression share a common virtue although they differ technically. The most notable advantage of ALS over quantile regression is that the former employs a smooth differentiable loss, which considerably alleviates the computational effort involved and also makes the theoretical analysis more amenable. These two are desirable properties under high dimensionality.

In this chapter, we develop the methodology and theory for the Sparse Asymmetric LEast Squares (SALES) regression and show its applications in detecting heteroscedasticity in a general class of sparse models in which the set of relevant covariates may vary from segment to segment on the conditional distribution. For the nonconvex penalized SALES regression, we prove its strong oracle property. We then discuss an important issue overlooked by existing methods dealing with heteroscedasticity in high dimensional data, that is, how to

exactly differentiate the sets of relevant covariates for the mean and scale when they have overlaps. To resolve this issue, we propose a novel COupled Sparse Asymmetric LEast Squares (COSALES) regression method to select important variables for the mean and scale of the conditional distribution simultaneously. The strong oracle property is also shown for the nonconvex penalized COSALES estimator. We develop novel efficient algorithms for computing both SALES and COSALES.

The remainder of the chapter is organized as follows. We study SALES in Section 2.2 and demonstrate its application in detecting heteroscedasticity in Section 2.3. In Section 2.4, we introduce and study COSALES. The performance of COSALES is illustrated by two simulation examples. In Section 2.5, we apply SALES and COSALES to analyze a real microarray dataset. The proofs of all main theoretical results are relegated to Section 2.6.

2.2 High-Dimensional SALES Regression

2.2.1 Background and setup

We start by defining the τ-mean of a random variable $Z \in \mathbb{R}$,

$$\mathscr{E}^{\tau}(Z) \equiv \arg\min_{a\in\mathbb{R}} \mathbb{E}\{\Psi_{\tau}(Z-a)\},\ \tau \in (0,1), \tag{2.1}$$

where $\Psi_{\tau}(u) = |\tau - I(u<0)|u^2$ is the asymmetric squared error loss (see e.g. Newey and Powell, 1987; Efron, 1991) and $I(\cdot)$ represents the indicator function. Similar definition can be found in Efron (1991). As a matter of fact, our τ-mean corresponds to Efron's w-mean, where $w = \tau/(1-\tau)$. Hereafter, we call $\mathscr{E}^{\tau}$ the asymmetric expectation operator (with asymmetry coefficient τ). Note that $\mathscr{E}^{0.5}$ coincides with the usual expectation operator $\mathbb{E}$. The τ-mean is also called the τ-expectile in the econometrics literature (Newey and Powell, 1987). By varying τ, the τ-mean quantifies different "locations" of a distribution, and thus it can be viewed as a generalization of the mean and an alternative measure of "location" of a

distribution.

The asymmetric squared error loss $\Psi_\tau(\cdot)$ gives rise to the ALS regression, in which the squared error loss is given different weights depending on whether the residual is positive or negative. Let $\mathbf{X} = (X_1, \ldots, X_p)$ be the $n \times p$ design matrix with $X_j = (x_{1j}, \ldots, x_{nj})^\top$, $j = 1, \ldots, p$, and $\mathbf{y} = (y_1, \ldots, y_n)^\top$ be the n-dimensional response vector. The design matrix may also be written as $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$, where $\mathbf{x}_i = (x_{i1} \ldots, x_{ip})^\top$, $i = 1, \ldots, n$. The ALS regression is done via

$$\widehat{\boldsymbol{\beta}}_\tau^{\mathrm{ALS}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} \Psi_\tau(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}).$$

When $\tau = 0.5$, the ALS regression reduces to the OLS regression. When $\tau \neq 0.5$, due to the asymmetric nature and relative smoothness of $\Psi_\tau(\cdot)$, the ALS regression provides a convenient and computationally efficient way of summarizing the conditional distribution of a response variable given the covariates (Newey and Powell, 1987; Efron, 1991). Applications of the ALS regression include estimation of the value at risk and expected shortfall (Taylor, 2008; Kuan et al., 2009), medical baseline correction (Eilers and Boelens, 2005), and small area estimation (Chambers and Tzavidis, 2006; Salvati et al., 2012) among others.

In the literature, the underlying model considered for studying the theoretical property of the ALS regression is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^\tau + \boldsymbol{\varepsilon}^\tau, \tag{2.2}$$

where $\boldsymbol{\beta}^\tau$ is a p-dimensional vector of unknown parameters and $\boldsymbol{\varepsilon}^\tau$ is the vector of n independent errors, which satisfy $\mathscr{E}^\tau(\varepsilon_i^\tau | \mathbf{x}_i) = 0$, $i = 1, \ldots, n$ for some $\tau \in (0, 1)$. It follows that $\mathscr{E}^\tau(y_i | \mathbf{x}_i) = \mathbf{x}_i^\top \boldsymbol{\beta}^\tau$, which means that the conditional τ-mean of y_i is a linear combination of $\mathbf{x}_i$, $i = 1, \ldots, n$. A similar model to (2.2) was considered in Wang et al. (2012) for quantile regression, where the conditional quantile of the response variable was

modeled as a linear combination of the covariates. In model (2.2), it is important to realize that the coefficient vector $\boldsymbol{\beta}^\tau$ is allowed to change with τ, which makes modeling for different "locations" of the conditional distribution possible, and as a result heteroscedasticity in the data, when it exists, can be inspected by this model. For convenience, we will drop the superscript for $\boldsymbol{\beta}^\tau$ and $\boldsymbol{\varepsilon}^\tau$ when no confusion arises.

To accommodate high-dimensional data in model (2.2), we allow the number of covariates p to increase with the sample size n, and moreover, we are primarily interested in cases where p exceeds n ($p > n$). We adopt the sparsity assumption that only a small number of covariates contribute to the response. Suppose $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_p^*)^{\mathrm{T}}$ is the parameter vector of the true underlying model that generates the data and assume $\boldsymbol{\beta}^*$ is s-sparse, where $s = |A|$ with $A \equiv \operatorname{supp}(\boldsymbol{\beta}^*) = \{j : \beta_j^* \neq 0\}$.

2.2.2 Methodology

To select important variables and estimate $\boldsymbol{\beta}$ in model (2.2) when the dimension is high, let us consider the following penalized SALES regression:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \; n^{-1} \sum_{i=1}^{n} \Psi_\tau(y_i - \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(\beta_j), \tag{2.3}$$

where $\Psi_\tau(\cdot)$ is the asymmetric squared error loss and $p_\lambda(\cdot)$ is a nonnegative penalty function with regularization parameter $\lambda \in (0, \infty)$. In the remainder of this chapter, we mainly focus on the lasso and nonconvex penalties.

L_1-penalized SALES regression

For ease of notation, let $\mathcal{L}_n(\boldsymbol{\beta}) = n^{-1} \sum_{i=1}^{n} \Psi_\tau(y_i - \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta})$. The L_1-penalized SALES estimator or SALES lasso estimator $\widehat{\boldsymbol{\beta}}^{\text{lasso}}$ is defined as the solution to the minimization

problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \mathcal{L}_n(\boldsymbol{\beta}) + \lambda_{\text{lasso}} \sum_{j=1}^{p} |\beta_j|, \ \lambda_{\text{lasso}} \in (0, \infty). \tag{2.4}$$

This is to take $p_\lambda(u) = \lambda|u|$ in (2.3). The lasso is computationally attractive and can be solved by efficient algorithms such as the LARS (Efron et al., 2004), the coordinate descent method (Friedman et al., 2010) and the generalized coordinate descent algorithm (Yang and Zou, 2013).

For efficient computation of $\widehat{\boldsymbol{\beta}}^{\text{lasso}}$ in (2.4), we propose an algorithm called SALES which combines the cyclic coordinate descent (Tseng, 2001) and proximal gradient algorithms (Parikh and Boyd, 2013). Our algorithm solves the following more general "weighted" L_1-minimization problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \mathcal{L}_n(\boldsymbol{\beta}) + \sum_{j=1}^{p} w_j|\beta_j| \tag{2.5}$$

with constants $w_j \geq 0$ for all j. Our consideration of formulation (2.5) is twofold. First, it not only can be directly applied to the SALES lasso problem (2.4) by setting $w_j = \lambda_{\text{lasso}}$ for all j, but also can be used to solve the convex approximations to the nonconvex penalized SALES estimation (see step (a) of Algorithm 2). Second, leaving some coefficients unpenalized is simply a matter of setting their corresponding weights to zero. Doing so gives us the flexibility to decide which covariates should always be kept in the model. The algorithm is described as follows.

For $\mathbf{v} = (v_1, \ldots, v_d)^{\mathrm{T}} \in \mathbb{R}^d$, denote $\mathbf{v}_{-k} = (v_1, \ldots, v_{k-1}, v_{k+1}, \ldots, v_d)^{\mathrm{T}}$ the subvector of $\mathbf{v}$ with its kth component removed. Recover $\mathbf{v}$ from $\mathbf{v}_{-k}$ by writing $\mathbf{v} = [v_k, \mathbf{v}_{-k}]$. Let $\boldsymbol{\beta}^r = (\beta_1^r, \ldots, \beta_p^r)^{\mathrm{T}}$ be the update of $\boldsymbol{\beta}$ after the rth ($r \geq 0$) cycle of the coordinate descent

algorithm. For ease of notation, denote

$$\mathbf{b}_{-k}^{r+1} = (\beta_1^{r+1}, \ldots, \beta_{k-1}^{r+1}, \beta_{k+1}^{r}, \ldots, \beta_p^{r})^{\mathrm{T}}, 1 \leq k \leq p,\ r \geq 0.$$

Applying the coordinate descent method, to update β_k in the $(r+1)$th cycle, we solve the following minimization problem:

$$\min_{\beta_k \in \mathbb{R}} \ell_n(\beta_k; \mathbf{b}_{-k}^{r+1}) + w_k|\beta_k|, \tag{2.6}$$

where $\ell_n(\beta_k; \mathbf{b}_{-k}^{r+1}) = \mathcal{L}_n([\beta_k, \mathbf{b}_{-k}^{r+1}]) = n^{-1}\sum_{i=1}^n \Psi_\tau(y_i - \mathbf{x}_{i,-k}^{\mathrm{T}}\mathbf{b}_{-k}^{r+1} - x_{ik}\beta_k)$. One can show that $\ell_n'(\beta_k; \mathbf{b}_{-k}^{r+1})$ is Lipschitz continuous with constant $L_k = 2\bar{c}n^{-1}\|X_k\|_2^2$, where $\|\cdot\|_2$ is the Euclidean norm. Thus, the proximal gradient method can be employed to solve problem (2.6)

$$\beta_k^{r,0} := \beta_k^r,\ \beta_k^{r,s+1} := \mathbb{S}_{L_k^{-1}w_k}(\beta_k^{r,s} - L_k^{-1}\ell_n'(\beta_k^{r,s}; \mathbf{b}_{-k}^{r+1})),\ s \geq 0, \tag{2.7}$$

where $\mathbb{S}_v(u) = \mathrm{sgn}(u)(|u| - v)^+$ denotes the soft thresholding operator with $u^+ = uI(u > 0)$. We let (2.7) run for s_k^r iterations and set $\beta_k^{r+1} := \beta_k^{r,s_k^r}$. Our algorithm is summarized in Algorithm 1. We prove in Appendix A that Algorithm 1 converges at least linearly.

Nonconvex penalized SALES regression

Nonconvex penalties have been used in a broad type of sparse regression models (Fan and Lv, 2011; Wang et al., 2013; Fan et al., 2014b). The most popular nonconvex penalties include the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001) and the minimax concave penalty (MCP, Zhang, 2010). For some constant $\gamma > 2$, the SCAD

Algorithm 1: SALES — Cyclic coordinate descent plus proximal gradient algorithm for solving the weighted L_1-minimization problem (2.5)

1. Initialize the algorithm with $\boldsymbol{\beta}^0 = (\beta_1^0, \ldots, \beta_p^0)^{\mathrm{T}}$.
2. For $r = 0, 1, 2, \ldots, m-1$,
   - (2.1) For $k = 1, \ldots, p$,
     - (2.1.1) Initialize $\beta_k^{r,0} := \beta_k^r$.
     - (2.1.2) For $s = 0, 1, 2, \ldots, s_k^r - 1$,
       - (2.1.2.1) Calculate $\beta_k^{r,s+1} := \mathbb{S}_{L_k^{-1} w_k}(\beta_k - L_k^{-1} \ell_n'(\beta_k^{r,s}; \mathbf{b}_{-k}^{r+1}))$.
     - (2.1.3) Set $\beta_k^{r+1} := \beta_k^{r,s_k^r}$.
   - (2.2) Set $\boldsymbol{\beta}^{r+1} := (\beta_1^{r+1}, \ldots, \beta_p^{r+1})^{\mathrm{T}}$.
3. Output $\widehat{\boldsymbol{\beta}} := \boldsymbol{\beta}^m$.

penalty is given by

$$p_\lambda(u) = \lambda|u| I(|u| \le \lambda) + \left\{\lambda|u| - \frac{(\lambda - |u|)^2}{2(\gamma - 1)}\right\} I(\lambda < |u| \le \gamma\lambda) + \frac{(\gamma + 1)\lambda^2}{2} I(|u| > \gamma\lambda). \tag{2.8}$$

The use of $\gamma = 3.7$ for the SCAD penalty is recommended in Fan and Li (2001) from a Bayesian perspective. The MCP is characterized by

$$p_\lambda(u) = \lambda \left(|u| - \frac{u^2}{2\gamma\lambda}\right) I(|u| \le \gamma\lambda) + \frac{\gamma\lambda^2}{2} I(|u| > \gamma\lambda) \tag{2.9}$$

for some $\gamma > 1$. The use of $\gamma = 2$ is suggested in Zhang (2010). In this chapter, we consider both SCAD and MCP penalized SALES regression.

The main motivation for using the nonconvex penalties is to achieve the oracle property. For the SALES regression, the oracle estimator is

$$\widehat{\boldsymbol{\beta}}^{\text{oracle}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p : \boldsymbol{\beta}_{A^c} = \mathbf{0}}{\arg\min} \; \mathcal{L}_n(\boldsymbol{\beta}). \tag{2.10}$$

In practice, the oracle estimator is infeasible, but it sets a benchmark for evaluation of other estimators. Many papers have shown that the nonconvex penalized least squares can find the oracle estimator with high probability (Wang et al., 2013; Fan et al., 2014b). In particular, Fan et al. (2014b) showed that the local linear approximation (LLA) algorithm (Zou and Li, 2008) converges to the oracle estimator under regularity conditions. The LLA algorithm fits a sequence of weighted L_1-regularization problems. Since we already have Algorithm 1 for computing any weighted L_1-penalized SALES regression, we adopt the LLA algorithm for solving the nonconvex penalized SALES estimation problem (2.3). The details of the LLA algorithm are shown in Algorithm 2. Note that step (a) can be readily solved by Algorithm 1.

Algorithm 2: Local linear approximation (LLA) algorithm for solving the nonconvex penalized SALES estimation problem (2.3)

1. Initialize $\widehat{\boldsymbol{\beta}}^0 := \widehat{\boldsymbol{\beta}}^{\text{initial}}$. Compute weights $\hat{w}_j^0 = p'_\lambda(|\hat{\beta}_j^0|),\ j = 1, \ldots, p$.
2. For $m = 1, 2, \ldots,$ repeat the LLA iteration in (a) and (b) until convergence
   (a) Solve the following convex optimization problem for $\widehat{\boldsymbol{\beta}}^m$

$$\widehat{\boldsymbol{\beta}}^m := \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min}\ \mathcal{L}_n(\boldsymbol{\beta}) + \sum_{j=1}^{p} \hat{w}_j^{m-1} |\beta_j|.$$

   (b) Update the weights $\hat{w}_j^m = p'_\lambda(|\hat{\beta}_j^m|),\ j = 1, \ldots, p$.

In our numerical examples, we tried using both the SALES lasso estimator and zero as the initial values of the LLA algorithm for computing the nonconvex penalized SALES estimator. Our practice is based on theoretical results in Section 2.2.3.

2.2.3 Theory

In this section, we theoretically analyze the SALES regression. We consider the case where the covariates are from a fixed design.

The following notation will be used. For any vector $\mathbf{v} = (v_1, \ldots, v_p)^{\mathrm{T}} \in \mathbb{R}^p$ and an

arbitrary index set $I \subset \{1, \ldots, p\}$, we write $\mathbf{v}_I = (v_j, j \in I)^{\mathrm{T}}$ and denote by $\mathbf{X}_I = (\mathbf{x}_j, j \in I)$ the submatrix consisting of the columns of $\mathbf{X}$ with indices in I. The complement of I is denoted by $I^{\mathrm{c}} = \{1, \ldots, p\} \backslash I$. For $q \in [1, \infty]$, the L_q-norm of $\mathbf{v}$ is denoted by $\|\mathbf{v}\|_q$. Sub-Gaussian norm (Rudelson and Vershynin, 2013) of a random variable Z is denoted by $\|Z\|_{\mathrm{SG}} = \sup_{k \geq 1} k^{-1/2} (\mathbb{E}|Z|^k)^{1/k}$. Let $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$ for real numbers a and b. For a differentiable function $f : \mathbb{R}^p \to \mathbb{R}$, we write $\nabla f(\mathbf{v}) = \partial f(\mathbf{v}) / \partial \mathbf{v}$ and $\nabla_I f(\mathbf{v}) = (\partial f(\mathbf{v}) / \partial v_j, \ j \in I)^{\mathrm{T}}$. We use $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ to represent respectively the smallest and largest eigenvalues of a symmetric matrix. We also let $\underline{c} = \tau \wedge (1 - \tau)$ and $\bar{c} = \tau \vee (1 - \tau)$.

L_1-penalized SALES regression

The estimation accuracy of the lasso has been extensively studied in the literature; see, for example, Negahban et al. (2012) and Ye and Zhang (2010). Let $\mathscr{C} = \{\boldsymbol{\delta} \in \mathbb{R}^p : \|\boldsymbol{\delta}_{A^{\mathrm{c}}}\|_1 \leq 3\|\boldsymbol{\delta}_A\|_1 \neq 0\}$ be a cone in $\mathbb{R}^p$. Let $\rho_{\min} = \lambda_{\min}(n^{-1}\mathbf{X}_A^{\mathrm{T}}\mathbf{X}_A)$ and $\rho_{\max} = \lambda_{\max}(n^{-1}\mathbf{X}_A^{\mathrm{T}}\mathbf{X}_A)$. We assume $\rho_{\min} > 0$ so that the important variables are not linearly dependent. To study the estimation accuracy of the SALES lasso, we impose the following conditions on the design matrix $\mathbf{X}$ and the random errors $\boldsymbol{\varepsilon}$.

(C1) The columns of $\mathbf{X}$ are normalizable, that is, $M_0 = \max_{1 \leq j \leq p} \frac{\|X_j\|_2}{\sqrt{n}} \in (0, \infty)$.

(C2) The random errors ε_i are i.i.d. sub-Gaussian random variables satisfying $\mathscr{E}^\tau(\varepsilon_i) = 0$, $i = 1, \ldots, n$.

(C3) $\kappa = \inf_{\boldsymbol{\delta} \in \mathscr{C}} \frac{\|\mathbf{X}\boldsymbol{\delta}\|_2^2}{n\|\boldsymbol{\delta}\|_2^2} \in (0, \infty)$.

(C4) $\varrho = \inf_{\boldsymbol{\delta} \in \mathscr{C}} \frac{\|\mathbf{X}\boldsymbol{\delta}\|_2^2}{n\|\boldsymbol{\delta}_A\|_1 \|\boldsymbol{\delta}\|_\infty} \in (0, \infty)$.

Condition (C3) is called the restricted eigenvalue condition and has been frequently assumed in the literature to study the lasso and Dantzig selectors. See Bickel et al.

(2009), Meier et al. (2009), and Negahban et al. (2012). Condition (C4), the generalized invertability factor (GIF) condition, is closely related to condition (C3) and has also been often adopted to study the lasso and Dantzig selectors. See discussion of these conditions in Ye and Zhang (2010) and Huang and Zhang (2012). Both conditions (C3) and (C4) are crucial assumptions to establish estimation consistency of the lasso for high-dimensional data.

Theorem 2.1

Suppose in model (2.2) the true coefficients $\boldsymbol{\beta}^*$ are s-sparse and assume conditions (C1-C2) hold. Let $\widehat{\boldsymbol{\beta}}^{\text{lasso}}$ be any optimal solution to the SALES lasso problem (2.4). Then with probability at least $1 - p_1^{\text{ALS}}$, $\|\widehat{\boldsymbol{\beta}}^{\text{lasso}} - \boldsymbol{\beta}^*\|_2 \leq 3s^{1/2}\lambda_{\text{lasso}}(4\kappa\underline{c})^{-1}$ if condition (C3) holds, and $\|\widehat{\boldsymbol{\beta}}^{\text{lasso}} - \boldsymbol{\beta}^*\|_\infty \leq 3\lambda_{\text{lasso}}(4\varrho\underline{c})^{-1}$ if condition (C4) holds, where

$$p_1^{\text{ALS}} = 2p\exp\left(-\frac{Cn\lambda_{\text{lasso}}^2}{4K_0^2M_0^2}\right),$$

$K_0 = \|\Psi_\tau'(\varepsilon_i)\|_{\text{SG}}$ with $\Psi_\tau'(\cdot)$ being the derivative of $\Psi_\tau(\cdot)$, and $C > 0$ is an absolute constant. □

Remark 2.1 In some applications, it is natural to leave a given subset of the parameters unpenalized in the penalized framework (2.3). Let $\mathscr{R}$ denote the index set of such parameters. For example, when X_1 is a vector consisting of all ones, $\mathscr{R} = \{1\}$ reflects the common practice of leaving the intercept term not penalized. In this case, it is natural to modify the penalized SALES estimation problem (2.3) to be

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \mathcal{L}_n(\boldsymbol{\beta}) + \sum_{j\in\mathscr{R}^{\text{c}}} p_\lambda(\beta_j).$$

With lasso penalty, the SALES algorithm can be readily used to solve the above case. Moreover, similar theoretical analysis can be carried out with slight modifications. For

instance, in the SALES lasso problem (2.4) we can define $A' \equiv \text{supp}(\boldsymbol{\beta}^*_{\mathscr{R}^c})$ and $\mathscr{C}' = \{\boldsymbol{\delta} \in \mathbb{R}^p : \|\boldsymbol{\delta}_{(A'\cup\mathscr{R})^c}\|_1 \le 3\|\boldsymbol{\delta}_{A'\cup\mathscr{R}}\|_1 \ne 0\}$. Conditions (C3) and (C4) can be then modified respectively as

$$\kappa' = \inf_{\boldsymbol{\delta}\in\mathscr{C}'} \frac{\|\mathbf{X}\boldsymbol{\delta}\|_2^2}{n\|\boldsymbol{\delta}\|_2^2} \in (0,\infty) \quad \text{and} \quad \varrho' = \inf_{\boldsymbol{\delta}\in\mathscr{C}'} \frac{\|\mathbf{X}\boldsymbol{\delta}\|_2^2}{n\|\boldsymbol{\delta}_{A'\cup\mathscr{R}}\|_1\|\boldsymbol{\delta}\|_\infty} \in (0,\infty). \qquad \square$$

To establish the selection consistency of the lasso, it is almost necessary to impose the irrepresentable condition; see Zou (2006) and Zhao and Yu (2006). When the focus is on identifying the underlying sparsity pattern, the nonconvex penalized regression is a competitive alternative as it requires weaker conditions to achieve selection consistency.

Nonconvex penalized SALES regression

To offer a unified treatment of the SCAD and MCP penalized SALES regression, our theoretical analysis handles the following class of nonconvex penalties:

(P1) $p_\lambda(u) = p_\lambda(-u)$;

(P2) $p_\lambda(u)$ is nondecreasing and concave in $u \in [0,\infty)$ and $p_\lambda(0) = 0$;

(P3) $p_\lambda(u)$ is differentiable in $u \in (0,\infty)$;

(P4) $p'_\lambda(u) \ge a_1\lambda$ for $u \in (0, a_2\lambda]$ and $p'_\lambda(0) := p'_\lambda(0+) \ge a_1\lambda$;

(P5) $p'_\lambda(u) = 0$ for $u \in [a\lambda,\infty)$ with some prespecified constant $a > a_2$,

where a_1 and a_2 are fixed constants characteristic of the penalty functions. It is easy to verify that both the SCAD penalty and MCP are in the above class.

We show that the sparse solutions obtained by the LLA algorithm in Section 2.2.2 possess the oracle property. Assume sufficient signal strength in the nonzero components of $\boldsymbol{\beta}^*$

(A1) $\min_{j\in A}|\beta_j^*| > (a+1)\lambda$.

Theorem 2.2

Suppose in model (2.2) the true coefficients $\boldsymbol{\beta}^*$ are s-sparse and satisfy assumption (A1). Assume conditions (C1-C2) hold and take $\widehat{\boldsymbol{\beta}}^{\text{lasso}}$ as the initial value. Let $a_0 = 1 \wedge a_2$. Take $\lambda \geq 3s^{1/2}\lambda_{\text{lasso}}(4a_0\kappa\underline{c})^{-1}$ when (C3) holds, or take $\lambda \geq 3\lambda_{\text{lasso}}(4a_0\varrho\underline{c})^{-1}$ when (C4) holds, or take $\lambda \geq [3s^{1/2}\lambda_{\text{lasso}}(4a_0\kappa\underline{c})^{-1}] \wedge [3\lambda_{\text{lasso}}(4a_0\varrho\underline{c})^{-1}]$ when both (C3) and (C4) hold. The LLA algorithm (Algorithm 2) converges to $\widehat{\boldsymbol{\beta}}^{\text{oracle}}$ after two iterations with probability at least $1 - p_1^{\text{ALS}} - p_2^{\text{ALS}} - p_3^{\text{ALS}}$, where p_1^{ALS} is given in Theorem 2.1,

$$p_2^{\text{ALS}} = 2(p-s)\exp\left(-\frac{Ca_1^2 n\lambda^2}{4K_0^2M_0^2}\right) + \Gamma(Q_1\lambda; n, s, K_0, M_0, \rho_{\max}, \nu_0)$$

and

$$p_3^{\text{ALS}} = \Gamma(2\underline{c}\rho_{\min}R; n, s, K_0, M_0, \rho_{\max}, \nu_0),$$

where $Q_1 = a_1\underline{c}\rho_{\min}\left(2\bar{c}\rho_{\max}^{1/2}M_0\right)^{-1}$, $\nu_0 = \text{var}(\Psi_\tau'(\varepsilon_i))$, $R = \min_{j\in A}|\beta_j^*| - a\lambda$, K_0 is defined in Theorem 2.1 and $\Gamma(\cdot)$ is a function defined by

$$\Gamma(x; n, s, K, M, \rho, \nu) = 2s\exp\left(-\frac{Cnx^2}{K^2M^2s}\right) \wedge 2\exp\left(-\frac{C\nu^2[(n^{1/2}x - \nu\rho^{1/2}s^{1/2})^+]^2}{K^4\rho}\right),$$

and $C > 0$ is an absolute constant. □

It is interesting to note that with the SCAD penalty or MCP, a three-step LLA algorithm starting from the zero vector may also work. Indeed, for these two penalties we have $p_\lambda'(0) = \lambda$, so if we can take $\lambda = \lambda_{\text{lasso}}$, this would give us the SALES lasso estimator in the second step.

Corollary 2.1

Assume the same framework of Theorem 2.2 and suppose the SCAD penalty (2.8) or MCP (2.9) is used. If condition (C3) holds and $4a_0\kappa\underline{c} \geq 3s^{1/2}$, or if condition (C4) holds and $4a_0\varrho\underline{c} \geq 3$, or if both (C3) and (C4) hold and $[3s^{1/2}(\kappa)^{-1}] \wedge [3(\varrho)^{-1}] \leq 4a_0\underline{c}$, the LLA algorithm (Algorithm 2) initialized by zero converges to the oracle estimator after three iterations with probability at least $1 - 2p\exp\{-Cn\lambda^2(4K_0^2M_0^2)^{-1}\} - p_2^{\mathrm{ALS}} - p_3^{\mathrm{ALS}}$, where p_2^{ALS} and p_2^{ALS} are given in Theorem 2.2. □

2.3 Application of SALES: Detecting Heteroscedasticity

Due to asymmetry of the squared error loss, the SALES regression (2.3) can be employed to detect heteroscedasticity in high-dimensional data. In the following, we use a simulation example to illustrate this application. For the nonconvex penalty functions used in the simulation, we fix $\gamma = 3.7$ for the SCAD penalty (2.8) and $\gamma = 2$ for the MCP (2.9).

EXAMPLE 1. We adopt a model from Wang et al. (2012). In the model, the covariates are generated in two steps. First, we generate copies of $(z_1, \ldots, z_p)^{\mathrm{T}}$ from the multivariate normal distribution $N(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} = (0.5^{|i-j|})_{p\times p}$. In the second step, for each copy of $(z_1, \ldots, z_p)^{\mathrm{T}}$, we set $x_1 = \Phi(z_1)$ and $x_j = z_j$ for $j = 2, 3, \ldots, p$, where $\Phi(\cdot)$ is the standard normal CDF. The response is then simulated from the following normal linear heteroscedastic model:

$$y = x_6 + x_{12} + x_{15} + x_{20} + (0.7x_1)\varepsilon, \tag{2.11}$$

where $\varepsilon \sim N(0, 1)$ is independent of the covariates. This model was considered in Wang et al. (2012) for the sparse quantile regression, where a sample size $n = 300$ and covariate dimensions $p = 400$ and 600 were considered. We apply the SALES regression (2.3) instead to select active variables and estimate the coefficients for this model. For the purpose

of demonstration, we choose $n = 300$ and $p = 600$. A validation set of size $n = 300$ is generated independently to tune the regularization parameter by minimizing the validation error $\sum_{i \in \text{validation}} \Psi_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}})$ for the computed estimate $\widehat{\boldsymbol{\beta}}$, where $\tau = 0.5$ and 0.85 are considered.

For comparison purpose, we included in this simulation the SALES lasso (2.4) and two variations of the LLA algorithm for each nonconvex penalized SALES regression: the two-step LLA algorithm initialized by the lasso estimator (SCAD*, MCP*), and the three-step LLA algorithm initialized by zero (SCAD^0, MCP^0).

Let $\widehat{\boldsymbol{\beta}}$ be the coefficient estimates from a given method. Based on 100 replicates, the following measurements are calculated to evaluate the sparsity recovery and estimation performance of that method:

$|\hat{A}|$: the average size of the active set $\hat{A} = \{j : \hat{\beta}_j \neq 0\}$ of $\widehat{\boldsymbol{\beta}}$.

p_a : proportion of the event $A \subset \hat{A}$, where A is the active set of $\boldsymbol{\beta}^*$. When $\tau = 0.5$, $A = \{6, 12, 15, 20\}$ and when $\tau \neq 0.5$, $A = \{1, 6, 12, 15, 20\}$.

p_1 : proportion of the event that $\{1\} \subset \hat{A}$.

R_1 : the average L_1 risk $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1$.

R_2 : the average L_2 risk $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$.

The simulation results are shown in Table 2.1. The following conclusions can be made:

(1) The variable x_1 in the scale function is often not recovered by penalized least-squares ($\tau = 0.5$). However, when several τ-means (e.g., $\tau = 0.85$) are inspected together, it is possible to detect this variable with high probability. This shows that indeed the SALES regression can be used to detect heteroscedasticity.

(2) Compared to the SALES lasso, the nonconvex penalized SALES regression selects much fewer irrelevant covariates and has better estimation accuracy.

(3) The three-step LLA algorithm starting from zero produces similar results to the two-step LLA algorithm starting from the lasso solution.

Table 2.1: Numerical summary of simulation results from the lasso, SCAD and MCP penalized SALES regression for model (2.11): $y = x_6 + x_{12} + x_{15} + x_{20} + (0.7x_1)\varepsilon$. The sparsity recovery performance is measured by the selected active set size $|\hat{A}|$, the proportion p_a of covering the true active set and the proportion p_1 of selecting the signature variable X_1. The estimation accuracy is measured by the L_1 risk R_1 and the L_2 risk R_2. The results are shown as averages over 100 replicates with standard errors listed in the parentheses when available

	Method	$\lvert\hat{A}\rvert$	p_a	p_1	R_1	R_2
$\tau = 0.5$	SALES-lasso	25.82 (1.15)	100%	0%	0.399 (0.015)	0.120 (0.003)
	SALES-SCAD*	7.75 (0.68)	100%	0%	0.103 (0.006)	0.049 (0.002)
	SALES-SCAD^0	6.65 (0.68)	100%	0%	0.100 (0.006)	0.050 (0.002)
	SALES-MCP*	6.39 (0.48)	100%	0%	0.099 (0.005)	0.049 (0.002)
	SALES-MCP^0	5.75 (0.29)	100%	0%	0.093 (0.004)	0.049 (0.002)
$\tau = 0.85$	SALES-lasso	34.17 (1.26)	100%	100%	0.714 (0.016)	0.249 (0.005)
	SALES-SCAD*	7.52 (0.51)	100%	100%	0.160 (0.009)	0.083 (0.005)
	SALES-SCAD^0	8.19 (0.59)	100%	100%	0.166 (0.007)	0.084 (0.003)
	SALES-MCP*	6.30 (0.25)	100%	100%	0.148 (0.005)	0.079 (0.003)
	SALES-MCP^0	6.35 (0.23)	100%	100%	0.147 (0.005)	0.078 (0.003)

2.4 High-Dimensional COSALES Regression

In Section 2.3, we showed that the SALES regression provides a means of detecting heteroscedasticity in high-dimensional data. Indeed, in the linear heteroscedastic model (2.11), the signature variable x_1, which appears in the scale function, was detected through comparison of different τ-means. However, in high-dimensional heteroscedastic models, often of more interest are the sparsity patterns in both the mean and the scale functions of the conditional distribution. The SALES regression and methods proposed by other authors, for example, Wang et al. (2012), are not sufficient to fulfill this task. To see it, consider a linear

heteroscedastic model in which the active set for the mean is $\{1, 2\}$ and the active set for the scale is $\{1, 3\}$. Suppose the SALES regression can exactly recover the active variables. Then the method picks x_1 and x_2 when $\tau = 0.5$ and hopefully x_1, x_2, and x_3 when $\tau \neq 0.5$. A natural question is whether the scale function depends on x_1. With the SALES regression, we cannot answer this question. This motivates us to consider the COSALES regression for a general class of models and gain some insight into analyzing heteroscedasticity in high-dimensional data.

2.4.1 Formulation and computation

Consider the following model of systematic heteroscedasticity,

$$y_i = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\gamma} + (\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\omega})\varepsilon_i,\ i = 1, \ldots, n, \tag{2.12}$$

where ε_i are i.i.d. random errors that are independent of the covariates and that have distribution F_0 with $\mathbb{E}(\varepsilon_i) = \int_{\mathbb{R}} x \mathrm{d}F_0(x) = 0$; $\boldsymbol{\gamma}$ and $\boldsymbol{\omega}$ are unknown p-dimensional parameter vectors controlling the conditional mean and scale; and $\boldsymbol{\omega}$ is assumed to satisfy $\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\omega} > 0$ for all i. The intercept can be included by letting $x_{i1} = 1$. The linear scale model of heteroscedasticity (2.12) is an important model considered by many authors (Koenker and Bassett, 1982; Efron, 1991; Koenker and Zhao, 1994) for analyzing heteroscedasticity.

Let $A_1 \equiv \mathrm{supp}(\boldsymbol{\gamma}^*) = \{j : \gamma_j^* \neq 0\}$ and $A_2 \equiv \mathrm{supp}(\boldsymbol{\omega}^*) = \{j : \omega_j^* \neq 0\}$ be the active sets of $\boldsymbol{\gamma}^*$ and of $\boldsymbol{\omega}^*$, respectively. Suppose $|A_1| = s_1$ and $|A_2| = s_2$. Let $e_\tau = \mathscr{E}^\tau(\varepsilon_1)$ be the τ-mean of the random error for $\tau \in (0, 1)$. It follows that the τ-mean of y_i given $\mathbf{x}_i$ is $\mathscr{E}^\tau(y_i|\mathbf{x}_i) = \mathbf{x}_i^{\mathrm{T}}(\boldsymbol{\gamma} + \boldsymbol{\omega}e_\tau)$. To select significant variables in both the mean and the scale functions, we now propose the COSALES regression. Write $\boldsymbol{\varphi} = \boldsymbol{\omega}e_\tau$. Note that we omit the dependency of $\boldsymbol{\varphi}$ on τ to ease exposition. In the COSALES regression, we will deal with $\boldsymbol{\varphi}$ instead of $\boldsymbol{\omega}$. However, when $e_\tau \neq 0$, it should be noted that since $\mathrm{supp}(\boldsymbol{\varphi}) = \mathrm{supp}(\boldsymbol{\omega})$, the selection result on $\boldsymbol{\varphi}$ applies to $\boldsymbol{\omega}$. Moreover, $\boldsymbol{\omega}$ can be estimated up to a scale from the

estimate of $\boldsymbol{\varphi}$. Ideally, if the distribution F_0 of ε_i is known, exact estimation of $\boldsymbol{\omega}$ is possible.

For some $\tau \in (0, 1)$ and $\tau \neq 0.5$, let

$$S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) = n^{-1} \sum_{i=1}^{n} \{\Psi_{0.5}(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\gamma}) + \Psi_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\gamma} - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\varphi})\}.$$

The COSALES regression tries to minimize

$$Q_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) = S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) + \sum_{j=1}^{p} p_{\lambda_1}(\gamma_j) + \sum_{j=1}^{p} p_{\lambda_2}(\varphi_j), \tag{2.13}$$

over $\boldsymbol{\gamma}, \boldsymbol{\varphi} \in \mathbb{R}^p$, where $p_{\lambda_1}(\cdot)$ and $p_{\lambda_2}(\cdot)$ are penalty functions with regularization parameters $\lambda_1, \lambda_2 \in (0, \infty)$, respectively. Let $\widehat{\boldsymbol{\gamma}}^{\text{oracle}}$ and $\widehat{\boldsymbol{\varphi}}^{\text{oracle}}$ be the oracle estimators of $\boldsymbol{\gamma}$ and $\boldsymbol{\varphi} = \boldsymbol{\omega} e_\tau$, respectively, in model (2.12),

$$(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}}) = \underset{\boldsymbol{\gamma}, \boldsymbol{\varphi} \in \mathbb{R}^p : \boldsymbol{\gamma}_{A_1^{\mathrm{c}}} = \mathbf{0}, \boldsymbol{\varphi}_{A_2^{\mathrm{c}}} = \mathbf{0}}{\arg\min} S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}). \tag{2.14}$$

In what follows, let us focus on the lasso and nonconvex penalties.

L_1-penalized COSALES regression

For $\lambda_1^{\text{lasso}}, \lambda_2^{\text{lasso}} \in (0, \infty)$, the L_1-penalized COSALES estimators or the COSALES lasso estimators of $\boldsymbol{\gamma}$ and $\boldsymbol{\varphi}$ can be achieved simultaneously by

$$(\widehat{\boldsymbol{\gamma}}^{\text{lasso}}, \widehat{\boldsymbol{\varphi}}^{\text{lasso}}) = \underset{\boldsymbol{\gamma}, \boldsymbol{\varphi} \in \mathbb{R}^p}{\arg\min}\, S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) + \lambda_1^{\text{lasso}} \|\boldsymbol{\gamma}\|_1 + \lambda_2^{\text{lasso}} \|\boldsymbol{\varphi}\|_1. \tag{2.15}$$

We note that problem (2.15) is a special case of the minimization problem in step (a) of Algorithm 4 (Section 2.4.1) and efficient computation of the solutions can be carried out by an algorithm similar to Algorithm 1. The algorithm applies the cyclic coordinate descent and proximal gradient descent methods to $\boldsymbol{\gamma}$ and $\boldsymbol{\varphi}$ alternately. We call this algorithm COSALES

and display it in Algorithm 3. Note that COSALES solves the general coupled weighted L_1-minimization problem

$$\min_{\boldsymbol{\gamma},\boldsymbol{\varphi}\in\mathbb{R}^p} S_n(\boldsymbol{\gamma},\boldsymbol{\varphi}) + \sum_{j=1}^{p} w_j|\gamma_j| + \sum_{j=1}^{p} v_j|\varphi_j|. \tag{2.16}$$

To facilitate the presentation, in Algorithm 3, we let $\boldsymbol{\gamma}^r$ and $\boldsymbol{\varphi}^r$ be the updates of $\boldsymbol{\gamma}$ and $\boldsymbol{\varphi}$ respectively after the rth cycle of the coordinate descent algorithm and denote

$$\mathbf{g}_{-k}^{r+1} = (\gamma_1^{r+1},\ldots,\gamma_{k-1}^{r+1},\gamma_{k+1}^{r},\ldots,\gamma_p^{r}),\ 1\le k\le p,\ r\ge 0,$$

and

$$\mathbf{p}_{-k}^{r+1} = (\varphi_1^{r+1},\ldots,\varphi_{k-1}^{r+1},\varphi_{k+1}^{r},\ldots,\varphi_p^{r}),\ 1\le k\le p,\ r\ge 0.$$

Theoretical justification of the estimation accuracy of the COSALES lasso will be deferred to the next section.

Nonconvex penalized COSALES regression

In (2.13), let $p_{\lambda_1}(\cdot)$ and $p_{\lambda_2}(\cdot)$ be nonconvex penalties having properties (P1-P5). This nonconvex penalized COSALES estimation problem can be solved by the LLA algorithm shown in Algorithm 4. Note that the minimization problem in step (a) was solved in Algorithm 3. Oracle properties of the sparse solutions will be established in the following section.

2.4.2 Theory

In this section, we show the selection and estimation accuracy of the COSALES regression for both lasso and nonconvex penalties.

Algorithm 3: COSALES — Coordinate descent plus proximal gradient algorithm for solving the coupled weighted L_1-minimization problem (2.16)

1. Initialize the algorithm with $\boldsymbol{\gamma}^0 = (\gamma_1^0, \ldots, \gamma_p^0)^{\mathrm{T}}$ and $\boldsymbol{\varphi}^0 = (\varphi_1^0, \ldots, \varphi_p^0)^{\mathrm{T}}$.
2. For $r = 1, \ldots, m-1$,
   - (2.1) For $k = 1, \ldots, p$,
     - (2.1.1) Initialize $\gamma_k^{r,0} := \gamma_k^r$.
     - (2.1.2) For $s = 0, 1, \ldots, s_{1k}^r - 1$,
       - (2.1.2.1) Compute $\gamma_k^{r,s+1} := \mathbb{S}_{L_{1k}^{-1} w_k}(\gamma_k^{r,s} - L_{1k}^{-1} h_n'(\gamma_k^{r,s}; \mathbf{g}_{-k}^{r+1}, \boldsymbol{\varphi}^r))$, where $L_{1k} = (2\bar{c}+1)n^{-1}\|X_k\|_2^2$; $h_n(\gamma_k; \mathbf{g}_{-k}^{r+1}, \boldsymbol{\varphi}^r) = S_n([\gamma_k, \mathbf{g}_{-k}^{r+1}], \boldsymbol{\varphi}^r)$.
     - (2.1.3) Set $\gamma_k^{r+1} := \gamma_k^{r, s_{1k}^r}$.
   - (2.2) Set $\boldsymbol{\gamma}^{r+1} := (\gamma_1^{r+1}, \ldots, \gamma_p^{r+1})^{\mathrm{T}}$.
   - (2.3) For $k = 1, \ldots, p$,
     - (2.3.1) Initialize $\varphi_k^{r,0} := \varphi_k^r$.
     - (2.3.2) For $s = 0, 1, \ldots, s_{2k}^r - 1$,
       - (2.3.2.1) Compute $\varphi_k^{r,s+1} := \mathbb{S}_{L_{2k}^{-1} v_k}(\varphi_k^{r,s} - L_{2k}^{-1} \hbar_n'(\varphi_k^{r,s}; \boldsymbol{\gamma}^{r+1}, \mathbf{p}_{-k}^{r+1}))$, where $L_{2k} = 2\bar{c}n^{-1}\|X_k\|_2^2$; $\hbar_n(\varphi_k; \boldsymbol{\gamma}^{r+1}, \mathbf{p}_{-k}^{r+1}) = S_n(\boldsymbol{\gamma}^{r+1}, [\varphi_k, \mathbf{p}_{-k}^{r+1}])$.
     - (2.3.3) Set $\varphi_k^{r+1} := \varphi_k^{r, s_{2k}^r}$.
   - (2.4) Set $\boldsymbol{\varphi}^{r+1} := (\varphi_1^{r+1}, \ldots, \varphi_p^{r+1})^{\mathrm{T}}$.
3. Output $\widehat{\boldsymbol{\gamma}} := \boldsymbol{\gamma}^m$ and $\widehat{\boldsymbol{\varphi}} := \boldsymbol{\varphi}^m$.

Algorithm 4: Local linear approximation (LLA) algorithm for solving the nonconvex penalized COSALES estimation problem (2.13)

1. Initialize $\widehat{\boldsymbol{\gamma}}^0 = \widehat{\boldsymbol{\gamma}}^{\text{initial}}$ and $\widehat{\boldsymbol{\varphi}}^0 = \widehat{\boldsymbol{\varphi}}^{\text{initial}}$. Compute weights
$$\hat{w}_j^0 = p_{\lambda_1}'(|\hat{\gamma}_j^0|),\ \bar{w}_j^0 = p_{\lambda_2}'(|\hat{\varphi}_j^0|),\ j = 1, \ldots, p.$$
2. For $m = 1, 2, \ldots$, repeat the LLA iteration in (a) and (b) until convergence.
   - (a) Solve the following convex optimization problem for $\widehat{\boldsymbol{\gamma}}^m$ and $\widehat{\boldsymbol{\varphi}}^m$
$$\min_{\boldsymbol{\gamma}, \boldsymbol{\varphi} \in \mathbb{R}^p} S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) + \sum_{j=1}^p \hat{w}_j^{m-1}|\gamma_j| + \sum_{j=1}^p \bar{w}_j^{m-1}|\varphi_j|.$$
   - (b) Update the weights
$$\hat{w}_j^m = p_{\lambda_1}'(|\hat{\gamma}_j^m|),\ \bar{w}_j^m = p_{\lambda_2}'(|\hat{\varphi}_j^m|),\ j = 1, \ldots, p.$$

L_1-penalized COSALES regression

For the lasso problem (2.15), let $\check{M} = (\lambda_1^{\text{lasso}}/\lambda_2^{\text{lasso}}) \vee (\lambda_2^{\text{lasso}}/\lambda_1^{\text{lasso}})$. Define set $A_0 = (A_1, A_2')$, where $A_2' = \{j + p : \omega_j^* \neq 0\}$. Let $\mathscr{C}_M = \{\boldsymbol{\delta} \in \mathbb{R}^{2p} : \|\boldsymbol{\delta}_{A_0^c}\|_1 \leq M\|\boldsymbol{\delta}_{A_0}\|_1 \neq 0\}$ for $M \geq 1$. For $k = 1, 2$, let $\rho_{k\bullet\min} = \lambda_{\min}(n^{-1}\mathbf{X}_{A_k}^{\mathrm{T}}\mathbf{X}_{A_k})$ and $\rho_{k\bullet\max} = \lambda_{\max}(n^{-1}\mathbf{X}_{A_k}^{\mathrm{T}}\mathbf{X}_{A_k})$. Denote $\phi_{\min} = \rho_{1\bullet\min} \wedge \rho_{2\bullet\min}$ and $\phi_{\max} = \rho_{1\bullet\max} \vee \rho_{2\bullet\max}$. Assume $\phi_{\min} > 0$. Let $\mathbf{I}_2$ be a 2×2 identity matrix and let $\otimes$ denote the Kronecker product. To establish an error bound on the COSALES lasso estimators, the following conditions on the design matrix $\mathbf{X}$ and the random errors $\boldsymbol{\varepsilon}$ are imposed:

(C1′) The columns of $\mathbf{X}$ is normalizable, that is, $M_0 = \max_{1 \leq j \leq p} \frac{\|X_j\|_2}{\sqrt{n}} \in (0, \infty)$.

(C2′) $M_1 = \|\mathbf{X}^{\mathrm{T}}\boldsymbol{\omega}^*\|_\infty \in (0, \infty)$.

(C3′) The random errors ε_i are i.i.d. mean zero sub-Gaussian random variables.

(C4′) $\bar{\kappa} = \kappa(3\check{M}) \in (0, \infty)$, where $\kappa(M) = \inf_{\boldsymbol{\delta} \in \mathscr{C}_M} \boldsymbol{\delta}^{\mathrm{T}}[\mathbf{I}_2 \otimes (n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X})]\boldsymbol{\delta}/\|\boldsymbol{\delta}\|_2^2$.

(C5′) $\bar{\varrho} = \varrho(3\check{M}) \in (0, \infty)$, where $\varrho(M) = \inf_{\boldsymbol{\delta} \in \mathscr{C}_M} \frac{\boldsymbol{\delta}^{\mathrm{T}}[\mathbf{I}_2 \otimes (n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X})]\boldsymbol{\delta}}{\|\boldsymbol{\delta}_{A_0}\|_1 \|\boldsymbol{\delta}\|_\infty}$.

Theorem 2.3

In model (2.12), suppose the true parameter vectors $\boldsymbol{\gamma}^*$ and $\boldsymbol{\omega}^*$ are respectively s_1-sparse and s_2-sparse and assume conditions (C1′-C3′) hold. Let $\widehat{\boldsymbol{\gamma}}^{\text{lasso}}$ and $\widehat{\boldsymbol{\varphi}}^{\text{lasso}}$ be any optimal solutions to the L_1-penalized COSALES estimation problem (2.15). Then with probability at least $1 - \pi_1^{\text{ALS}}$,

$$\left\| \begin{pmatrix} \widehat{\boldsymbol{\gamma}}^{\text{lasso}} \\ \widehat{\boldsymbol{\varphi}}^{\text{lasso}} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\gamma}^* \\ \boldsymbol{\varphi}^* \end{pmatrix} \right\|_2 \leq 3(s_1 + s_2)^{1/2}(\lambda_1^{\text{lasso}} \vee \lambda_2^{\text{lasso}})(2\bar{\kappa}c_0)^{-1}$$

if condition (C4′) holds and

$$\left\| \begin{pmatrix} \widehat{\boldsymbol{\gamma}}^{\text{lasso}} \\ \widehat{\boldsymbol{\varphi}}^{\text{lasso}} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\gamma}^* \\ \boldsymbol{\varphi}^* \end{pmatrix} \right\|_\infty \leq 3(\lambda_1^{\text{lasso}} \vee \lambda_2^{\text{lasso}})(2\bar{\varrho}c_0)^{-1}$$

if condition (C5′) holds, where

$$\pi_1^{\text{ALS}} = 2p \exp\left(-\frac{Cn(\lambda_1^{\text{lasso}})^2}{4M_0^2 M_1^2 (K_1+K_2)^2}\right) + 2p \exp\left(-\frac{Cn(\lambda_2^{\text{lasso}})^2}{4M_0^2 M_1^2 K_2^2}\right),$$

$c_0 = 2^{-1}[(1+4\underline{c}) - (1+16\underline{c}^2)^{1/2}]$, $K_1 = \|\varepsilon_i\|_{\text{SG}}$, $K_2 = \|\Psi_\tau'(\varepsilon_i - e_\tau)\|_{\text{SG}}$, and $C > 0$ is an absolute constant. □

Nonconvex penalized COSALES regression

We show that the oracle estimators $\widehat{\boldsymbol{\gamma}}^{\text{oracle}}$ and $\widehat{\boldsymbol{\varphi}}^{\text{oracle}}$ can be achieved with overwhelming probability by Algorithm 4 under rather general conditions. Indeed, suppose the minimal signal strength of $\boldsymbol{\gamma}^*$ and $\boldsymbol{\omega}^*$ satisfies

(A0′) $\min_{j\in A_1} |\gamma_j^*| > (a+1)\lambda_1$ and $\min_{j\in A_2} |\omega_j^*| > (a+1)|e_\tau|^{-1}\lambda_2$.

Theorem 2.4

Suppose in model (2.12) $\boldsymbol{\gamma}^*$ and $\boldsymbol{\omega}^*$ are respectively s_1-sparse and s_2-sparse and satisfy assumption (A0′). Take $\widehat{\boldsymbol{\gamma}}^{\text{lasso}}$ and $\widehat{\boldsymbol{\varphi}}^{\text{lasso}}$ as the initial values and assume conditions (C1′-C3′) hold. Take $\lambda \geq 3s^{1/2}(\lambda_1^{\text{lasso}} \vee \lambda_2^{\text{lasso}})(2a_0c_0\bar{\kappa})^{-1}$ when (C4′) holds, or take $\lambda \geq 3(\lambda_1^{\text{lasso}} \vee \lambda_2^{\text{lasso}})(2a_0c_0\bar{\varrho})^{-1}$ when (C5′) holds, or take $\lambda \geq 3(\lambda_1^{\text{lasso}} \vee \lambda_2^{\text{lasso}})(2a_0c_0)^{-1}[(s^{1/2}\bar{\kappa}^{-1}) \wedge \bar{\varrho}^{-1}]$ when both (C4′) and (C5′) hold. The LLA algorithm (Algorithm 4) converges to the oracle estimators $\widehat{\boldsymbol{\gamma}}^{\text{oracle}}$ and $\widehat{\boldsymbol{\varphi}}^{\text{oracle}}$ in two iterations with probability at least $1-\pi_1^{\text{ALS}}-\pi_2^{\text{ALS}}-\pi_3^{\text{ALS}}$,

where π_1^{ALS} is given in Theorem 2.3,

$$\begin{aligned}\pi_2^{\mathrm{ALS}} &= \Gamma(2^{-1}Q_2\lambda; n, s_1, K_1+K_2, M_0M_1, M_1^2\rho_{1\bullet\max}, \nu_1)\\ &+ \Gamma(2^{-1}Q_2\lambda; n, s_2, K_2, M_0M_1, M_1^2\rho_{2\bullet\max}, \nu_2)\\ &+ 2(p-s_1)\exp\bigg(-\frac{Ca_1^2n\lambda^2}{4M_0^2M_1^2(K_1+K_2)^2}\bigg) + 2(p-s_2)\exp\bigg(-\frac{Ca_1^2n\lambda^2}{4M_0^2M_1^2K_2^2}\bigg),\end{aligned}$$

and

$$\begin{aligned}\pi_3^{\mathrm{ALS}} &= \Gamma(2^{-1}c_0\phi_{\min}\bar{R}; n, s_1, K_1+K_2, M_0M_1, M_1^2\rho_{1\bullet\max}, \nu_1)\\ &\quad + \Gamma(2^{-1}c_0\phi_{\min}\bar{R}; n, s_2, K_2, M_0M_1, M_1^2\rho_{2\bullet\max}, \nu_2),\end{aligned}$$

where $s = s_1 + s_2$, $\lambda = \lambda_1 \wedge \lambda_2$, $Q_2 = a_1c_0\phi_{\min}[2(1+2\bar{c})M_0\phi_{\max}^{1/2}]^{-1}$, $\nu_1 = \mathrm{var}(\varepsilon_i + \Psi_\tau'(\varepsilon_i - e_\tau))$, $\nu_2 = \mathrm{var}(\Psi_\tau'(\varepsilon_i - e_\tau))$, $\bar{R} = (\min_{j\in A_1}|\gamma_j^*| - a\lambda_1) \wedge (\min_{j\in A_2}|\varphi_j^*| - a\lambda_2)$, $C > 0$ is an absolute constant, c_0, K_1, K_2 are given in Theorem 2.3, and $\Gamma(\cdot)$ is given in Theorem 2.2. □

For SCAD and MCP penalized COSALES regressions, the LLA algorithm (Algorithm 4) starting from the zero vector can also be used as long as we can take $\lambda_k = \lambda_k^{\mathrm{lasso}}, k = 1, 2$.

Corollary 2.2

Assume the same framework of Theorem 2.4 and suppose the SCAD penalty (2.8) or MCP (2.9) is used. If condition (C4′) holds and $2a_0c_0\bar{\kappa} \geq 3\check{M}s^{1/2}$, or if condition (C5′) holds and $2a_0c_0\bar{\varrho} \geq 3\check{M}$, or if both (C4′) and (C5′) hold and $3\check{M}[(s^{1/2}\bar{\kappa}^{-1}) \wedge \bar{\varrho}^{-1}] \leq 2a_0c_0$, then the LLA algorithm (Algorithm 4) initialized by zero converges to the oracle estimators $\widehat{\boldsymbol{\gamma}}^{\mathrm{oracle}}$ and $\widehat{\boldsymbol{\varphi}}^{\mathrm{oracle}}$ after three iterations with probability at least $1 - \check{\pi}_1^{\mathrm{ALS}} - \pi_2^{\mathrm{ALS}} - \pi_3^{\mathrm{ALS}}$, where

$$\check{\pi}_1^{\mathrm{ALS}} = 2p\exp\bigg(-\frac{Cn\lambda_1^2}{4M_0^2M_1^2(K_1+K_2)^2}\bigg) + 2p\exp\bigg(-\frac{Cn\lambda_2^2}{4M_0^2M_1^2K_2^2}\bigg),$$

π_2^{ALS} and π_3^{ALS} are given in Theorem 2.4, and $s = s_1 + s_2$. □

Remark 2.2 We can easily modify (2.13) to allow certain subsets of coefficients not to be penalized. Let $\mathscr{R}_1$ and $\mathscr{R}_2$ be the index sets of unpenalized components of $\boldsymbol{\gamma}$ and $\boldsymbol{\varphi}$, respectively. Then (2.13) can be modified as

$$\min_{\boldsymbol{\gamma},\boldsymbol{\varphi}\in\mathbb{R}^p} S_n(\boldsymbol{\gamma},\boldsymbol{\varphi}) + \sum_{j\in\mathscr{R}_1^{\mathrm{c}}} p_{\lambda_1}(\gamma_j) + \sum_{j\in\mathscr{R}_2^{\mathrm{c}}} p_{\lambda_2}(\varphi_j).$$

The COSALES algorithm can be readily used to solve the above problem. Moreover, similar theoretical results can be established with slight modifications. □

2.4.3 Simulation examples

We demonstrate the selection and estimation accuracy of the COSALES regression through two numerical simulations. For the nonconvex penalties used in both simulations, we fix $\gamma = 3.7$ for the SCAD penalty and $\gamma = 2$ for the MCP.

EXAMPLE 2. We consider the same model (2.11) that was used in Example 1, but different from the approach used there, we estimate the coefficients through the nonconvex penalized COSALES regression (2.13). Again we choose $p = 600$ and independently simulate a training set of size $n = 300$ for fitting and a validation set of size $n = 300$ for tuning. The tuning parameter is selected by minimizing the validation error $\sum_{i\in\text{validation}}\{\Psi_{0.5}(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\gamma}}) + \Psi_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\gamma}} - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\varphi}})\}$ for the computed estimates $\widehat{\boldsymbol{\gamma}}$ and $\widehat{\boldsymbol{\varphi}}$. We pick a fairly extreme τ-value ($\tau = 0.95$) for easy separation of the conditional mean and scale functions. Both the COSALES lasso and two variations of the LLA algorithm for each of the SCAD and MCP penalized COSALES regressions are implemented.

Based on 100 independent runs, the following measurements are calculated to evaluate the sparsity recovery and estimation performance of the COSALES estimators:

$|\hat{A}_1|, |\hat{A}_2|$: the average size of the active sets for $\widehat{\boldsymbol{\gamma}}$ and $\widehat{\boldsymbol{\varphi}}$, respectively, $\hat{A}_1 = \{j : \hat{\gamma}_j \neq 0\}$ and $\hat{A}_2 = \{j : \hat{\varphi}_j \neq 0\}$.

p_{a_1}, p_{a_2} : proportions of the events $A_1 \subset \hat{A}_1$ and $A_2 \subset \hat{A}_2$, respectively, where $A_1 = \{6, 12, 15, 20\}$ denotes the active set of $\boldsymbol{\gamma}^*$ and $A_2 = \{1\}$ denotes the active set of $\boldsymbol{\varphi}^*$.

$R_1^{\gamma}, R_1^{\varphi}$: the average L_1 risks, $R_1^{\gamma} = \|\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*\|_1$ and $R_1^{\varphi} = \|\widehat{\boldsymbol{\varphi}} - \boldsymbol{\varphi}^*\|_1$.

$R_2^{\gamma}, R_2^{\varphi}$: the average L_2 risks, $R_2^{\gamma} = \|\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}^*\|_2$ and $R_2^{\varphi} = \|\widehat{\boldsymbol{\varphi}} - \boldsymbol{\varphi}^*\|_2$.

The results are summarized in Table 2.2, from which we can draw the following conclusions:

(1) The COSALES regression (with lasso or nonconvex penalties) can recover the sparse patterns in both the mean and scale functions with overwhelming probabilities.

(2) The COSALES lasso tends to select a lot more irrelevant covariates and has much larger estimation errors than the nonconvex penalized COSALES regression (with the SCAD penalty or MCP).

(3) The three-step LLA algorithm starting from zero produces similar results to the two-step LLA algorithm starting from the lasso solution.

EXAMPLE 3. In this example, we simulate data from the following normal linear heteroscedastic model

$$y = x_6 + x_{12} + x_{15} + x_{20} + (0.7x_1 + 0.7x_{12})\varepsilon, \tag{2.17}$$

where the covariates are simulated by setting $x_1 = \Phi(z_1), x_{12} = \Phi(z_{12})$, and $x_j = z_j, j \neq 1, 12$, where $(z_1, \ldots, z_p)^{\mathrm{T}} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = (0.5^{|i-j|})_{p\times p}$, and $\Phi(\cdot)$ is the CDF of the standard normal distribution. The random error $\varepsilon \sim N(0, 1)$. Note that in model (2.11), the active sets of the true parameter vectors do not overlap, so the SALES

Table 2.2: Numerical summary of simulation results from the lasso, SCAD and MCP penalized COSALES regression for model (2.11): $y = x_6 + x_{12} + x_{15} + x_{20} + (0.7x_1)\varepsilon$. The selection accuracy is measured by the number of selected variables $|\hat{A}_1|$ and $|\hat{A}_2|$, and the proportions p_{a_1} and p_{a_2} of covering the true active sets. The estimation accuracy is measured by the L_1 risks R_1^{γ} and R_1^{φ}, and the L_2 risks R_2^{γ} and R_2^{φ}. The results are shown as averages over 100 replicates with standard errors listed in the parentheses. A fairly extreme τ-value ($\tau = 0.95$) is used in the simulation for easy separation of the mean and scale

Method	$\lvert\hat{A}_1\rvert$	$\lvert\hat{A}_2\rvert$	p_{a_1}	p_{a_2}	R_1^{γ}	R_1^{φ}	R_2^{γ}	R_2^{φ}
COSALES-lasso	26.88	13.36	100%	100%	0.407	0.378	0.124	0.294
	(1.04)	(0.45)	(0)	(0)	(0.012)	(0.008)	(0.002)	(0.006)
COSALES-SCAD*	7.24	1.01	100%	100%	0.095	0.072	0.048	0.072
	(0.10)	(0.01)	(0)	(0)	(0.004)	(0.005)	(0.002)	(0.005)
COSALES-SCAD^0	8.85	1.01	100%	100%	0.107	0.065	0.049	0.065
	(0.57)	(0.01)	(0)	(0)	(0.005)	(0.005)	(0.002)	(0.005)
COSALES-MCP*	6.46	1.01	100%	100%	0.089	0.070	0.045	0.070
	(0.38)	(0.01)	(0)	(0)	(0.004)	(0.005)	(0.002)	(0.005)
COSALES-MCP^0	7.08	1.01	100%	100%	0.102	0.067	0.052	0.067
	(0.44)	(0.01)	(0)	(0)	(0.006)	(0.005)	(0.003)	(0.005)

regression can detect active variables in the scale. However, in model (2.17) the active set for the mean, $A_1 = \{6, 12, 15, 20\}$, overlaps with the active set for the scale, $A_2 = \{1, 12\}$. Thus, the SALES regression cannot recover the variable x_{12} in the scale function. We show by this Monte Carlo simulation that the COSALES regression can recover the sparse patterns in both the mean and scale functions. We fix $p = 600$ and independently simulate a training set of size $n = 500$ for fitting and a validation set of the same size for tuning. We select the regularization parameter by minimizing the validation error $\sum_{i \in \text{validation}} \{\Psi_{0.5}(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\gamma}}) + \Psi_{\tau}(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\gamma}} - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\varphi}})\}$ for the computed estimate $\widehat{\boldsymbol{\gamma}}$ and $\widehat{\boldsymbol{\varphi}}$. In order to separate the mean and scale easily, we again pick $\tau = 0.95$. We implement the COSALES lasso and two variations of the LLA algorithm as were done in Examples 2 for each of the SCAD and MCP penalized COSALES regressions.

Based on 100 independent runs, the same measurements of performance as in Example 2 are calculated to evaluate the sparsity recovery and estimation accuracy of the COSALES

estimation. The results are summarized in Table 2.3. Same conclusions in Example 2 can be drawn here.

Table 2.3: Numerical summary of simulation results from the the lasso, SCAD and MCP penalized COSALES regression for model (2.17): $y = x_6 + x_{12} + x_{15} + x_{20} + (0.7x_1 + 0.7x_{12})\varepsilon$. The selection accuracy is measured by the number of selected variables $|\hat{A}_1|$ and $|\hat{A}_2|$, and the proportions p_{a_1} and p_{a_2} of covering the true active sets. The estimation accuracy is measured by the L_1 risks R_1^{γ} and R_1^{φ}, and the L_2 risks R_2^{γ} and R_2^{φ}. The results are shown as averages over 100 replicates with standard errors listed in the parentheses. A fairly extreme τ-value ($\tau = 0.95$) is used in the simulation for easy separation of the mean and scale

Method	$\lvert\hat{A}_1\rvert$	$\lvert\hat{A}_2\rvert$	p_{a_1}	p_{a_2}	R_1^{γ}	R_1^{φ}	R_2^{γ}	R_2^{φ}
COSALES-lasso	27.92	12.67	100%	100%	0.719	0.450	0.249	0.282
	(0.98)	(0.49)	(0)	(0)	(0.018)	(0.011)	(0.006)	(0.008)
COSALES-SCAD*	6.80	2.06	100%	100%	0.167	0.210	0.089	0.161
	(0.52)	(0.04)	(0)	(0)	(0.008)	(0.014)	(0.004)	(0.010)
COSALES-SCAD^0	5.70	2.02	100%	100%	0.157	0.199	0.090	0.148
	(0.25)	(0.01)	(0)	(0)	(0.006)	(0.013)	(0.003)	(0.009)
COSALES-MCP*	5.95	2.06	100%	100%	0.153	0.221	0.086	0.165
	(0.35)	(0.03)	(0)	(0)	(0.006)	(0.015)	(0.003)	(0.010)
COSALES-MCP^0	6.00	2.04	100%	100%	0.180	0.205	0.098	0.154
	(0.36)	(0.02)	(0)	(0)	(0.009)	(0.014)	(0.004)	(0.010)

2.5 Real Data Example

We apply the SALES and COSALES regressions to a real data set reported in Scheetz et al. (2006). The data set consists of gene expression levels of more than 31,000 probes obtained from 120 rats. The expressions are analyzed on a logarithmic scale (base 2). As was done in Scheetz et al. (2006), we exclude the probes that were not expressed in the eye or that lacked sufficient variation. Among those 18,976 probes left, we study how the expressions of other genes are associated with the gene *TRIM32* (probe 1389163_at). This gene was found to be associated with Bardet–Biedl syndrome, which is a disorder that affects many parts of the body including the retina. For all the other genes, we first standardize them

and select the 3,000 probes with the largest variances. These 3,000 probes are then ranked according to the magnitude of the correlations between their expressions and that of probe 1389163_at. We choose the top 300 probes with the largest correlations in magnitude for the analysis.

The third column of Table 2.4 lists the number of active variables selected by the SALES regressions with lasso, SCAD and MCP penalties, fitted on the whole data set of 120 subjects. For both SCAD and MCP penalized SALES regressions, the two variations of the LLA algorithm were used. The tuning parameter for each method is selected by five-fold cross-validation. The last two columns of Table 2.4 summarize the results from 50 random partitions. Each partition randomly splits the data into a training set with 80 observations and a validation set with 40 observations. We fit the model with the training set using five-fold cross-validation for tuning and calculate the predicted loss $(1/40)\sum_{i\in\text{validation}}\Psi_\tau(y_i-\hat{\beta}_0-\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}})$ based on the validation set. The average number of active variables selected and the average predicted loss are calculated from the 50 partitions with their respective standard errors listed in the parentheses. Table 2.4 reveals two interesting findings. First, the nonconvex penalized SALES regression selects less variables than the SALES lasso, but there is no obvious improvement of the nonconvex penalized SALES regression over the SALES lasso in terms of predicted loss. Second, for all SALES regressions, the number of variables selected is different at different values of τ (0.3, 0.5 and 0.7). This is an indication of heteroscedasticity in the data.

To further explore the heterogeneous scale, we also apply the COSALES regression to the data. The results are summarized in Table 2.5. Columns 2 and 3 display the number of variables selected for the mean ($|\hat{A}_1|$) and scale ($|\hat{A}_2|$), and the number of variables that overlap ($|\hat{A}_1\cap\hat{A}_2|$) for each method. For all penalties, τ is set to be 0.7 in the COSALES regression. Random partitions are done in the same way as the SALES regression and the predicted loss is calculated via $(1/40)\sum_{i\in\text{validation}}\Psi_{0.5}(y_i-\hat{\gamma}_0-\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\gamma}})+\Psi_\tau(y_i-\hat{\gamma}_0-\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\gamma}}-\hat{\varphi}_0-\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\varphi}})$. The results for the random partitions are shown in columns 4 to 6. It can

be seen that the COSALES regression reveals more information about the heterogeneous scale which cannot be otherwise detected in the SALES regression or the sparse quantile regression (Wang et al., 2012) due to overlaps.

Table 2.4: Analysis of microarray data reported in Scheetz et al. (2006) using SALES regressions with lasso, SCAD and MCP penalties. Three different values of τ (0.3, 0.5 and 0.7) are used for each method. The number of active variables selected using the whole data set is given in column 3. The average number of active variables selected and average predicted loss $(1/40)\sum_{i\in\text{validation}}\Psi_\tau(y_i-\hat\beta_0-\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol\beta})$ listed in columns 4 and 5 are calculated from 50 random partitions of the original data with standard errors listed in parentheses

		All data	Random partition	
Method	τ	$\lvert\hat{A}\rvert$	$\lvert\hat{A}\rvert$	Predicted loss
SALES-lasso	0.3	22	22.00 (1.51)	0.007 (0.00055)
	0.5	25	25.38 (1.94)	0.005 (0.00036)
	0.7	20	21.90 (1.66)	0.005 (0.00022)
SALES-SCAD*	0.3	19	16.02 (2.09)	0.006 (0.00048)
	0.5	13	15.52 (1.80)	0.006 (0.00043)
	0.7	11	13.54 (1.98)	0.005 (0.00037)
SALES-SCAD^0	0.3	16	16.60 (2.03)	0.006 (0.00054)
	0.5	17	17.22 (2.36)	0.007 (0.00048)
	0.7	14	14.82 (2.18)	0.005 (0.00030)
SALES-MCP*	0.3	14	15.82 (2.56)	0.006 (0.00053)
	0.5	12	12.66 (2.58)	0.008 (0.00054)
	0.7	10	9.66 (1.78)	0.006 (0.00035)
SALES-MCP^0	0.3	11	11.74 (1.47)	0.006 (0.00057)
	0.5	13	13.24 (2.75)	0.007 (0.00058)
	0.7	13	14.18 (3.36)	0.006 (0.00034)

Table 2.5: Analysis of microarray data reported in Scheetz et al. (2006) using COSALES regressions with lasso, SCAD and MCP penalties. In this analysis, $\tau = 0.7$ is used. The number of active variables selected for the mean and scale using the whole data set is given in columns 2 and 3. The average number of active variables selected for the mean and scale and average predicted loss $(1/40)\sum_{i\in\text{validation}} \Psi_{0.5}(y_i-\hat{\gamma}_0-\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\gamma}})+\Psi_\tau(y_i-\hat{\gamma}_0-\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\gamma}}-\hat{\varphi}_0-\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\varphi}})$ listed in columns 4 to 6 are calculated from 50 random partitions of the original data with standard errors listed in parentheses

	All data			Random partition			
Method	$\lvert\hat{A}_1\rvert$	$\lvert\hat{A}_2\rvert$	$\lvert\hat{A}_1\cap\hat{A}_2\rvert$	$\lvert\hat{A}_1\rvert$	$\lvert\hat{A}_2\rvert$	$\lvert\hat{A}_1\cap\hat{A}_2\rvert$	Predicted loss
COSALES-lasso	22	10	9	22.62 (1.21)	9.80 (1.10)	7.86 (0.93)	0.010 (0.00056)
COSALES-SCAD*	19	7	6	18.92 (0.79)	5.58 (0.50)	3.90 (0.31)	0.011 (0.00067)
COSALES-SCAD^0	20	5	4	20.22 (0.98)	5.82 (0.64)	3.92 (0.44)	0.011 (0.00072)
COSALES-MCP*	10	3	1	10.96 (2.32)	3.08 (1.40)	1.38 (0.74)	0.014 (0.00096)
COSALES-MCP^0	10	4	3	12.94 (1.83)	4.56 (1.04)	1.46 (0.42)	0.012 (0.00083)

2.6 Proofs

In this section, we give the proofs of the main theoretical results stated in previous sections. First of all, let us state two lemmas on the properties of the asymmetric squared error loss $\Psi_\tau(\cdot)$ given in (2.1). These properties play an important role in the proofs of many results to be presented below. Let $w_\tau(u) = |\tau - I(u < 0)|$ and recall that $\underline{c} = \tau \wedge (1-\tau)$ and $\bar{c} = \tau \vee (1-\tau)$.

Lemma 2.1

The asymmetric squared error loss $\Psi_\tau(\cdot)$ is continuously differentiable, but is not twice differentiable at zero when $\tau \neq 0.5$. Moreover, for any $u, u_0 \in \mathbb{R}$ and $\tau \in (0,1)$, we have

$$\underline{c}(u-u_0)^2 \leq \Psi_\tau(u) - \Psi_\tau(u_0) - \Psi'_\tau(u_0)(u-u_0) \leq \bar{c}(u-u_0)^2.$$

It follows that $\Psi_\tau(\cdot)$ is strongly convex. □

Lemma 2.2

For any $u, u_0 \in \mathbb{R}$ and $\tau \in (0,1)$, we have

$$2\underline{c}|u-u_0| \leq |\Psi'_\tau(u) - \Psi'_\tau(u_0)| \leq 2\bar{c}|u-u_0|.$$

It follows immediately that $\Psi'_\tau(\cdot)$ is Lipschitz continuous. □

Proof 2.1 (Proof of Lemma 2.1)

It is easy to see that $\underline{c} \leq w_\tau(u) \leq \bar{c}$ for any $u \in \mathbb{R}$. Note that $\Psi'_\tau(u) = 2w_\tau(u)u$, which is continuous and which is not differentiable at $u = 0$ when $\tau \neq 0.5$. To show the inequalities, consider the following situations. If $w_\tau(u) \geq w_\tau(u_0)$, it follows that

$$\begin{aligned}
&\Psi_\tau(u) - \Psi_\tau(u_0) - \Psi'_\tau(u_0)(u-u_0) \\
&= w_\tau(u)u^2 - w_\tau(u_0)u_0^2 - 2w_\tau(u_0)u_0(u-u_0) \\
&= w_\tau(u_0)(u-u_0)^2 + \{w_\tau(u) - w_\tau(u_0)\}u^2 \\
&\geq w_\tau(u_0)(u-u_0)^2 \geq \underline{c}(u-u_0)^2.
\end{aligned}$$

Otherwise, if $w_\tau(u) < w_\tau(u_0)$, then we know that $\underline{c} = w_\tau(u)$, $\bar{c} = w_\tau(u_0)$ and $u_0 u \leq 0$. It follows that

$$\begin{aligned}
&\Psi_\tau(u) - \Psi_\tau(u_0) - \Psi'_\tau(u_0)(u-u_0) \\
&= \underline{c}u^2 - \bar{c}u_0^2 - 2\bar{c}u_0(u-u_0) \\
&\geq \underline{c}u^2 - 2\underline{c}u_0 u + \underline{c}u_0^2 = \underline{c}(u-u_0)^2.
\end{aligned}$$

Therefore, the first inequality holds. Similarly, we can show the second inequality. □

Proof 2.2 (Proof of Lemma 2.2)

If $u = 0$ or $u_0 = 0$, then the inequalities hold trivially. If $uu_0 > 0$, we know that

$w_\tau(u) = w_\tau(u_0)$. It follows that

$$2\underline{c}|u - u_0| \leq |\Psi'_\tau(u) - \Psi'_\tau(u_0)| = 2w_\tau(u)|u - u_0| \leq 2\bar{c}|u - u_0|.$$

If instead, $uu_0 < 0$, there are two cases: $u > 0, u_0 < 0$ or $u < 0, u_0 > 0$. For the first case, we have

$$2\underline{c}|u - u_0| \leq |\Psi'_\tau(u) - \Psi'_\tau(u_0)| = 2\tau u - 2(1-\tau)u_0 \leq 2\bar{c}|u - u_0|.$$

For the second case, we have

$$2\underline{c}|u - u_0| \leq |\Psi'_\tau(u) - \Psi'_\tau(u_0)| = -2(1-\tau)u + 2\tau u_0 \leq 2\bar{c}|u - u_0|.$$

This completes the proof. □

The following lemma deals with sub-Gaussian random variables.

Lemma 2.3

Suppose that $Z, Z_1, \ldots, Z_n \in \mathbb{R}$ are i.i.d. sub-Gaussian random variables. Let $\mathbf{Z} = (Z_1, \ldots, Z_n)^{\mathrm{T}}$, $K = \|Z\|_{\mathrm{SG}}$, $Z^+ = \max(Z, 0)$ and $Z^- = \max(-Z, 0)$.

(1) If $\mathbb{E}(Z) = 0$, then there exists an absolute constant $C > 0$ such that for any $\mathbf{a} = (a_1, \ldots, a_n)^{\mathrm{T}} \in \mathbb{R}^n$ and any $t \geq 0$,

$$\mathrm{P}(|\mathbf{a}^{\mathrm{T}}\mathbf{Z}| \geq t) \leq 2\exp\left(-\frac{Ct^2}{K^2\|\mathbf{a}\|_2^2}\right).$$

(2) Let $\mathbf{A}$ be a fixed $m \times n$ matrix. If $\mathbb{E}(Z) = 0$ and $\mathrm{var}(Z) = 1$, then there exists an absolute constant $C > 0$ such that for any $t \geq 0$,

$$\mathrm{P}(|\|\mathbf{AZ}\|_2 - \|\mathbf{A}\|_F| \geq t) \leq 2\exp\left(-\frac{Ct^2}{K^4\|\mathbf{A}\|_2^2}\right),$$

where $\|\mathbf{A}\|_F$ and $\|\mathbf{A}\|_2$ represent the Frobenius and L_2 norms of matrix $\mathbf{A}$ respectively.

(3) Let $\mathbf{A}$ be a fixed $m \times n$ matrix. Let $\mathbf{e}_j \in \mathbb{R}^m$ be the unit vector with its jth component one, $j = 1, \ldots, m$. Suppose $M \equiv \max_{1 \le j \le m} n^{-1/2} \|\mathbf{A}^{\mathrm{T}} \mathbf{e}_j\|_2 \in (0, \infty)$ and $\rho \equiv \lambda_{\max}(n^{-1}\mathbf{A}\mathbf{A}^{\mathrm{T}}) \in (0, \infty)$. If $\mathbb{E}(Z) = 0$ and $\nu = \mathrm{var}(Z) \in (0, \infty)$, then there exists an absolute constant $C > 0$ such that for any $t \ge 0$,

$$\begin{aligned} &\mathrm{P}(\|n^{-1}\mathbf{A}\mathbf{Z}\|_2 \ge t) \le \Gamma(t; n, m, K, M, \rho, \nu) \\ &= 2m \exp\left(-\frac{Cnt^2}{K^2 M^2 m}\right) \wedge 2\exp\left(-\frac{C\nu^2[(n^{1/2}t - \nu m^{1/2}\rho^{1/2})^+]^2}{K^4 \rho}\right). \end{aligned}$$

(4) The random variables Z^+ and Z^- are also sub-Gaussian. Moreover, for any $c_1, c_2 \in \mathbb{R}$, $c_1 Z^+ + c_2 Z^-$ is sub-Gaussian. □

Proof 2.3 (Proof of Lemma 2.3)

(1) This part follows directly from Proposition 5.10 of Vershynin (2010).

(2) This part follows from Theorem 2.1 of Rudelson and Vershynin (2013).

(3) On one hand, we have by part (1) that

$$\mathrm{P}\left(\left\|\frac{\mathbf{A}}{n}\mathbf{Z}\right\|_2 \ge t\right) \le \mathrm{P}\left(\left\|\frac{\mathbf{A}}{\sqrt{n}}\mathbf{Z}\right\|_\infty \ge \frac{t\sqrt{n}}{\sqrt{m}}\right) \le 2m \exp\left(-\frac{Cnt^2}{K^2 M^2 m}\right).$$

One the other hand, note that $\|n^{-1/2}\mathbf{A}\|_F = \sqrt{\mathrm{Tr}(\mathbf{A}\mathbf{A}^{\mathrm{T}}/n)} \le \sqrt{m\rho}$ and $\|n^{-1/2}\mathbf{A}\|_2^2 = \lambda_{\max}(\mathbf{A}^{\mathrm{T}}\mathbf{A}/n) = \lambda_{\max}(\mathbf{A}\mathbf{A}^{\mathrm{T}}/n) = \rho$. We have by part (2)

$$\begin{aligned} &\mathrm{P}\left(\left\|\frac{\mathbf{A}}{n}\mathbf{Z}\right\|_2 \ge t\right) \le \mathrm{P}\left(\left\|\frac{\mathbf{A}}{\sqrt{n}}\frac{\mathbf{Z}}{\nu}\right\|_2 - \left\|\frac{\mathbf{A}}{\sqrt{n}}\right\|_F \ge \frac{t\sqrt{n}}{\nu} - \sqrt{m\rho}\right) \\ &\le \mathrm{P}\left(\left|\left\|\frac{\mathbf{A}}{\sqrt{n}}\frac{\mathbf{Z}}{\nu}\right\|_2 - \left\|\frac{\mathbf{A}}{\sqrt{n}}\right\|_F\right| \ge \left(\frac{t\sqrt{n}}{\nu} - \sqrt{m\rho}\right)^+\right) \\ &\le 2\exp\left(-\frac{C\nu^2[(n^{1/2}t - \nu m^{1/2}\rho^{1/2})^+]^2}{K^4\rho}\right). \end{aligned}$$

(4) Note that by definition, we have $K \in (0, \infty)$ and $(\mathbb{E}|Z|^p)^{1/p} \leq K\sqrt{p}$, $\forall p \geq 1$. It follows immediately that $(\mathbb{E}|Z^+|^p)^{1/p} \leq (\mathbb{E}|Z|^p)^{1/p} \leq K\sqrt{p}$ and $(\mathbb{E}|Z^-|^p)^{1/p} \leq (\mathbb{E}|Z|^p)^{1/p} \leq K\sqrt{p}$, $\forall p \geq 1$. Now by Lemma 5.5 of Vershynin (2010), we conclude that Z^+ and Z^- are both sub-Gaussian. For any $c_1, c_2 \in \mathbb{R}$, by Minkowski inequality,

$$\begin{aligned}(\mathbb{E}|c_1 Z^+ + c_2 Z^-|^p)^{1/p} &\leq |c_1|(\mathbb{E}|Z^+|^p)^{1/p} + |c_2|(\mathbb{E}|Z^-|^p)^{1/p} \\ &\leq (|c_1| + |c_2|)K\sqrt{p}, \quad \forall p \geq 1.\end{aligned}$$

By Lemma 5.5 of Vershynin (2010) again, we can see that $c_1 Z^+ + c_2 Z^-$ is also sub-Gaussian. This completes the proof. □

Now we are ready to prove Theorems 2.1 and 2.2. Lemmas 2.4 and 2.5 are presented to facilitate the proofs.

Lemma 2.4

Let $\boldsymbol{\zeta} = (\zeta_i, 1 \leq i \leq n)^{\mathrm{T}}$ with $\zeta_i = \Psi'_\tau(\varepsilon_i) = 2|\tau - I(\varepsilon_i < 0)|\varepsilon_i$.

(1) For any $\boldsymbol{\beta}, \boldsymbol{\delta} \in \mathbb{R}^p$, $\langle \nabla \mathcal{L}_n(\boldsymbol{\beta} + \boldsymbol{\delta}) - \nabla \mathcal{L}_n(\boldsymbol{\beta}), \boldsymbol{\delta} \rangle \geq 2\underline{c}\|\mathbf{X}\boldsymbol{\delta}\|_2^2/n$.

(2) For any $d > 0$, $\mathrm{P}(\|\widehat{\boldsymbol{\beta}}^{\mathrm{oracle}} - \boldsymbol{\beta}^*\|_2 \geq d) \leq \mathrm{P}(\|n^{-1}\mathbf{X}_A^{\mathrm{T}}\boldsymbol{\zeta}\|_2 \geq 2\underline{c}\rho_{\min} d)$. □

Proof 2.4 (Proof of Lemma 2.4)

The first part follows from the strong convexity of $\Psi_\tau(\cdot)$. Specifically, by Lemma 2.1, we have

$$\begin{aligned}&\mathcal{L}_n(\boldsymbol{\beta} + \boldsymbol{\delta}) - \mathcal{L}_n(\boldsymbol{\beta}) - \langle \nabla \mathcal{L}_n(\boldsymbol{\beta}), \boldsymbol{\delta} \rangle \geq \underline{c}\|\mathbf{X}\boldsymbol{\delta}\|_2^2/n, \\ &\mathcal{L}_n(\boldsymbol{\beta}) - \mathcal{L}_n(\boldsymbol{\beta} + \boldsymbol{\delta}) - \langle \nabla \mathcal{L}_n(\boldsymbol{\beta} + \boldsymbol{\delta}), -\boldsymbol{\delta} \rangle \geq \underline{c}\|\mathbf{X}\boldsymbol{\delta}\|_2^2/n.\end{aligned}$$

Summing up the above two inequalities yields the desired result in part (1).

For the second part, let $\widehat{\boldsymbol{\delta}} = \widehat{\boldsymbol{\beta}}^{\text{oracle}} - \boldsymbol{\beta}^*$. By definition of $\widehat{\boldsymbol{\beta}}^{\text{oracle}}$, we have $\widehat{\boldsymbol{\delta}}_{A^c} = \mathbf{0}$ and $\nabla_A \mathcal{L}_n(\widehat{\boldsymbol{\beta}}^{\text{oracle}}) = \mathbf{0}$. Now by part (1) we have

$$
\begin{aligned}
&2\underline{c}\rho_{\min}\|\widehat{\boldsymbol{\delta}}\|_2^2 = 2\underline{c}\rho_{\min}\|\widehat{\boldsymbol{\delta}}_A\|_2^2 \le 2\underline{c}\widehat{\boldsymbol{\delta}}_A^{\mathrm{T}}(\mathbf{X}_A^{\mathrm{T}}\mathbf{X}_A/n)\widehat{\boldsymbol{\delta}}_A = 2\underline{c}\|\mathbf{X}\widehat{\boldsymbol{\delta}}\|_2^2/n \\
&\le \langle \nabla \mathcal{L}_n(\widehat{\boldsymbol{\beta}}^{\text{oracle}}) - \nabla \mathcal{L}_n(\boldsymbol{\beta}^*), \widehat{\boldsymbol{\delta}} \rangle = \langle -\nabla_A \mathcal{L}_n(\boldsymbol{\beta}^*), \widehat{\boldsymbol{\delta}}_A \rangle \\
&\le \|\nabla_A \mathcal{L}_n(\boldsymbol{\beta}^*)\|_2 \|\widehat{\boldsymbol{\delta}}_A\|_2 = \|n^{-1}\mathbf{X}_A^{\mathrm{T}}\boldsymbol{\zeta}\|_2 \|\widehat{\boldsymbol{\delta}}\|_2,
\end{aligned}
$$

which implies that $2\underline{c}\rho_{\min}\|\widehat{\boldsymbol{\beta}}^{\text{oracle}} - \boldsymbol{\beta}^*\|_2 \le \|n^{-1}\mathbf{X}_A^{\mathrm{T}}\boldsymbol{\zeta}\|_2$. The result of part (2) then follows.□

Proof 2.5 (Proof of Theorem 2.1)
Let $\widehat{\boldsymbol{\delta}} = \widehat{\boldsymbol{\beta}}^{\text{lasso}} - \boldsymbol{\beta}^*$ and $z_\infty^* = \|\nabla \mathcal{L}_n(\boldsymbol{\beta}^*)\|_\infty$. Note that $\widehat{\boldsymbol{\beta}}^{\text{lasso}}$ satisfies the Karush–Kuhn–Tucker (KKT) condition

$$
\nabla \mathcal{L}_n(\widehat{\boldsymbol{\beta}}^{\text{lasso}}) + \mathbf{g} = \mathbf{0},
$$

where $g_j = \lambda_{\text{lasso}} \text{sgn}(\hat{\beta}_j^{\text{lasso}})$ if $\hat{\beta}_j^{\text{lasso}} \neq 0$ and $g_j \in [-\lambda_{\text{lasso}}, \lambda_{\text{lasso}}]$ if $\hat{\beta}_j^{\text{lasso}} = 0$. It follows that $\hat{\beta}_j^{\text{lasso}} g_j = \lambda_{\text{lasso}} |\hat{\beta}_j^{\text{lasso}}|, \forall j$. Since $\boldsymbol{\beta}_{A^c}^* = \mathbf{0}$, we have $\widehat{\boldsymbol{\delta}}_{A^c} = \widehat{\boldsymbol{\beta}}_{A^c}^{\text{lasso}}$. By Lemma 2.4 and Hölder's inequality, we get

$$
\begin{aligned}
&0 \le 2\underline{c}\|\mathbf{X}\widehat{\boldsymbol{\delta}}\|_2^2/n \le \langle \nabla \mathcal{L}_n(\widehat{\boldsymbol{\beta}}^{\text{lasso}}) - \nabla \mathcal{L}_n(\boldsymbol{\beta}^*), \widehat{\boldsymbol{\delta}} \rangle = \langle -\mathbf{g} - \nabla \mathcal{L}_n(\boldsymbol{\beta}^*), \widehat{\boldsymbol{\delta}} \rangle \\
&= \langle \widehat{\boldsymbol{\delta}}_A, -\mathbf{g}_A - \nabla_A \mathcal{L}_n(\boldsymbol{\beta}^*) \rangle + \langle \widehat{\boldsymbol{\beta}}_{A^c}^{\text{lasso}}, -\mathbf{g}_{A^c} - \nabla_{A^c} \mathcal{L}_n(\boldsymbol{\beta}^*) \rangle \qquad (2.18) \\
&\le (z_\infty^* + \lambda_{\text{lasso}}) \|\widehat{\boldsymbol{\delta}}_A\|_1 + (z_\infty^* - \lambda_{\text{lasso}}) \|\widehat{\boldsymbol{\delta}}_{A^c}\|_1.
\end{aligned}
$$

Under the event $\mathcal{E} = \{z_\infty^* \le 2^{-1}\lambda_{\text{lasso}}\}$, from (2.18) we get

$$
\|\widehat{\boldsymbol{\delta}}_{A^c}\|_1 \le \frac{z_\infty^* + \lambda_{\text{lasso}}}{z_\infty^* - \lambda_{\text{lasso}}} \|\widehat{\boldsymbol{\delta}}_A\|_1 \le 3\|\widehat{\boldsymbol{\delta}}_A\|_1,
$$

which implies that $\widehat{\boldsymbol{\delta}} \in \mathscr{C}$. Now under $\mathcal{E}$, by condition (C3), it follows from (2.18) that

$$2\underline{c}\kappa\|\widehat{\boldsymbol{\delta}}\|_2^2 \leq (3/2)\lambda_{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_A\|_1 \leq (3/2)\lambda_{\text{lasso}}s^{1/2}\|\widehat{\boldsymbol{\delta}}_A\|_2 \leq (3/2)\lambda_{\text{lasso}}s^{1/2}\|\widehat{\boldsymbol{\delta}}\|_2,$$

and similarly by condition (C4) and (2.18), we get

$$2\underline{c}\varrho\|\widehat{\boldsymbol{\delta}}\|_\infty \leq 2\underline{c}\|\mathbf{X}\widehat{\boldsymbol{\delta}}\|_2^2/(n\|\widehat{\boldsymbol{\delta}}_A\|_1) \leq (3/2)\lambda_{\text{lasso}}.$$

Thus, we have

$$\begin{aligned}&\mathrm{P}(\|\widehat{\boldsymbol{\delta}}\|_2 \leq 3s^{1/2}\lambda_{\text{lasso}}(4\kappa\underline{c})^{-1} \cap \|\widehat{\boldsymbol{\delta}}\|_\infty \leq 3\lambda_{\text{lasso}}(4\varrho\underline{c})^{-1})\\ &\geq \mathrm{P}(z_\infty^* \leq 2^{-1}\lambda_{\text{lasso}}) \geq 1 - \mathrm{P}(\|n^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\zeta}\|_\infty \geq 2^{-1}\lambda_{\text{lasso}}).\end{aligned}$$

Note that $\zeta_i = \Psi_\tau'(\varepsilon_i) = 2\tau\varepsilon_i^+ - 2(1-\tau)\varepsilon_i^-$. It follows from Lemma 2.3 and $\mathscr{E}^\tau(\varepsilon_i) = 0$ that ζ_i are i.i.d. mean zero sub-Gaussian random variables. Now by the union bound argument and Lemma 2.3 again

$$\mathrm{P}(\|n^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\zeta}\|_\infty \geq 2^{-1}\lambda_{\text{lasso}}) \leq 2p\exp\left(-\frac{Cn\lambda_{\text{lasso}}^2}{4K_0^2M_0^2}\right) = 1 - p_1^{\text{ALS}}.$$

This completes the proof. □

Lemma 2.5

Under the assumptions of Theorem 2.2, the probability that the LLA algorithm (Algorithm 2) initialized by $\widehat{\boldsymbol{\beta}}^{\text{lasso}}$ converges to $\widehat{\boldsymbol{\beta}}^{\text{oracle}}$ after two iterations is at least $1 - p_1 - p_2 - p_3$, where

$$\begin{aligned}p_1 &= \mathrm{P}(\|\widehat{\boldsymbol{\beta}}^{\text{lasso}} - \boldsymbol{\beta}^*\|_\infty > a_0\lambda),\\ p_2 &= \mathrm{P}(\|\nabla_{A^{\mathrm{c}}}\mathcal{L}_n(\widehat{\boldsymbol{\beta}}^{\text{oracle}})\|_\infty \geq a_1\lambda),\\ p_3 &= \mathrm{P}(\min_{j\in A}|\hat{\beta}_j^{\text{oracle}}| < a\lambda).\end{aligned}$$

□

Proof 2.6 (Proof of Lemma 2.5)

The convexity of $\mathcal{L}_n(\boldsymbol{\beta})$ follows from Lemma 2.1. Let $\mathcal{S} = \{\boldsymbol{\beta} \in \mathbb{R}^p : \boldsymbol{\beta}_{A^c} = \mathbf{0}\}$. Note that $\widehat{\boldsymbol{\beta}}^{\text{oracle}} \in \mathcal{S}$. For any $\boldsymbol{\beta} \in \mathcal{S}$, let $\bar{\mathcal{L}}_n(\boldsymbol{\beta}_A) \equiv n^{-1} \sum_{i=1}^n \Psi_\tau(y_i - \mathbf{x}_{iA}^{\mathrm{T}} \boldsymbol{\beta}_A) = \mathcal{L}_n(\boldsymbol{\beta})$. Then $\nabla \bar{\mathcal{L}}_n(\boldsymbol{\beta}_A) = -n^{-1} \sum_{i=1}^n \mathbf{x}_{iA} \Psi'_\tau(y_i - \mathbf{x}_{iA}^{\mathrm{T}} \boldsymbol{\beta}_A)$. Now for any $\boldsymbol{\beta}$ and $\boldsymbol{\beta}' \in \mathcal{S}$, by Lemma 2.1 again, we get

$$\bar{\mathcal{L}}_n(\boldsymbol{\beta}_A) \geq \bar{\mathcal{L}}_n(\boldsymbol{\beta}'_A) + \langle \nabla \bar{\mathcal{L}}_n(\boldsymbol{\beta}'_A), \boldsymbol{\beta}_A - \boldsymbol{\beta}'_A \rangle + \underline{c}(\boldsymbol{\beta}_A - \boldsymbol{\beta}'_A)^{\mathrm{T}} \frac{\mathbf{X}_A^{\mathrm{T}} \mathbf{X}_A}{n} (\boldsymbol{\beta}_A - \boldsymbol{\beta}'_A).$$

Since $\mathbf{X}_A$ is of full column rank by assumption, we can see that $\bar{\mathcal{L}}_n(\boldsymbol{\beta}_A)$ is strongly convex with respect to $\boldsymbol{\beta}_A$ and, therefore, $\widehat{\boldsymbol{\beta}}^{\text{oracle}}$ is the unique solution of problem (2.10) with $\nabla \bar{\mathcal{L}}_n(\widehat{\boldsymbol{\beta}}_A^{\text{oracle}}) = \mathbf{0}$. The lemma then follows from Theorems 1 and 2 in Fan et al. (2014b).□

Proof 2.7 (Proof of Theorem 2.2)

Let $\widehat{\boldsymbol{\delta}} = \widehat{\boldsymbol{\beta}}^{\text{lasso}} - \boldsymbol{\beta}^*$. Assume both (C3) and (C4) hold. The other cases where either (C3) or (C4) holds are similar. From Lemma 2.5 and Theorem 2.1, we immediately get

$$\begin{aligned} p_1 &\leq \mathrm{P}\big(\|\widehat{\boldsymbol{\delta}}\|_\infty > [3s^{1/2}\lambda_{\text{lasso}}(4\kappa\underline{c})^{-1}] \wedge [3\lambda_{\text{lasso}}(4\varrho\underline{c})^{-1}]\big) \\ &\leq \mathrm{P}(\|\widehat{\boldsymbol{\delta}}\|_2 > 3s^{1/2}\lambda_{\text{lasso}}(4\kappa\underline{c})^{-1}) \vee \mathrm{P}(\|\widehat{\boldsymbol{\delta}}\|_\infty > 3\lambda_{\text{lasso}}(4\varrho\underline{c})^{-1}) \leq p_1^{\text{ALS}}. \end{aligned}$$

To derive the bound for p_2, by the triangular inequality, it suffices to show bounds for $\mathrm{P}(\|\nabla_{A^c}\mathcal{L}_n(\boldsymbol{\beta}^*)\|_\infty \geq 2^{-1}a_1\lambda)$ and $\mathrm{P}(\|\nabla_{A^c}\mathcal{L}_n(\widehat{\boldsymbol{\beta}}^{\text{oracle}}) - \nabla_{A^c}\mathcal{L}_n(\boldsymbol{\beta}^*)\|_\infty \geq 2^{-1}a_1\lambda)$. By the union bound argument and Lemma 2.3,

$$\begin{aligned} \mathrm{P}(\|\nabla_{A^c}\mathcal{L}_n(\boldsymbol{\beta}^*)\|_\infty \geq 2^{-1}a_1\lambda) &= \mathrm{P}(\| - n^{-1}X_{A^c}^{\mathrm{T}}\boldsymbol{\zeta}\|_\infty \geq 2^{-1}a_1\lambda) \\ &\leq 2(p-s)\exp\left(-\frac{Ca_1^2 n\lambda^2}{4M_0^2K_0^2}\right). \end{aligned}$$

Let $\boldsymbol{d} = (d_i,\ i = 1,\ldots,n)^{\mathrm{T}}$ with $d_i = \Psi'_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{\text{oracle}}) - \Psi'_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*)$. By Cauchy–

Schwarz inequality and Lemma 2.2, we get

$$\begin{aligned}
&\|\nabla_{A^c}\mathcal{L}_n(\widehat{\boldsymbol{\beta}}^{\text{oracle}}) - \nabla_{A^c}\mathcal{L}_n(\boldsymbol{\beta}^*)\|_\infty \\
&= n^{-1}\max_{j\in A^c}|\textstyle\sum_{i=1}^n d_i x_{ij}| \le n^{-1}\max_{j\in A^c}(\|\boldsymbol{d}\|_2\|X_j\|_2) \\
&\le (2\bar{c}M_0)\big[(\widehat{\boldsymbol{\beta}}_A^{\text{oracle}} - \boldsymbol{\beta}_A^*)^{\mathrm{T}}(n^{-1}\mathbf{X}_A^{\mathrm{T}}\mathbf{X}_A)(\widehat{\boldsymbol{\beta}}_A^{\text{oracle}} - \boldsymbol{\beta}_A^*)\big]^{1/2} \\
&\le (2\bar{c}\rho_{\max}^{1/2}M_0)\big\|\widehat{\boldsymbol{\beta}}^{\text{oracle}} - \boldsymbol{\beta}^*\big\|_2.
\end{aligned}$$

It follows from Lemma 2.4 and Lemma 2.3 that

$$\begin{aligned}
&\mathrm{P}(\|\nabla_{A^c}\mathcal{L}_n(\widehat{\boldsymbol{\beta}}^{\text{oracle}}) - \nabla_{A^c}\mathcal{L}_n(\boldsymbol{\beta}^*)\|_\infty \ge 2^{-1}a_1\lambda) \\
&\le \mathrm{P}\bigg(\|\widehat{\boldsymbol{\beta}}^{\text{oracle}} - \boldsymbol{\beta}^*\|_2 \ge \frac{a_1\lambda}{4\bar{c}\rho_{\max}^{1/2}M_0}\bigg) \le \mathrm{P}(\|n^{-1}\mathbf{X}_A^{\mathrm{T}}\boldsymbol{\zeta}\|_2 \ge Q_1\lambda) \\
&\le \Gamma(Q_1\lambda; n, s, K_0, M_0, \rho_{\max}, \nu_0).
\end{aligned}$$

This establishes the desired upper bound for p_2. To show the upper bound for p_3, let $R = \min_{j\in A}|\beta_j^*| - a\lambda$ and observe that

$$\begin{aligned}
p_3 &= \mathrm{P}(\min_{j\in A}|\hat{\beta}_j^{\text{oracle}}| < a\lambda) \le \mathrm{P}(\|\widehat{\boldsymbol{\beta}}^{\text{oracle}} - \boldsymbol{\beta}^*\|_\infty > R) \\
&\le \mathrm{P}(\|\widehat{\boldsymbol{\beta}}^{\text{oracle}} - \boldsymbol{\beta}^*\|_2 > R) \le \mathrm{P}(\|n^{-1}\mathbf{X}_A^{\mathrm{T}}\boldsymbol{\zeta}\|_2 \ge 2\underline{c}\rho_{\min}R).
\end{aligned}$$

Similarly, by Lemma 2.3 we obtain

$$\mathrm{P}(\|n^{-1}\mathbf{X}_A^{\mathrm{T}}\boldsymbol{\zeta}\|_2 \ge 2\underline{c}\rho_{\min}R) \le \Gamma(2\underline{c}\rho_{\min}R; n, s, K_0, M_0, \rho_{\max}, \nu_0),$$

which completes the proof. □

Let us now prove the results for the COSALES estimation. To simplify notation, let $\boldsymbol{\varpi} = (\boldsymbol{\gamma}^{\mathrm{T}}, \boldsymbol{\varphi}^{\mathrm{T}})^{\mathrm{T}}$. It follows that $\mathrm{supp}(\boldsymbol{\varpi}^*) = A_0$. Let $\lambda_{\text{lasso}} = \lambda_1^{\text{lasso}} \wedge \lambda_2^{\text{lasso}}$ and $\Lambda_{\text{lasso}} = \lambda_1^{\text{lasso}} \vee \lambda_2^{\text{lasso}}$. We first present a lemma to facilitate the proofs.

Lemma 2.6

Let $\boldsymbol{\varepsilon} = (\varepsilon_i,\ 1 \le i \le n)^{\mathrm{T}}$ and $\boldsymbol{\eta} = (\eta_i, 1 \le i \le n)^{\mathrm{T}}$, where $\eta_i = \Psi'_\tau(\varepsilon_i - e_\tau)$. Also, let $\mathbf{W} = \mathrm{diag}\{\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\omega}^*, 1 \le i \le n\}$.

(1) For $\boldsymbol{\varpi}, \boldsymbol{\delta} \in \mathbb{R}^{2p}$, $\langle \nabla S_n(\boldsymbol{\varpi} + \boldsymbol{\delta}) - \nabla S_n(\boldsymbol{\varpi}), \boldsymbol{\delta}\rangle \ge n^{-1}c_0\|(\mathbf{I}_2 \otimes \mathbf{X})\boldsymbol{\delta}\|_2^2$, where $\mathbf{I}_2$ is a 2×2 identity matrix and $c_0 = 2^{-1}[(1 + 4\underline{c}) - (1 + 16\underline{c}^2)^{1/2}] > 0$.

(2) For $d > 0$, $\mathrm{P}(\|\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}} - \boldsymbol{\varpi}^*\|_2 > d) \le \mathrm{P}(\|\nabla_{A_0} S_n(\boldsymbol{\varpi}^*)\|_2 \ge c_0\phi_{\min} d)$, where

$$\nabla_{A_0} S_n(\boldsymbol{\varpi}^*) = -n^{-1}\begin{pmatrix} \mathbf{X}_{A_1}^{\mathrm{T}}\mathbf{W}(\boldsymbol{\varepsilon} + \boldsymbol{\eta}) \\ \mathbf{X}_{A_2}^{\mathrm{T}}\mathbf{W}\boldsymbol{\eta} \end{pmatrix}. \qquad \square$$

Proof 2.8 (Proof of Lemma 2.6)

The first part follows directly from the strong convexity of the (asymmetric) squared error loss. Specifically, note that since c_0 is the smaller eigenvalue of the 2×2 matrix $\begin{pmatrix} 1 + 2\underline{c} & 2\underline{c} \\ 2\underline{c} & 2\underline{c} \end{pmatrix}$, we have

$$\begin{aligned} S_n(\boldsymbol{\varpi} + \boldsymbol{\delta}) - S_n(\boldsymbol{\varpi}) - \langle \nabla S_n(\boldsymbol{\varpi}), \boldsymbol{\delta}\rangle &\ge \frac{1}{2n}\boldsymbol{\delta}^{\mathrm{T}}\left[\begin{pmatrix} 1 + 2\underline{c} & 2\underline{c} \\ 2\underline{c} & 2\underline{c} \end{pmatrix} \otimes (\mathbf{X}^{\mathrm{T}}\mathbf{X})\right]\boldsymbol{\delta} \\ &\ge (2n)^{-1}c_0\|(\mathbf{I}_2 \otimes \mathbf{X})\boldsymbol{\delta}\|_2^2. \end{aligned}$$

Similarly, $S_n(\boldsymbol{\varpi}) - S_n(\boldsymbol{\varpi} + \boldsymbol{\delta}) - \langle \nabla S_n(\boldsymbol{\varpi} + \boldsymbol{\delta}), -\boldsymbol{\delta}\rangle \ge (2n)^{-1}c_0\|(\mathbf{I}_2 \otimes \mathbf{X})\boldsymbol{\delta}\|_2^2$. Result (1) then follows by summing up the above two inequalities.

Let $\widehat{\boldsymbol{\delta}} = \widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}} - \boldsymbol{\varpi}^*$. Note that $\widehat{\boldsymbol{\delta}}_{A_0^{\mathrm{c}}} = \mathbf{0}$ and $\nabla_{A_0} S_n(\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}}) = \mathbf{0}$. From result (1) we

have

$$
\begin{aligned}
&c_0\phi_{\min}\|\widehat{\boldsymbol{\delta}}\|_2^2 = c_0\phi_{\min}\|\widehat{\boldsymbol{\delta}}_{A_0}\|_2^2 \le n^{-1}c_0\|(\mathbf{I}_2\otimes\mathbf{X})\widehat{\boldsymbol{\delta}}\|_2^2 \\
&\le \langle \nabla S_n(\widehat{\boldsymbol{\varpi}}^{\text{oracle}}) - \nabla S_n(\boldsymbol{\varpi}^*), \widehat{\boldsymbol{\delta}}\rangle = \langle -\nabla_{A_0} S_n(\boldsymbol{\varpi}^*), \widehat{\boldsymbol{\delta}}_{A_0}\rangle \\
&\le \|\nabla_{A_0} S_n(\boldsymbol{\varpi}^*)\|_2^2\|\widehat{\boldsymbol{\delta}}\|_2^2.
\end{aligned}
$$

Result (2) follows immediately. □

Proof 2.9 (Proof of Theorem 2.3)

Let $\widehat{\boldsymbol{\delta}}_1 = \widehat{\boldsymbol{\gamma}}^{\text{lasso}} - \boldsymbol{\gamma}^*$, $\widehat{\boldsymbol{\delta}}_2 = \widehat{\boldsymbol{\varphi}}^{\text{lasso}} - \boldsymbol{\varphi}^*$, $\widehat{\boldsymbol{\delta}} = (\widehat{\boldsymbol{\delta}}_1^{\mathrm{T}}, \widehat{\boldsymbol{\delta}}_2^{\mathrm{T}})^{\mathrm{T}}$, $z_{1\infty}^* = \|\partial S_n(\boldsymbol{\varpi}^*)/\partial\boldsymbol{\gamma}\|_\infty$, and $z_{2\infty}^* = \|\partial S_n(\boldsymbol{\varpi}^*)/\partial\boldsymbol{\varphi}\|_\infty$. By Lemma 2.6 and similar arguments in the proof of Theorem 2.1, it can shown that

$$
\begin{aligned}
0 &\le n^{-1}c_0\|(\mathbf{I}_2\otimes\mathbf{X})\widehat{\boldsymbol{\delta}}\|_2^2 \le \langle \nabla S_n(\widehat{\boldsymbol{\varpi}}^{\text{lasso}}) - \nabla S_n(\boldsymbol{\varpi}^*), \widehat{\boldsymbol{\delta}}\rangle \\
&\le (z_{1\infty}^* + \lambda_1^{\text{lasso}})\|\widehat{\boldsymbol{\delta}}_{1A_1}\|_1 + (z_{1\infty}^* - \lambda_1^{\text{lasso}})\|\widehat{\boldsymbol{\delta}}_{1A_1^{\mathrm{c}}}\|_1 \\
&\quad + (z_{2\infty}^* + \lambda_2^{\text{lasso}})\|\widehat{\boldsymbol{\delta}}_{2A_2}\|_1 + (z_{2\infty}^* - \lambda_2^{\text{lasso}})\|\widehat{\boldsymbol{\delta}}_{2A_2^{\mathrm{c}}}\|_1.
\end{aligned} \tag{2.19}
$$

Under events $\mathcal{E}_1 = \{z_{1\infty}^* \le 2^{-1}\lambda_1^{\text{lasso}}\}$ and $\mathcal{E}_2 = \{z_{2\infty}^* \le 2^{-1}\lambda_2^{\text{lasso}}\}$, it follows from (2.19) that

$$
\begin{aligned}
&2^{-1}\lambda_{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_{A_0^{\mathrm{c}}}\|_1 \le 2^{-1}\lambda_1^{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_{1A_1^{\mathrm{c}}}\|_1 + 2^{-1}\lambda_2^{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_{2A_2^{\mathrm{c}}}\|_1 \\
&\le (3/2)\lambda_1^{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_{1A_1}\|_1 + (3/2)\lambda_2^{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_{2A_2}\|_1 \le (3/2)\Lambda_{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_{A_0}\|_1,
\end{aligned}
$$

which implies that $\widehat{\boldsymbol{\delta}} \in \mathscr{C}_{3\check{M}}$. Now under conditions (C4′-C5′), we have from (2.19) that

$$
\begin{aligned}
c_0\bar{\kappa}\|\widehat{\boldsymbol{\delta}}\|_2^2 &\le n^{-1}c_0\|(\mathbf{I}_2\otimes\mathbf{X})\widehat{\boldsymbol{\delta}}\|_2^2 \le (3/2)\Lambda_{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_{A_0}\|_1 \\
&\le (3/2)\Lambda_{\text{lasso}}(s_1+s_2)^{1/2}\|\widehat{\boldsymbol{\delta}}\|_2
\end{aligned}
$$

and that

$$c_0\bar{\varrho}\|\widehat{\boldsymbol{\delta}}\|_\infty\|\widehat{\boldsymbol{\delta}}_{A_0}\|_1 \le n^{-1}c_0\|(\mathbf{I}_2\otimes\mathbf{X})\widehat{\boldsymbol{\delta}}\|_2^2 \le (3/2)\Lambda_{\text{lasso}}\|\widehat{\boldsymbol{\delta}}_{A_0}\|_1.$$

It follows that under events $\mathcal{E}_1$ and $\mathcal{E}_2$, we have $\|\widehat{\boldsymbol{\delta}}\|_2 \le 3(s_1+s_2)^{1/2}\Lambda_{\text{lasso}}(2\bar{\kappa}c_0)^{-1}$ and $\|\widehat{\boldsymbol{\delta}}\|_\infty \le 3\Lambda_{\text{lasso}}(2\bar{\varrho}c_0)^{-1}$. Recall that in Lemma 2.6 ε_i and $\eta_i = \Psi'_\tau(\varepsilon_i - e_\tau)$ are both mean zero sub-Gaussian random variables with $K_1 = \|\varepsilon_i\|_{\text{SG}}$ and $K_2 = \|\eta_i\|_{\text{SG}}$. It follows that $\varepsilon_i + \eta_i$ is also sub-Gaussian, and moreover, $\|\varepsilon_i+\eta_i\|_{\text{SG}} \le K_1 + K_2$. Since $M_1 = \|\mathbf{X}\boldsymbol{\omega}^*\|_\infty$, we have

$$\begin{aligned}
&\mathrm{P}(\|\widehat{\boldsymbol{\delta}}\|_2 \le 3(s_1+s_2)^{1/2}\Lambda_{\text{lasso}}(2\bar{\kappa}c_0)^{-1} \cap \|\widehat{\boldsymbol{\delta}}\|_\infty \le 3\Lambda_{\text{lasso}}(2\bar{\varrho}c_0)^{-1}) \\
&\ge \mathrm{P}(\mathcal{E}_1\cap\mathcal{E}_2) \ge 1 - \mathrm{P}(\mathcal{E}_1^{\text{c}}) - \mathrm{P}(\mathcal{E}_2^{\text{c}}) \\
&= 1 - \mathrm{P}(\|n^{-1}\mathbf{X}^{\text{T}}\mathbf{W}(\boldsymbol{\varepsilon}+\boldsymbol{\eta})\|_\infty > 2^{-1}\lambda_1^{\text{lasso}}) - \mathrm{P}(\|n^{-1}\mathbf{X}^{\text{T}}\mathbf{W}\boldsymbol{\eta}\|_\infty > 2^{-1}\lambda_2^{\text{lasso}}) \\
&\ge 1 - 2p\exp\left(-\frac{Cn(\lambda_1^{\text{lasso}})^2}{4M_0^2M_1^2(K_1+K_2)^2}\right) - 2p\exp\left(-\frac{Cn(\lambda_2^{\text{lasso}})^2}{4M_0^2M_1^2K_2^2}\right).
\end{aligned}$$

Theorem 2.3 then follows. □

The proof of Theorem 2.4 relies on the following lemma.

Lemma 2.7

Under assumptions of Theorem 2.4, the LLA algorithm (Algorithm 4) initialized by $\widehat{\boldsymbol{\gamma}}^{\text{lasso}}$ and $\widehat{\boldsymbol{\varphi}}^{\text{lasso}}$ converges to the oracle estimators $\widehat{\boldsymbol{\gamma}}^{\text{oracle}}$ and $\widehat{\boldsymbol{\varphi}}^{\text{oracle}}$ in two iterations with probability at least $1-\pi_1-\pi_2-\pi_3$, where

$$\begin{aligned}
\pi_1 &= \mathrm{P}(\|\widehat{\boldsymbol{\gamma}}^{\text{lasso}} - \boldsymbol{\gamma}^*\|_\infty > a_0\lambda_1, \|\widehat{\boldsymbol{\varphi}}^{\text{lasso}} - \boldsymbol{\varphi}^*\|_\infty > a_0\lambda_2), \\
\pi_2 &= \mathrm{P}(\|\partial S_n(\widehat{\boldsymbol{\varpi}}^{\text{oracle}})/\partial\boldsymbol{\gamma}_{A_1^{\text{c}}}\|_\infty \ge a_1\lambda_1, \|\partial S_n(\widehat{\boldsymbol{\varpi}}^{\text{oracle}})/\partial\boldsymbol{\varphi}_{A_2^{\text{c}}}\|_\infty \ge a_1\lambda_2), \\
\pi_3 &= \mathrm{P}(\min_{j\in A_1}|\hat{\gamma}_j^{\text{oracle}}| < a\lambda_1, \min_{j\in A_2}|\hat{\varphi}_j^{\text{oracle}}| < a\lambda_2).
\end{aligned}$$

□

Proof 2.10 (Proof of Lemma 2.7)

The convexity of $S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi})$ follows immediately from Lemma 2.1,

$$S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) \geq S_n(\boldsymbol{\gamma}', \boldsymbol{\varphi}') + \langle \nabla_{\boldsymbol{\gamma}} S_n(\boldsymbol{\gamma}', \boldsymbol{\varphi}'), \boldsymbol{\gamma} - \boldsymbol{\gamma}' \rangle + \langle \nabla_{\boldsymbol{\varphi}} S_n(\boldsymbol{\gamma}', \boldsymbol{\varphi}'), \boldsymbol{\varphi} - \boldsymbol{\varphi}' \rangle$$
$$+ 2^{-1} \begin{pmatrix} \boldsymbol{\gamma} - \boldsymbol{\gamma}' \\ \boldsymbol{\varphi} - \boldsymbol{\varphi}' \end{pmatrix}^{\mathrm{T}} \left[\begin{pmatrix} 1 + 2\underline{c} & 2\underline{c} \\ 2\underline{c} & 2\underline{c} \end{pmatrix} \otimes (n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}) \right] \begin{pmatrix} \boldsymbol{\gamma} - \boldsymbol{\gamma}' \\ \boldsymbol{\varphi} - \boldsymbol{\varphi}' \end{pmatrix}.$$

Restrict $S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi})$ to the set $\mathcal{S} = \{\boldsymbol{\gamma}, \boldsymbol{\varphi} \in \mathbb{R}^p : \boldsymbol{\gamma}_{A_1^{\mathrm{c}}} = \mathbf{0}, \boldsymbol{\varphi}_{A_2^{\mathrm{c}}} = \mathbf{0}\}$ and define for any $(\boldsymbol{\gamma}, \boldsymbol{\varphi}) \in \mathcal{S}$

$$\breve{S}_n(\boldsymbol{\gamma}_{A_1}, \boldsymbol{\varphi}_{A_2}) = n^{-1} \sum_{i=1}^{n} \{\Psi_{0.5}(y_i - \mathbf{x}_{iA_1}^{\mathrm{T}} \boldsymbol{\gamma}_{A_1}) + \Psi_{\tau}(y_i - \mathbf{x}_{iA_1}^{\mathrm{T}} \boldsymbol{\gamma}_{A_1} - \mathbf{x}_{iA_2}^{\mathrm{T}} \boldsymbol{\varphi}_{A_2})\}.$$

It follows immediately that for any $(\boldsymbol{\gamma}, \boldsymbol{\varphi}), (\boldsymbol{\gamma}', \boldsymbol{\varphi}') \in \mathcal{S}$,

$$\begin{aligned} \breve{S}_n(\boldsymbol{\gamma}_{A_1}, \boldsymbol{\varphi}_{A_2}) \geq \breve{S}_n(\boldsymbol{\gamma}'_{A_1}, \boldsymbol{\varphi}'_{A_2}) &+ \langle \nabla_{\boldsymbol{\gamma}_{A_1}} \breve{S}_n(\boldsymbol{\gamma}'_{A_1}, \boldsymbol{\varphi}'_{A_2}), \boldsymbol{\gamma}_{A_1} - \boldsymbol{\gamma}'_{A_1} \rangle \\ &+ \langle \nabla_{\boldsymbol{\varphi}_{A_2}} \breve{S}_n(\boldsymbol{\gamma}'_{A_1}, \boldsymbol{\varphi}'_{A_2}), \boldsymbol{\varphi}_{A_2} - \boldsymbol{\varphi}'_{A_2} \rangle \\ &+ 2^{-1} c_0 (\boldsymbol{\gamma}_{A_1} - \boldsymbol{\gamma}'_{A_1})^{\mathrm{T}} (n^{-1} \mathbf{X}_{A_1}^{\mathrm{T}} \mathbf{X}_{A_1}) (\boldsymbol{\gamma}_{A_1} - \boldsymbol{\gamma}'_{A_1}) \\ &+ 2^{-1} c_0 (\boldsymbol{\varphi}_{A_2} - \boldsymbol{\varphi}'_{A_2})^{\mathrm{T}} (n^{-1} \mathbf{X}_{A_2}^{\mathrm{T}} \mathbf{X}_{A_2}) (\boldsymbol{\varphi}_{A_2} - \boldsymbol{\varphi}'_{A_2}), \end{aligned}$$

where $c_0 = 2^{-1}[(1 + 4\underline{c}) - (1 + 16\underline{c}^2)^{1/2}]$. Since both $\mathbf{X}_{A_1}$ and $\mathbf{X}_{A_2}$ are of full column ranks by assumption, we can see that $\breve{S}_n(\boldsymbol{\gamma}_{A_1}, \boldsymbol{\varphi}_{A_2})$ is strongly convex and thus the oracle estimators $\widehat{\boldsymbol{\gamma}}^{\mathrm{oracle}}$ and $\widehat{\boldsymbol{\varphi}}^{\mathrm{oracle}}$ are the unique solution of problem (2.14).

Let $\mathcal{E}_1$ be the event that $\|\widehat{\boldsymbol{\gamma}}^{\mathrm{lasso}} - \boldsymbol{\gamma}^*\|_\infty \leq a_0\lambda_1$ and $\|\widehat{\boldsymbol{\varphi}}^{\mathrm{lasso}} - \boldsymbol{\varphi}^*\|_\infty \leq a_0\lambda_2$. Under $\mathcal{E}_1$ and Assumption (A0′), on one hand we have $\min_{j \in A_1} |\hat{\gamma}_j^{\mathrm{lasso}}| \geq \min_{j \in A_1} |\gamma_j^*| - \|\widehat{\boldsymbol{\gamma}}^{\mathrm{lasso}} - \boldsymbol{\gamma}^*\|_\infty > a\lambda_1$, implying that $p'_{\lambda_1}(|\hat{\gamma}_j^{\mathrm{lasso}}|) = 0$ for $j \in A_1$. On the other hand, we have $\|\widehat{\boldsymbol{\gamma}}^{\mathrm{lasso}}_{A^{\mathrm{c}}}\|_\infty \leq \|\widehat{\boldsymbol{\gamma}}^{\mathrm{lasso}} - \boldsymbol{\gamma}^*\|_\infty \leq a_2\lambda_1$, indicating that $p'_{\lambda_1}(|\hat{\gamma}_j^{\mathrm{lasso}}|) \geq a_1\lambda_1$ for $j \in A_1^{\mathrm{c}}$. Similarly, we can show that $p'_{\lambda_2}(|\hat{\varphi}_j^{\mathrm{lasso}}|) = 0$ for $j \in A_2$ and $p'_{\lambda_2}(|\hat{\varphi}_j^{\mathrm{lasso}}|) \geq a_1\lambda_2$ for $j \in A_2^{\mathrm{c}}$.

Let $\widehat{\boldsymbol{\gamma}}^1$ and $\widehat{\boldsymbol{\varphi}}^1$ be the update after the first iteration of the LLA algorithm. Then under $\mathcal{E}1$, $\widehat{\boldsymbol{\gamma}}^1$ and $\widehat{\boldsymbol{\varphi}}^1$ are minimizers of

$$\mathcal{Q}_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) = S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) + \sum_{j \in A_1^c} p'_{\lambda_1}(|\hat{\gamma}_j^{\text{lasso}}|)|\gamma_j| + \sum_{j \in A_2^c} p'_{\lambda_2}(|\hat{\varphi}_j^{\text{lasso}}|)|\varphi_j|.$$

By definition of the oracle estimators, $\partial S_n(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}})/\partial\gamma_j = 0$ for $j \in A_1$ and $\partial S_n(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}})/\partial\varphi_j = 0$ for $j \in A_2^c$. Also, $\widehat{\boldsymbol{\gamma}}^{\text{oracle}}_{A_1^c} = \mathbf{0}$ and $\widehat{\boldsymbol{\varphi}}^{\text{oracle}}_{A_2^c} = \mathbf{0}$. Now let $\mathcal{E}_2$ be the event $\max_{j \in A_1^c} |\partial\mathcal{L}(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}})/\partial\gamma_j| < a_1\lambda_1$ and $\max_{j \in A_2^c} |\partial\mathcal{L}(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}})/\partial\varphi_j| < a_1\lambda_2$. It follows from the convexity of $S_n(\boldsymbol{\gamma}, \boldsymbol{\varphi})$ that

$$\begin{aligned}
&\mathcal{Q}_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) - \mathcal{Q}_n(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}}) \\
\geq &\sum_{j \in A_1^c} \frac{\partial}{\partial\gamma_j} S_n(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}})\gamma_j + \sum_{j \in A_2^c} \frac{\partial}{\partial\varphi_j} S_n(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}})\varphi_j \\
&+ \sum_{j \in A_1^c} p'_{\lambda_1}(|\hat{\gamma}_j^{\text{lasso}}|)|\gamma_j| + \sum_{j \in A_2^c} p'_{\lambda_2}(|\hat{\varphi}_j^{\text{lasso}}|)|\varphi_j|.
\end{aligned}$$

Under $\mathcal{E}_2$, this implies that $Q_n(\boldsymbol{\gamma}, \boldsymbol{\varphi}) \geq Q_n(\widehat{\boldsymbol{\gamma}}^{\text{oracle}}, \widehat{\boldsymbol{\varphi}}^{\text{oracle}})$ for any $\boldsymbol{\gamma} \in \mathbb{R}^p$ and $\boldsymbol{\varphi} \in \mathbb{R}^p$. The strict inequality holds unless $\gamma_j = 0$ for all $j \in A_1^c$ and $\varphi_j = 0$ for all $j \in A_2^c$. By the uniqueness of the oracle estimators, we must have $\widehat{\boldsymbol{\gamma}}^1 = \widehat{\boldsymbol{\gamma}}^{\text{oracle}}$ and $\widehat{\boldsymbol{\varphi}}^1 = \widehat{\boldsymbol{\varphi}}^{\text{oracle}}$.

Let $\mathcal{E}_3$ be the event that $\min_{j \in A_1} |\hat{\gamma}_j^{\text{oracle}}| \geq a\lambda_1$ and $\min_{j \in A_2} |\hat{\varphi}_j^{\text{oracle}}| \geq a\lambda_2$. Once the oracle estimators are obtained after the first iteration, under $\mathcal{E}_3$, we can see that $p'_{\lambda_1}(|\hat{\gamma}_j^{\text{oracle}}|) = 0$ for $j \in A_1$, $p'_{\lambda_1}(|\hat{\gamma}_j^{\text{oracle}}|) \geq a_1\lambda_1$ for $j \in A_1^c$ and $p'_{\lambda_2}(|\hat{\varphi}_j^{\text{oracle}}|) = 0$ for $j \in A_2$, $p'_{\lambda_2}(|\hat{\varphi}_j^{\text{oracle}}|) \geq a_1\lambda_2$ for $j \in A_2^c$. By similar arguments, it can be shown that the second iteration of the LLA algorithm will still yield the oracle estimators, which means the algorithm converges to the oracle estimators hereafter. This completes the proof. □

Proof 2.11 (Proof of Theorem 2.4)

Let $\widehat{\boldsymbol{\delta}} = \widehat{\boldsymbol{\varpi}}^{\text{lasso}} - \boldsymbol{\varpi}^*$. Assume both (C4′) and (C5′) hold. The other cases where either (C4′)

or (C5′) holds are similar. It follows from Theorem 2.3 that

$$\begin{aligned}
&\pi_1 \le \mathrm{P}(\|\widehat{\boldsymbol{\delta}}\|_\infty > a_0\lambda) \le \mathrm{P}(\|\widehat{\boldsymbol{\delta}}\|_\infty > 3\Lambda_{\mathrm{lasso}}(2c_0)^{-1}[(s^{1/2}\bar{\kappa}^{-1}) \wedge \bar{\varrho}^{-1}]) \\
&\le \mathrm{P}(\|\widehat{\boldsymbol{\delta}}\|_2 > 3s^{1/2}\Lambda_{\mathrm{lasso}}(2c_0\bar{\kappa})^{-1}) \vee \mathrm{P}(\|\widehat{\boldsymbol{\delta}}\|_\infty > 3\Lambda_{\mathrm{lasso}}(2c_0\bar{\varrho})^{-1}) \le \pi_1^{\mathrm{ALS}}.
\end{aligned}$$

Next, note that $\pi_2 \le \mathrm{P}(\|\nabla_{A_0^{\mathrm{c}}} S_n(\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}})\|_\infty \ge a_1\lambda)$. By the triangular inequality, it suffices to show the upper bounds for respectively $\mathrm{P}(\|\nabla_{A_0^{\mathrm{c}}} S_n(\boldsymbol{\varpi}^*)\|_\infty \ge 2^{-1}a_1\lambda)$ and $\mathrm{P}(\|\nabla_{A_0^{\mathrm{c}}} S_n(\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}}) - \nabla_{A_0^{\mathrm{c}}} S_n(\boldsymbol{\varpi}^*)\|_\infty \ge 2^{-1}a_1\lambda)$. First, by the union bound argument we have

$$\begin{aligned}
&\mathrm{P}(\|\nabla_{A_0^{\mathrm{c}}} S_n(\boldsymbol{\varpi}^*)\|_\infty \ge 2^{-1}a_1\lambda) \\
&\le \mathrm{P}(\|n^{-1}\mathbf{X}_{A_1^{\mathrm{c}}}^{\mathrm{T}}\mathbf{W}(\boldsymbol{\varepsilon}+\boldsymbol{\eta})\|_\infty \ge 2^{-1}a_1\lambda) + \mathrm{P}(\|n^{-1}\mathbf{X}_{A_2^{\mathrm{c}}}^{\mathrm{T}}\mathbf{W}\boldsymbol{\eta}\|_\infty \ge 2^{-1}a_1\lambda) \\
&\le 2(p-s_1)\exp\left(-\frac{Ca_1^2 n\lambda^2}{4M_0^2M_1^2(K_1+K_2)^2}\right) + 2(p-s_2)\exp\left(-\frac{Ca_1^2 n\lambda^2}{4M_0^2M_1^2K_2^2}\right).
\end{aligned}$$

Now let $\bar{d}_i = \Psi_\tau'(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\gamma}}^{\mathrm{oracle}} - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\varphi}}^{\mathrm{oracle}}) - \Psi_\tau'(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\gamma}^* - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\varphi}^*)$ and $\bar{\boldsymbol{d}} = (\bar{d}_i, 1 \le i \le n)^{\mathrm{T}}$. It follows that

$$\begin{aligned}
&\|\nabla_{A_0^{\mathrm{c}}} S_n(\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}}) - \nabla_{A_0^{\mathrm{c}}} S_n(\boldsymbol{\varpi}^*)\|_\infty \le M_0(\|\mathbf{X}(\widehat{\boldsymbol{\gamma}}^{\mathrm{oracle}} - \boldsymbol{\gamma}^*)\|_2 + \|\bar{\boldsymbol{d}}\|_2)/\sqrt{n} \\
&\le M_0[(1+2\bar{c})\|\mathbf{X}_{A_1}(\widehat{\boldsymbol{\gamma}}_{A_1}^{\mathrm{oracle}} - \boldsymbol{\gamma}_{A_1}^*)\|_2 + (2\bar{c})\|\mathbf{X}_{A_2}(\widehat{\boldsymbol{\varphi}}_{A_2}^{\mathrm{oracle}} - \boldsymbol{\varphi}_{A_2}^*)\|_2]/\sqrt{n} \\
&\le (1+2\bar{c})M_0\phi_{\max}^{1/2}\|\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}} - \boldsymbol{\varpi}^*\|_2.
\end{aligned}$$

By Lemma 2.6 and Lemma 2.3, we get

$$
\begin{aligned}
&\mathrm{P}(\|\nabla_{A_0^{\mathrm{c}}} S_n(\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}}) - \nabla_{A_0^{\mathrm{c}}} S_n(\boldsymbol{\varpi}^*)\|_\infty \geq 2^{-1} a_1 \lambda) \\
&\leq \mathrm{P}\bigg(\|\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}} - \boldsymbol{\varpi}^*\|_2 \geq \frac{a_1 \lambda}{2(1+2\underline{c}) M_0 \phi_{\max}^{1/2}} \bigg) \\
&\leq \mathrm{P}\bigg(\bigg\| \frac{1}{n} \begin{pmatrix} \mathbf{X}_{A_1}^{\mathrm{T}} \mathbf{W}(\boldsymbol{\varepsilon} + \boldsymbol{\eta}) \\ \mathbf{X}_{A_2}^{\mathrm{T}} \mathbf{W} \boldsymbol{\eta} \end{pmatrix} \bigg\|_2 \geq Q_2 \lambda \bigg) \\
&\leq \mathrm{P}(\|n^{-1} \mathbf{X}_{A_1}^{\mathrm{T}} \mathbf{W}(\boldsymbol{\varepsilon} + \boldsymbol{\eta})\|_2 \geq 2^{-1} Q_2 \lambda) + \mathrm{P}(\|n^{-1} \mathbf{X}_{A_2}^{\mathrm{T}} \mathbf{W} \boldsymbol{\eta}\|_2 \geq 2^{-1} Q_2 \lambda) \\
&\leq \Gamma(2^{-1} Q_2 \lambda; n, s_1, K_1 + K_2, M_0 M_1, M_1^2 \rho_{1\bullet\max}, \nu_1) \\
&\quad + \Gamma(2^{-1} Q_2 \lambda; n, s_2, K_2, M_0 M_1, M_1^2 \rho_{2\bullet\max}, \nu_2).
\end{aligned}
$$

This completes the upper bound for π_2. To derive the upper bound for π_3, note that by Assumption (A0′) we have $\min_{j \in A_1} |\gamma_j^*| \geq (a+1)\lambda_1$ and $\min_{j \in A_2} |\varphi_j^*| \geq (a+1)\lambda_2$. Observe that $\min_{j \in A_1} |\hat{\gamma}_j^{\mathrm{oracle}}| \geq \min_{j \in A_1} |\gamma_j^*| - \|\widehat{\boldsymbol{\gamma}}^{\mathrm{oracle}} - \boldsymbol{\gamma}^*\|_\infty$ and $\min_{j \in A_2} |\hat{\varphi}_j^{\mathrm{oracle}}| \geq \min_{j \in A_1} |\varphi_j^*| - \|\widehat{\boldsymbol{\varphi}}^{\mathrm{oracle}} - \boldsymbol{\varphi}^*\|_\infty$, and it follows that

$$
\begin{aligned}
\pi_3 &\leq \mathrm{P}\bigg(\|\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}} - \boldsymbol{\varpi}^*\|_\infty > \bar{R} \bigg) \leq \mathrm{P}\bigg(\|\widehat{\boldsymbol{\varpi}}^{\mathrm{oracle}} - \boldsymbol{\varpi}^*\|_2 > \bar{R} \bigg) \\
&\leq \mathrm{P}\bigg(\bigg\| \frac{1}{n} \begin{pmatrix} \mathbf{X}_{A_1}^{\mathrm{T}} \mathbf{W}(\boldsymbol{\varepsilon} + \boldsymbol{\eta}) \\ \mathbf{X}_{A_2}^{\mathrm{T}} \mathbf{W} \boldsymbol{\eta} \end{pmatrix} \bigg\|_2 \geq c_0 \phi_{\min} \bar{R} \bigg) \\
&\leq \mathrm{P}(\|n^{-1} \mathbf{X}_{A_1}^{\mathrm{T}} \mathbf{W}(\boldsymbol{\varepsilon} + \boldsymbol{\eta})\|_2 \geq \frac{1}{2} c_0 \phi_{\min} \bar{R}) + \mathrm{P}(\|n^{-1} \mathbf{X}_{A_2}^{\mathrm{T}} \mathbf{W} \boldsymbol{\eta}\|_2 \geq \frac{1}{2} c_0 \phi_{\min} \bar{R}) \\
&\leq \Gamma(2^{-1} c_0 \phi_{\min} \bar{R}; n, s_1, K_1 + K_2, M_0 M_1, M_1^2 \rho_{1\bullet\max}, \nu_1) \\
&\quad + \Gamma(2^{-1} c_0 \phi_{\min} \bar{R}; n, s_2, K_2, M_0 M_1, M_1^2 \rho_{2\bullet\max}, \nu_2).
\end{aligned}
$$

□

Chapter 3

ADMM for High-Dimensional Sparse Penalized Quantile Regression

Sparse penalized quantile regression is a useful tool for variable selection, robust estimation and heteroscedasticity detection in high-dimensional data analysis. The computational issue of the sparse penalized quantile regression has not yet been fully resolved in the literature, due to nonsmoothness of the quantile regression loss function. We introduce fast alternating direction method of multipliers algorithms for computing sparse penalized quantile regression. The convergence properties of the proposed algorithms are established. Numerical examples demonstrate the competitive performance of our algorithm: it significantly outperforms several other fast solvers for high-dimensional penalized quantile regression.

3.1 Introduction

High-dimensional data are frequently collected in a wide variety of research areas such as genomics, functional magnetic resonance imaging, tomography, economics, and finance. Analysis of high-dimensional data poses many challenges and has attracted tremendous recent interests in a number of fields such as econometrics, applied mathematics, electronic engineering, and statistics. Sparse penalized least squares regression has become a widely used method for analyzing high-dimensional data. The least squares regression can be

regularized with various penalties, such as the bridge penalty (Frank and Friedman, 1993), lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), elastic net (Zou and Hastie, 2005), adaptive lasso (Zou, 2006), and so on. Many researchers have also considered regression methods other than the least squares for high-dimensional data analysis. For example, quantile regression introduced by Koenker and Bassett (1978) has gained a lot of attention in the high-dimensional statistics literature, owing to its robustness property and its ability to offer unique insights into the relation between the response variable and the covariates that is not available in doing least squares regression which only estimates the conditional mean function. The classical least absolute deviation (LAD) regression can be viewed as a special case of the quantile regression. A comprehensive treatment of the quantile regression can be found in Koenker (2005). Recently, many studies on quantile regression have been focusing on high-dimensional scenarios where the number of parameters exceeds the number of observations; see, for example, Wu and Liu (2009), Belloni and Chernozhukov (2011), Wang et al. (2012), Wang (2013), Fan et al. (2014a), and Fan et al. (2014b). Belloni and Chernozhukov (2011) studied the L_1-penalized quantile regression in high-dimensional sparse models where the dimensionality could be larger than the sample size. They showed that the lasso penalized quantile regression estimator is consistent at near-oracle rate, and gave conditions under which the selected model includes the true model. Wang (2013) studied the L_1-penalized LAD regression and showed that its estimator achieves near-oracle risk performance with a universal penalty parameter. Fan et al. (2014a) studied the penalized quantile regression with the weighted L_1-penalty. Fan et al. (2014b) provided a general framework for solving folded concave penalized regression, including the quantile regression as a special case, via a two-step local linear approximation (LLA) approach. They showed that with high probability, the oracle estimator can be directly obtained within two iterations of the LLA algorithm. This property is often referred to as the strong oracle property (Fan and Lv, 2011).

Compared to the least squares method, fitting quantile regression requires more so-

phisticated computational algorithm. Numerical computation is particularly important in high-dimensional scenarios. Several algorithms have been developed in the literature to deal with regularized quantile regression. A standard method for solving the quantile lasso is to transform the corresponding optimization problem into a linear program, which can then be solved by many existing optimization software packages. Koenker and Ng (2005) proposed an interior-point method for quantile regression and penalized quantile regression. Li and Zhu (2008) proposed an algorithm for computing the solution path of the lasso penalized quantile regression following the LARS/lasso (Efron et al., 2004) algorithm. Wu and Lange (2008) proposed a greedy coordinate descent algorithm for lasso penalized LAD regression. A similar coordinate descent algorithm for the penalized quantile regression was studied in Peng and Wang (2015). Yi and Huang (2016) proposed a coordinate descent algorithm for solving the elastic-net penalized Huber regression and used that to approximate the penalized quantile regression. Hunter and Lange (2000) presented a majorization-minimization (MM) algorithm which successively finds quadratic majorizing functions for a perturbed version of the quantile loss function. For the kernel quantile regression under smoothness-sparsity constraint, Lv et al. (2016) developed their algorithm by combining the MM technique in Hunter and Lange (2000) and the proximal gradient method (Parikh and Boyd, 2013). Yang et al. (2013) considered a randomized algorithm for solving large scale quantile regression with small to moderate dimensions.

The alternating direction method of multipliers (ADMM) algorithm has found many successful applications in high-dimensional statistics and machine learning, such as comprehensive sensing (Yin et al., 2008; Goldstein and Osher, 2009), optimal control (O'Donoghue et al., 2013), and statistics (Xue et al., 2012; Bien et al., 2013; Bogdan et al., 2013; Zhang et al., 2014), to name a few. Boyd et al. (2011) argued that ADMM is well suited for distributed convex optimization and for large-scale problems arising in statistics, machine learning, and related areas. As an important variant of ADMM, the proximal ADMM has also attracted many research efforts in the fields of optimization; see, for example, Eckstein

(1994), He et al. (2002), and Fazel et al. (2013).

In this chapter, we propose a proximal ADMM (pADMM) algorithm and a sparse coordinate descent ADMM (scdADMM) algorithm to solve the penalized quantile regression with the lasso, adaptive lasso and folded concave penalties. Global convergence results are established for the proposed methods. In numerical experiments, we demonstrate that our algorithms can efficiently solve the sparse penalized quantile regression and the solutions produced by the algorithms are of high statistical accuracy. The chapter is organized as follows. In Section 3.2, we introduce the sparse penalized quantile regression and set up a uniform framework to include various regularization types, such as the lasso, adaptive lasso and folded concave penalties. We present the ADMM algorithms for solving the sparse penalized quantile regression in Section 3.3. The numerical and statistical efficiency of the proposed algorithms is demonstrated by simulation studies and real data analysis in Section 3.4. Technical proofs can be found in the supplementary file.

3.2 Sparse Penalized Quantile Regression

Quantile regression is a popular method for studying the influence of a set of covariates on the conditional distribution of a response variable. Besides the well-known property of being robust to outliers, quantile regression has also been widely applied to handling heteroscedasticity (Koenker and Bassett, 1982; Wang et al., 2012). Given univariate response $Y \in \mathbb{R}$ and a vector of covariates $\mathbf{X} \in \mathbb{R}^p$, let $F_Y(y|\mathbf{x}) = \Pr(Y \leq y|\mathbf{X} = \mathbf{x})$ be the conditional cumulative distribution function and $Q_Y(\tau|\mathbf{x}) = \inf\{y: F_Y(y|\mathbf{x}) \geq \tau\}$ be the τth conditional quantile for $\tau \in (0, 1)$. The linear quantile regression model assumes $Q_Y(\tau|\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}(\tau)$ for some unknown coefficient vector $\boldsymbol{\beta}(\tau)$. Given observations $(\mathbf{x}_i, y_i)_{i=1}^n$, the quantile regression estimator of $\boldsymbol{\beta}(\tau)$ is obtained through minimization of the empirical loss function $\sum_{i=1}^n \rho_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})$ over $\boldsymbol{\beta} \in \mathbb{R}^p$, where $\rho_\tau(u) = u\{\tau - I(u < 0)\}$ is the check loss. Asymptotic properties for the regression quantiles under fixed dimension

have been well studied (Koenker and Bassett, 1978; Chen et al., 1990; Pollard, 1991). When the dimension is allowed to increase, but with $p = o(n)$, the asymptotic behaviors of the regression quantiles can be investigated directly using results from Welsh (1989), Bai and Wu (1994) and He and Shao (2000). With even higher dimensions, especially when $p > n$, the sparse penalized quantile regression has been proposed to encourage sparsity in the coefficient estimates where we consider minimizing

$$\frac{1}{n}\sum_{i=1}^{n}\rho_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \sum_{j=1}^{n} p_\lambda(|\beta_j|)$$

over $\boldsymbol{\beta} \in \mathbb{R}^p$, where $p_\lambda(\cdot)$, $\lambda > 0$ is the penalty function introduced to control the model complexity. A popular choice of $p_\lambda(\cdot)$ is the lasso penalty. Under some sparsity assumption of $\boldsymbol{\beta}(\tau)$, the lasso penalized regression estimator is shown to be consistent at near-oracle rate $\mathcal{O}(\sqrt{s \log p/n})$ by Belloni and Chernozhukov (2011), where $s = \|\boldsymbol{\beta}(\tau)\|_0 = \sum_{j=1}^{p} I(\beta_j(\tau) \neq 0)$. To alleviate the bias phenomenon of the lasso, adaptive lasso and folded concave penalties have been used in, for example, Wang et al. (2012), Fan et al. (2014a) and Fan et al. (2014b).

Sparse penalized quantile regression is computationally challenging due to the nonsmooth nature of the check loss. An added layer of complexity comes from the nonsmoothness of the penalty functions, let alone the issues arising from nonconvex optimization when folded concave penalties are used. In this chapter, we propose fast alternating direction method of multipliers algorithms for computing penalized quantile regression with various penalties. To facilitate the discussion, let us consider the following weighted L_1-penalized quantile regression

$$\min_{\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n}\rho_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \lambda\|\mathbf{w} \circ \boldsymbol{\beta}\|_1, \tag{3.1}$$

where $\lambda > 0$ is the regularization parameter, $\mathbf{w} = (w_1, \ldots, w_p)^{\mathrm{T}}$ is the vector of nonnegative

weights, $w_j \geq 0$, $j = 1, \ldots, p$, and $\|\mathbf{w} \circ \boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |w_j \beta_j| = \sum_{j=1}^{p} w_j |\beta_j|$ with $\circ$ denoting the Hadamard product. We note that in formulation (3.1), if $x_{i1} = 1$ and β_1 represents the intercept term, one can set $w_1 = 0$ to respect the practice of leaving the intercept term unpenalized.

To see why formulation (3.1) is general, note that for the lasso penalized quantile regression, one can choose $\mathbf{w} = \mathbf{1}_p$, a vector of all ones. While for the adaptive lasso penalized quantile regression, the typical choice, $w_j = (|\hat{\beta}_j^{\text{lasso}}| + 1/n)^{-1}$, $j = 1 \ldots, p$, is often employed, where $\widehat{\boldsymbol{\beta}}^{\text{lasso}} = (\hat{\beta}_j^{\text{lasso}},\ j = 1, \ldots, p)^{\mathrm{T}}$ denotes the quantile lasso estimator.

Once the problem in (3.1) is efficiently solved, the nonconvex penalized quantile regression can then be solved by combining the local linear approximation (LLA, Zou and Li, 2008) algorithm and the efficient algorithm for solving (3.1). Specifically, let p_λ be a folded concave penalty (Fan and Lv, 2011; Fan et al., 2014b). The LLA algorithm solves the folded concave penalized quantile regression,

$$\min_{\boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(|\beta_j|),$$

via the following iterations:

(a) Initialize $\boldsymbol{\beta}$ with $\widehat{\boldsymbol{\beta}}^0$.

(b) For $k = 1, 2, \ldots, M$,

(b.1) Compute the weights $w_j = \hat{w}_j^{k-1} = \lambda^{-1} p'_\lambda(|\hat{\beta}_j^{k-1}|)$, $j = 1, \ldots, p$.

(b.2) Solve problem (3.1) using the weights from step (b.1) to obtain the update $\widehat{\boldsymbol{\beta}}^k$.

It can be seen that the folded concave penalized quantile regression is solved by a sequence of weighted L_1-penalized quantile regression. In fact, Fan et al. (2014b) showed that theoretically two or three iterations are good enough to yield a solution with high statistical

accuracy. As an example, the SCAD penalty has derivative

$$p'_\lambda(u) = \lambda I(|u| \le \lambda) + \frac{\max(a\lambda - |u|, 0)}{a-1} I(|u| > \lambda)$$

for some $a > 2$. A typical choice is $a = 3.7$ as suggested by Fan and Li (2001). It was shown in Fan et al. (2014b) that one only needs to take $\widehat{\boldsymbol{\beta}}^0 = \widehat{\boldsymbol{\beta}}^{\text{lasso}}$ and run the LLA algorithm for two iterations to obtain the quantile SCAD estimator.

3.3 Alternating Direction Algorithm

3.3.1 Review of two existing algorithms

A typical approach to solving the weighted L_1-penalized quantile regression is to cast it as a linear program and then solve the linear program using the interior point method. The popular R package `quantreg` (Koenker, 2016) is based on an interior-point method specifically designed for solving the (penalized) quantile regression (Koenker and Ng, 2005). Note that the weighted L_1-penalized quantile regression (3.1) is equivalent to the linear program

$$\begin{aligned} \text{minimize} \quad & \tau \mathbf{1}_n^{\mathrm{T}} \mathbf{u} + (1-\tau)\mathbf{1}_n^{\mathrm{T}} \mathbf{v} + (n\lambda)\mathbf{w}^{\mathrm{T}} \boldsymbol{\beta}^+ + (n\lambda)\mathbf{w}^{\mathrm{T}} \boldsymbol{\beta}^- \\ \text{subject to} \quad & \mathbf{u} - \mathbf{v} + \mathbf{X}\boldsymbol{\beta}^+ - \mathbf{X}\boldsymbol{\beta}^- = \mathbf{y} \\ & \mathbf{u}, \mathbf{v} \in \mathbb{R}_+^n, \ \boldsymbol{\beta}^+, \boldsymbol{\beta}^- \in \mathbb{R}_+^p, \end{aligned} \tag{3.2}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^{\mathrm{T}}$. Problem (3.2) is often solved with the interior point method (Koenker and Ng, 2005) in its dual domain

$$\begin{aligned} \text{minimize} \quad & (-\mathbf{y}^{\mathrm{T}}, \ \mathbf{0}_p^{\mathrm{T}})\boldsymbol{d} \\ \text{subject to} \quad & [\mathbf{X}^{\mathrm{T}} \ \ (2n\lambda)\mathrm{diag}(\mathbf{w})]\boldsymbol{d} = (1-\tau)\mathbf{X}^{\mathrm{T}}\mathbf{1}_n + (n\lambda)\mathbf{w} \\ & 0 \le d_k \le 1, \ k = 1, \ldots, n+p, \end{aligned} \tag{3.3}$$

where $\mathrm{diag}(\mathbf{w})$ denotes the diagonal matrix with the components of $\mathbf{w}$ on its diagonals. Note that the dual problem (3.3) involves p equality constraints. We notice that the interior point algorithm is the state-of-the-art method for fitting quantile regression for low or moderate dimension, but it fails to scale well with high dimensions. For numerical evidence, see Section 3.4. This observation motivates us to consider an efficient alternative for fitting the high-dimensional quantile regression.

During the revision, one reviewer pointed out the algorithm by Yi and Huang (2016). Specifically, Yi and Huang (2016) proposed a coordinate descent algorithm to solve the penalized Huber regression and used its solutions to approximate those of the penalized quantile regression. Their algorithm is implemented in the R package `hqreg` (Yi, 2016). We include this algorithm in our numerical comparisons. It is worth mentioning that both interior-point algorithm and our algorithm solve the exact quantile regression problem in theory, while `hgreg` offers an approximate solution.

3.3.2 Two ADMM algorithms

We now introduce two ADMM algorithms for solving the weighted L_1-penalized quantile regression. These new algorithms can be combined with the LLA algorithm to solve the SCAD penalized quantile regression.

For ease of notation, we denote $\mathbb{Q}_\tau(\mathbf{z}) = (1/n)\sum_{i=1}^n \rho_\tau(z_i)$ for $\mathbf{z} = (z_1, \ldots, z_n)^{\mathrm{T}}$. In order to handle the nonsmoothness of the check loss, we introduce new variables $\mathbf{z} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$.

By convexity, problem (3.1) is equivalent to

$$\begin{aligned} &\min_{\boldsymbol{\beta},\mathbf{z}} && \mathbb{Q}_\tau(\mathbf{z}) + \lambda\|\mathbf{w}\circ\boldsymbol{\beta}\|_1 \\ &\text{subject to} && \mathbf{X}\boldsymbol{\beta} + \mathbf{z} = \mathbf{y}. \end{aligned} \tag{3.4}$$

Fix $\sigma > 0$ and the augmented Lagrangian function of (3.4) is

$$\mathcal{L}_\sigma(\boldsymbol{\beta},\mathbf{z},\boldsymbol{\theta}) := \mathbb{Q}_\tau(\mathbf{z}) + \lambda\|\mathbf{w}\circ\boldsymbol{\beta}\|_1 - \langle\boldsymbol{\theta},\mathbf{X}\boldsymbol{\beta}+\mathbf{z}-\mathbf{y}\rangle + \frac{\sigma}{2}\|\mathbf{X}\boldsymbol{\beta}+\mathbf{z}-\mathbf{y}\|_2^2,$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ is the Lagrangian multiplier, and $\langle\cdot,\cdot\rangle$ and $\|\cdot\|_2$ denote the inner product and L_2-norm in the Euclidean space, respectively. Following Boyd et al. (2011), the iterations for the standard ADMM algorithm are given by

$$\begin{aligned} \boldsymbol{\beta}^{k+1} &:= \arg\min_{\boldsymbol{\beta}} \mathcal{L}_\sigma(\boldsymbol{\beta},\mathbf{z}^k,\boldsymbol{\theta}^k) \\ \mathbf{z}^{k+1} &:= \arg\min_{\mathbf{z}} \mathcal{L}_\sigma(\boldsymbol{\beta}^{k+1},\mathbf{z},\boldsymbol{\theta}^k) \\ \boldsymbol{\theta}^{k+1} &:= \boldsymbol{\theta}^k - \sigma(\mathbf{X}\boldsymbol{\beta}^{k+1}+\mathbf{z}^{k+1}-\mathbf{y}), \end{aligned}$$

where $(\boldsymbol{\beta}^k,\mathbf{z}^k,\boldsymbol{\theta}^k)$ denotes the kth iteration of the algorithm for $k \geq 0$. More specifically, the iterations are

$$\begin{aligned} \boldsymbol{\beta}\text{ step}: &\quad \boldsymbol{\beta}^{k+1} := \arg\min_{\boldsymbol{\beta}} \lambda\|\mathbf{w}\circ\boldsymbol{\beta}\|_1 - \langle\boldsymbol{\theta}^k,\mathbf{X}\boldsymbol{\beta}\rangle + \frac{\sigma}{2}\|\mathbf{X}\boldsymbol{\beta}+\mathbf{z}^k-\mathbf{y}\|_2^2 \\ \mathbf{z}\text{ step}: &\quad \mathbf{z}^{k+1} := \arg\min_{\mathbf{z}} \mathbb{Q}_\tau(\mathbf{z}) - \langle\boldsymbol{\theta}^k,\mathbf{z}\rangle + \frac{\sigma}{2}\|\mathbf{z}+\mathbf{X}\boldsymbol{\beta}^{k+1}-\mathbf{y}\|_2^2 \\ \boldsymbol{\theta}\text{ step}: &\quad \boldsymbol{\theta}^{k+1} := \boldsymbol{\theta}^k - \sigma(\mathbf{X}\boldsymbol{\beta}^{k+1}+\mathbf{z}^{k+1}-\mathbf{y}). \end{aligned} \tag{3.5}$$

Note that in the $\mathbf{z}$ step, the update of $\mathbf{z}^{k+1}$ has a closed form solution which is very easy to compute. This property directly addresses the computational difficulty caused by the nonsmoothness of the quantile regression check loss. In fact, the update of $\mathbf{z}^{k+1}$ can be

carried out component-wisely. For $i = 1, \ldots, n$, we have

$$\begin{aligned} z_i^{k+1} &:= \arg\min_{z_i} \frac{1}{n}\rho_\tau(z_i) - \theta_i^k z_i + \frac{\sigma}{2}(z_i + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{k+1} - y_i)^2 \\ &= \arg\min_{z_i} \rho_\tau(z_i) + \frac{n\sigma}{2}\left[z_i - \left(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \frac{1}{\sigma}\theta_i^k\right)\right]^2. \end{aligned}$$

To solve the above univariate minimization problems, we consider a slightly more general form

$$\mathrm{Prox}_{\rho_\tau}[\xi, \alpha] := \arg\min_{u \in \mathbb{R}} \rho_\tau(u) + \frac{\alpha}{2}(u - \xi)^2, \tag{3.6}$$

whose solution is given in the following lemma.

Lemma 3.1

Given $\tau \in (0, 1)$ and $\alpha > 0$, the proximal mapping $\mathrm{Prox}_{\rho_\tau}[\xi, \alpha]$ in (3.6) has explicit expression: $\mathrm{Prox}_{\rho_\tau}[\xi, \alpha] = \xi - \max\left((\tau - 1)/\alpha,\ \min(\xi, \tau/\alpha)\right)$, or equivalently,

$$\mathrm{Prox}_{\rho_\tau}[\xi, \alpha] = \begin{cases} \xi - \frac{\tau}{\alpha}, & \text{if } \xi > \frac{\tau}{\alpha} \\ 0, & \text{if } \frac{\tau-1}{\alpha} \le \xi \le \frac{\tau}{\alpha} \\ \xi - \frac{\tau-1}{\alpha}, & \text{if } \xi < \frac{\tau-1}{\alpha}. \end{cases}$$ □

The operator $\mathrm{Prox}_{\rho_\tau}$ is called proximal mapping. We now apply the proximal mapping formula to the $\mathbf{z}$ step and obtain

$$z_i^{k+1} = \mathrm{Prox}_{\rho_\tau}\left[y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \frac{1}{\sigma}\theta_i^k,\ n\sigma\right],\ i = 1, \ldots, n. \tag{3.7}$$

Unlike the $\mathbf{z}$ step, the $\boldsymbol{\beta}$ step does not have a simple closed-form formula with a general design matrix $\mathbf{X}$. It would be nice to use a simple closed-form update formula for $\boldsymbol{\beta}$ as well, then the resulting algorithm is more transparent and easy to code. To this end, we

adopt a widely used trick known as "linearization" from the operational research literature. Specifically, we consider adding a proximal term to the objective function in the $\boldsymbol{\beta}$ step and replace the $\boldsymbol{\beta}$ step in the standard ADMM (3.5) with the following augmented $\boldsymbol{\beta}$ step:

$$\begin{aligned}\text{Augmented } \boldsymbol{\beta} \text{ step}: \quad \boldsymbol{\beta}^{k+1} := \arg\min_{\boldsymbol{\beta}} \lambda\|\mathbf{w}\circ\boldsymbol{\beta}\|_1 - \langle\boldsymbol{\theta}^k, \mathbf{X}\boldsymbol{\beta}\rangle + \frac{\sigma}{2}\|\mathbf{X}\boldsymbol{\beta} + \mathbf{z}^k - \mathbf{y}\|_2^2 \\ + \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}^k\|_{\mathbf{S}}^2,\end{aligned}$$

where $\mathbf{S}$ is a positive semi-definite matrix. We let $\mathbf{S} = \sigma(\eta\mathbf{I}_p - \mathbf{X}^{\mathrm{T}}\mathbf{X})$ with $\eta \geq \Lambda_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})$, where $\Lambda_{\max}(\cdot)$ denotes the largest eigenvalue of a real symmetric matrix. Here $\|\mathbf{v}\|_{\mathbf{S}}^2 := \langle\mathbf{v}, \mathbf{S}\mathbf{v}\rangle$ is the semi-norm induced by the semi-inner product defined via $\mathbf{S}$. In the augmented $\boldsymbol{\beta}$ step, the update of $\boldsymbol{\beta}$ can be also carried out component-wisely,

$$\begin{aligned}\boldsymbol{\beta}^{k+1} &= \arg\min_{\boldsymbol{\beta}} \lambda\|\mathbf{w}\circ\boldsymbol{\beta}\|_1 + \frac{\sigma\eta}{2}\left\|\boldsymbol{\beta} - \frac{\sigma\eta\boldsymbol{\beta}^k + \mathbf{X}^{\mathrm{T}}(\boldsymbol{\theta}^k + \sigma\mathbf{y} - \sigma\mathbf{X}\boldsymbol{\beta}^k - \sigma\mathbf{z}^k)}{\sigma\eta}\right\|_2^2 \\ &= \left(\mathrm{Shrink}\Big[\beta_j^k + \frac{1}{\sigma\eta}X_j^{\mathrm{T}}(\boldsymbol{\theta}^k + \sigma\mathbf{y} - \sigma\mathbf{X}\boldsymbol{\beta}^k - \sigma\mathbf{z}^k), \frac{\lambda w_j}{\sigma\eta}\Big]\right)_{1\leq j\leq p},\end{aligned} \tag{3.8}$$

where $\mathrm{Shrink}[u, \alpha] = \mathrm{sgn}(u)\max(|u| - \alpha, 0)$ denotes the soft shrinkage operator and X_j denotes the jth column of $\mathbf{X}$, $j = 1, \ldots, p$.

Based on (3.7) and (3.8), we present the proximal ADMM (pADMM) algorithm for solving the penalized quantile regression

$$\begin{aligned}\text{Augmented } \boldsymbol{\beta} \text{ step}: \quad \boldsymbol{\beta}^{k+1} &:= \arg\min_{\boldsymbol{\beta}} \lambda\|\mathbf{w}\circ\boldsymbol{\beta}\|_1 - \langle\boldsymbol{\theta}^k, \mathbf{X}\boldsymbol{\beta}\rangle + \frac{\sigma}{2}\|\mathbf{X}\boldsymbol{\beta} + \mathbf{z}^k - \mathbf{y}\|_2^2 \\ &\quad + \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}^k\|_{\mathbf{S}}^2 \\ \mathbf{z} \text{ step}: \quad \mathbf{z}^{k+1} &:= \arg\min_{\mathbf{z}} \mathbb{Q}_\tau(\mathbf{z}) - \langle\boldsymbol{\theta}^k, \mathbf{z}\rangle + \frac{\sigma}{2}\|\mathbf{z} + \mathbf{X}\boldsymbol{\beta}^{k+1} - \mathbf{y}\|_2^2 \\ \boldsymbol{\theta} \text{ step}: \quad \boldsymbol{\theta}^{k+1} &:= \boldsymbol{\theta}^k - \gamma\sigma(\mathbf{X}\boldsymbol{\beta}^{k+1} + \mathbf{z}^{k+1} - \mathbf{y}),\end{aligned}$$

where γ is a constant controlling the step length for the $\boldsymbol{\theta}$ step. We summarize the proximal

ADMM algorithm in Algorithm 5.

Note that the $\boldsymbol{\beta}$ step in the ADMM can be also solved with successive linearization minimization, which is equivalent to a proximal gradient method, such as FISTA (Beck and Teboulle, 2009; Parikh and Boyd, 2013). In that sense, our augmented $\boldsymbol{\beta}$ step can be viewed as a one-step iterate of FISTA with step length $1/(\sigma\eta)$. Just like FISTA, on one hand, the proximal ADMM algorithm can be really fast with a reasonable step size, while on the other hand, it can become quite slow when the step size is small. Therefore, when η is large, the step size for the update becomes really small which could result in too many iterations of the algorithm. However, η is indeed large when the dimension p is high. To address this concern, we investigate the ADMM algorithm and notice that the $\boldsymbol{\beta}$ step in (3.5) can be viewed as a lassoed least squares problem. Although the lassoed least squares problem does not have a closed form solution in general, it can be directly solved very efficiently by coordinate descent (Friedman et al., 2007). In other words, we can afford to call a lassoed least squares solver based on coordinate descent to handle the $\boldsymbol{\beta}$ step in the ADMM algorithm. We use scdADMM to denote the combination of sparse coordinate descent and ADMM. We summarize the scdADMM algorithm in Algorithm 6.

Algorithm 5: pADMM – Proximal ADMM algorithm for solving the weighted L_1-penalized quantile regression.

---

1. Initialize the algorithm with $(\boldsymbol{\beta}^0, \mathbf{z}^0, \boldsymbol{\theta}^0)$.
2. For $k = 0, 1, 2, \ldots,$ repeat steps (2.1) – (2.3) until the convergence criterion is met.

   (2.1) Update $\boldsymbol{\beta}^{k+1} \leftarrow \left(\text{Shrink}\left[\beta_j^k + \frac{1}{\sigma\eta} X_j^{\mathrm{T}}(\boldsymbol{\theta}^k + \sigma\mathbf{y} - \sigma\mathbf{X}\boldsymbol{\beta}^k - \sigma\mathbf{z}^k), \frac{\lambda w_j}{\sigma\eta}\right]\right)_{1\le j\le p}$.

   (2.2) Update $\mathbf{z}^{k+1} \leftarrow \left(\text{Prox}_{\rho_\tau}\left[y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{k+1} + \sigma^{-1}\theta_i^k, n\sigma\right]\right)_{1\le i\le n}$.

   (2.3) Update $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k - \gamma\sigma(\mathbf{X}\boldsymbol{\beta}^{k+1} + \mathbf{z}^{k+1} - \mathbf{y})$.

Algorithm 6: scdADMM – Sparse coordinate descent ADMM algorithm for solving the weighted L_1-penalized quantile regression with coordinate descent steps.

1. Initialize the algorithm with $(\boldsymbol{\beta}^0, \mathbf{z}^0, \boldsymbol{\theta}^0)$.
2. For $k = 0, 1, 2, \ldots,$ repeat steps (2.1) – (2.3) until the convergence criterion is met.

   (2.1) Carry out the coordinate descent steps (2.1.1) – (2.1.3).

   (2.1.1) Initialize $\boldsymbol{\beta}^{k,0} = \boldsymbol{\beta}^k$.

   (2.1.2) For $m = 0, 1, 2, \ldots,$ repeat step (2.1.2.1) until convergence.

   (2.1.2.1) For $j = 1, \ldots, p$, update

$$\beta_j^{k,m+1} \leftarrow \frac{\text{Shrink}\left[\sum_{i=1}^n x_{ij}\left\{\theta_i^k + \sigma\left(y_i - z_i^k - \sum_{t\neq j} x_{it}\beta_t^{k,m+I(t<j)}\right)\right\}, \lambda w_j\right]}{\sigma\|X_j\|_2^2}.$$

   (2.1.3) Set $\boldsymbol{\beta}^{k+1} \leftarrow \boldsymbol{\beta}^{k,m+1}$.

   (2.2) Update $\mathbf{z}^{k+1} \leftarrow \left(\text{Prox}_{\rho_\tau}\left[y_i - \mathbf{x}_i^\top\boldsymbol{\beta}^{k+1} + \sigma^{-1}\theta_i^k, n\sigma\right]\right)_{1\le i\le n}$.

   (2.3) Update $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k - \sigma(\mathbf{X}\boldsymbol{\beta}^{k+1} + \mathbf{z}^{k+1} - \mathbf{y})$.

3.3.3 Convergence theory

In this section, we establish the convergence properties of scdADMM and pADMM. Note that the convergence of the scdADMM algorithm (Algorithm 6) can be directly obtained from Boyd et al. (2011). Therefore, we only establish the convergence result for the pADMM algorithm (Algorithm 5). We show that with proper choice of the step length γ, the pADMM algorithm yields a sequence $\{(\boldsymbol{\beta}^k, \mathbf{z}^k), k = 1, 2, \ldots\}$ that converges to a global minimizer of problem (3.4).

Theorem 3.1

For given $\lambda > 0, \sigma > 0, 0 < \tau < 1, 0 < \gamma < (\sqrt{5}+1)/2$ and a component-wisely nonnegative weight vector $\mathbf{w}$, let $\{(\boldsymbol{\beta}^k, \mathbf{z}^k, \boldsymbol{\theta}^k)\}$ be generated by the pADMM algorithm as described in Algorithm 5. Then, the sequence $\{(\boldsymbol{\beta}^k, \mathbf{z}^k), k = 0, 1, 2, \ldots\}$ converges to an optimal solution $(\boldsymbol{\beta}^*, \mathbf{z}^*)$ to (3.4) and $\{\boldsymbol{\theta}^k, k = 0, 1, 2, \ldots\}$ converges to an optimal solution $\boldsymbol{\theta}^*$ to the dual problem of (3.4). Equivalently, $\{\boldsymbol{\beta}^k, k = 0, 1, 2, \ldots\}$ converges

to a global minimizer of problem (3.1). Moreover, when $\gamma = 1$, the sequence of norms $\{\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_{\mathbf{S}}^2 + \sigma\|\mathbf{z}^k - \mathbf{z}^*\|_2^2 + \sigma^{-1}\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\|_2^2,\ k \geq 0\}$ is non-increasing and satisfies $\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_{\mathbf{S}}^2 + \sigma\|\mathbf{z}^k - \mathbf{z}^*\|_2^2 + \sigma^{-1}\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^*\|_2^2 = \mathcal{O}(1/k)$ as $k \to \infty$. □

The proof of the theorem can be found in the supplementary file. Note that the convergence of the algorithm is guaranteed regardless of the value σ takes. According to the theorem, when $\gamma = 1$, the worst-case convergence rate of the algorithm is at least of order $1/k$ in terms of the iterate norms defined in the theorem, where k is the iteration number. Moreover, by setting $\gamma = 1$ and $\mathbf{S} = \mathbf{0}$, the convergence results in Theorem 3.1 can be naturally applied to scdADMM.

3.3.4 Implementation details

We implement Algorithms 5–6 in an R package `FHDQR`, where `FHDQR` stands for fast high-dimensional quantile regression. In this section we describe some important implementation details of the package.

When no λ value is specified, the package will use a default λ sequence that is calculated based on the Karush–Kuhn–Tucker (KKT) condition. This λ sequence is determined by its largest element $\lambda_{\max}$, a factor δ and the number of elements M in the sequence such that the smallest element is given by $\lambda_{\min} = \delta\lambda_{\max}$ and the kth element of the sequence is calculated by

$$\lambda_k = \lambda_{\max}^{\frac{M-k}{M-1}} \lambda_{\min}^{\frac{k-1}{M-1}},\ k = 1, \ldots, M.$$

This makes the λ sequence a decreasing arithmetic progression on the logarithmic scale. By default, M is 100 and δ is 0.001 when $n \geq p$ and 0.05 when $n < p$. We select $\lambda_{\max}$ to make sure that all coefficients $\beta_j, 1 \leq j \leq p$, are shrunk to zero. One such $\lambda_{\max}$ can be

derived from the KKT condition. Specifically, $\widehat{\boldsymbol{\beta}}$ is an optimal solution to problem (3.1) if

$$0 \in -\frac{1}{n}\sum_{i=1}^{n} \partial\rho_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}})x_{ij} + \lambda w_j \partial|\hat{\beta}_j| \tag{3.9}$$

for all $j = 1, \ldots, p$, where $\partial\rho_\tau(u) = (\tau - 1/2) + (1/2)\partial|u|$ and

$$\partial|u| = \begin{cases} \operatorname{sgn}(u), & \text{if } u \neq 0 \\ [-1, 1], & \text{if } u = 0. \end{cases}$$

Here, $\partial f(x)$ denotes the sub-differential of a convex function f at x and $\operatorname{sgn}(\cdot)$ denotes the sign function. For simplicity, assume that all w_j's are positive. Then it follows directly from (3.9) that the choice

$$\lambda_{\max} = \max_{1\le j\le p} w_j^{-1}\left\{\left|\frac{2\tau - 1}{2n}\sum_{i=1}^{n} x_{ij} + \frac{1}{2n}\sum_{i\notin\mathscr{Z}} \operatorname{sgn}(y_i)x_{ij}\right| + \frac{1}{2n}\sum_{i\in\mathscr{Z}} |x_{ij}|\right\},$$

shrinks all coefficients toward exact zero, where $\mathscr{Z} = \{i : y_i = 0, 1 \le i \le n\}$.

We also implement the warm-start technique (Friedman et al., 2010, 2007), which uses the solution at the current λ value as the initial value for the solution at the next λ value.

The ADMM algorithm is iterated until some stopping criterion is met. We adopt the stopping criterion from Boyd et al. (2011), Subsection 3.3.1. Specifically, the algorithm is terminated either when the sequence $\{(\boldsymbol{\beta}^k, \mathbf{z}^k, \boldsymbol{\theta}^k)\}$ meets the following criterion

$$\begin{aligned} &\|\mathbf{X}\boldsymbol{\beta}^k + \mathbf{z}^k - \mathbf{y}\|_2 \le \sqrt{n}\epsilon_1 + \epsilon_2 \max\{\|\mathbf{X}\boldsymbol{\beta}^k\|_2, \|\mathbf{z}^k\|_2, \|\mathbf{y}\|_2\}, \\ &\sigma\|\mathbf{X}^{\mathrm{T}}(\mathbf{z}^k - \mathbf{z}^{k-1})\|_2 \le \sqrt{p}\epsilon_1 + \epsilon_2\|\mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}^k\|_2, \end{aligned}$$

where typical choices are $\epsilon_1 = 10^{-3}$ and $\epsilon_2 = 10^{-3}$, or when the number of ADMM iterations exceeds a certain number, say 10^5, at each λ value along the sequence.

3.4 Numerical Experiments

In this section, we first compare the running times of the ADMM algorithms with those of the R packages `quantreg` and `hqreg` for fitting penalized quantile regression and then investigate the finite-sample statistical performance of penalized quantile regression as compared to the penalized least squares.

3.4.1 Timing comparisons

We conduct extensive timing comparisons for various scenarios of high-dimensional models. Through these timing comparisons, we demonstrate that `FHDQR` compares favorably against `quantreg` and `hqreg`. For the timing comparison, we only consider the lasso penalty for demonstration purpose. All timings reported are performed on an Intel Core i5-3210M processor (single-core, 2.5 GHz).

The first study. In the first setup, we consider a popular simulation model from Friedman et al. (2010) to generate data for timing comparison. We simulate data with n observations from the linear model

$$Y = \sum_{j=1}^{p} X_j \beta_j + k \cdot \varepsilon, \tag{3.10}$$

where $(X_1, \ldots, X_p)^{\mathrm{T}} \sim N(\mathbf{0}, \mathbf{\Sigma})$, $\mathbf{\Sigma} = (\alpha + (1-\alpha) I(i = j))_{p \times p}$, $\beta_j = (-1)^j \exp(-(2j-1)/20)$, $\varepsilon \sim N(0,1)$, and k is chosen such that the signal-noise ratio of the data is 3.0. For our timing comparison, we focus on the high-dimensional situation, where $n = 100$ and $p = 1000$ or 5000, with various choices of the correlation $\alpha \in \{0, 0.1, 0.2, 0.5, 0.9, 0.95\}$. Under each scenario, the timings in seconds are recorded by accumulating the overall time spent on fitting the lasso penalized quantile regression over the same sequence of one hundred λ values. For illustration purpose, three different τ values, 0.25, 0.50 and 0.75, are considered.

The average timings over three runs are reported in Tables 3.1–3.2. We see that the ADMM algorithms and `hqreg` are a lot faster than `quantreg` and scdADMM is the fastest. When the correlation α is small, pADMM is very fast. When the correlation grows, pADMM becomes slower. This can be understood by observing that $\Lambda_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})$ becomes larger as the correlation grows. It is nice to see that scdADMM and `hqreg` are insensitive to the correlation.

Note that in order to do a meaningful timing comparison, we need to check that the objective function values of problem (3.1) at the optimal solutions computed by the different algorithms and make sure different algorithms all yield the same (numerically speaking) objective function values. See Appendix B for a graphical illustration.

The second study. The second model setup is inspired by the simulation studies in Fan et al. (2014a). Specifically, the model for the simulated data is

$$y_i = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \varepsilon_i, \quad \mathbf{x}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{x}}), \quad i = 1, \ldots, n, \tag{3.11}$$

where the true coefficient vector is fixed at

$$\boldsymbol{\beta}^* = (2, 0, 1.5, 0, 0.8, 0, 0, 1, 0, 1.75, 0, 0, 0.75, 0, 0, 0.3, \mathbf{0}_{p-16}^{\mathrm{T}})^{\mathrm{T}}.$$

In our numerical experiments, a variety of error distributions are considered, including: (1) the normal distribution $N(0, 2)$ with variance 2; (2) the mixture normal distribution $0.9N(0, 1) + 0.1N(0, 25)$, denoted by MN_1; (3) the mixture normal distribution $N(0, \sigma^2)$ with $\sigma \sim \mathrm{Unif}(1, 5)$, denoted by MN_2; (4) the Laplace distribution with density $d(u) = 0.5\exp(-|u|)$; (5) the scaled Student's t-distribution with 4 degrees of freedom, $\sqrt{2} \times t_4$; and (6) the Cauchy distribution with density $d(u) = \pi^{-1}(1 + u^2)^{-1}$. For the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}}$, several scenarios are also considered, from the independence structure $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbf{I}_p$, to the autoregressive structures $\boldsymbol{\Sigma}_{\mathbf{x}} = (0.5^{|i-j|})$ and $(0.8^{|i-j|})$, denoted by $\mathrm{AR}_{0.5}$ and $\mathrm{AR}_{0.8}$ respectively, to the compound symmetric structures $\boldsymbol{\Sigma}_{\mathbf{x}} = (\alpha + (1 - \alpha)I(i = j))$ with

$\alpha = 0.5$ and 0.8, denoted by $\text{CS}_{0.5}$ and $\text{CS}_{0.8}$ respectively.

For all of the above settings, we fix $n = 200$ and $p = 1000$ in the timing comparison. The timings, which are accumulated over one hundred pre-chosen λ values, are reported in Table 3.3. We report results at levels $\tau = 0.50$ and $\tau = 0.75$ for demonstration purpose. Several observations can be readily made from this timing comparison. First of all, it is clear from Table 3.3 that the ADMM algorithms are very fast. Secondly, pADMM works fairly well for covariance structures $\mathbf{I}$, $\text{AR}_{0.5}$ and $\text{AR}_{0.8}$ and becomes slower for $\text{CS}_{0.5}$ and $\text{CS}_{0.8}$. Thirdly, the timings for scdADMM exhibit certain insensitivity to the covariance structures and error distributions. This robustness in timing is also observed in `quantreg`. Lastly, `hqreg` takes longer time to fit the data under the Cauchy distribution even when the covariance structures have small correlations.

Table 3.1: Timings (in seconds) for running lasso penalized quantile regression ($\tau = 0.25$, 0.5 and 0.75) on model (3.10) with $n = 100$ and $p = 1000$ over one hundred λ values. Timings reported are averaged over three runs. quantreg: timing by the `quantreg` package (300+: above 300 seconds); hqreg: timing by the `hqreg` package; scdADMM and pADMM: timing by our package `FHDQR`.

	Correlation (α)					
	0.00	0.10	0.20	0.50	0.90	0.95
			$\tau = 0.25$			
quantreg	300+	300+	300+	300+	300+	300+
hqreg	9.52	9.32	9.82	9.86	7.05	5.63
pADMM	0.57	4.38	5.85	12.14	18.43	11.62
scdADMM	1.41	1.37	1.36	1.23	1.00	0.94
			$\tau = 0.50$			
quantreg	300+	300+	300+	300+	300+	300+
hqreg	6.88	6.64	6.89	7.92	8.65	5.29
pADMM	0.62	5.36	7.85	15.47	30.34	21.66
scdADMM	1.26	1.20	1.26	1.18	1.19	0.91
			$\tau = 0.75$			
quantreg	300+	300+	300+	300+	300+	300+
hqreg	8.65	8.34	8.81	8.87	8.37	5.76
pADMM	0.55	4.45	6.26	12.12	21.49	16.40
scdADMM	1.42	1.39	1.48	1.34	1.15	1.20

Table 3.2: Timings (in seconds) for running lasso penalized quantile regression ($\tau = 0.25$, 0.5 and 0.75) on model (3.10) with $n = 100$ and $p = 5000$ over one hundred λ values. Timings reported are averaged over three runs. quantreg: timing by the `quantreg` package (20000+: above 20000 seconds); hqreg: timing by the `hqreg` package; scdADMM and pADMM: timing by our package `FHDQR`.

	Correlation (α)					
	0.00	0.10	0.20	0.50	0.90	0.95
			$\tau = 0.25$			
quantreg	20000+	20000+	20000+	20000+	20000+	20000+
hqreg	14.99	14.89	14.60	15.08	13.63	17.58
pADMM	12.57	44.35	62.06	107.52	168.39	147.23
scdADMM	6.19	6.17	5.89	5.93	5.89	5.39
			$\tau = 0.50$			
quantreg	20000+	20000+	20000+	20000+	20000+	20000+
hqreg	9.82	9.76	10.26	11.13	14.34	17.55
pADMM	12.69	51.30	72.92	135.98	187.20	167.19
scdADMM	5.71	5.50	5.32	5.50	6.13	5.98
			$\tau = 0.75$			
quantreg	20000+	20000+	20000+	20000+	20000+	20000+
hqreg	14.47	15.19	14.82	15.96	14.33	13.61
pADMM	12.76	44.15	58.99	104.38	151.69	115.43
scdADMM	6.21	6.36	6.23	5.44	5.65	5.44

Table 3.3: Timings (in seconds) for running lasso penalized quantile regression ($\tau = 0.5$ and 0.75) on model (3.11) with $n = 200$ and $p = 1000$ over one hundred λ values. All timings reported are averaged over three runs. quantreg: timing by the `quantreg` package (400+: above 400 seconds); hqreg: timing by the `hqreg` package; scdADMM and pADMM: timing by our package `FHDQR`. **I**: independent structure; $AR_{0.5}$ ($AR_{0.8}$): autoregressive structure with correlation 0.5 (0.8); $CS_{0.5}$ ($CS_{0.8}$): compound symmetric structure with correlation 0.5 (0.8).

		Error					
Covariance	Method	$N(0,2)$	MN_1	MN_2	Laplace	$\sqrt{2} \times t_4$	Cauchy
				$\tau = 0.50$			
I	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	10.45	14.35	21.10	10.45	13.95	32.96
	scdADMM	3.03	3.06	4.31	2.88	3.39	5.88
	pADMM	1.52	1.46	1.45	1.47	1.46	0.46
$AR_{0.5}$	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	11.19	12.73	21.61	10.48	14.11	24.56
	scdADMM	3.76	3.89	4.99	3.47	4.11	5.85
	pADMM	1.83	1.80	1.76	1.76	1.77	0.55
$AR_{0.8}$	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	9.28	8.61	20.03	8.17	11.31	16.06
	scdADMM	5.63	5.46	6.91	5.16	5.62	6.80
	pADMM	2.63	2.42	2.82	2.43	2.53	0.78
$CS_{0.5}$	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	13.70	11.64	21.96	10.11	14.32	15.04
	scdADMM	7.11	7.34	9.60	6.77	7.68	8.49
	pADMM	19.91	18.58	21.49	17.96	20.12	5.65
$CS_{0.8}$	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	16.88	12.86	19.17	11.01	13.27	10.04
	scdADMM	9.11	9.14	10.67	9.18	9.61	9.92
	pADMM	13.96	13.45	15.39	14.24	12.83	3.84
				$\tau = 0.75$			
I	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	14.34	15.11	29.34	10.68	14.07	37.02
	scdADMM	3.53	3.58	5.07	3.28	3.62	6.48
	pADMM	1.33	1.34	1.30	1.22	1.34	0.40
$AR_{0.5}$	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	11.99	13.12	25.51	10.66	14.48	30.96
	scdADMM	4.00	4.22	5.42	3.96	4.38	6.98
	pADMM	1.43	1.50	1.55	1.37	1.55	0.45
$AR_{0.8}$	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	9.95	10.92	19.24	8.30	9.98	13.51
	scdADMM	6.54	6.09	7.38	6.03	6.30	7.67
	pADMM	2.03	2.19	2.20	2.06	2.08	0.66
$CS_{0.5}$	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	14.91	13.07	25.15	10.48	13.48	14.55
	scdADMM	8.23	8.10	9.57	7.28	8.84	10.79
	pADMM	17.66	16.33	18.70	15.76	18.6	4.92
$CS_{0.8}$	quantreg	400+	400+	400+	400+	400+	400+
	hqreg	15.23	12.13	19.75	9.64	13.54	8.32
	scdADMM	9.74	9.87	11.18	9.49	11.59	6.12
	pADMM	14.24	13.13	16.67	13.24	14.54	4.11

3.4.2 Finite sample performance

We investigate the finite-sample performance of the penalized quantile regression. The purpose is to compare penalized quantile regression with penalized least squares using the same penalty function. Many researchers have done simulation studies to show that the penalized quantile regression has some unique advantages over the penalized least squares. Our simulation study is more extensive than the existing results.

We adopt the six error distributions and five covariance structures that we used in the second timing study in section 4.1. Under each scenario, we investigate the estimation and selection performance of the penalized least squares regression and the penalized quantile regression. Since we observe similar statistical performance for the penalized quantile regression with $\tau = 0.25$ and $\tau = 0.75$, we only present the results for $\tau = 0.50$ and $\tau = 0.75$. All three types of penalties, the lasso, adaptive lasso and folded concave (specifically, SCAD), are considered in the simulation. The results are summarized in Tables 3.4–3.8. It is clearly shown that the penalized quantile regression performs better than the penalized least squares under heavy-tailed error distributions, such as t and Cauchy.

3.4.3 A real data example

The microarray data of Scheetz et al. (2006) comprise gene expression levels of 31,042 probes on 120 twelve-week-old laboratory rats. The data were used to understand the gene regulation in mammalian eyes and to gain insight into genetic variation related to human eyes. We apply the penalized quantile regression to analyze this set of microarray data.

Following Scheetz et al. (2006) and Huang et al. (2008), we select the 18,976 probes that exhibited sufficient variation. Among those probes, there is one probe, 1389163_at, corresponding to gene *TRIM32*, that was found to be associated with the Bardet–Biedl syndrome (Chiang et al., 2006), a human genetic disorder that affects many parts of the body and primarily the retina. We study how the expression of this gene depends on the

Table 3.4: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (3.11) with independent covariates $\mathbf{\Sigma} = \mathbf{I}$. The estimation accuracy is measured by the L_1 and L_2 losses and the selection accuracy is measured by the number of false positives (FP) and false negatives (FN). Numbers reported are averaged over 100 independent runs with their respective standard errors listed in the parentheses.

		$\mathbf{\Sigma} = \mathbf{I}$		
		Lasso	Alasso	SCAD
			L_1, L_2 losses	
$N(0, 2)$	LS	3.492 (0.088), 0.783 (0.011)	2.142 (0.081), 0.583 (0.014)	0.806 (0.020), 0.383 (0.007)
	QR(0.50)	4.941 (0.180), 0.906 (0.014)	1.317 (0.038), 0.535 (0.013)	1.331 (0.059), 0.462 (0.010)
	QR(0.75)	4.581 (0.125), 0.995 (0.018)	1.344 (0.045), 0.570 (0.018)	1.390 (0.051), 0.493 (0.012)
MN_1	LS	4.217 (0.097), 0.977 (0.018)	3.662 (0.180), 0.869 (0.027)	1.105 (0.043), 0.510 (0.017)
	QR(0.50)	4.133 (0.141), 0.750 (0.011)	0.877 (0.027), 0.385 (0.010)	0.939 (0.030), 0.349 (0.007)
	QR(0.75)	3.877 (0.106), 0.867 (0.016)	1.094 (0.034), 0.473 (0.013)	1.069 (0.038), 0.400 (0.008)
MN_2	LS	6.770 (0.164), 1.596 (0.023)	8.990 (0.340), 1.948 (0.039)	2.724 (0.097), 1.173 (0.033)
	QR(0.50)	8.124 (0.320), 1.714 (0.027)	3.281 (0.080), 1.256 (0.026)	3.198 (0.135), 1.147 (0.031)
	QR(0.75)	7.737 (0.277), 1.918 (0.031)	4.185 (0.111), 1.596 (0.033)	4.190 (0.158), 1.520 (0.037)
Laplace	LS	3.257 (0.079), 0.759 (0.012)	2.095 (0.080), 0.579 (0.014)	0.784 (0.028), 0.366 (0.010)
	QR(0.50)	3.732 (0.118), 0.723 (0.014)	0.807 (0.026), 0.367 (0.010)	0.864 (0.029), 0.315 (0.008)
	QR(0.75)	4.102 (0.144), 0.861 (0.016)	1.171 (0.041), 0.495 (0.015)	1.293 (0.057), 0.454 (0.012)
$\sqrt{2} \times t_4$	LS	4.722 (0.120), 1.058 (0.020)	4.275 (0.200), 0.992 (0.029)	1.337 (0.057), 0.606 (0.024)
	QR(0.50)	5.489 (0.179), 1.043 (0.017)	1.493 (0.044), 0.593 (0.016)	1.491 (0.068), 0.518 (0.014)
	QR(0.75)	5.706 (0.172), 1.220 (0.022)	1.866 (0.076), 0.771 (0.029)	1.794 (0.066), 0.645 (0.021)
Cauchy	LS	11.262 (0.951), 3.442 (0.071)	25.593 (4.345), 8.310 (3.370)	325.098 (78.041), 32.315 (7.091)
	QR(0.50)	5.326 (0.185), 1.098 (0.021)	1.611 (0.072), 0.662 (0.027)	1.543 (0.077), 0.521 (0.020)
	QR(0.75)	7.015 (0.213), 1.642 (0.038)	2.962 (0.124), 1.167 (0.042)	2.924 (0.154), 0.992 (0.037)
			FP, FN	
$N(0, 2)$	LS	42.74 (1.81), 0.38 (0.05)	12.60 (0.69), 0.43 (0.05)	0.22 (0.05), 0.84 (0.04)
	QR(0.50)	62.19 (3.28), 0.37 (0.05)	2.43 (0.27), 0.87 (0.04)	9.68 (1.11), 0.62 (0.05)
	QR(0.75)	46.48 (2.38), 0.51 (0.05)	1.68 (0.16), 0.93 (0.05)	8.83 (0.82), 0.67 (0.05)
MN_1	LS	39.86 (1.49), 0.53 (0.05)	16.85 (0.98), 0.57 (0.05)	0.49 (0.11), 1.02 (0.04)
	QR(0.50)	63.10 (3.18), 0.23 (0.04)	1.18 (0.15), 0.56 (0.05)	8.27 (0.84), 0.44 (0.05)
	QR(0.75)	42.40 (1.83), 0.41 (0.05)	1.26 (0.15), 0.81 (0.04)	7.87 (0.80), 0.52 (0.05)
MN_2	LS	36.98 (1.50), 1.14 (0.08)	21.76 (1.21), 1.55 (0.08)	1.32 (0.14), 2.15 (0.09)
	QR(0.50)	47.99 (3.05), 1.31 (0.09)	3.76 (0.30), 2.07 (0.10)	8.01 (0.88), 1.61 (0.09)
	QR(0.75)	32.99 (2.33), 2.04 (0.11)	3.17 (0.28), 2.83 (0.10)	7.30 (0.81), 2.59 (0.10)
Laplace	LS	39.42 (1.77), 0.22 (0.04)	11.79 (0.66), 0.46 (0.05)	0.31 (0.08), 0.67 (0.05)
	QR(0.50)	55.55 (2.31), 0.20 (0.04)	1.11 (0.14), 0.62 (0.05)	9.19 (0.92), 0.31 (0.05)
	QR(0.75)	47.49 (2.64), 0.35 (0.05)	1.54 (0.17), 0.83 (0.05)	9.58 (0.90), 0.61 (0.05)
$\sqrt{2} \times t_4$	LS	42.71 (1.77), 0.59 (0.05)	17.76 (1.01), 0.67 (0.05)	0.56 (0.10), 1.11 (0.06)
	QR(0.50)	58.34 (2.77), 0.42 (0.05)	2.88 (0.24), 0.90 (0.05)	9.86 (1.07), 0.75 (0.05)
	QR(0.75)	47.93 (2.50), 0.67 (0.06)	2.26 (0.18), 1.26 (0.07)	9.15 (0.67), 0.83 (0.06)
Cauchy	LS	13.35 (3.15), 6.07 (0.15)	30.01 (5.62), 5.71 (0.18)	108.66 (5.89), 5.00 (0.13)
	QR(0.50)	51.16 (2.46), 0.57 (0.05)	2.32 (0.21), 1.11 (0.07)	11.17 (0.95), 0.66 (0.05)
	QR(0.75)	43.46 (2.22), 1.21 (0.07)	3.02 (0.23), 2.01 (0.11)	11.00 (1.04), 1.34 (0.09)

Table 3.5: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (3.11) with covariance matrix $\mathbf{\Sigma} = (0.5^{|i-j|})$. The estimation accuracy is measured by the L_1 and L_2 losses and the selection accuracy is measured by the number of false positives (FP) and false negatives (FN). Numbers reported are averaged over 100 independent runs with their respective standard errors listed in the parentheses.

		$\mathbf{\Sigma} = (0.5^{\|i-j\|})$		
		Lasso	Alasso	SCAD
			L_1, L_2 losses	
$N(0,2)$	LS	2.787 (0.081), 0.679 (0.011)	1.718 (0.070), 0.539 (0.014)	0.873 (0.024), 0.404 (0.009)
	QR(0.50)	3.896 (0.141), 0.803 (0.012)	1.199 (0.039), 0.499 (0.013)	1.250 (0.049), 0.455 (0.009)
	QR(0.75)	3.612 (0.111), 0.853 (0.014)	1.372 (0.044), 0.591 (0.017)	1.369 (0.054), 0.511 (0.015)
MN_1	LS	3.395 (0.102), 0.821 (0.016)	2.850 (0.135), 0.790 (0.025)	1.149 (0.038), 0.519 (0.017)
	QR(0.50)	3.027 (0.107), 0.628 (0.011)	0.858 (0.026), 0.384 (0.009)	1.042 (0.046), 0.364 (0.009)
	QR(0.75)	3.369 (0.114), 0.781 (0.016)	1.002 (0.032), 0.449 (0.013)	1.081 (0.040), 0.395 (0.008)
MN_2	LS	5.764 (0.163), 1.429 (0.024)	7.052 (0.285), 1.705 (0.038)	2.676 (0.100), 1.169 (0.032)
	QR(0.50)	6.559 (0.230), 1.464 (0.025)	2.961 (0.095), 1.136 (0.028)	3.140 (0.131), 1.148 (0.031)
	QR(0.75)	6.780 (0.204), 1.650 (0.030)	3.581 (0.114), 1.391 (0.032)	3.785 (0.132), 1.417 (0.034)
Laplace	LS	2.720 (0.075), 0.667 (0.012)	1.799 (0.074), 0.549 (0.015)	0.814 (0.021), 0.380 (0.008)
	QR(0.50)	3.018 (0.108), 0.639 (0.013)	0.788 (0.022), 0.363 (0.009)	0.848 (0.032), 0.312 (0.009)
	QR(0.75)	3.345 (0.116), 0.762 (0.014)	1.081 (0.028), 0.472 (0.010)	1.093 (0.038), 0.418 (0.010)
$\sqrt{2} \times t_4$	LS	3.794 (0.118), 0.934 (0.019)	3.700 (0.168), 0.961 (0.030)	1.312 (0.062), 0.595 (0.023)
	QR(0.50)	4.114 (0.153), 0.880 (0.017)	1.237 (0.038), 0.523 (0.014)	1.408 (0.059), 0.510 (0.016)
	QR(0.75)	4.467 (0.138), 1.044 (0.020)	1.760 (0.058), 0.721 (0.022)	1.731 (0.076), 0.635 (0.021)
Cauchy	LS	15.047 (2.087), 3.779 (0.166)	21.601 (3.035), 5.217 (0.469)	299.790 (82.662), 29.588 (7.306)
	QR(0.50)	3.924 (0.127), 0.902 (0.019)	1.310 (0.051), 0.546 (0.018)	1.324 (0.063), 0.476 (0.016)
	QR(0.75)	5.677 (0.201), 1.334 (0.032)	2.369 (0.080), 0.959 (0.031)	2.825 (0.133), 1.009 (0.040)
			FP, FN	
$N(0,2)$	LS	32.75 (1.51), 0.29 (0.05)	7.82 (0.53), 0.46 (0.05)	0.31 (0.07), 0.76 (0.04)
	QR(0.50)	45.12 (2.60), 0.40 (0.05)	1.88 (0.19), 0.72 (0.05)	8.02 (0.94), 0.69 (0.05)
	QR(0.75)	34.69 (1.98), 0.52 (0.05)	1.24 (0.14), 0.93 (0.05)	6.72 (0.74), 0.61 (0.05)
MN_1	LS	33.39 (1.66), 0.46 (0.05)	11.19 (0.76), 0.61 (0.05)	0.46 (0.06), 0.96 (0.04)
	QR(0.50)	44.69 (2.27), 0.13 (0.03)	1.11 (0.13), 0.59 (0.05)	9.94 (1.16), 0.40 (0.05)
	QR(0.75)	36.53 (2.02), 0.35 (0.05)	0.75 (0.09), 0.79 (0.05)	7.67 (0.84), 0.48 (0.05)
MN_2	LS	29.89 (1.43), 1.11 (0.07)	15.78 (0.97), 1.27 (0.07)	0.97 (0.16), 2.22 (0.08)
	QR(0.50)	38.64 (2.15), 1.20 (0.06)	3.44 (0.29), 1.81 (0.08)	7.56 (0.84), 1.69 (0.08)
	QR(0.75)	31.64 (1.72), 1.43 (0.08)	2.95 (0.27), 2.35 (0.09)	6.12 (0.65), 2.38 (0.10)
Laplace	LS	31.80 (1.34), 0.32 (0.05)	8.86 (0.51), 0.50 (0.05)	0.43 (0.07), 0.70 (0.05)
	QR(0.50)	43.52 (2.21), 0.21 (0.04)	0.71 (0.11), 0.57 (0.05)	8.20 (0.88), 0.26 (0.04)
	QR(0.75)	38.20 (2.15), 0.32 (0.05)	1.12 (0.11), 0.77 (0.04)	6.32 (0.64), 0.61 (0.05)
$\sqrt{2} \times t_4$	LS	31.88 (1.51), 0.62 (0.05)	13.59 (0.91), 0.69 (0.05)	0.52 (0.09), 1.14 (0.04)
	QR(0.50)	42.01 (2.28), 0.43 (0.05)	1.51 (0.15), 0.76 (0.05)	7.94 (0.78), 0.73 (0.05)
	QR(0.75)	35.98 (1.93), 0.68 (0.06)	1.84 (0.19), 1.11 (0.06)	7.12 (0.72), 0.89 (0.05)
Cauchy	LS	19.81 (4.54), 5.79 (0.17)	25.02 (4.51), 5.35 (0.19)	102.83 (5.77), 4.42 (0.14)
	QR(0.50)	37.45 (1.66), 0.51 (0.05)	1.99 (0.19), 0.90 (0.04)	8.60 (0.77), 0.73 (0.04)
	QR(0.75)	38.92 (2.17), 1.04 (0.06)	2.50 (0.21), 1.54 (0.09)	8.71 (0.77), 1.48 (0.09)

Table 3.6: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (3.11) with covariance matrix $\mathbf{\Sigma} = (0.8^{|i-j|})$. The estimation accuracy is measured by the L_1 and L_2 losses and the selection accuracy is measured by the number of false positives (FP) and false negatives (FN). Numbers reported are averaged over 100 independent runs with their respective standard errors listed in the parentheses.

		$\mathbf{\Sigma} = (0.8^{\|i-j\|})$		
		Lasso	Alasso	SCAD
			L_1, L_2 losses	
$N(0,2)$	LS	2.497 (0.081), 0.718 (0.018)	1.603 (0.061), 0.583 (0.019)	1.272 (0.061), 0.554 (0.023)
	QR(0.50)	3.151 (0.125), 0.829 (0.023)	1.726 (0.077), 0.726 (0.028)	1.684 (0.067), 0.660 (0.025)
	QR(0.75)	3.477 (0.105), 0.927 (0.020)	1.798 (0.078), 0.765 (0.030)	1.960 (0.085), 0.741 (0.025)
MN_1	LS	3.453 (0.107), 0.949 (0.024)	2.496 (0.116), 0.856 (0.032)	1.707 (0.091), 0.733 (0.036)
	QR(0.50)	2.731 (0.097), 0.697 (0.016)	1.244 (0.046), 0.541 (0.019)	1.229 (0.066), 0.460 (0.013)
	QR(0.75)	2.981 (0.111), 0.809 (0.023)	1.375 (0.056), 0.610 (0.023)	1.407 (0.054), 0.558 (0.020)
MN_2	LS	5.380 (0.141), 1.552 (0.032)	6.176 (0.219), 1.781 (0.042)	3.859 (0.131), 1.592 (0.048)
	QR(0.50)	5.656 (0.200), 1.478 (0.036)	3.746 (0.146), 1.454 (0.048)	3.949 (0.154), 1.512 (0.048)
	QR(0.75)	6.397 (0.197), 1.760 (0.041)	4.564 (0.158), 1.801 (0.053)	4.531 (0.157), 1.733 (0.055)
Laplace	LS	2.592 (0.082), 0.736 (0.019)	1.499 (0.057), 0.568 (0.019)	1.272 (0.059), 0.560 (0.023)
	QR(0.50)	2.417 (0.087), 0.652 (0.016)	1.058 (0.043), 0.474 (0.018)	1.139 (0.054), 0.437 (0.017)
	QR(0.75)	3.132 (0.108), 0.844 (0.023)	1.497 (0.068), 0.640 (0.026)	1.638 (0.078), 0.624 (0.023)
$\sqrt{2} \times t_4$	LS	3.378 (0.098), 0.983 (0.025)	2.805 (0.127), 0.926 (0.035)	2.175 (0.113), 0.896 (0.041)
	QR(0.50)	3.795 (0.146), 0.961 (0.024)	1.778 (0.071), 0.770 (0.029)	1.836 (0.082), 0.716 (0.028)
	QR(0.75)	4.120 (0.138), 1.123 (0.028)	2.377 (0.092), 0.989 (0.033)	2.392 (0.108), 0.921 (0.035)
Cauchy	LS	24.121 (6.956), 5.008 (0.750)	18.301 (1.326), 5.339 (0.241)	1105.656 (391.026), 96.732 (29.233)
	QR(0.50)	3.745 (0.117), 1.004 (0.026)	2.042 (0.098), 0.853 (0.039)	1.777 (0.088), 0.689 (0.031)
	QR(0.75)	5.360 (0.177), 1.408 (0.035)	3.380 (0.152), 1.324 (0.050)	3.531 (0.155), 1.308 (0.051)
			FP, FN	
$N(0,2)$	LS	21.47 (1.16), 0.26 (0.04)	4.37 (0.34), 0.51 (0.05)	0.57 (0.10), 0.86 (0.04)
	QR(0.50)	26.49 (1.92), 0.42 (0.05)	1.05 (0.11), 1.04 (0.05)	5.83 (0.73), 0.99 (0.05)
	QR(0.75)	25.73 (1.56), 0.53 (0.05)	1.10 (0.11), 1.02 (0.06)	6.42 (0.82), 0.89 (0.05)
MN_1	LS	24.03 (1.17), 0.48 (0.05)	5.17 (0.37), 0.70 (0.06)	0.69 (0.10), 1.04 (0.06)
	QR(0.50)	28.67 (1.79), 0.24 (0.04)	0.63 (0.08), 0.78 (0.05)	7.21 (1.24), 0.65 (0.05)
	QR(0.75)	24.29 (1.44), 0.38 (0.05)	0.69 (0.09), 0.81 (0.05)	5.07 (0.60), 0.86 (0.04)
MN_2	LS	19.53 (1.08), 1.24 (0.07)	10.02 (0.69), 1.58 (0.08)	1.33 (0.14), 2.42 (0.08)
	QR(0.50)	27.12 (1.74), 1.10 (0.06)	2.73 (0.23), 1.90 (0.10)	4.54 (0.51), 2.12 (0.09)
	QR(0.75)	23.18 (1.43), 1.25 (0.07)	2.37 (0.21), 2.38 (0.09)	3.89 (0.42), 2.43 (0.09)
Laplace	LS	21.84 (1.16), 0.29 (0.05)	3.46 (0.28), 0.44 (0.05)	0.53 (0.08), 0.89 (0.05)
	QR(0.50)	24.47 (1.51), 0.21 (0.04)	0.63 (0.08), 0.69 (0.05)	7.14 (0.89), 0.57 (0.05)
	QR(0.75)	24.56 (1.42), 0.44 (0.05)	0.79 (0.10), 0.91 (0.05)	5.69 (0.80), 0.86 (0.04)
$\sqrt{2} \times t_4$	LS	19.68 (0.90), 0.51 (0.06)	6.70 (0.40), 0.83 (0.05)	1.26 (0.16), 1.24 (0.07)
	QR(0.50)	29.32 (1.89), 0.45 (0.05)	1.13 (0.13), 1.04 (0.06)	5.21 (0.60), 1.04 (0.04)
	QR(0.75)	24.13 (1.50), 0.70 (0.06)	1.55 (0.15), 1.30 (0.07)	4.97 (0.53), 1.23 (0.05)
Cauchy	LS	17.76 (3.45), 5.33 (0.19)	12.48 (2.26), 5.38 (0.17)	116.48 (6.97), 4.88 (0.11)
	QR(0.50)	27.86 (1.63), 0.50 (0.06)	1.55 (0.14), 1.15 (0.06)	5.43 (0.68), 0.95 (0.06)
	QR(0.75)	30.64 (1.98), 0.83 (0.07)	2.11 (0.18), 1.78 (0.09)	5.78 (0.60), 1.73 (0.09)

Table 3.7: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (3.11) with covariance matrix $\mathbf{\Sigma} = (0.5 + 0.5I(i = j))$. The estimation accuracy is measured by the L_1 and L_2 losses and the selection accuracy is measured by the number of false positives (FP) and false negatives (FN). Numbers reported are averaged over 100 independent runs with their respective standard errors listed in the parentheses.

		$\mathbf{\Sigma} = (0.5 + 0.5I(i = j))$		
		Lasso	Alasso	SCAD
			L_1, L_2 losses	
$N(0, 2)$	LS	4.229 (0.109), 0.959 (0.017)	1.717 (0.056), 0.610 (0.016)	1.286 (0.060), 0.570 (0.022)
	QR(0.50)	5.744 (0.216), 1.177 (0.018)	1.710 (0.060), 0.718 (0.023)	1.681 (0.078), 0.674 (0.024)
	QR(0.75)	5.630 (0.152), 1.236 (0.020)	2.016 (0.069), 0.835 (0.026)	2.170 (0.111), 0.824 (0.028)
MN_1	LS	5.427 (0.114), 1.250 (0.023)	2.974 (0.107), 0.940 (0.027)	1.997 (0.092), 0.873 (0.036)
	QR(0.50)	4.425 (0.118), 0.937 (0.014)	1.212 (0.039), 0.526 (0.015)	1.120 (0.045), 0.454 (0.012)
	QR(0.75)	4.664 (0.133), 1.058 (0.021)	1.542 (0.055), 0.652 (0.021)	1.388 (0.062), 0.560 (0.018)
MN_2	LS	8.820 (0.229), 2.028 (0.031)	7.923 (0.272), 2.039 (0.046)	4.931 (0.149), 1.821 (0.041)
	QR(0.50)	8.815 (0.244), 1.989 (0.032)	5.067 (0.177), 1.749 (0.046)	5.678 (0.306), 1.865 (0.049)
	QR(0.75)	9.878 (0.253), 2.292 (0.033)	6.584 (0.210), 2.193 (0.054)	6.539 (0.249), 2.179 (0.055)
Laplace	LS	4.217 (0.100), 0.973 (0.018)	1.753 (0.059), 0.620 (0.017)	1.294 (0.051), 0.578 (0.021)
	QR(0.50)	4.120 (0.135), 0.887 (0.016)	1.110 (0.042), 0.494 (0.017)	0.991 (0.049), 0.408 (0.012)
	QR(0.75)	5.207 (0.168), 1.103 (0.022)	1.667 (0.067), 0.700 (0.025)	1.609 (0.081), 0.666 (0.024)
$\sqrt{2} \times t_4$	LS	6.027 (0.158), 1.342 (0.025)	3.706 (0.147), 1.114 (0.032)	2.265 (0.107), 0.975 (0.038)
	QR(0.50)	5.941 (0.164), 1.276 (0.022)	2.045 (0.074), 0.842 (0.027)	1.900 (0.079), 0.780 (0.028)
	QR(0.75)	6.485 (0.165), 1.456 (0.025)	2.819 (0.108), 1.090 (0.034)	2.959 (0.125), 1.112 (0.035)
Cauchy	LS	20.931 (2.691), 4.650 (0.241)	32.018 (3.774), 8.487 (0.627)	472.891 (139.968), 46.890 (12.323)
	QR(0.50)	5.905 (0.191), 1.324 (0.028)	2.497 (0.131), 0.978 (0.040)	2.257 (0.113), 0.868 (0.035)
	QR(0.75)	8.193 (0.223), 1.904 (0.038)	4.563 (0.211), 1.598 (0.057)	4.904 (0.198), 1.684 (0.051)
			FP, FN	
$N(0, 2)$	LS	38.01 (1.37), 0.53 (0.05)	4.77 (0.26), 0.75 (0.04)	0.56 (0.11), 1.02 (0.04)
	QR(0.50)	44.74 (2.64), 0.58 (0.05)	1.44 (0.13), 1.06 (0.05)	2.76 (0.43), 1.05 (0.05)
	QR(0.75)	38.27 (1.67), 0.67 (0.05)	1.89 (0.14), 1.29 (0.06)	3.90 (0.81), 1.24 (0.06)
MN_1	LS	37.55 (1.10), 0.74 (0.06)	7.95 (0.35), 1.01 (0.06)	0.57 (0.08), 1.58 (0.07)
	QR(0.50)	41.34 (1.72), 0.44 (0.05)	0.82 (0.10), 0.90 (0.04)	2.68 (0.47), 0.72 (0.05)
	QR(0.75)	35.76 (1.55), 0.57 (0.05)	1.13 (0.12), 1.05 (0.05)	2.39 (0.44), 0.90 (0.04)
MN_2	LS	36.00 (1.38), 1.86 (0.07)	15.80 (0.89), 2.16 (0.09)	4.80 (0.65), 2.87 (0.09)
	QR(0.50)	34.48 (1.46), 1.94 (0.07)	5.08 (0.31), 2.75 (0.10)	6.23 (0.99), 2.84 (0.08)
	QR(0.75)	32.74 (1.38), 2.46 (0.10)	5.61 (0.29), 3.48 (0.09)	5.29 (0.50), 3.34 (0.11)
Laplace	LS	37.45 (1.05), 0.45 (0.05)	5.16 (0.25), 0.74 (0.05)	0.43 (0.07), 1.07 (0.04)
	QR(0.50)	38.10 (1.96), 0.42 (0.05)	0.65 (0.08), 0.86 (0.05)	2.94 (0.45), 0.71 (0.05)
	QR(0.75)	41.16 (1.93), 0.62 (0.05)	1.44 (0.13), 1.13 (0.05)	2.29 (0.46), 1.03 (0.06)
$\sqrt{2} \times t_4$	LS	40.64 (1.53), 0.84 (0.06)	9.13 (0.42), 1.03 (0.06)	0.92 (0.20), 1.68 (0.08)
	QR(0.50)	41.34 (1.78), 0.83 (0.06)	2.23 (0.17), 1.24 (0.06)	2.03 (0.26), 1.19 (0.06)
	QR(0.75)	35.86 (1.43), 0.99 (0.05)	2.90 (0.21), 1.83 (0.08)	3.83 (0.48), 1.71 (0.07)
Cauchy	LS	23.75 (3.68), 6.02 (0.12)	19.74 (3.82), 6.42 (0.10)	96.64 (6.00), 5.51 (0.13)
	QR(0.50)	40.24 (2.18), 0.80 (0.06)	2.64 (0.24), 1.47 (0.08)	3.49 (0.47), 1.36 (0.07)
	QR(0.75)	38.41 (1.69), 1.74 (0.09)	4.46 (0.27), 2.60 (0.12)	4.47 (0.35), 2.60 (0.10)

Table 3.8: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (3.11) with covariance matrix $\mathbf{\Sigma} = (0.8 + 0.2I(i = j))$. The estimation accuracy is measured by the L_1 and L_2 losses and the selection accuracy is measured by the number of false positives (FP) and false negatives (FN). Numbers reported are averaged over 100 independent runs with their respective standard errors listed in the parentheses.

		$\mathbf{\Sigma} = (0.8 + 0.2I(i = j))$		
		Lasso	Alasso	SCAD
			L_1, L_2 losses	
$N(0, 2)$	LS	6.199 (0.119), 1.440 (0.022)	3.354 (0.131), 1.143 (0.030)	3.009 (0.108), 1.258 (0.037)
	QR(0.50)	7.938 (0.228), 1.716 (0.024)	3.656 (0.110), 1.419 (0.035)	4.195 (0.246), 1.425 (0.041)
	QR(0.75)	7.884 (0.176), 1.820 (0.030)	4.301 (0.146), 1.617 (0.044)	4.960 (0.207), 1.653 (0.033)
MN_1	LS	7.969 (0.171), 1.853 (0.030)	5.295 (0.201), 1.609 (0.041)	4.083 (0.184), 1.606 (0.056)
	QR(0.50)	6.745 (0.199), 1.415 (0.025)	2.689 (0.089), 1.124 (0.033)	2.635 (0.124), 1.003 (0.032)
	QR(0.75)	6.973 (0.151), 1.603 (0.026)	3.421 (0.125), 1.345 (0.037)	3.768 (0.167), 1.311 (0.039)
MN_2	LS	11.557 (0.209), 2.766 (0.039)	11.552 (0.358), 3.005 (0.065)	9.011 (0.245), 3.005 (0.061)
	QR(0.50)	11.892 (0.385), 2.688 (0.048)	9.125 (0.242), 2.996 (0.061)	9.467 (0.394), 2.839 (0.065)
	QR(0.75)	13.183 (0.294), 3.110 (0.040)	10.657 (0.234), 3.431 (0.061)	10.788 (0.262), 3.218 (0.055)
Laplace	LS	6.021 (0.132), 1.416 (0.024)	3.593 (0.167), 1.196 (0.034)	2.993 (0.120), 1.260 (0.038)
	QR(0.50)	5.902 (0.138), 1.317 (0.024)	2.608 (0.086), 1.087 (0.033)	2.394 (0.154), 0.906 (0.040)
	QR(0.75)	7.431 (0.182), 1.703 (0.029)	3.557 (0.120), 1.398 (0.038)	3.645 (0.159), 1.309 (0.037)
$\sqrt{2} \times t_4$	LS	7.969 (0.172), 1.878 (0.029)	5.866 (0.221), 1.715 (0.041)	5.089 (0.208), 1.942 (0.056)
	QR(0.50)	8.922 (0.259), 1.957 (0.033)	4.689 (0.141), 1.742 (0.043)	4.704 (0.186), 1.652 (0.038)
	QR(0.75)	9.250 (0.213), 2.121 (0.030)	5.468 (0.161), 1.971 (0.045)	6.322 (0.239), 2.024 (0.045)
Cauchy	LS	23.053 (3.151), 5.664 (0.438)	33.802 (7.390), 8.318 (0.940)	473.486 (133.365), 53.634 (12.914)
	QR(0.50)	8.023 (0.243), 1.872 (0.039)	4.608 (0.187), 1.706 (0.054)	4.987 (0.235), 1.727 (0.056)
	QR(0.75)	11.009 (0.260), 2.635 (0.046)	8.118 (0.265), 2.714 (0.075)	8.619 (0.331), 2.589 (0.063)
			FP, FN	
$N(0, 2)$	LS	36.49 (0.96), 0.93 (0.05)	6.50 (0.58), 1.45 (0.07)	0.96 (0.12), 2.09 (0.08)
	QR(0.50)	38.46 (1.83), 1.33 (0.09)	2.11 (0.14), 2.26 (0.08)	4.95 (1.03), 2.01 (0.09)
	QR(0.75)	33.41 (1.16), 1.58 (0.09)	2.61 (0.15), 2.70 (0.09)	5.95 (0.95), 2.36 (0.10)
MN_1	LS	35.91 (0.99), 1.44 (0.09)	9.39 (0.65), 2.05 (0.09)	1.69 (0.36), 2.55 (0.09)
	QR(0.50)	41.13 (1.81), 0.87 (0.06)	1.11 (0.11), 1.77 (0.08)	2.80 (0.49), 1.47 (0.07)
	QR(0.75)	34.83 (1.12), 1.05 (0.07)	2.22 (0.18), 2.09 (0.08)	4.27 (0.48), 1.89 (0.09)
MN_2	LS	31.06 (0.81), 3.27 (0.10)	16.10 (0.71), 3.74 (0.10)	9.30 (0.99), 4.12 (0.10)
	QR(0.50)	32.37 (1.56), 3.09 (0.10)	5.61 (0.25), 4.36 (0.09)	7.15 (0.72), 4.08 (0.10)
	QR(0.75)	30.37 (1.14), 3.90 (0.10)	6.10 (0.22), 4.95 (0.09)	7.75 (0.40), 4.66 (0.11)
Laplace	LS	34.77 (0.90), 0.89 (0.05)	6.60 (0.72), 1.55 (0.07)	0.85 (0.13), 2.12 (0.08)
	QR(0.50)	35.76 (1.28), 0.77 (0.06)	1.04 (0.10), 1.68 (0.07)	2.86 (0.53), 1.39 (0.07)
	QR(0.75)	33.92 (1.22), 1.19 (0.07)	1.97 (0.16), 2.32 (0.09)	3.80 (0.47), 1.88 (0.08)
$\sqrt{2} \times t_4$	LS	34.02 (1.01), 1.64 (0.08)	10.81 (0.89), 2.17 (0.09)	1.78 (0.27), 3.10 (0.09)
	QR(0.50)	38.86 (1.61), 1.71 (0.10)	2.80 (0.18), 2.81 (0.08)	4.09 (0.55), 2.50 (0.08)
	QR(0.75)	34.06 (1.29), 2.09 (0.10)	3.59 (0.21), 3.10 (0.08)	5.95 (0.71), 2.90 (0.09)
Cauchy	LS	17.01 (2.41), 6.61 (0.08)	16.47 (2.97), 6.62 (0.08)	55.88 (5.62), 6.22 (0.10)
	QR(0.50)	40.24 (2.29), 1.54 (0.09)	3.05 (0.30), 2.65 (0.09)	7.41 (3.33), 2.52 (0.10)
	QR(0.75)	41.00 (4.63), 2.97 (0.11)	5.09 (0.23), 4.05 (0.10)	10.78 (2.95), 3.74 (0.09)

expressions of all other 18,975 genes. We first standardize the 18,975 gene expressions and select the 3,000 probes with the largest variances. Those 3,000 expressions are then analyzed on a logarithmic scale with base two.

In our analysis, we also conduct timing comparison on the processed data aforementioned ($n = 120$, $p = 3000$) among `quantreg`, `hqreg` and `FHDQR` for the lasso penalized quantile regression with $\tau = 0.25,\ 0.50$ and 0.75. The timings are reported in Table 3.9 and they demonstrate the efficiency of scdADMM. One possible reason why pADMM takes longer time than scdADMM and `hqreg` is because of the high correlation among gene expressions.

We also fit the lasso penalized quantile regression on the data to select genes that are most relevant to *TRIM32*. Specifically, we first analyze the data on all 120 rats using the lasso penalized quantile regression with quantile indices $\tau = 0.25,\ 0.50$ and 0.75. The tuning parameter is selected using five-fold cross-validation. The number of relevant genes that are selected is reported in the second column of Table 3.10. The difference in the number of selected genes by different quantile indices is a sign of heteroscedasticity in the data, as explained in Wang et al. (2012). We then conduct 50 random partitions on the data. Each partition has 80 rats in the training set and 40 rats in the validation set. We apply the lasso penalized quantile regression to the training set using five-fold cross-validation and evaluate its prediction error on the validation set by calculating $(1/40)\sum_{i\in\text{validation}}\rho_\tau(y_i-\hat{\beta}_0-\mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}})$. The average number of selected genes and prediction errors over the 50 partitions are reported in the third and fourth columns of Table 3.10. We observe that the genes selected by $\tau = 0.25$ and 0.75 are fewer than those by $\tau = 0.5$. This agrees with the observation we made from the fit on the full data.

Table 3.9: Timings (in seconds) for running lasso penalized quantile regression (with $\tau = 0.25,\ 0.50$ and 0.75) on the microarray data reported in Scheetz et al. (2006) over one hundred λ values by `quantreg` (5000+: above 5000 seconds), scdADMM, pADMM and `hqreg`. All timings reported are averaged over three runs.

τ	0.25	0.50	0.75
quantreg	5000+	5000+	5000+
hqreg	4.97	4.09	4.56
pADMM	351.93	401.76	347.89
scdADMM	1.68	1.15	1.09

Table 3.10: Analysis of the microarray data reported in Scheetz et al. (2006) by lasso penalized quantile regression with the `FHDQR` package. The number of genes selected and prediction errors are averaged over 50 runs for the random partition columns. Numbers in the parentheses are standard errors of their corresponding averages.

	All data	Random partition	
τ	#genes	Ave. #genes	Prediction error
0.25	14	15.00 (1.26)	0.0351 (0.0014)
0.50	23	24.16 (2.38)	0.0395 (0.0010)
0.75	14	11.22 (1.07)	0.0671 (0.0196)

Chapter 4

Ultrahigh-Dimensional Composite Quantile Regression

Composite quantile regression (CQR) provides efficient estimation of the coefficients in linear models, regardless of the error distributions. We consider penalized CQR for both variable selection and efficient coefficient estimation in a linear model under ultrahigh dimensionality and possibly heavy-tailed error distribution. Both lasso and folded concave penalties are discussed. An L_2-risk bound is derived for the lasso estimator to establish its estimation consistency and strong oracle property of the folded concave penalized CQR is shown for a feasible solution via the LLA algorithm. The nonsmooth nature of the penalized CQR poses great numerical challenges for high-dimensional data. To that end, we provide a unified and effective numerical optimization algorithm for computing penalized CQR via ADMM. We demonstrate the superior efficiency of penalized CQR estimator, as compared to the penalized least squares estimator, through simulated data under various error distributions.

4.1 Introduction

Coefficient estimation in linear models is routinely done via the least squares (LS) regression. Under normal error distributions, the LS estimator has the likelihood interpretation and is the

most efficient estimator. It is still reasonably efficient under other light-tailed error distributions besides the normal distribution. However, it is usually less efficient than the maximum likelihood estimator (MLE) that exploits the distributional information. Moreover, when the error distribution exhibits heavy-tailedness, the LS estimator may not even be consistent. Our numerical studies on the LS oracle estimator in Section 4.5 clearly demonstrate this point. Ideally, the MLE is the most desired estimator since it is asymptotically efficient, but one has to impose distributional assumptions in order to write down the likelihood. This may not be feasible in practice. Therefore, we need to consider tractable solutions when distributional assumptions are too restrictive.

As a practically feasible method and a robust alternative to the LS regression, the quantile regression (Koenker and Bassett, 1978) was proposed to consistently estimate coefficients even under heavy-tailed error distributions, such as the Cauchy distribution. Moreover, it relies on very minimal distributional assumptions. The robustness of the quantile regression estimator really comes from the fact that its asymptotic variance does not depend on the moments of the error distribution, upon which the asymptotic variance of the LS estimator relies, however. Therefore, the quantile regression estimator enjoys the asymptotic normality even when the error distribution is heavy-tailed. In terms of efficiency, it is well known that the asymptotic variance of a quantile regression estimator is inversely proportional to the error density evaluated at the true quantile of the error distribution (Knight, 1998; Koenker, 2005). Therefore, under some error distributions, the quantile regression estimator can be more efficient than the LS estimator. For example, the least absolute deviation (LAD) regression estimator is the most efficient under the double exponential error distribution. However, since the quantile regression considers only one quantile at a time, it may not fully grasp the information from a distribution to always produce efficient estimation. As an extreme example, when the error density at a specified quantile approaches zero, the asymptotic variance of the corresponding quantile regression estimator may explode to infinity, which results in an estimator with arbitrarily small efficiency.

To gain efficient coefficient estimation in linear models that is safe against efficiency decay under such error distributions for quantile regression estimators, several methods based on the idea of combining quantile regression across multiple quantiles have been proposed in the literature. The idea is natural: as more quantiles are used, we have more information about the distribution and can hence obtain more efficient estimation. One notable approach by Zhao and Xiao (2014) seeks an optimal weighting scheme to convexly combine a fixed number of quantile regression estimators at given levels to achieve as much efficiency gain as possible. It was shown that as the number of quantiles increases, the asymptotic variance of their proposed estimator can achieve the Cramér–Rao lower bound under some regularity conditions. Another approach called composite quantile regression (CQR) by Zou and Yuan (2008) is to combine the information over different quantiles via the quantile loss function. The relative efficiency of the CQR estimator with respect to the LS estimator is shown to be greater than 70% regardless of the error distribution. Under the normal error distribution, the relative efficiency is as high as 99.5%. Moreover, the CQR estimator can be much more (or arbitrarily more) efficient than the LS estimator under heavy-tailed error distributions.

In high-dimensional linear models where the number of covariates may be huge, in order to obtain meaningful coefficient estimation, many regularization techniques have been proposed to augment regular regressions. Examples include the lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), elastic net (Zou and Hastie, 2005), adaptive lasso (Zou, 2006), and so on. Under certain sparsity assumptions, many penalized methods target the so called oracle property where the penalized estimators are shown to be asymptotically equivalent to the oracle estimators which are obtained as if we knew the underlying sparsity structure of the linear models. For folded concave penalized LS regression, it can be shown that the penalized LS estimator enjoys the oracle property (Fan and Li, 2001; Fan et al., 2014b) under the light-tail assumption of the error distribution. When the error distribution is heavy-tailed, the oracle property of the penalized LS no longer holds. To this end, in

order to estimate the coefficients efficiently even under heavy-tailed error distributions for high-dimensional linear models, we consider the penalized CQR. Our motivation is to target the oracle CQR estimator, which was shown by Zou and Yuan (2008) to enjoy very nice theoretical properties. We point out that the approach by Zhao and Xiao (2014) cannot be easily regularized to obtain desired sparse solutions since their estimator is a convex combination of multiple estimators.

The rest of this chapter is organized as follows. In Section 4.2, we introduce the framework for penalized CQR, followed by a discussion of the theoretical properties of the L_1 and folded concave penalized CQR estimators in Section 4.3. We propose a unified and efficient sparse coordinate descent ADMM algorithm for numerically solving the penalized CQR in Section 4.4. Numerical studies are conducted in Section 4.5 to show the superior finite sample performance of penalized CQR over penalized LS. All proofs are relegated to Section 4.6.

4.2 Penalized Composite Quantile Regression

We consider the problem of variable selection and coefficient estimation in the following linear model

$$y = \beta_0 + \sum_{j=1}^{p} x_j \beta_j + \varepsilon, \tag{4.1}$$

where ε is independent of $\mathbf{x} = (x_1, \ldots, x_p)^{\mathrm{T}}$. Suppose β_0^* and $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_p^*)^{\mathrm{T}}$ are the true coefficients in model (4.1) that generates the data $(\mathbf{x}_i, y_i)_{i=1}^n$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$. We will denote the response vector by $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ and the design matrix by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^{\mathrm{T}}$. The design matrix may also be denoted as $\mathbf{X} = (X_1, \ldots, X_p)$ in terms of the covariates. Let $\mathbb{X} = (\mathbf{1}_n, \mathbf{X})$ be the augmented design matrix by adding the column corresponding to an intercept term, where $\mathbf{1}_n$ is the n-dimensional vector of all ones.

We will denote $X_0 = \mathbf{1}_n$ so that $\mathbb{X} = (X_0, X_1, \ldots, X_p)$.

In the high-dimensional regime, the number of parameters p can be much greater than the number of observations n. However, we will assume that many components in $\boldsymbol{\beta}^*$ are effectively zero, so that $\boldsymbol{\beta}^*$ is sparse. Specifically, let $\mathscr{A} = \{1 \le j \le p : \beta_j^* \neq 0\}$ be the active set of $\boldsymbol{\beta}^*$. It is usually assumed that the effective dimension $s = |\mathscr{A}|$ is less than n.

We consider the penalized CQR as an effective way of estimating the coefficient vector $\boldsymbol{\beta}$ in model (4.1). Assume that the random error ε has cumulative distribution F and probability density function f. Given an ordered sequence of quantile levels $0 < \tau_1 < \tau_2 < \cdots < \tau_K < 1$, let $\alpha_k^* = \beta_0^* + F^{-1}(\tau_k)$, where $F^{-1}(\tau_k)$ denotes the τ_kth quantile of ε, $1 \le k \le K$. In this chapter, we assume $K > 1$ is a fixed integer. In practice, one often sets $K = 9$ or $K = 19$. Without loss of generality, we assume $\beta_0^* = 0$. Without penalization, the regular composite quantile regression estimates the coefficient vector $\boldsymbol{\beta}$ by minimizing

$$\sum_{i=1}^{n} \sum_{k=1}^{K} \rho_{\tau_k}(y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta})$$

jointly over $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K) \in \mathbb{R}^K$ and $\boldsymbol{\beta} \in \mathbb{R}^p$, where $\rho_{\tau_k}(u) = \{\tau_k - I(u < 0)\}u$ denotes the check loss for $k = 1, \ldots, K$. For more details about the regular CQR, the readers are referred to the article by Zou and Yuan (2008). In the remainder of this chapter, we will be focusing on the penalized CQR.

4.3 Theory

In this section, we show the theoretical properties of the sparse penalized CQR. Specifically, we consider sparse penalized CQR with both lasso and folded concave penalties. For lasso penalized CQR, we show the estimation consistency in terms of an L_2 error bound on the lasso estimator. We also prove the strong oracle property (Fan and Lv, 2011; Fan et al., 2014b) for a feasible solution of the folded concave penalized CQR. For ease of exposition,

we introduce the following notation.

Notation. For $u \in \mathbb{R}$, let $u_+ = uI(u > 0)$ and $u_- = -uI(u < 0)$ be the positive and negative parts of u, respectively. Moreover, let $\text{sgn}(u) = I(u > 0) - I(u < 0)$ be the sign function. The largest and smallest eigenvalues of a square matrix $\mathbf{A}$ are denoted respectively by $\Lambda_{\max}(\mathbf{A})$ and $\Lambda_{\min}(\mathbf{A})$. We also let ∂f be the subdifferential of a convex function f. For two matrices $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{m\times n}$, we let $\langle \mathbf{A}_1, \mathbf{A}_2 \rangle = \text{tr}(\mathbf{A}_1^{\mathrm{T}}\mathbf{A}_2)$ be the inner product and $\|\mathbf{A}_1\|_F = \langle \mathbf{A}_1, \mathbf{A}_1 \rangle^{1/2}$ be the Frobenius norm. For any vector $\mathbf{v} = (v_1, \ldots, v_p)^{\mathrm{T}} \in \mathbb{R}^p$ and an arbitrary index set $I \subset \{1, \ldots, p\}$, we write $\mathbf{v}_I = (v_j, j \in I)^{\mathrm{T}}$ and denote by $\mathbf{X}_I = (\mathbf{x}_j, j \in I)$ the submatrix consisting of the columns of $\mathbf{X}$ with indices in I. The complement of I is denoted by $I^{\mathrm{c}} = \{1, \ldots, p\}\backslash I$. For $q \in [1, \infty]$, the L_q-norm of $\mathbf{v}$ is denoted by $\|\mathbf{v}\|_q$.

4.3.1 L_1-penalized composite quantile regression

For $\lambda > 0$, define the L_1-penalized CQR estimator

$$(\widehat{\boldsymbol{\alpha}}_\lambda, \widehat{\boldsymbol{\beta}}_\lambda) := \arg\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{4.2}$$

where $Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta}) = (nK)^{-1} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k}(y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})$. In the sequel, we call $(\widehat{\boldsymbol{\alpha}}_\lambda, \widehat{\boldsymbol{\beta}}_\lambda)$ the CQR lasso estimator for short. Moreover, define the restricted set $\mathscr{C} = \big\{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathbb{R}^K \times \mathbb{R}^p : \|\boldsymbol{\Delta}_{\mathscr{A}^{\mathrm{c}}}\|_1 \le 3\|\boldsymbol{\Delta}_{\mathscr{A}}\|_1 + \|\boldsymbol{\delta}\|_1\big\}$.

For $\boldsymbol{\Delta} \in \mathbb{R}^p$ and integer $m \ge 0$, let $\overline{\mathscr{A}}(\boldsymbol{\Delta}, m) \subset \mathscr{A}^{\mathrm{c}}$ be the support of the m largest in absolute value components of $\boldsymbol{\Delta}_{\mathscr{A}^{\mathrm{c}}}$. When $m = 0$, we take $\overline{\mathscr{A}}(\boldsymbol{\Delta}, m) = \emptyset$. The following assumption is imposed on the data and the error distribution:

C0. *(Sampling and smoothness).* The observations $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$ are i.i.d. with $\min(n, p) \ge 3$. The density function satisfies that $f(u) \le \bar{f} < \infty$ for all u in the support of ε and that f is continuously differentiable and $f'(u) \le \bar{f}' \in (0, \infty)$ for all

u in the support of ε. Moreover, there exists a constant $\mathcal{U}_0 > 0$ such that $f(\alpha_k^* + u) \geq \underline{f} > 0$ for all $1 \leq k \leq K$ and $|u| \leq \mathcal{U}_0$. Also, $(\mathbf{x}_{i\mathscr{A}}, y_i)$, $i = 1, \ldots, n$ are in general positions (Koenker, 2005, Section 2.2) and there is at least one continuous covariate in the true model.

Note that we do not impose any moment or light-tail assumptions on the error distribution and the assumptions on the error density are mild and can be satisfied by many commonly seen distributions, including heavy-tailed distributions like Cauchy. We will also assume that $\bar{f}$, $\bar{f}'$ and $\underline{f}$ are all positive constants. The assumptions on $(\mathbf{x}_{i\mathscr{A}}, y_i)$'s make sure the CQR oracle estimator is unique. This is a fairly common assumption in quantile regression (see Koenker, 2005). More discussions of the CQR oracle estimator can be found in Appendix C.

To establish the estimation consistency of the CQR lasso estimator, we additionally assume two conditions. For the sake of brevity, we only consider fixed design.

C1. *(Restricted identifiability).* The design matrix $\mathbf{X}$ satisfies

$$\kappa_m = \min_{1\leq k\leq K} \inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in\mathscr{C},(\boldsymbol{\delta},\boldsymbol{\Delta})\neq\mathbf{0}} \frac{\sum_{i=1}^n (\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2}{n(\|\boldsymbol{\Delta}_{\mathscr{A}\cup\overline{\mathscr{A}}(\boldsymbol{\Delta},m)}\|_2^2 + \delta_k^2)} > 0.$$

C2. *(Restricted nonlinearity).* The design matrix $\mathbf{X}$ satisfies

$$q = \frac{3}{8}\frac{\underline{f}^{3/2}}{\bar{f}'} \inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in\mathscr{C},(\boldsymbol{\delta},\boldsymbol{\Delta})\neq\mathbf{0}} \frac{\left[n^{-1}\sum_{i=1}^n \sum_{k=1}^K (\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2\right]^{3/2}}{n^{-1}\sum_{i=1}^n \sum_{k=1}^K |\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}|^3} > 0.$$

Condition (C1) is an extension of the restricted identifiability property (RIP), also known as the restricted eigenvalue (RE) condition, to the case of the penalized composite quantile regression. RIP is often assumed in the literature for sparse penalized regressions, such as the penalized least squares, Dantzig selector (Candes and Tao, 2007; Bickel et al., 2009) and the penalized quantile regression (Belloni and Chernozhukov, 2011). The restricted nonlinearity assumption in Condition (C2) is similar to the one in Belloni and Chernozhukov

(2011). The quantity q, referred to as the restricted nonlinear impact (RNI) coefficient by Belloni and Chernozhukov (2011), describes how well the composite quantile regression empirical loss function can be minorized by a quadratic function over the restricted set.

Theorem 4.1

Under conditions (C0), (C1) and (C2), with probability at least $1 - p_1(\lambda)$, where

$$p_1(\lambda) = 2K \exp\Big(-\frac{1}{2} n K^2 \lambda^2\Big) + 2p \exp\Big(-\frac{n\lambda^2}{2M_0}\Big) + \exp\Big[-16 M_0 \frac{s(1+\log p)}{\kappa_0}\Big],$$

the lasso estimator $(\widehat{\boldsymbol{\alpha}}_\lambda, \widehat{\boldsymbol{\beta}}_\lambda)$ of the composite quantile regression satisfies

$$\|\widehat{\boldsymbol{\alpha}}_\lambda - \boldsymbol{\alpha}^*\|_2 \le \frac{8}{\underline{f}} \sqrt{\frac{K}{\kappa_m}} \Bigg[8 \sqrt{\frac{2M_0}{\kappa_0}} \sqrt{\frac{1+\log p}{n}} (4\sqrt{s} + K + 1) + \lambda \sqrt{\frac{s}{\kappa_0}} \Bigg]$$

and

$$\begin{aligned} \|\widehat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}^*\|_2 \le{}& \frac{8}{\underline{f}\sqrt{\kappa_m}} \sqrt{1 + \frac{18s}{m} + \frac{2K^2}{m}} \\ & \cdot \Bigg[8 \sqrt{\frac{2M_0}{\kappa_0}} \sqrt{\frac{1+\log p}{n}} (4\sqrt{s} + K + 1) + \lambda \sqrt{\frac{s}{\kappa_0}} \Bigg], \end{aligned}$$

provided that the growth condition

$$16 \sqrt{\frac{2M_0}{\kappa_0}} \sqrt{\frac{1+\log p}{n}} (4\sqrt{s} + K + 1) + 2\lambda \sqrt{\frac{s}{\kappa_0}} \le q \sqrt{\frac{\underline{f}}{K}}.$$

holds, where $M_0 = \max_{0 \le j \le p} \|X_j\|_2^2 / n$. □

Remark 4.1 By Theorem 4.1, a typical choice of the tuning parameter is $\lambda = 2\sqrt{M_0 \log p / n}$.

With such choice of λ, we can see that

$$\|\widehat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}^*\|_2 = \mathcal{O}_P\left(\frac{1}{\sqrt{\kappa_0 \kappa_s}}\sqrt{\frac{s\log p}{n}}\right)$$

provided that s satisfies $q^{-1}\sqrt{s\log p/(n\kappa_0)} = o(1)$ and $\kappa_0 = o(s\log p)$. When κ_0 and κ_s are bounded away from zero, then the CQR lasso estimator achieves the near-optimal rate $\sqrt{s\log p/n}$, which implies that p can be of exponential order of n, i.e., $\log p = \mathcal{O}(n^\gamma)$ for some $0 < \gamma < 1$. □

4.3.2 Folded concave penalized composite quantile regression

Folded concave penalized regression has been widely adopted in the statistical analysis of high-dimensional data due to its strong oracle optimality (Fan and Lv, 2011; Fan et al., 2014b). In order to establish the oracle property of folded concave penalized CQR estimators, let us first define the oracle estimator of the composite quantile regression,

$$(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}) := \underset{\boldsymbol{\alpha}, \boldsymbol{\beta}:\boldsymbol{\beta}_{\mathscr{A}^{\mathrm{c}}}=\mathbf{0}}{\arg\min} \sum_{i=1}^{n}\sum_{k=1}^{K} \rho_{\tau_k}(y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}). \tag{4.3}$$

The oracle estimator $(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}})$ is not feasible in practice since $\mathscr{A}$ is unknown, but it serves as a benchmark estimator to which one can compare the penalized CQR estimator. In the following, we only show the rate of convergence of the CQR oracle estimator. More asymptotic properties of the CQR oracle can be found in Zou and Yuan (2008) and some numerical properties of the CQR oracle are shown in Appendix C.

Let $\mathscr{A}_0 = \{0\} \cup \mathscr{A}$ and $\mathbb{X}_{\mathscr{A}_0} = (\mathbf{1}_n, \mathbf{X}_{\mathscr{A}})$. Denote $\underline{\mu} = \Lambda_{\min}(n^{-1}\mathbb{X}_{\mathscr{A}_0}^{\mathrm{T}}\mathbb{X}_{\mathscr{A}_0})$ and $\overline{\mu} = \Lambda_{\max}(n^{-1}\mathbb{X}_{\mathscr{A}_0}^{\mathrm{T}}\mathbb{X}_{\mathscr{A}_0})$. Moreover, let $M_{\mathscr{A}} = \max_{1\le i\le n}(s+1)^{-1}\big(1 + \|\mathbf{x}_{i\mathscr{A}}\|_2^2\big)$ and $M_{\mathscr{A}^{\mathrm{c}}} = \max_{1\le i\le n,\, j\in\mathscr{A}^{\mathrm{c}}}|x_{ij}|$. In this chapter, we will assume $M_{\mathscr{A}}$ and $M_{\mathscr{A}^{\mathrm{c}}}$ are positive constants.

Lemma 4.1

Under conditions (C0) – (C3) and the assumptions that $(s+1)/(\underline{\mu}\sqrt{n}) = \mathcal{O}(1)$ and that $\underline{\mu}/(s+1) = o(1)$, the CQR oracle estimator satisfies

$$\|\widehat{\boldsymbol{\alpha}}^{\mathrm{o}} - \boldsymbol{\alpha}^*\|_2 = \mathcal{O}_P\big(\underline{\mu}^{-1}\sqrt{(s+1)/n}\big),$$
$$\|\widehat{\boldsymbol{\beta}}^{\mathrm{o}} - \boldsymbol{\beta}^*\|_2 = \mathcal{O}_P\big(\underline{\mu}^{-1}\sqrt{(s+1)/n}\big)$$

as $n \to \infty$. □

The folded concave penalized CQR targets the oracle estimator even if $\mathscr{A}$ is unknown. Specifically, the folded concave penalized CQR at penalty level $\lambda > 0$ solves the following minimization problem

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} Q_n(\boldsymbol{\alpha},\boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(|\beta_j|), \tag{4.4}$$

where $p_\lambda(t)$, $t \geq 0$ belongs to a class of folded concave penalties satisfying the following properties:

(P1) $p_\lambda(t)$ is nondecreasing and concave in $t \geq 0$ and $p_\lambda(0) = 0$;

(P2) $p_\lambda(t)$ is differentiable in $t > 0$;

(P3) $p'_\lambda(t) \geq a_1\lambda$, $0 < t \leq a_2\lambda$ and $p'_\lambda(0) := p'_\lambda(0+) \geq a_1\lambda$, where $a_1, a_2 > 0$ are fixed constants;

(P4) $p'_\lambda(t) = 0$, $t \geq a\lambda$ for a fixed constant $a > a_2$.

It has been shown that both the SCAD penalty (Fan and Li, 2001) and MCP (Zhang, 2010) belong to this class; see, for instance, Lv and Fan (2009). To establish the strong oracle optimality, Fan et al. (2014b) proposed to use the local linear approximation (LLA) algorithm (Zou and Li, 2008) for solving the folded concave penalized CQR. The LLA algorithm is shown in Algorithm 7.

Algorithm 7: The local linear approximation (LLA) algorithm for solving the folded concave penalized composite quantile regression

1. Initialize $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with respectively $\widehat{\boldsymbol{\alpha}}^{(0)}$ and $\widehat{\boldsymbol{\beta}}^{(0)}$. Compute weights

$$\hat{w}_j^{(0)} = p'_\lambda(|\hat{\beta}_j^{(0)}|),\ j = 1, \ldots, p.$$

2. For $m = 1, 2, \ldots, M$, repeat the LLA iteration in (2.a) and (2.b).

   (2.a) Solve the following convex optimization problem for $\widehat{\boldsymbol{\alpha}}^{(m)}$ and $\widehat{\boldsymbol{\beta}}^{(m)}$

$$(\widehat{\boldsymbol{\alpha}}^{(m)}, \widehat{\boldsymbol{\beta}}^{(m)}) := \underset{\boldsymbol{\alpha}, \boldsymbol{\beta}}{\arg\min}\ Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta}) + \sum_{j=1}^{p} \hat{w}_j^{(m-1)} |\beta_j|.$$

   (2.b) Calculate the weights

$$\hat{w}_j^{(m)} = p'_\lambda(|\hat{\beta}_j^{(m)}|),\ j = 1, \ldots, p.$$

An advantage of the LLA algorithm is that it can find the oracle estimator in a few iterations with high probability under mild conditions. Specifically, assume that the true coefficient vector $\boldsymbol{\beta}^*$ exhibits sufficient signal

(C3) $\min_{j \in \mathscr{A}} |\beta_j^*| > (a+1)\lambda$.

Theorem 4.2

Suppose the folded concave penalized CQR (4.4) is solved with LLA (Algorithm 7) that is initialized with the CQR lasso estimator (4.2) at penalty level $\lambda_0 > 0$. Let $r_0 = \min_{j \in \mathscr{A}} |\beta_j^*| - a\lambda$ and $r_* = \sqrt{(s+1) M_{\mathscr{A}} \log n / n}$. Assume the folded concave penalty $p_\lambda(\cdot)$ satisfies properties (P1) – (P4), where λ is taken to satisfy

$$\begin{aligned} \lambda \geq & \frac{8}{a_0 \underline{f} \sqrt{\kappa_m}} \sqrt{1 + \frac{18s}{m} + \frac{2K^2}{m}} \\ & \cdot \left[8 \sqrt{\frac{2M_0}{\kappa_0}} \sqrt{\frac{1 + \log p}{n}} (4\sqrt{s} + K + 1) + \lambda_0 \sqrt{\frac{s}{\kappa_0}} \right]. \end{aligned} \tag{4.5}$$

Under conditions (C0) – (C3) and the assumptions that $r_0\sqrt{(s+1)M_{\mathscr{A}}} \leq \mathcal{U}_0$ and $\lambda > 8K\underline{f}^{-1}\underline{\mu}^{-1}\sqrt{(s+1)M_{\mathscr{A}}/n}$, with probability at least $1-p_1(\lambda_0)-p_2(r_0)-p_2(r_*)-2(p-s)\exp(-2nB^2/M_0)-p_3$, the LLA algorithm converges to the oracle estimator $(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}})$ in two iterations, where $p_1(\cdot)$ is given in Theorem 4.1, $p_2(\cdot)$ is defined as

$$p_2(r) = \exp\left\{-\frac{n(t(r))^2}{32\overline{\mu}r^2}\right\}, \text{ where } t(r) = \frac{\underline{f}}{4K}\underline{\mu}r^2 - 2r\sqrt{\frac{(s+1)M_{\mathscr{A}}}{n}},$$

and

$$\begin{aligned} p_3 = {} & 2(p-s)n^{2(K+s)}\exp\left(-\frac{3nB^2}{24\bar{f}M_{\mathscr{A}^{\mathrm{c}}}^2\overline{\mu}^{1/2}r_* + 8M_{\mathscr{A}^{\mathrm{c}}}B}\right) \\ & + 2(p-s)n^{2(K+s)}\exp\left(-\frac{3nB_0^2}{24\bar{f}M_{\mathscr{A}^{\mathrm{c}}}^2M_{\mathscr{A}}^{1/2}(s+1)^{1/2}n^{-2}r_* + 4M_{\mathscr{A}^{\mathrm{c}}}B_0}\right) \end{aligned}$$

with $B = \frac{1}{2}\big[a_1\lambda - \frac{K+s}{n}M_{\mathscr{A}^{\mathrm{c}}} - \bar{f}M_{\mathscr{A}^{\mathrm{c}}}\overline{\mu}^{1/2}r_*\big]$ and $B_0 = \big[\frac{B}{2} - \frac{8\bar{f}\sqrt{s+1}}{n^2}M_{\mathscr{A}^{\mathrm{c}}}M_{\mathscr{A}}^{1/2}r_*\big]_+$. □

Remark 4.2 With the choice $\lambda_0 = 2\sqrt{M_0\log p/n}$ and $\lambda = C(\underline{f}\sqrt{\kappa_0\kappa_s})^{-1}\sqrt{s\log p/n}$ for some constant $C > 0$ such that (4.5) is satisfied, the probability lower bound will approach one as $n \to \infty$ provided that $(\sqrt{\kappa_0\kappa_s})^{-1}\sqrt{s^2\log p/n} = \mathcal{O}(1)$, $\sqrt{\kappa_0\kappa_s}/(\underline{\mu}\sqrt{\log p}) = o(1)$, $(\underline{\mu}\log n)^{-1} = o(1)$, $\kappa_0 = o(s\log p)$, $\sqrt{\kappa_0\kappa_s}\sqrt{\log p/(ns)} = o(1)$, $\sqrt{\kappa_0\kappa_s}\sqrt{s/(n\log p)} = o(1)$, and $\sqrt{\underline{\mu}\kappa_0\kappa_s}\sqrt{\log n/\log p} = o(1)$. It can be see that when κ_0, κ_s and $\underline{\mu}$ are bounded away from zero, with proper choice of the orders of s and p, the folded concave penalized CQR estimator enjoys the strong oracle property even under ultrahigh dimensionality, $\log p = \mathcal{O}(n^\gamma)$ for some $0 < \gamma < 1$. □

4.4 Numerical Optimization

In this section, we propose an efficient numerical algorithm for solving the penalized composite quantile regression. The algorithm is based on the alternating direction method

of multipliers (ADMM) and the coordinate descent algorithm. Before introducing the algorithm, we note that both the L_1-penalized and folded concave penalized composite quantile regression can be solved with one or more applications of the following weighted L_1-penalized composite quantile regression:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k}(y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \lambda_1 \sum_{j=1}^{p} d_j |\beta_j|, \tag{4.6}$$

where $\lambda_1 > 0$ and $d_j \geq 0$, $j = 1, \ldots, p$. Specifically, the CQR lasso estimator can be achieved by letting $d_j = 1$ for all $j = 1, \ldots, p$, while by Algorithm 7, the folded concave penalized CQR estimator can be obtained by iteratively minimizing (4.6) with $d_j = \hat{w}_j^{(m-1)}$ in the mth iteration. As a result, in the sequel, the ADMM algorithm will be discussed in term of the minimization of (4.6).

Our proposal of using the ADMM algorithm is motivated from the fact that both the check loss $\rho_\tau(u) = [\tau - I(u < 0)]u$, $\tau \in (0, 1)$ and the L_1-norm are non-smooth. This makes it difficult to optimize over them jointly via gradient-based methods. The ADMM algorithm, however, tackles the two non-smooth functions one at a time, through the proximity operators, and splits the optimization problem into sub-problems that are easy to solve. Thus, it can effectively optimize over the highly non-smooth objective function in (4.6). To elaborate on the ADMM algorithm, let $z_{ik} = y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}$, $i = 1, \ldots, n$, $k = 1, \ldots, K$. Next, define matrix $\mathbf{Z} = (z_{ik})_{n\times K}$ in terms of z_{ik}'s and denote the kth column of $\mathbf{Z}$ by Z_k, $k = 1, \ldots, K$. By convexity, it can be seen immediately that minimization problem (4.6) can be recast as

$$\begin{aligned} &\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k}(z_{ik}) + \lambda_1 \sum_{j=1}^{p} d_j |\beta_j| \\ &\text{subject to} \quad \mathbf{1}_n \otimes \boldsymbol{\alpha}^{\mathrm{T}} + \mathbf{1}_K^{\mathrm{T}} \otimes (\mathbf{X}\boldsymbol{\beta}) + \mathbf{Z} - \mathbf{1}_K^{\mathrm{T}} \otimes \mathbf{y} = \mathbf{0}, \end{aligned} \tag{4.7}$$

where $\otimes$ denotes the Kronecker product. For ease of notation, let $\mathbf{Y} = \mathbf{1}_K^{\mathrm{T}} \otimes \mathbf{y}$, $\mathbb{X} = (\mathbf{1}_n, \mathbf{X})$

and $\mathbf{\Phi} = \begin{pmatrix} \boldsymbol{\alpha}^{\mathrm{T}} \\ \mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{\beta} \end{pmatrix}$. The constraint in (4.7) can be then rewritten as $\mathbb{X}\mathbf{\Phi} + \mathbf{Z} - \mathbf{Y} = \mathbf{0}$, and therefore, (4.6) can be equivalently solved by the following constrained convex optimization problem

$$\begin{aligned} &\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k}(z_{ik}) + \lambda_1 \sum_{j=1}^{p} d_j |\beta_j| \\ &\text{subject to} \quad \mathbb{X}\mathbf{\Phi} + \mathbf{Z} - \mathbf{Y} = \mathbf{0}. \end{aligned} \tag{4.8}$$

Now introduce the augmented Lagrangian of problem (4.8)

$$\begin{aligned} L_\sigma(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{Z}, \mathbf{\Theta}) := &\frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k}(z_{ik}) + \lambda_1 \sum_{j=1}^{p} d_j |\beta_j| \\ &- \langle \mathbf{\Theta}, \mathbb{X}\mathbf{\Phi} + \mathbf{Z} - \mathbf{Y} \rangle + \frac{\sigma}{2} \|\mathbb{X}\mathbf{\Phi} + \mathbf{Z} - \mathbf{Y}\|_F^2, \end{aligned}$$

where $\mathbf{\Theta} = (\theta_{ik})_{n \times K}$ is the Lagrangian multiplier and $\sigma > 0$ is the parameter for the augmented quadratic term. Similarly, we will denote the kth column of $\mathbf{\Theta}$ by Θ_k, $k = 1, \ldots, K$. To apply the ADMM algorithm to problem (4.8), let $\boldsymbol{\alpha}^r$, $\boldsymbol{\beta}^r$, $\mathbf{Z}^r$, and $\mathbf{\Theta}^r$ be the iterates after the rth iteration of the algorithm. The algorithm updates the parameters in the $(r+1)$th iteration as follows

$$\begin{cases} (\boldsymbol{\alpha}^{r+1}, \boldsymbol{\beta}^{r+1}) := \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} L_\sigma(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{Z}^r, \mathbf{\Theta}^r) \\ \mathbf{Z}^{r+1} := \arg\min_{\mathbf{Z}} L_\sigma(\boldsymbol{\alpha}^{r+1}, \boldsymbol{\beta}^{r+1}, \mathbf{Z}, \mathbf{\Theta}^r), \\ \mathbf{\Theta}^{r+1} := \mathbf{\Theta}^r - \sigma[\mathbf{1}_n \otimes (\boldsymbol{\alpha}^{r+1})^{\mathrm{T}} + \mathbf{1}_K^{\mathrm{T}} \otimes (\mathbf{X}\boldsymbol{\beta}^{r+1}) + \mathbf{Z}^{r+1} - \mathbf{1}_K^{\mathrm{T}} \otimes \mathbf{y}], \end{cases} \tag{4.9}$$

More specifically, the updates in (4.9) can be formulated as

$$\begin{cases} (\boldsymbol{\alpha}^{r+1}, \boldsymbol{\beta}^{r+1}) = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \lambda_1 \sum_{j=1}^{p} d_j |\beta_j| - \sum_{k=1}^{K} \mathbf{1}_n^{\mathrm{T}} \Theta_k^r \alpha_k - \boldsymbol{\beta}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \sum_{k=1}^{K} \Theta_k^r \\ \qquad\qquad + \frac{\sigma}{2} \sum_{k=1}^{K} \|\alpha_k \mathbf{1}_n + \mathbf{X}\boldsymbol{\beta} + Z_k^r - \mathbf{y}\|_2^2 \\ \mathbf{Z}^{r+1} = \arg\min_{\mathbf{Z}} \sum_{i=1}^{n} \sum_{k=1}^{K} \big[\frac{1}{nK} \rho_{\tau_k}(z_{ik}) - \theta_{ik}^r z_{ik} + \frac{\sigma}{2} (\alpha_k^{r+1} + \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}^{r+1} + z_{ik} - y_i)^2 \big] \\ \boldsymbol{\Theta}^{r+1} = \boldsymbol{\Theta}^r - \sigma[\mathbf{1}_n \otimes (\boldsymbol{\alpha}^{r+1})^{\mathrm{T}} + \mathbf{1}_K^{\mathrm{T}} \otimes (\mathbf{X}\boldsymbol{\beta}^{r+1}) + \mathbf{Z}^{r+1} - \mathbf{1}_K^{\mathrm{T}} \otimes \mathbf{y}]. \end{cases}$$

Note that the update of $\mathbf{Z}^{r+1}$ can be carried out component-wisely. To be specific, by Lemma 3.1, we can obtain

$$z_{ik}^{r+1} = \mathrm{Prox}_{\rho_{\tau_k}} \left(\frac{\theta_{ik}^r}{\sigma} + y_i - \alpha_k^{r+1} - \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}^{r+1}, \, nK\sigma \right), \; 1 \le i \le n, \; 1 \le k \le K.$$

To update $\boldsymbol{\alpha}^{r+1}$ and $\boldsymbol{\beta}^{r+1}$, we propose to use the coordinate descent algorithm. We call the resulting algorithm the sparse coordinate descent ADMM (scdADMM) algorithm. The details of the scdADMM algorithm for solving the weighted L_1-penalized quantile regression is summarized in Algorithm 8.

Algorithm 8: scdADMM – Sparse coordinate descent ADMM algorithm for solving the weighted L_1-penalized composite quantile regression.

1. Initialize the algorithm with $(\boldsymbol{\alpha}^0, \boldsymbol{\beta}^0, \mathbf{Z}^0, \boldsymbol{\Theta}^0)$.

2. For $r = 0, 1, 2, \ldots$, repeat steps (2.1) – (2.3) until the convergence criterion is met.

   (2.1) Carry out the coordinate descent steps (2.1.1) – (2.1.3).

   (2.1.1) Initialize $\boldsymbol{\alpha}^{r,0} = \boldsymbol{\alpha}^r$ and $\boldsymbol{\beta}^{r,0} = \boldsymbol{\beta}^r$.

   (2.1.2) For $m = 0, 1, 2, \ldots$, repeat steps (2.1.2.1) – (2.1.2.2) until convergence.

   (2.1.2.1) For $k = 1, \ldots, K$, update

$$\alpha_k^{r,m+1} \leftarrow (n\sigma)^{-1}\mathbf{1}^{\mathrm{T}}[\Theta_k^r + \sigma(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{r,m} - Z_k^r)].$$

   (2.1.2.2) For $j = 1, \ldots, p$, update

$$\beta_j^{r,m+1} \leftarrow \frac{\text{Shrink}\left[X_j^{\mathrm{T}} \sum_{k=1}^K \left\{\sigma\left(\mathbf{y} - \alpha_k^{r,m+1}\mathbf{1} - \sum_{t\neq j} X_t \beta_t^{r,m+I(t<j)} - Z_k^r\right) + \Theta_k^r\right\},\ \lambda_1 w_j\right]}{K\sigma\|X_j\|_2^2}.$$

   (2.1.3) Set $\boldsymbol{\alpha}^{r+1} \leftarrow \boldsymbol{\alpha}^{r,m+1}$ and $\boldsymbol{\beta}^{r+1} \leftarrow \boldsymbol{\beta}^{r,m+1}$.

   (2.2) Update $\mathbf{Z}^{r+1} \leftarrow \left(\text{Prox}_{\rho_{\tau_k}}\left(y_i - \alpha_k^{r+1} - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{r+1} + \frac{\theta_{ik}^r}{\sigma},\ nK\sigma\right)\right)_{1\le i\le n,\, 1\le k\le K}$.

   (2.3) Update $\boldsymbol{\Theta}^{r+1} \leftarrow \left(\theta_{ik}^r - \sigma[\alpha_k^{r+1} + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{r+1} - y_i + z_{ik}^{r+1}]\right)_{1\le i\le n,\, 1\le k\le K}$.

4.5 Numerical Experiments

We conduct Monte Carlo studies to assess the finite sample performance of the proposed method. We will compare the estimators from the penalized least squares, penalized composite quantile regression and the ideal oracle least squares and oracle composite quantile regression. Recall that the oracle estimators are obtained through applying the canonical least squares and composite quantile regression to the true underlying model. We focus on both the model selection and estimation accuracy of those estimators.

Our simulated data are from the linear model

$$y = \beta_0^* + \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}^* + \varepsilon, \tag{4.10}$$

where $\beta_0^* = 0$ and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, 2, \mathbf{0}_{p-5})^{\mathrm{T}}$. The covariates are drawn from the multivariate normal distribution, $\mathbf{x} \sim N_p(\mathbf{0}, \boldsymbol{\Sigma})$, where two different covariance matrices $\boldsymbol{\Sigma} = (0.5^{|i-j|})$ and $\boldsymbol{\Sigma} = (0.8^{|i-j|})$ are considered. For the error distribution, we refer to Zou and Yuan (2008) and consider five different shapes:

(a) the normal distribution, $\varepsilon \sim N(0, 3)$;

(b) the mixture normal distribution (MN), $\varepsilon \sim \sqrt{6} \times \varepsilon^*$, where $\varepsilon^* \sim 0.5N(0, 1) + 0.5N(0, 0.5^6)$;

(c) the mixture double gamma distribution (MDG), $\varepsilon \sim \frac{1}{9}\varepsilon^*$, where $\varepsilon^* \sim f(\varepsilon) = e^{-14} \cdot \frac{1}{2}e^{-|\varepsilon|} + (1 - e^{-14}) \cdot \frac{1}{\Gamma(15)}|\varepsilon|^{14}e^{-|\varepsilon|}$;

(d) the t-distribution with 3 degrees of freedom $\varepsilon \sim t_3$; and,

(e) the Cauchy distribution, $\varepsilon \sim$ Cauchy.

In the simulation study, our training data are composed of n observations $(\mathbf{x}_i, y_i)_{i=1}^n$, independently generated from model (4.10). An independent set of n observations is

also simulated from the same model for parameter tuning of the training model. We evaluate the variable selection performance of the estimated coefficients $\widehat{\boldsymbol{\beta}}$ by the number of false positives FP $= |\hat{A}\backslash A^*|$ and the number of false negatives FN $= |A^*\backslash\hat{A}|$, where $A^* = \{1 \le j \le p : \beta_j^* \ne 0\}$ and $\hat{A} = \{1 \le j \le p : \hat{\beta}_j \ne 0\}$. The estimation accuracy of $\widehat{\boldsymbol{\beta}}$ is measured by the model error $(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)^{\mathrm{T}}\boldsymbol{\Sigma}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)$. Two sets of data dimensions $(n, p) = (100, 600)$ and $(n, p) = (200, 1200)$ are used in our simulations. In all settings, we use $K = 19$ quantile levels $\tau_k = 0.05k$, $k = 1, \ldots, 19$. The simulation results are summarized in Tables 4.1 – 4.4.

It can be seen from the tables that the CQR oracle estimator has very similar model error to the LS estimator under normal error, while is more efficient under the other error distributions. In particular, the model error of the LS estimator is not stable under the Cauchy error. In theory, it can be arbitrarily large. SCAD penalized CQR estimators have very close model errors to the CQR oracle estimator under most error distributions and outperform the penalized LS estimators. In terms of model selection accuracy, the SCAD penalized CQR estimator also outperform all the other penalized estimators.

Table 4.1: Simulation results for model (4.10) with $n = 100$, $p = 600$ and $\mathbf{\Sigma} = (0.5^{|i-j|})$. Numbers listed are averages over 100 independent runs, with standard errors reported in the parentheses

	$N(0, 3)$	MN	MDG	t_3	Cauchy
			Model error		
LS-oracle	0.093	0.093	0.084	0.098	9350.072
	(0.008)	(0.007)	(0.007)	(0.009)	(6837.507)
CQR-oracle	0.105	0.004	0.025	0.047	0.094
	(0.008)	(0.002)	(0.003)	(0.004)	(0.011)
LS-lasso	0.664	0.620	0.588	0.663	18.963
	(0.035)	(0.025)	(0.031)	(0.054)	(1.513)
LS-SCAD	0.671	0.646	0.523	0.578	31.738
	(0.038)	(0.036)	(0.033)	(0.036)	(7.959)
CQR-lasso	0.792	0.272	0.465	0.374	1.672
	(0.041)	(0.029)	(0.034)	(0.022)	(0.144)
CQR-SCAD	0.122	0.006	0.032	0.064	0.438
	(0.019)	(0.002)	(0.004)	(0.006)	(0.098)
			FP, FN		
LS-lasso	16.55, 0	16.91, 0	16.53, 0	15.83, 0	13.37, 1.79
	(1.28), (0)	(0.92), (0)	(1.07), (0)	(1.08), (0)	(2.41), (0.12)
LS-SCAD	18.04, 0	18.00, 0	17.04, 0	16.81, 0	17.11, 1.78
	(1.54), (0)	(1.44), (0)	(1.58), (0)	(1.10), (0)	(2.93), (0.13)
CQR-lasso	15.33, 0	15.29, 0	14.49, 0	12.89, 0	38.75, 0.01
	(0.67), (0)	(0.67), (0)	(0.53), (0)	(0.55), (0)	(2.93), (0.01)
CQR-SCAD	1.68, 0.01	1.62, 0	2.27, 0	2.33, 0	2.18, 0.01
	(0.25), (0.01)	(0.29), (0)	(0.32), (0)	(0.34), (0)	(0.38), (0.01)

Table 4.2: Simulation results for model (4.10) with $n = 100$, $p = 600$ and $\mathbf{\Sigma} = (0.8^{|i-j|})$. Numbers listed are averages over 100 independent runs, with standard errors reported in the parentheses

	$N(0,3)$	MN	MDG	t_3	Cauchy
			Model error		
LS-oracle	0.097	0.092	0.079	0.088	184.788
	(0.008)	(0.008)	(0.006)	(0.008)	(91.476)
CQR-oracle	0.097	0.005	0.023	0.046	0.134
	(0.008)	(0.002)	(0.002)	(0.004)	(0.011)
LS-lasso	0.488	0.493	0.441	0.422	19.649
	(0.025)	(0.028)	(0.021)	(0.031)	(1.623)
LS-SCAD	0.524	0.443	0.422	0.442	27.706
	(0.026)	(0.020)	(0.019)	(0.029)	(5.775)
CQR-lasso	0.498	0.152	0.259	0.269	0.993
	(0.029)	(0.023)	(0.029)	(0.014)	(0.087)
CQR-SCAD	0.123	0.005	0.032	0.060	0.355
	(0.013)	(0.001)	(0.003)	(0.005)	(0.055)
			FP, FN		
LS-lasso	13.06, 0	12.34, 0	11.97, 0	13.59, 0	9.21, 1.60
	(0.85), (0)	(0.87), (0)	(0.91), (0)	(0.99), (0)	(1.77), (0.11)
LS-SCAD	13.89, 0	11.97, 0	11.28, 0	14.04, 0	13.53, 1.62
	(0.90), (0)	(0.78), (0)	(0.64), (0)	(0.99), (0)	(2.27), (0.11)
CQR-lasso	12.15, 0	13.28, 0	12.09, 0	13.06, 0	28.26, 0.03
	(0.68), (0)	(0.62), (0)	(0.57), (0)	(0.66), (0)	(2.46), (0.02)
CQR-SCAD	2.09, 0.02	1.29, 0	1.83, 0	1.97, 0	2.38, 0.06
	(0.40), (0.01)	(0.24), (0)	(0.32), (0)	(0.30), (0)	(0.32), (0.03)

Table 4.3: Simulation results for model (4.10) with $n = 200$, $p = 1200$ and $\mathbf{\Sigma} = (0.5^{|i-j|})$. Numbers listed are averages over 100 independent runs, with standard errors reported in the parentheses

	$N(0,3)$	MN	MDG	t_3	Cauchy
			Model error		
LS-oracle	0.051	0.045	0.041	0.048	1136.066
	(0.004)	(0.004)	(0.003)	(0.004)	(965.520)
CQR-oracle	0.047	0.001	0.011	0.023	0.060
	(0.005)	(0)	(0.001)	(0.002)	(0.005)
LS-lasso	0.340	0.337	0.281	0.284	28.450
	(0.015)	(0.014)	(0.011)	(0.013)	(7.987)
LS-SCAD	0.061	0.061	0.055	0.062	41.685
	(0.006)	(0.005)	(0.005)	(0.005)	(24.654)
CQR-lasso	0.394	0.072	0.180	0.239	0.830
	(0.018)	(0.011)	(0.014)	(0.013)	(0.073)
CQR-SCAD	0.046	0.001	0.011	0.023	0.137
	(0.004)	(0)	(0.001)	(0.002)	(0.030)
			FP, FN		
LS-lasso	19.62, 0	20.09, 0	20.15, 0	19.59, 0	21.41, 1.66
	(1.41), (0)	(1.25), (0)	(1.22), (0)	(1.27), (0)	(3.94), (0.13)
LS-SCAD	5.24, 0	5.76, 0	4.56, 0	6.53, 0	25.49, 1.54
	(1.08), (0)	(0.94), (0)	(0.92), (0)	(1.10), (0)	(4.01), (0.12)
CQR-lasso	18.76, 0	19.05, 0	19.11, 0	18.59, 0	60.56, 0
	(0.95), (0)	(0.77), (0)	(0.78), (0)	(0.95), (0)	(6.35), (0)
CQR-SCAD	2.21, 0	2.29, 0	2.99, 0	2.31, 0	1.50, 0
	(0.30), (0)	(0.48), (0)	(0.44), (0)	(0.36), (0)	(0.30), (0)

Table 4.4: Simulation results for model (4.10) with $n = 200$, $p = 1200$ and $\mathbf{\Sigma} = (0.8^{|i-j|})$. Numbers listed are averages over 100 independent runs, with standard errors reported in the parentheses

	$N(0,3)$	MN	MDG	t_3	Cauchy
	Model error				
LS-oracle	0.042	0.047	0.034	0.046	71435.826
	(0.004)	(0.005)	(0.003)	(0.004)	(68875.639)
CQR-oracle	0.049	0.001	0.011	0.022	0.055
	(0.004)	(0)	(0.001)	(0.002)	(0.005)
LS-lasso	0.252	0.235	0.219	0.208	22.598
	(0.012)	(0.011)	(0.009)	(0.013)	(2.334)
LS-SCAD	0.071	0.073	0.050	0.065	23.726
	(0.006)	(0.007)	(0.004)	(0.009)	(2.856)
CQR-lasso	0.255	0.030	0.099	0.153	0.730
	(0.014)	(0.004)	(0.008)	(0.009)	(0.070)
CQR-SCAD	0.086	0.001	0.023	0.048	0.539
	(0.008)	(0)	(0.003)	(0.004)	(0.099)
	FP, FN				
LS-lasso	14.83, 0	16.17, 0	15.24, 0	14.80, 0	20.44, 1.63
	(1.03), (0)	(1.1), (0)	(1.09), (0)	(1.23), (0)	(3.83), (0.12)
LS-SCAD	6.36, 0	5.68, 0	5.64, 0	4.74, 0.01	15.22, 1.84
	(0.99), (0)	(0.72), (0)	(0.83), (0)	(0.76), (0.01)	(2.92), (0.10)
CQR-lasso	16.86, 0	14.89, 0	15.83, 0	16.47, 0	42.75, 0
	(1.02), (0)	(0.7), (0)	(0.78), (0)	(0.98), (0)	(4.13), (0)
CQR-SCAD	2.19, 0	1.93, 0	2.70, 0	2.59, 0	1.85, 0
	(0.32), (0)	(0.36), (0)	(0.50), (0)	(0.44), (0)	(0.24), (0)

4.6 Proofs

We provide proofs of all previously stated results in this section. For the sake of brevity, some auxiliary results are relegated to the appendix.

Lemma 4.2

Under condition (C0), with probability at least

$$1 - 2K \exp\Big(-\frac{1}{2} n K^2 \lambda^2\Big) - 2p \exp\Big(-\frac{n\lambda^2}{2M_0}\Big),$$

the lasso estimator $(\widehat{\boldsymbol{\alpha}}_\lambda, \widehat{\boldsymbol{\beta}}_\lambda)$ of the composite quantile regression satisfies

$$(\widehat{\boldsymbol{\delta}}^\lambda, \widehat{\boldsymbol{\Delta}}^\lambda) \in \mathscr{C} = \{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathbb{R}^K \times \mathbb{R}^p \colon \|\boldsymbol{\Delta}_{\mathscr{A}^c}\|_1 \le 3\|\boldsymbol{\Delta}_{\mathscr{A}}\|_1 + \|\boldsymbol{\delta}\|_1\},$$

where $\widehat{\boldsymbol{\delta}}^\lambda = \widehat{\boldsymbol{\alpha}}_\lambda - \boldsymbol{\alpha}^*$ and $\widehat{\boldsymbol{\Delta}}^\lambda = \widehat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}^*$. □

Proof 4.1 (Proof of Lemma 4.2)

Let $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_K)^{\mathrm{T}}$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p)^{\mathrm{T}}$, where

$$\zeta_k = -\frac{1}{nK} \sum_{i=1}^{n} \big[\tau_k - I(\varepsilon_i < \alpha_k^*)\big],\ 1 \le k \le K,$$

and

$$\xi_j = -\frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} \big[\tau_k - I(\varepsilon_i < \alpha_k^*)\big] x_{ij},\ 1 \le j \le p.$$

Note that $(\boldsymbol{\zeta}^{\mathrm{T}}, \boldsymbol{\xi}^{\mathrm{T}})^{\mathrm{T}} \in \partial Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$, where the subdifferential is taken with respect to $\boldsymbol{\alpha}$ and

$\boldsymbol{\beta}$. By convexity of $Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and optimality of $(\widehat{\boldsymbol{\alpha}}_\lambda, \widehat{\boldsymbol{\beta}}_\lambda)$, we have

$$\begin{aligned}
0 &\geq Q_n(\widehat{\boldsymbol{\alpha}}_\lambda, \widehat{\boldsymbol{\beta}}_\lambda) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) + \lambda(\|\widehat{\boldsymbol{\beta}}_\lambda\|_1 - \|\boldsymbol{\beta}^*\|_1) \\
&\geq \boldsymbol{\zeta}^{\mathrm{T}}(\widehat{\boldsymbol{\alpha}}_\lambda - \boldsymbol{\alpha}^*) + \boldsymbol{\xi}^{\mathrm{T}}(\widehat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}^*) + \lambda(\|\widehat{\boldsymbol{\beta}}_\lambda\|_1 - \|\boldsymbol{\beta}^*\|_1) \\
&\geq -\|\boldsymbol{\zeta}\|_\infty \cdot \|\widehat{\boldsymbol{\alpha}}_\lambda - \boldsymbol{\alpha}^*\|_1 - \|\boldsymbol{\xi}\|_\infty \cdot \|\widehat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}^*\|_1 \\
&\qquad + \lambda\big(\|\widehat{\boldsymbol{\beta}}_{\lambda,\mathscr{A}^{\mathrm{c}}} - \boldsymbol{\beta}^*_{\mathscr{A}^{\mathrm{c}}}\|_1 - \|\widehat{\boldsymbol{\beta}}_{\lambda,\mathscr{A}} - \boldsymbol{\beta}^*_{\mathscr{A}}\|_1\big),
\end{aligned}$$

which implies that

$$\begin{aligned}
(\lambda - \|\boldsymbol{\xi}\|_\infty)\|\widehat{\boldsymbol{\beta}}_{\lambda,\mathscr{A}^{\mathrm{c}}} - \boldsymbol{\beta}^*_{\mathscr{A}^{\mathrm{c}}}\|_1 \leq (\lambda + \|\boldsymbol{\xi}\|_\infty)\|\widehat{\boldsymbol{\beta}}_{\lambda,\mathscr{A}} - \boldsymbol{\beta}^*_{\mathscr{A}}\|_1 \\
+ \|\boldsymbol{\zeta}\|_\infty \cdot \|\widehat{\boldsymbol{\alpha}}_\lambda - \boldsymbol{\alpha}^*\|_1.
\end{aligned} \tag{4.11}$$

Under event $\mathcal{E} = \{\|\boldsymbol{\zeta}\|_\infty \leq \lambda/2,\ \|\boldsymbol{\xi}\|_\infty \leq \lambda/2\}$, it follows from (4.11) that

$$\|\widehat{\boldsymbol{\Delta}}^\lambda_{\mathscr{A}^{\mathrm{c}}}\|_1 \leq 3\|\widehat{\boldsymbol{\Delta}}^\lambda_{\mathscr{A}}\|_1 + \|\widehat{\boldsymbol{\delta}}^\lambda\|_1.$$

The lemma then follows from Hoeffding's inequality

$$\begin{aligned}
\Pr(\mathcal{E}) &\geq 1 - \Pr\Big(\|\boldsymbol{\zeta}\|_\infty > \frac{\lambda}{2}\Big) - \Pr\Big(\|\boldsymbol{\xi}\|_\infty > \frac{\lambda}{2}\Big) \\
&\geq 1 - \sum_{k=1}^{K} \Pr\Big(\Big|-\frac{1}{nK}\sum_{i=1}^{n}[\tau_k - I(\varepsilon_i \leq \alpha_k^*)]\Big| > \frac{\lambda}{2}\Big) \\
&\qquad - \sum_{j=1}^{p} \Pr\Big(\Big|-\frac{1}{nK}\sum_{i=1}^{n} x_{ij}\sum_{k=1}^{K}[\tau_k - I(\varepsilon_i \leq \alpha_k^*)]\Big| > \frac{\lambda}{2}\Big) \\
&\geq 1 - 2K\exp\Big(-\frac{1}{2}nK^2\lambda^2\Big) - 2p\exp\Big(-\frac{n\lambda^2}{2M_0}\Big).
\end{aligned}$$

This proves the lemma. □

Now let $\nu_n(\boldsymbol{\alpha}, \boldsymbol{\beta}) = Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) - \mathbb{E}\big[Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big]$. For $r > 0$, set $\mathscr{C}_r = \{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}: (nK)^{-1}\sum_{k=1}^{K}\sum_{i=1}^{n}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2 \leq r^2\}$ and define $e(r) =$

$\sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}_r} |\nu_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta})|$.

Lemma 4.3

For $r,\ t > 0$, under conditions (C0) and (C1), with probability at least $1 - \exp\big[-nt^2/(32r^2)\big]$, we have

$$e(r) \leq 4\sqrt{\frac{2M_0}{\kappa_0}}\sqrt{\frac{1+\log p}{n}}\big(4\sqrt{s} + K + 1\big)r + t.$$

It follows immediately that, if one takes

$$t = 4\sqrt{\frac{2M_0}{\kappa_0}}\sqrt{\frac{1+\log p}{n}}\big(4\sqrt{s} + K + 1\big)r,$$

then with probability at least $1 - \exp\big[-16M_0\kappa_0^{-1}s(1+\log p)\big]$, we have

$$e(r) \leq 8\sqrt{\frac{2M_0}{\kappa_0}}\sqrt{\frac{1+\log p}{n}}\big(4\sqrt{s} + K + 1\big)r. \qquad \square$$

Proof 4.2 (Proof of Lemma 4.3)

First, let us show that the check loss $\rho_\tau(\cdot)$ is Lipschitz continuous with Lipschitz constant $\max(\tau,\ 1-\tau)$. To see it, note that for any $u_1,\ u_2 \in \mathbb{R}$, we have

$$\begin{aligned}
&|\rho_\tau(u_1) - \rho_\tau(u_2)| = |(\tau - 0.5)(u_1 - u_2) + 0.5(|u_1| - |u_2|)| \\
&\leq (|\tau - 0.5| + 0.5)|u_1 - u_2| = \max(\tau,\ 1-\tau)|u_1 - u_2|.
\end{aligned}$$

Now let $\boldsymbol{\delta} = \boldsymbol{\alpha} - \boldsymbol{\alpha}^*$, $\boldsymbol{\Delta} = \boldsymbol{\beta} - \boldsymbol{\beta}^*$ and define

$$\begin{aligned} U_i(\boldsymbol{\delta}, \boldsymbol{\Delta}) &= \frac{1}{K}\sum_{k=1}^{K} \rho_{\tau_k}(y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) - \frac{1}{K}\sum_{k=1}^{K} \rho_{\tau_k}(y_i - \alpha_k^* - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*) \\ &= \frac{1}{K}\sum_{k=1}^{K} \rho_{\tau_k}(r_{ik}^* - \delta_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}) - \frac{1}{K}\sum_{k=1}^{K} \rho_{\tau_k}(r_{ik}^*), \end{aligned}$$

where $r_{ik}^* = y_i - \alpha_k^* - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* = \varepsilon_i - \alpha_k^*$, $1 \le i \le n$, $1 \le k \le K$. It follows immediately that

$$e(r) = \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}_r} \left| \frac{1}{n}\sum_{i=1}^{n} \big[U_i(\boldsymbol{\delta}, \boldsymbol{\Delta}) - \mathbb{E}U_i(\boldsymbol{\delta}, \boldsymbol{\Delta}) \big] \right|.$$

By Lipschitz continuity of the check loss, it follows that

$$\begin{aligned} |U_i(\boldsymbol{\delta}, \boldsymbol{\Delta})| &\le \frac{1}{K}\sum_{k=1}^{K} |\rho_{\tau_k}(r_{ik}^* - \delta_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}) - \rho_{\tau_k}(r_{ik}^*)| \\ &\le \frac{1}{K}\sum_{k=1}^{K} \max(\tau_k, 1-\tau_k)|\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}| \le \frac{1}{K}\sum_{k=1}^{K} |\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}|, \ 1 \le i \le n. \end{aligned} \tag{4.12}$$

Now applying Massart's concentration inequality (Theorem 14.2, Bühlmann and van de Geer, 2011), we obtain

$$\Pr(e(r) \ge \mathbb{E}[e(r)] + t) \le \exp\left(-\frac{n^2 t^2}{8 b_n^2(r)} \right), \tag{4.13}$$

where $b_n^2(r) = \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}_r} \sum_{i=1}^{n} \operatorname{var}(U_i(\boldsymbol{\delta}, \boldsymbol{\Delta}))$. Now let us derive the upper bound on $b_n^2(r)$. Note that by (4.12) and Cauchy–Schwarz inequality

$$\begin{aligned} b_n^2(r) &= \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}_r} \sum_{i=1}^{n} \mathbb{E}\big[U_i(\boldsymbol{\delta}, \boldsymbol{\Delta}) - \mathbb{E}U_i(\boldsymbol{\delta}, \boldsymbol{\Delta}) \big]^2 \le 4 \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}_r} \sum_{i=1}^{n} \left[\sum_{k=1}^{K} \frac{1}{K} |\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}| \right]^2 \\ &\le 4 \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}_r} \sum_{i=1}^{n} \left(\sum_{k=1}^{K} \frac{1}{K} \right) \left[\sum_{k=1}^{K} \frac{1}{K} (\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2 \right] \le 4nr^2. \end{aligned}$$

We next show the upper bound on $\mathbb{E}\big[e(r)\big]$. Applying the symmetrization procedure (van der Vaart and Wellner, 1996) and the contraction principle (Ledoux and Talagrand, 1991), we have

$$\begin{aligned}
\mathbb{E}[e(r)] &\le 2\mathbb{E}\bigg[\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in\mathscr{C}_r}\frac{1}{n}\bigg|\sum_{i=1}^{n}\xi_i U_i(\boldsymbol{\delta},\boldsymbol{\Delta})\bigg|\bigg] \\
&\le \frac{2}{nK}\sum_{k=1}^{K}\mathbb{E}\bigg[\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in\mathscr{C}_r}\bigg|\sum_{i=1}^{n}\xi_i\{\rho_{\tau_k}(r_{ik}^{*}-\delta_k-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})-\rho_{\tau_k}(r_{ik}^{*})\}\bigg|\bigg] \qquad (4.14)\\
&\le \frac{4}{nK}\sum_{k=1}^{K}\mathbb{E}\bigg[\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in\mathscr{C}_r}\bigg|\sum_{i=1}^{n}\xi_i(\delta_k+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})\bigg|\bigg],
\end{aligned}$$

where $\xi_1,\ldots,\xi_n$ are i.i.d. Rademacher random variables that satisfy $\Pr(\xi_i=-1)=\Pr(\xi_i=1)=1/2$ and are independent of $\varepsilon_1,\ldots,\varepsilon_n$.

For $(\boldsymbol{\delta},\ \boldsymbol{\Delta})\in\mathscr{C}_r$, by condition (C1) and Cauchy–Schwarz inequality, we have

$$r^2 \ge \frac{\kappa_0}{K}\sum_{k=1}^{K}(\delta_k^2+\|\boldsymbol{\Delta}_{\mathscr{A}}\|_2^2) \ge \frac{\kappa_0}{K^2}\|\boldsymbol{\delta}\|_1^2+\frac{\kappa_0}{s}\|\boldsymbol{\Delta}_{\mathscr{A}}\|_1^2, \qquad (4.15)$$

which implies that $\|\boldsymbol{\delta}\|_1\le rK/\sqrt{\kappa_0}$ and $\|\boldsymbol{\Delta}_{\mathscr{A}}\|_1\le r\sqrt{s/\kappa_0}$. Now let $\boldsymbol{\xi}=(\xi_1,\ldots,\xi_n)^{\mathrm{T}}$. Note that for any $t\in\mathbb{R}$, we have

$$\begin{aligned}
\mathbb{E}\big[\exp(tX_j^{\mathrm{T}}\boldsymbol{\xi})\big] &= \prod_{i=1}^{n}\bigg[\frac{1}{2}(e^{tx_{ij}}+e^{-tx_{ij}})\bigg] \\
&\le \prod_{i=1}^{n}\exp\Big(\frac{1}{2}t^2x_{ij}^2\Big)=\exp\bigg(\frac{t^2}{2}\sum_{i=1}^{n}x_{ij}^2\bigg),\ 0\le j\le p.
\end{aligned}$$

Letting $t > 0$, by Jensen's inequality, we have

$$\begin{aligned}
&\exp\big(t\mathbb{E}\big[\|\mathbb{X}^{\mathrm{T}}\boldsymbol{\xi}\|_\infty\big]\big) = \exp\Big(t\mathbb{E}\max_{0\le j\le p}|X_j^{\mathrm{T}}\boldsymbol{\xi}|\Big) \le \mathbb{E}\exp\Big(t\max_{0\le j\le p}|X_j^{\mathrm{T}}\boldsymbol{\xi}|\Big)\\
&= \mathbb{E}\Big[\max_{0\le j\le p}\exp(t|X_j^{\mathrm{T}}\boldsymbol{\xi}|)\Big] \le \mathbb{E}\max_{0\le j\le p}\Big(e^{tX_j^{\mathrm{T}}\boldsymbol{\xi}} + e^{-tX_j^{\mathrm{T}}\boldsymbol{\xi}}\Big)\\
&\le \sum_{j=0}^{p}\mathbb{E}\Big(e^{tX_j^{\mathrm{T}}\boldsymbol{\xi}} + e^{-tX_j^{\mathrm{T}}\boldsymbol{\xi}}\Big) \le 2\sum_{j=0}^{p}\exp\Big(\frac{t^2}{2}\|X_j\|_2^2\Big)\\
&\le 2(1+p)\exp\Big(\frac{t^2}{2}\max_{0\le j\le p}\|X_j\|_2^2\Big) = 2(1+p)\exp\Big(\frac{1}{2}nM_0t^2\Big),
\end{aligned}$$

which implies that

$$\mathbb{E}\big(\|\mathbb{X}^{\mathrm{T}}\boldsymbol{\xi}\|_\infty\big) \le \frac{1}{t}\big[\log 2 + \log(1+p)\big] + \frac{nM_0}{2}t,\ t > 0.$$

Taking $t = \sqrt{2[\log 2 + \log(1+p)]/(nM_0)}$ and noting that $p \ge 3$ by condition (C0), we obtain

$$\mathbb{E}\big(\|\mathbb{X}^{\mathrm{T}}\boldsymbol{\xi}\|_\infty\big) \le \sqrt{2nM_0[\log 2 + \log(1+p)]} \le \sqrt{2M_0}\cdot\sqrt{n(1+\log p)}. \tag{4.16}$$

It then follows from (4.14), (4.16) and Hölder's inequality that

$$\begin{aligned}
\mathbb{E}[e(r)] &\le \frac{4}{nK}\mathbb{E}\big(\|\mathbb{X}^{\mathrm{T}}\boldsymbol{\xi}\|_\infty\big)\cdot\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in\mathscr{C}_r}\sum_{k=1}^{K}\big(|\delta_k| + \|\boldsymbol{\Delta}\|_1\big)\\
&\le \frac{4\sqrt{2M_0}}{n}\sqrt{n(1+\log p)}\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in\mathscr{C}_r}\big(K^{-1}\|\boldsymbol{\delta}\|_1 + \|\boldsymbol{\Delta}\|_1\big)\\
&\le \frac{4\sqrt{2M_0}}{n}\sqrt{n(1+\log p)}\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in\mathscr{C}_r}\big[(1+K^{-1})\|\boldsymbol{\delta}\|_1 + 4\|\boldsymbol{\Delta}_{\mathscr{A}}\|_1\big]\\
&\le 4\sqrt{\frac{2M_0}{\kappa_0}}\sqrt{\frac{1+\log p}{n}}\Big[4\sqrt{s} + (1+K)\Big]r.
\end{aligned}$$

The lemma then follows from (4.13). □

Lemma 4.4

Under conditions (C0) and (C2), for any $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}$, we have

$$\mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \geq \min\big\{\underline{f} r^2/4,\, q\big(\underline{f}/K\big)^{1/2} r\big\},$$

where $r^2 = (nK)^{-1} \sum_{i=1}^{n} \sum_{k=1}^{K} (\delta_k + \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\Delta})^2$. □

Proof 4.3 (Proof of Lemma 4.4)

By Knight's identity (Knight, 1998), we have for any two scalars r and s,

$$|r - s| - |r| = -s\big[I(r > 0) - I(r < 0)\big] + 2\int_0^s \big[I(r \leq t) - I(r \leq 0)\big]\mathrm{d}\, t.$$

It follows that for any $\tau \in (0, 1)$,

$$\begin{aligned}
&\rho_\tau(r - s) - \rho_\tau(r) = (\tau - 0.5)\big[(r - s) - r\big] + 0.5\big[|r - s| - |r|\big] \\
&= (0.5 - \tau)s - 0.5s\big[I(r > 0) - I(r < 0)\big] + \int_0^s \big[I(r \leq t) - I(r \leq 0)\big]\mathrm{d}\, t \\
&= s\big[I(r < 0) - \tau\big] + \int_0^s \big[I(r \leq t) - I(r \leq 0)\big]\mathrm{d}\, t.
\end{aligned} \tag{4.17}$$

Let $r_{ik}^* = y_i - \alpha_k^* - \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}^* = \varepsilon_i - \alpha_k^*$, $1 \leq i \leq n$, $1 \leq k \leq K$. By condition (C0), equation (4.17) and the mean value theorem, we have for some $\bar{u}_{ik,t}$ between 0 and t,

$$\begin{aligned}
&\mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \\
&= \frac{1}{nK} \sum_{i=1}^{n} \sum_{k=1}^{K} \int_0^{\delta_k + \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\Delta}} \big[F(\alpha_k^* + t) - F(\alpha_k^*)\big]\mathrm{d}\, t \\
&= \frac{1}{nK} \sum_{i=1}^{n} \sum_{k=1}^{K} \int_0^{\delta_k + \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\Delta}} \Big[t f(\alpha_k^*) + \frac{t^2}{2} f'(\alpha_k^* + \bar{u}_{ik,t})\Big]\mathrm{d}\, t \\
&\geq \frac{\underline{f}}{2nK} \sum_{i=1}^{n} \sum_{k=1}^{K} (\delta_k + \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\Delta})^2 - \frac{\bar{f}'}{6nK} \sum_{i=1}^{n} \sum_{k=1}^{K} |\delta_k + \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\Delta}|^3.
\end{aligned} \tag{4.18}$$

For $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}$, note that if

$$\left[\frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2\right]^{1/2} \leq \frac{4q}{K^{1/2}\underline{f}^{1/2}}, \tag{4.19}$$

then by condition (C2), this implies that

$$\frac{\bar{f}'}{6nK}\sum_{i=1}^{n}\sum_{k=1}^{K}|\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}|^3 \leq \frac{\underline{f}}{4nK}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2,$$

which, together with (4.18), implies that for all $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}_{4q(K\underline{f})^{-1/2}}$,

$$\mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \geq \frac{\underline{f}}{4nK}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2.$$

To show that the lemma holds for all $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}$, define

$$r_{\mathscr{C}} = \sup_{r>0}\Big\{r : \mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \geq \frac{\underline{f}}{4nK}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2,\ \forall (\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}_r\Big\}.$$

By previous arguments, we must have $r_{\mathscr{C}} \geq 4q(K\underline{f})^{-1/2}$. Now for any $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}$, let $r^2 = (nK)^{-1}\sum_{i=1}^{n}\sum_{k=1}^{K}\rho_{\tau_k}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2$. If $r < r_{\mathscr{C}}$, then by the definition of $r_{\mathscr{C}}$, we have

$$\mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \geq \frac{\underline{f}}{4nK}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2. \tag{4.20}$$

Otherwise, if $r \geq r_{\mathscr{C}}$, let $\boldsymbol{\delta}' = r_{\mathscr{C}}\boldsymbol{\delta}/r$ and $\boldsymbol{\Delta}' = r_{\mathscr{C}}\boldsymbol{\Delta}/r$. It can be seen immediately that

$(nK)^{-1}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta'_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}')^2 = r_{\mathscr{C}}^2$. By convexity of Q_n, we have

$$\begin{aligned}
&\mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \\
&\geq \frac{r}{r_{\mathscr{C}}}\mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}', \boldsymbol{\beta}^* + \boldsymbol{\Delta}') - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \\
&\geq \frac{r}{r_{\mathscr{C}}}\frac{f}{4}r_{\mathscr{C}}^2 \geq q\Big(\frac{f}{K}\Big)^{1/2}\Big[\frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2\Big]^{1/2}.
\end{aligned} \tag{4.21}$$

The lemma then follows from (4.20) and (4.21). □

Proof 4.4 (Proof of Theorem 4.1)

Let

$$r_* = 8\underline{f}^{-1}\left[8\sqrt{\frac{2M_0}{\kappa_0}}\sqrt{\frac{1+\log p}{n}}(4\sqrt{s} + K + 1) + \lambda\sqrt{\frac{s}{\kappa_0}}\right]$$

and set $\mathscr{C}^* = \{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C} : (nK)^{-1}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2 = r_*^2\}$. Moreover, define $\widehat{\boldsymbol{\delta}}^{\lambda} = \widehat{\boldsymbol{\alpha}}_{\lambda} - \boldsymbol{\alpha}^*$ and $\widehat{\boldsymbol{\Delta}}^{\lambda} = \widehat{\boldsymbol{\beta}}_{\lambda} - \boldsymbol{\beta}^*$. Under event $\mathcal{E}_1 = \{(\widehat{\boldsymbol{\delta}}^{\lambda}, \widehat{\boldsymbol{\Delta}}^{\lambda}) \in \mathscr{C}\}$, if we can show that

$$\inf_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}^*} Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) + \lambda(\|\boldsymbol{\beta}^* + \boldsymbol{\Delta}\|_1 - \|\boldsymbol{\beta}^*\|_1) > 0, \tag{4.22}$$

then by convexity of Q_n, this implies that $(\widehat{\boldsymbol{\delta}}^{\lambda}, \widehat{\boldsymbol{\Delta}}^{\lambda}) \in \mathscr{C}_{r_*}$. To show (4.22), first note that for all $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}^*$,

$$\begin{aligned}
&Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) + \lambda(\|\boldsymbol{\beta}^* + \boldsymbol{\Delta}\|_1 - \|\boldsymbol{\beta}^*\|_1) \\
&\qquad \geq \mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] - e(r_*) \\
&\qquad\quad + \lambda\big(\|\Delta_{\mathscr{A}^{\mathrm{c}}}\|_1 - \|\Delta_{\mathscr{A}}\|_1\big).
\end{aligned} \tag{4.23}$$

Now let $\mathcal{E}_2 = \big\{e(r_*) \leq 8\sqrt{2M_0/\kappa_0}\sqrt{(1+\log p)/n}\big(4\sqrt{s} + K + 1\big)r_*\big\}$. It follows from Lemma 4.3 that $\Pr(\mathcal{E}_2) \geq 1 - \exp\big[-16M_0\kappa_0^{-1}s(1+\log p)\big]$. By Lemma 4.4, for any

$(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}^*$, we have

$$\mathbb{E}\big[Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \geq \min\big\{\underline{f} r_*^2/4,\ q(\underline{f}/K)^{1/2} r_*\big\}.$$

Also, by condition (C1) and (4.15), for $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}^*$, we have $\|\boldsymbol{\Delta}_{\mathscr{A}}\|_1 \leq r_* \sqrt{s/\kappa_0}$. Thus, under event $\mathcal{E}_1 \cap \mathcal{E}_2$, for any $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathscr{C}^*$, it follows from (4.23) and the growth condition that

$$\begin{aligned}
&Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) + \lambda(\|\boldsymbol{\beta}^* + \boldsymbol{\Delta}\|_1 - \|\boldsymbol{\beta}^*\|_1) \\
&\geq \frac{\underline{f}}{4} r_*^2 - \left[8\sqrt{\frac{2M_0}{\kappa_0}}\sqrt{\frac{1+\log p}{n}}\big(4\sqrt{s} + K + 1\big) + \lambda\sqrt{\frac{s}{\kappa_0}}\right] r_* > 0
\end{aligned}$$

by our choice of r_*. Therefore, by Lemma 4.2 and 4.3, with probability at least

$$\Pr(\mathcal{E}_1 \cap \mathcal{E}_2) \geq 1 - \Pr(\mathcal{E}_1^{\mathrm{c}}) - \Pr(\mathcal{E}_2^{\mathrm{c}}) \geq 1 - p_1(\lambda),$$

we have $(\widehat{\boldsymbol{\delta}}^{\lambda}, \widehat{\boldsymbol{\Delta}}^{\lambda}) \in \mathscr{C}_{r_*}$. This, by condition (C1), further implies that

$$\begin{aligned}
r_*^2 &\geq \frac{1}{K}\sum_{k=1}^{K} \kappa_m \Big[|\hat{\delta}_k^{\lambda}|^2 + \|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}\cup\overline{\mathscr{A}}(\widehat{\boldsymbol{\Delta}}^{\lambda}, m)}\|_2^2\Big] \\
&\geq \frac{1}{K}\kappa_m \|\widehat{\boldsymbol{\delta}}^{\lambda}\|_2^2 + \kappa_m \|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}\cup\overline{\mathscr{A}}(\widehat{\boldsymbol{\Delta}}^{\lambda}, m)}\|_2^2.
\end{aligned}$$

As a result, we obtain that $\|\widehat{\boldsymbol{\delta}}^{\lambda}\|_2 \leq r_* \sqrt{K/\kappa_m}$ and that

$$\|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}\cup\overline{\mathscr{A}}(\widehat{\boldsymbol{\Delta}}^{\lambda}, m)}\|_2 \leq r_*/\sqrt{\kappa_m}. \tag{4.24}$$

Note that the jth largest in absolute value component of $\widehat{\boldsymbol{\Delta}}_{\mathscr{A}^{\mathrm{c}}}$ is bounded by $\|\widehat{\boldsymbol{\Delta}}_{\mathscr{A}^{\mathrm{c}}}\|_1/j$.

Therefore, it follows that

$$\begin{aligned}
&\Big\| \widehat{\boldsymbol{\Delta}}_{(\mathscr{A}\cup\overline{\mathscr{A}}(\widehat{\boldsymbol{\Delta}}^{\lambda},m))^{\mathrm{c}}} \Big\|_2^2 \le \sum_{j=m+1}^{p} \frac{\|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}^{\mathrm{c}}}\|_1^2}{j^2} \le \frac{1}{m}\|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}^{\mathrm{c}}}\|_1^2 \\
&\le \frac{1}{m}\big[3\|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}}\|_1 + \|\widehat{\boldsymbol{\delta}}^{\lambda}\|_1\big]^2 \le \frac{18s}{m}\|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}}\|_2^2 + \frac{2K}{m}\|\widehat{\boldsymbol{\delta}}^{\lambda}\|_2^2 \\
&\le \frac{18s}{m}\|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}\cup\overline{\mathscr{A}}(\widehat{\boldsymbol{\Delta}}^{\lambda},m)}\|_2^2 + \frac{2K}{m}\|\widehat{\boldsymbol{\delta}}^{\lambda}\|_2^2,
\end{aligned}$$

which, together with (4.24), implies that

$$\begin{aligned}
\|\widehat{\boldsymbol{\Delta}}^{\lambda}\|_2^2 &\le \Big(1 + \frac{18s}{m}\Big)\|\widehat{\boldsymbol{\Delta}}^{\lambda}_{\mathscr{A}\cup\overline{\mathscr{A}}(\widehat{\boldsymbol{\Delta}}^{\lambda},m)}\|_2^2 + \frac{2K}{m}\|\widehat{\boldsymbol{\delta}}^{\lambda}\|_2^2 \\
&\le \frac{r_*^2}{\kappa_m}\Big(1 + \frac{18s}{m} + \frac{2K^2}{m}\Big).
\end{aligned}$$

This completes the proof of Theorem 4.1. □

Lemma 4.5

Suppose the folded concave penalized quantile regression (4.4) is solved with the LLA algorithm (Algorithm 7). Let $a_0 = \min(a_2, 1)$ and define

$$\begin{aligned}
\mathcal{E}_1 &= \{\|\widehat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^*\|_\infty \le a_0\lambda\}, \\
\mathcal{E}_2 &= \{\|\nabla_{\mathscr{A}^{\mathrm{c}}} Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}})\|_\infty < a_1\lambda\}, \\
\mathcal{E}_3 &= \Big\{\min_{j\in\mathscr{A}} |\hat{\beta}_j^{\mathrm{o}}| > a\lambda\Big\},
\end{aligned}$$

where $\nabla_{\mathscr{A}^{\mathrm{c}}} Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}) = \big(\nabla_j Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}),\ j \in \mathscr{A}^{\mathrm{c}}\big)$ with

$$\nabla_j Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}) = \frac{1}{2n}\sum_{i=1}^{n} x_{ij}\Big(1 - \frac{2}{K}\sum_{k=1}^{K}\tau_k\Big) - \frac{1}{2nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\mathrm{Sgn}(\hat{r}_{ik})x_{ij},$$

$\hat{r}_{ik} = y_i - \hat{\alpha}_k^{\mathrm{o}} - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{\mathrm{o}}$, $1 \le i \le n$, $1 \le k \le K$, and

$$\mathrm{Sgn}(u) = \begin{cases} 1, & \text{if } u > 0 \\ [-1, 1], & \text{if } u = 0 \\ -1, & \text{if } u < 0. \end{cases}$$

Then under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ and condition (C0), the LLA algorithm converges to the oracle estimator. □

Proof 4.5 (Proof of Lemma 4.5)

Note that Q_n is convex, but not differentiable. Denote the subdifferential of $Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta})$ at $(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}})$ by

$$\partial Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}) = \Big\{(\boldsymbol{\zeta}, \boldsymbol{\xi}) \colon \zeta_k = \frac{1 - 2\tau_k}{2K} - \frac{1}{2nK}\sum_{i=1}^{n} \mathrm{Sgn}(\hat{r}_{ik}),\ 1 \le k \le K,$$
$$\xi_j = \frac{1}{2n}\sum_{i=1}^{n} x_{ij}\Big(1 - \frac{2}{K}\sum_{l=1}^{K}\tau_l\Big) - \frac{1}{2nK}\sum_{i=1}^{n}\sum_{l=1}^{K}\mathrm{Sgn}(\hat{r}_{il})x_{ij},\ 1 \le j \le p\Big\}.$$

By convexity of Q_n, for any $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and $(\boldsymbol{\zeta}, \boldsymbol{\xi}) \in \partial Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}})$, we have

$$Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta}) - Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}) \ge \boldsymbol{\zeta}^{\mathrm{T}}(\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}}^{\mathrm{o}}) + \boldsymbol{\xi}^{\mathrm{T}}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{\mathrm{o}}).$$

Now by optimality of $(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}})$, we can take $\boldsymbol{\zeta} = \mathbf{0}$ and $\boldsymbol{\xi}_{\mathscr{A}} = \mathbf{0}$. It follows that

$$Q_n(\boldsymbol{\alpha}, \boldsymbol{\beta}) \ge Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}) + \sum_{j \in \mathscr{A}^{\mathrm{c}}} \xi_j(\beta_j - \hat{\beta}_j^{\mathrm{o}}). \tag{4.25}$$

Under event $\mathcal{E}_1$, we have $\max_{j \in \mathscr{A}^{\mathrm{c}}} |\hat{\beta}_j^{(0)}| \le a_0\lambda \le a_2\lambda$. Moreover, by condition (C3), we

have

$$\min_{j\in\mathscr{A}}|\hat{\beta}^{(0)}| \geq \min_{j\in\mathscr{A}}|\beta_j^*| - \max_{j\in\mathscr{A}}|\hat{\beta}_j^{(0)} - \beta_j^*| \geq (a+1-a_0)\lambda \geq a\lambda.$$

Thus, under event $\mathcal{E}_1$, it follows from properties (P3) and (P4) of p_λ that

$$p'_\lambda(|\hat{\beta}_j^{(0)}|) \geq a_1\lambda,\ \forall\, j \in \mathscr{A}^{\mathrm{c}} \quad \text{and} \quad p'_\lambda(|\hat{\beta}_j^{(0)}|) = 0,\ \forall\, j \in \mathscr{A}.$$

Similarly, under event $\mathcal{E}_3$ and by the fact that $\widehat{\boldsymbol{\beta}}^{\mathrm{o}}_{\mathscr{A}^{\mathrm{c}}} = \mathbf{0}$, it can be shown that

$$p'_\lambda(|\hat{\beta}_j^{\mathrm{o}}|) = 0,\ \forall\, j \in \mathscr{A} \quad \text{and} \quad p'_\lambda(|\hat{\beta}_j^{\mathrm{o}}|) \geq a_1\lambda,\ \forall\, j \in \mathscr{A}^{\mathrm{c}}.$$

To this end, it can be seen from step (2.a) of Algorithm 7 that

$$(\widehat{\boldsymbol{\alpha}}^{(1)}, \widehat{\boldsymbol{\beta}}^{(1)}) = \operatorname*{arg\,min}_{\boldsymbol{\alpha},\boldsymbol{\beta}} Q_n(\boldsymbol{\alpha},\boldsymbol{\beta}) + \sum_{j\in\mathscr{A}^{\mathrm{c}}} p'_\lambda(|\hat{\beta}_j^{(0)}|)|\beta_j|.$$

Now under $\mathcal{E}_2 = \{\|\boldsymbol{\xi}_{\mathscr{A}^{\mathrm{c}}}\|_\infty < a_1\lambda\}$, it follows from (4.25) that for any $(\boldsymbol{\alpha},\boldsymbol{\beta})$,

$$\begin{aligned}
&\Big[Q_n(\boldsymbol{\alpha},\boldsymbol{\beta}) + \sum_{j\in\mathscr{A}^{\mathrm{c}}} p'_\lambda(|\hat{\beta}_j^{(0)}|)|\beta_j|\Big] - \Big[Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}},\widehat{\boldsymbol{\beta}}^{\mathrm{o}}) + \sum_{j\in\mathscr{A}^{\mathrm{c}}} p'_\lambda(|\hat{\beta}_j^{(0)}|)|\hat{\beta}_j^{\mathrm{o}}|\Big] \\
&\qquad \geq \sum_{j\in\mathscr{A}^{\mathrm{c}}} \xi_j(\beta_j - \hat{\beta}_j^{\mathrm{o}}) + \sum_{j\in\mathscr{A}^{\mathrm{c}}} p'_\lambda(|\hat{\beta}_j|^{(0)})|\beta_j| \\
&\qquad \geq \sum_{j\in\mathscr{A}^{\mathrm{c}}} \big[p'_\lambda(|\hat{\beta}_j^{(0)}|) - |\xi_j|\big]|\beta_j| \geq 0.
\end{aligned} \tag{4.26}$$

The leftmost hand side of the above inequality is strictly positive unless $\boldsymbol{\beta}_{\mathscr{A}^{\mathrm{c}}} = \mathbf{0}$. Note that condition (C0) implies the uniqueness of the oracle estimator (See Appendix C). It can be then seen that $(\widehat{\boldsymbol{\alpha}}^{(1)}, \widehat{\boldsymbol{\beta}}^{(1)})$ coincides with the oracle estimator. Now given that $(\widehat{\boldsymbol{\alpha}}^{(1)}, \widehat{\boldsymbol{\beta}}^{(1)})$ is the oracle estimator, we show that $(\widehat{\boldsymbol{\alpha}}^{(2)}, \widehat{\boldsymbol{\beta}}^{(2)})$ yielded by the LLA algorithm will still be the

oracle estimator. To see it, note that under event $\mathcal{E}_2$,

$$p'_\lambda(|\hat{\beta}_j^{(1)}|) = 0,\ \forall j \in \mathscr{A} \quad \text{and} \quad p'_\lambda(|\hat{\beta}_j^{(1)}|) \geq a_1\lambda,\ \forall j \in \mathscr{A}^{\mathrm{c}}.$$

By the LLA iteration, we have

$$(\widehat{\boldsymbol{\alpha}}^{(2)}, \widehat{\boldsymbol{\beta}}^{(2)}) = \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} Q_n(\boldsymbol{\alpha},\boldsymbol{\beta}) + \sum_{j\in\mathscr{A}^{\mathrm{c}}} p'_\lambda(|\hat{\beta}_j^{(1)}|)|\beta_j|.$$

Thus, we can follow similar arguments from (4.26) to show that under event $\mathcal{E}_3$, $(\widehat{\boldsymbol{\alpha}}^{(2)}, \widehat{\boldsymbol{\beta}}^{(2)})$ is still the oracle estimator. This proves the lemma. □

Note that the above proof is slightly different from the general result (Theorems 1 and 2) in Fan et al. (2014b) since we need to deal with the intercept terms additionally.

For $r > 0$, define $B_{\mathscr{A}}(r) = \{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathbb{R}^K \times \mathbb{R}^p : \|\boldsymbol{\delta}\|_2^2 + \|\boldsymbol{\Delta}_{\mathscr{A}}\|_2^2 \leq r^2,\ \boldsymbol{\Delta}_{\mathscr{A}^{\mathrm{c}}} = \mathbf{0}\}$ and $S_{\mathscr{A}}(r) = \{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in \mathbb{R}^K \times \mathbb{R}^p : \|\boldsymbol{\delta}\|_2^2 + \|\boldsymbol{\Delta}_{\mathscr{A}}\|_2^2 = r^2,\ \boldsymbol{\Delta}_{\mathscr{A}^{\mathrm{c}}} = \mathbf{0}\}$. Moreover, let $z(r) = \sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in B_{\mathscr{A}}(r)} |\nu_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta})|$.

Lemma 4.6

Under condition (C0), for any $t > 0$ and $r > 0$ satisfying $r\sqrt{(s+1)M_{\mathscr{A}}} \leq \mathcal{U}_0$, with probability at least $1 - \exp\big[-nt^2/(32\overline{\mu}r^2)\big]$, we have $z(r) \leq 4r\sqrt{(s+1)M_{\mathscr{A}}/n} + t$ and

$$\inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r)} \big[Q_n(\boldsymbol{\alpha}^*+\boldsymbol{\delta}, \boldsymbol{\beta}^*+\boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)\big] \geq \frac{1}{2K}\underline{f\mu} r^2 - 4r\sqrt{\frac{(s+1)M_{\mathscr{A}}}{n}} - t. \quad \square$$

Proof 4.6 (Proof of Lemma 4.6)

As with the proof of Lemma 4.3, define

$$U_i(\boldsymbol{\delta},\boldsymbol{\Delta}) = \frac{1}{K}\sum_{k=1}^K \rho_{\tau_k}(r_{ik}^* - \delta_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}) - \frac{1}{K}\sum_{k=1}^K \rho_{\tau_k}(r_{ik}^*),$$

where $r_{ik}^* = y_i - \alpha_k^* - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* = \varepsilon_i - \alpha^*$, $1 \le i \le n$, $1 \le k \le K$. By Massart's concentration inequality again, we get

$$\Pr(z(r) \ge \mathbb{E}[z(r)] + t) \le \exp\left(-\frac{n^2 t^2}{8 b_n^2(r)}\right), \tag{4.27}$$

where $b_n^2(r) = \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)} \sum_{i=1}^n \operatorname{var}(U_i(\boldsymbol{\delta}, \boldsymbol{\Delta}))$. For ease of notation, let us denote $\boldsymbol{\Delta}_{\mathscr{A}}^k = (\delta_k, \boldsymbol{\Delta}_{\mathscr{A}}^{\mathrm{T}})^{\mathrm{T}}$, $1 \le k \le K$. It follows from the Lipschitz continuity of the check loss that

$$\begin{aligned} b_n^2(r) &\le \frac{4}{K} \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)} \sum_{k=1}^K \sum_{i=1}^n (\delta_k + \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\Delta})^2 \\ &= \frac{4}{K} \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)} \sum_{k=1}^K (\boldsymbol{\Delta}_{\mathscr{A}}^k)^{\mathrm{T}} \mathbb{X}_{\mathscr{A}_0}^{\mathrm{T}} \mathbb{X}_{\mathscr{A}_0} \boldsymbol{\Delta}_{\mathscr{A}}^k \\ &\le \frac{4n}{K} \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)} \sum_{k=1}^K \overline{\mu} \big[\delta_k^2 + \|\boldsymbol{\Delta}_{\mathscr{A}}\|_2^2\big] \le 4n\overline{\mu} r^2. \end{aligned}$$

Moreover, by the symmetrization procedure and the contraction principle again, we obtain

$$\begin{aligned} \mathbb{E}[z(r)] &\le \frac{4}{nK} \sum_{k=1}^K \mathbb{E}\left[\sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)} \left| \sum_{i=1}^n \xi_i (\delta_k + \mathbf{x}_{\mathscr{A}}^{\mathrm{T}} \boldsymbol{\Delta}_{\mathscr{A}}) \right| \right] \\ &\le \frac{4}{nK} \mathbb{E}\big(\|\mathbb{X}_{\mathscr{A}_0}^{\mathrm{T}} \boldsymbol{\xi}\|_2\big) \cdot \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)} \sum_{k=1}^K \|\boldsymbol{\Delta}_{\mathscr{A}}^k\|_2 \le \frac{4r}{n} \mathbb{E}\big(\|\mathbb{X}_{\mathscr{A}_0}^{\mathrm{T}} \boldsymbol{\xi}\|_2\big), \end{aligned} \tag{4.28}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)^{\mathrm{T}}$ is a random vector of i.i.d. Rademacher variables that satisfy $\Pr(\xi_i = -1) = \Pr(\xi_i = 1) = 1/2$ and that are independent of $\varepsilon_1, \ldots, \varepsilon_n$. By Jensen's and

Khintchine inequalities (Haagerup, 1981), we have

$$\mathbb{E}\big(\|\mathbb{X}_{\mathscr{A}_0}^{\mathrm{T}}\boldsymbol{\xi}\|_2\big) \le \big[\mathbb{E}\big(\boldsymbol{\xi}^{\mathrm{T}}\mathbb{X}_{\mathscr{A}_0}\mathbb{X}_{\mathscr{A}_0}\boldsymbol{\xi}\big)\big]^{1/2} = \Big[\sum_{j\in\mathscr{A}_0}\mathbb{E}\Big(\sum_{i=1}^{n}\xi_i x_{ij}\Big)^2\Big]^{1/2}$$
$$\le \Big(\sum_{j\in\mathscr{A}_0}\sum_{i=1}^{n}x_{ij}^2\Big)^{1/2} = \Big(\sum_{i=1}^{n}\sum_{j\in\mathscr{A}_0}x_{ij}^2\Big)^{1/2} \le \sqrt{n(s+1)M_{\mathscr{A}}}.$$

It follows from (4.28) that $\mathbb{E}[z(r)] \le 4r\sqrt{(s+1)M_{\mathscr{A}}/n}$. The first part of the lemma then follows from (4.27). Let $F(\boldsymbol{\delta},\boldsymbol{\Delta}) = Q_n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)$. To prove the second inequality of the lemma, it suffices to note that

$$\inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r)} F(\boldsymbol{\delta},\boldsymbol{\Delta}) \ge \inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r)} \mathbb{E}\big[Q_n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\big] - z(r),$$

and that by (4.17) and the mean value theorem, we have for some $\bar{u}_{ik,t}$ between 0 and t such that

$$\begin{aligned}
&\inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r)} \mathbb{E}\big[Q_n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\big] \\
&= \inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r)} \frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\int_0^{\delta_k+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}} \big[F(\alpha_k^*+t) - F(\alpha_k^*)\big]\mathrm{d}t \qquad (4.29)\\
&= \inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r)} \frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\int_0^{\delta_k+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}} \big[t f(\alpha_k^*+\bar{u}_{ik,t})\big]\mathrm{d}t.
\end{aligned}$$

Now for any $1 \le i \le n$, $1 \le k \le K$ and $(\boldsymbol{\delta},\boldsymbol{\Delta}) \in S_{\mathscr{A}}(r)$, we have

$$|\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}| \le \sqrt{1+\|\mathbf{x}_{i\mathscr{A}}\|_2^2}\cdot\sqrt{\delta_k^2+\|\boldsymbol{\Delta}_{\mathscr{A}}\|_2^2} \le r\sqrt{(s+1)M_{\mathscr{A}}} \le \mathcal{U}_0.$$

It then follows from condition (C0) and (4.29) that

$$\begin{aligned}&\inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r)} \mathbb{E}\big[Q_n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta})-Q_n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\big]\\ &\geq \inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r)} \frac{\underline{f}}{2nK}\sum_{i=1}^{n}\sum_{k=1}^{K}(\delta_k+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})^2 \geq \frac{\underline{f}}{2K}\underline{\mu} r^2.\end{aligned}$$

This completes the proof. □

Proof 4.7 (Proof of Lemma 4.1)

For ease of notation, let $\widehat{\boldsymbol{\delta}}^{\mathrm{o}} = \widehat{\boldsymbol{\alpha}}^{\mathrm{o}} - \boldsymbol{\alpha}^*$ and $\widehat{\boldsymbol{\Delta}}^{\mathrm{o}} = \widehat{\boldsymbol{\beta}}^{\mathrm{o}} - \boldsymbol{\beta}^*$. In Lemma 4.6, let $r = r^* = 32K(\underline{f}\,\underline{\mu})^{-1}\sqrt{M_{\mathscr{A}}(s+1)/n}$ and take $t = 4r^*\sqrt{M_{\mathscr{A}}(s+1)/n}$. We can see that with the choice of r^*, there exists $\mathcal{U}_0$ such that $r^*\sqrt{(s+1)M_{\mathscr{A}}} \leq \mathcal{U}_0$. It follows immediately that with probability at least $1-\exp\big[-(s+1)M_{\mathscr{A}}/(2\bar{\mu})\big]$, we have

$$\inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r^*)}\big[Q_n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta})-Q_n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\big] \geq \frac{\underline{f}}{2K}\underline{\mu}(r^*)^2 - 8r^*\sqrt{\frac{(s+1)M_{\mathscr{A}}}{n}} > 0.$$

By convexity of Q_n and optimality of $(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}},\widehat{\boldsymbol{\beta}}^{\mathrm{o}})$, this implies that

$$\|\widehat{\boldsymbol{\delta}}^{\mathrm{o}}\|_2^2 + \|\widehat{\boldsymbol{\Delta}}^{\mathrm{o}}\|_2^2 \leq (r^*)^2.$$

This completes the lemma. □

For each $j \in \mathscr{A}^{\mathrm{c}}$, define $S_j^n(\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\big[I(y_i-\alpha_k-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}\leq 0)-\tau_k\big]x_{ij}$, and for $r>0$, let

$$\begin{aligned}\gamma_j(r) = \sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in B_{\mathscr{A}}(r)} \big| & S_j^n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta}) - S_j^n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\\ & - \mathbb{E}\big[S_j^n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta}) - S_j^n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\big]\big|.\end{aligned}$$

Lemma 4.7

For $r, t > 0$, $0 < \psi < r$ and $j \in \mathscr{A}^{\mathrm{c}}$, we have

$$\Pr(\gamma_j(r) > t) \le 2N_\psi \exp\left(-\frac{nt^2}{8\bar{f}M_{\mathscr{A}^{\mathrm{c}}}^2\overline{\mu}^{1/2}r + \frac{8}{3}M_{\mathscr{A}^{\mathrm{c}}}t}\right) + 2N_\psi \exp\left(-\frac{nt_0^2}{2\bar{f}M_{\mathscr{A}^{\mathrm{c}}}^2\big((s+1)M_{\mathscr{A}}\big)^{1/2}\psi + \frac{4}{3}M_{\mathscr{A}^{\mathrm{c}}}t_0}\right),$$

where N_ψ is the ψ-covering number (see, e.g., Pollard, 1990) of $B_{\mathscr{A}}(r)$ and $t_0 = \big[t/2 - 2\bar{f}M_{\mathscr{A}^{\mathrm{c}}}\big((s+1)M_{\mathscr{A}}\big)^{1/2}\psi\big]_+$. □

Proof 4.8 (Proof of Lemma 4.7)

Consider a minimal ψ-cover of $B_{\mathscr{A}}(r)$. Let us denote this covering net by $\{(\boldsymbol{\delta}^\ell, \boldsymbol{\Delta}^\ell), \ell = 1, \ldots, N_\psi\} \subset B_{\mathscr{A}}(r)$. For $j \in \mathscr{A}^{\mathrm{c}}$, define

$$U_{ij}(\boldsymbol{\delta}, \boldsymbol{\Delta}) = \frac{1}{K}\sum_{k=1}^{K}\big[I(r_{ik}^* \le \delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}) - \tau_k\big]x_{ij} - \frac{1}{K}\sum_{k=1}^{K}\big[I(r_{ik}^* \le 0) - \tau_k\big]x_{ij},$$

where $r_{ik}^* = y_i - \alpha_k^* - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* = \varepsilon_i - \alpha_k^*$, $1 \le i \le n$, $1 \le k \le K$. Then it can be seen that

$$\gamma_j(r) = \sup_{(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)} \left| \frac{1}{n}\sum_{i=1}^{n}\big[U_{ij}(\boldsymbol{\delta}, \boldsymbol{\Delta}) - \mathbb{E}U_{ij}(\boldsymbol{\delta}, \boldsymbol{\Delta})\big] \right|.$$

For any $(\boldsymbol{\delta}, \boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)$ and $j \in \mathscr{A}^{\mathrm{c}}$, note that

$$|U_{ij}(\boldsymbol{\delta}, \boldsymbol{\Delta})| \le \frac{1}{K}\sum_{k=1}^{K}|I(r_{ik}^* \le \delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}) - I(r_{ik}^* \le 0)| \cdot |x_{ij}| \le M_{\mathscr{A}^{\mathrm{c}}}.$$

Let $p_{ik} = \Pr\big(-(\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})_- < r_{ik}^* \le (\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})_+\big)$, $1 \le i \le n$, $1 \le k \le K$. It follows from the mean value theorem and condition (C0) that

$$p_{ik} = F(\alpha_k^* + (\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})_+) - F(\alpha_k^* - (\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})_-) \le \bar{f}|\delta_k + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}|.$$

By Cauchy–Schwarz inequality and the mean value theorem, we have

$$\begin{aligned}\operatorname{var}\big[U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta})\big] &\le \frac{x_{ij}^2}{K}\sum_{k=1}^{K}\operatorname{var}\big[I\big(-(\delta_k+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})_- < r_{ik}^* \le (\delta_k+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta})_+\big)\big]\\ &= \frac{x_{ij}^2}{K}\sum_{k=1}^{K} p_{ik}(1-p_{ik}) \le \frac{\bar f x_{ij}^2}{K}\sum_{k=1}^{K}|\delta_k+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}| \le \frac{\bar f M_{\mathscr{A}^{\mathrm{c}}}^2}{K}\sum_{k=1}^{K}|\delta_k+\mathbf{x}_{i\mathscr{A}}^{\mathrm{T}}\boldsymbol{\Delta}_{\mathscr{A}}|.\end{aligned}$$

Let $\boldsymbol{\Delta}_{\mathscr{A}}^k = (\delta_k, \boldsymbol{\Delta}_{\mathscr{A}}^{\mathrm{T}})^{\mathrm{T}}$, $1 \le k \le K$. By Cauchy–Schwarz inequality again, we get

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}\operatorname{var}\big[U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta})\big] \le \frac{1}{nK}\bar f M_{\mathscr{A}^{\mathrm{c}}}^2\sum_{i=1}^{n}\sum_{k=1}^{K}|\delta_k+\mathbf{x}_{i\mathscr{A}}^{\mathrm{T}}\boldsymbol{\Delta}_{\mathscr{A}}|\\ &\le \frac{\bar f M_{\mathscr{A}^{\mathrm{c}}}^2}{K}\sum_{k=1}^{K}\Big[\frac{1}{n}\big(\boldsymbol{\Delta}_{\mathscr{A}}^k\big)^{\mathrm{T}}\mathbb{X}_{\mathscr{A}_0}^{\mathrm{T}}\mathbb{X}_{\mathscr{A}_0}\boldsymbol{\Delta}_{\mathscr{A}}^k\Big]^{1/2} \le \bar f M_{\mathscr{A}^{\mathrm{c}}}^2\overline{\mu}^{1/2}r.\end{aligned}$$

Now applying Bernstein inequality, we have for any $(\boldsymbol{\delta},\boldsymbol{\Delta}) \in B_{\mathscr{A}}(r)$ and $t > 0$,

$$\Pr\bigg(\bigg|\frac{1}{n}\sum_{i=1}^{n}\big[U_{ij}(\delta,\boldsymbol{\Delta}) - \mathbb{E}U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta})\big]\bigg| > t\bigg) \le 2\exp\bigg(-\frac{nt^2}{2\bar f M_{\mathscr{A}^{\mathrm{c}}}^2\overline{\mu}^{1/2}r + \frac{4}{3}M_{\mathscr{A}^{\mathrm{c}}}t}\bigg).$$

Now for $1 \le \ell \le N_\psi$, let $B_\ell(\psi) = \{(\boldsymbol{\delta},\boldsymbol{\Delta})\colon \|\boldsymbol{\delta}-\boldsymbol{\delta}^\ell\|_2^2 + \|\boldsymbol{\Delta}-\boldsymbol{\Delta}^\ell\|_2^2 \le \psi^2,\ \boldsymbol{\Delta}_{\mathscr{A}^{\mathrm{c}}} = \mathbf{0}\}$ be the ball centered at $(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell) \in B_{\mathscr{A}}(r)$ with radius ψ. For any $1 \le i \le n$, $1 \le k \le K$ and $(\boldsymbol{\delta},\boldsymbol{\Delta}) \in B_\ell(\psi)$, note that

$$|(\delta_k+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}) - (\delta_k^\ell+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell)| \le \big(1+\|\mathbf{x}_{i\mathscr{A}}\|_2^2\big)^{1/2}\psi \le \big[(s+1)M_{\mathscr{A}}\big]^{1/2}\psi.$$

For $1 \le i \le n$ and $j \in \mathscr{A}^{\mathrm{c}}$, let

$$\begin{aligned}V_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell) = \frac{1}{K}|x_{ij}|\sum_{k=1}^{K}\big[&I\big(r_{ik}^* \le \delta_k^\ell + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell + ((s+1)M_{\mathscr{A}})^{1/2}\psi\big)\\ &- I(r_{ik}^* \le \delta_k^\ell + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell)\big].\end{aligned}$$

Since the indicator function $I(u \le t)$ is nondecreasing in t, we have

$$\begin{aligned}
&\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in B_\ell(\psi)}\left|\frac{1}{n}\sum_{i=1}^{n}\big(U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta})-U_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)-\mathbb{E}\big[U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta})-U_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)\big]\big)\right| \\
&\le \frac{1}{nK}\sum_{i=1}^{n}|x_{ij}|\sum_{k=1}^{K}\big[I\big(r_{ik}^{*}\le \delta_k^\ell+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell+((s+1)M_{\mathscr{A}})^{1/2}\psi\big)-I(r_{ik}^{*}\le \delta_k^\ell+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell) \\
&\qquad -\Pr\big(r_{ik}^{*}\le \delta_k^\ell+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell-((s+1)M_{\mathscr{A}})^{1/2}\psi\big)+\Pr(r_{ik}^{*}\le \delta_k^\ell+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell)\big] \\
&:= I_1+\frac{1}{n}\sum_{i=1}^{n}\big[V_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)-\mathbb{E}V_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)\big],
\end{aligned}$$

where

$$\begin{aligned}
I_1=\frac{1}{nK}\sum_{i=1}^{n}|x_{ij}|\sum_{k=1}^{K}\big[&\Pr\big(r_{ik}^{*}\le \delta_k^\ell+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell+((s+1)M_{\mathscr{A}})^{1/2}\psi\big) \\
&-\Pr\big(r_{ik}^{*}\le \delta_k^\ell+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Delta}^\ell-((s+1)M_{\mathscr{A}})^{1/2}\psi\big)\big].
\end{aligned}$$

By the mean value theorem, we have

$$I_1\le \frac{1}{nK}\sum_{i=1}^{n}|x_{ij}|\sum_{k=1}^{K}\cdot 2\big((s+1)M_{\mathscr{A}}\big)^{1/2}\bar{f}\psi\le 2\bar{f}M_{\mathscr{A}^{\mathrm{c}}}\big((s+1)M_{\mathscr{A}}\big)^{1/2}\psi.$$

Similarly, it can be shown that $|V_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)|\le M_{\mathscr{A}^{\mathrm{c}}}$ and

$$\frac{1}{n}\sum_{i=1}^{n}\mathrm{var}(V_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell))\le \bar{f}M_{\mathscr{A}^{\mathrm{c}}}^2\big((s+1)M_{\mathscr{A}}\big)^{1/2}\psi.$$

It then follows from Bernstein inequality that for $1\le \ell\le N_\psi$,

$$\begin{aligned}
\Pr\Bigg(\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in B_\ell(\psi)}&\left|\frac{1}{n}\sum_{i=1}^{n}\big(U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta})-U_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)-\mathbb{E}\big[U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta})-U_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)\big]\big)\right|>t\Bigg) \\
&\le 2\exp\Bigg(-\frac{nt_1^2}{2\bar{f}M_{\mathscr{A}^{\mathrm{c}}}^2\big((s+1)M_{\mathscr{A}}\big)^{1/2}\psi+\frac{4}{3}M_{\mathscr{A}^{\mathrm{c}}}t_1}\Bigg),
\end{aligned}$$

where $t_1 = \big[t - 2\bar{f} M_{\mathscr{A}^c}\big((s+1)M_{\mathscr{A}}\big)^{1/2}\psi\big]_+$. The lemma then follows by noting that

$$\begin{aligned}
\Pr(\gamma_j(r) > t) &= \Pr\bigg(\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in B_{\mathscr{A}}(r)}\bigg|\frac{1}{n}\sum_{i=1}^{n}\big[U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta}) - \mathbb{E}U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta})\big] > t\bigg|\bigg) \\
&\le \sum_{\ell=1}^{N_\psi}\Pr\bigg(\sup_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in B_\ell(\psi)}\bigg|\frac{1}{n}\sum_{i=1}^{n}\big(U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta}) - U_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell) \\
&\qquad\qquad - \mathbb{E}\big[U_{ij}(\boldsymbol{\delta},\boldsymbol{\Delta}) - U_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)\big]\big)\bigg| > \frac{t}{2}\bigg) \\
&\quad + \sum_{\ell=1}^{N_\psi}\Pr\bigg(\bigg|\frac{1}{n}\sum_{i=1}^{n}\big[U_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell) - \mathbb{E}U_{ij}(\boldsymbol{\delta}^\ell,\boldsymbol{\Delta}^\ell)\big]\bigg| > \frac{t}{2}\bigg).
\end{aligned}$$

This completes the proof. □

Proof 4.9 (Proof of Theorem 4.2)

Let $\widehat{\boldsymbol{\delta}}^{\mathrm{o}} = \widehat{\boldsymbol{\alpha}}^{\mathrm{o}} - \boldsymbol{\alpha}^*$ and $\widehat{\boldsymbol{\Delta}}^{\mathrm{o}} = \widehat{\boldsymbol{\beta}}^{\mathrm{o}} - \boldsymbol{\beta}^*$. For $1 \le i \le n$, $1 \le k \le K$, write $\hat{r}_{ik} = y_i - \hat{\alpha}_k^{\mathrm{o}} - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{\mathrm{o}}$ and $r_{ik}^* = y_i - \alpha_k^* - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*$. For ease of notation, let $F(\boldsymbol{\delta},\boldsymbol{\Delta}) = Q_n(\boldsymbol{\alpha}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^* + \boldsymbol{\Delta}) - Q_n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ for $(\boldsymbol{\delta},\boldsymbol{\Delta}) \in \mathbb{R}^K \times \mathbb{R}^p$. According to Lemma 4.5, with probability at least

$$\Pr(\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3) \ge 1 - \Pr(\mathcal{E}_1^{\mathrm{c}}) - \Pr(\mathcal{E}_2^{\mathrm{c}}) - \Pr(\mathcal{E}_3^{\mathrm{c}}),$$

the LLA algorithm will converge to the oracle estimator in two iterations. In the sequel, we will split the proof into three parts and provide the upper bound on each of $\Pr(\mathcal{E}_1^{\mathrm{c}})$, $\Pr(\mathcal{E}_2^{\mathrm{c}})$ and $\Pr(\mathcal{E}_3^{\mathrm{c}})$, separately.

(i) First, we deal with $\Pr(\mathcal{E}_1^{\mathrm{c}}) = \Pr(\|\widehat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^*\|_\infty > a_0\lambda)$. Since in the LLA algorithm (Algorithm 7), we take $(\widehat{\boldsymbol{\alpha}}^{(0)}, \widehat{\boldsymbol{\beta}}^{(0)})$ to be the lasso estimator $(\widehat{\boldsymbol{\alpha}}_{\lambda_0}, \widehat{\boldsymbol{\beta}}_{\lambda_0})$, by Theorem 4.1, we have

$$\Pr(\mathcal{E}_1) = \Pr\big(\|\widehat{\boldsymbol{\beta}}_{\lambda_0} - \boldsymbol{\beta}^*\|_\infty \le a_0\lambda\big) \ge \Pr\big(\|\widehat{\boldsymbol{\beta}}_{\lambda_0} - \boldsymbol{\beta}^*\|_2 \le a_0\lambda\big) \ge 1 - p_1(\lambda_0),$$

which implies that $\Pr(\mathcal{E}_1^{\mathrm{c}}) \le p_1(\lambda_0)$.

(ii) We next derive the upper bound on $\Pr(\mathcal{E}_3^{\mathrm{c}}) = \Pr\big(\min_{j\in\mathscr{A}} |\hat{\beta}_j^{\mathrm{o}}| \le a\lambda\big)$. Let $r_0 = \min_{j\in\mathscr{A}} |\beta_j^*| - a\lambda$. It can be seen that $\Pr(\mathcal{E}_3^{\mathrm{c}}) \le \Pr(\|\widehat{\boldsymbol{\Delta}}^{\mathrm{o}}\|_\infty > r_0)$. Note that by convexity of Q_n, $\|\widehat{\boldsymbol{\Delta}}^{\mathrm{o}}\|_2 \le r_0$ is implied by the event that $\inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r_0)} F(\boldsymbol{\delta},\boldsymbol{\Delta}) > 0$. Since $r_0\sqrt{(s+1)M_{\mathscr{A}}} \le \mathcal{U}_0$, it follows from Lemma 4.6 that for any $t > 0$,

$$\inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r_0)} F(\boldsymbol{\delta},\boldsymbol{\Delta}) \ge \frac{\underline{f}}{2K}\underline{\mu} r_0^2 - 4r_0\sqrt{\frac{(s+1)M_{\mathscr{A}}}{n}} - t$$

holds with probability at least $1-\exp\big[-nt^2/(32\overline{\mu}r_0^2)\big]$. By condition (C3), it can be seen that $r_0 > \lambda > 8K(\underline{f}\,\underline{\mu})^{-1}\sqrt{(s+1)M_{\mathscr{A}}/n}$. Now take $t = \underline{f}\,\underline{\mu} r_0^2/(4K) - 2r_0\sqrt{(s+1)M_{\mathscr{A}}/n}$. Then, we can see that $t > 0$. It follows immediately that $\inf_{(\boldsymbol{\delta},\boldsymbol{\Delta})\in S_{\mathscr{A}}(r_0)} F(\boldsymbol{\delta},\boldsymbol{\Delta}) > t > 0$. With this specific choice of t, we get

$$\Pr\big(\|\widehat{\boldsymbol{\Delta}}^{\mathrm{o}}\|_2 \le r_0\big) \ge 1 - \exp\big[-nt^2/(32\overline{\mu}r_0^2)\big],$$

which implies that

$$\Pr(\mathcal{E}_3^{\mathrm{c}}) \le \Pr(\|\widehat{\boldsymbol{\Delta}}^{\mathrm{o}}\|_\infty > r_0) \le \Pr(\|\widehat{\boldsymbol{\Delta}}^{\mathrm{o}}\|_2 > r_0) \le \exp\big[-nt^2/(32\overline{\mu}r_0^2)\big].$$

(iii) Finally, we look at $\Pr(\mathcal{E}_2^{\mathrm{c}}) = \Pr(\|\nabla_{\mathscr{A}^{\mathrm{c}}} Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}})\|_\infty \ge a_1\lambda)$. To this end, we set $r_* = \sqrt{(s+1)M_{\mathscr{A}}\log n/n}$ and let $\mathcal{R} = \{(i,k)\colon \hat{r}_{ik} = 0,\ 1 \le i \le n,\ 1 \le k \le K\}$ be the index set for zero residuals. From Section B of the appendix, we have $|\mathcal{R}| \le K(K+s)$. It can be seen that

$$\begin{aligned}\nabla_j Q_n(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}) &= \frac{1}{2nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\big[(1-2\tau_k) - \mathrm{Sgn}(\hat{r}_{ik})\big]x_{ij} \\ &= \frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\big[I(\hat{r}_{ik} \le 0) - \tau_k\big]x_{ij} - \frac{1}{2nK}\sum_{(i,k)\in\mathcal{R}}\big[\mathrm{Sgn}(\hat{r}_{ik}) + 1\big]x_{ij},\end{aligned}$$

where we have

$$\max_{j\in\mathscr{A}^{\mathrm{c}}}\left|\frac{1}{2nK}\sum_{(i,\,k)\in\mathcal{R}}\big[\mathrm{Sgn}(\hat{r}_{ik})+1\big]x_{ij}\right|\le\frac{(K+s)M_{\mathscr{A}^{\mathrm{c}}}}{n}:=B_1.$$

Now let $\mathcal{E}_0=\{(\widehat{\boldsymbol{\delta}}^{\mathrm{o}},\widehat{\boldsymbol{\Delta}}^{\mathrm{o}})\in B_{\mathscr{A}}(r_*)\}$. Under $\mathcal{E}_0$, note that by the triangular inequality, we have

$$\begin{aligned}\max_{j\in\mathscr{A}^{\mathrm{c}}}\left|\frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\big[I(\hat{r}_{ik}\le 0)-\tau_k\big]x_{ij}\right|&\le\max_{j\in\mathscr{A}^{\mathrm{c}}}\gamma_j(r_*)+\max_{j\in\mathscr{A}^{\mathrm{c}}}|S_j^n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)|\\&+\max_{j\in\mathscr{A}^{\mathrm{c}}}\sup_{(\boldsymbol{\delta},\,\boldsymbol{\Delta})\in B_{\mathscr{A}}(r_*)}\Big|\mathbb{E}\big[S_j^n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta})-S_j^n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\big]\Big|.\end{aligned}$$

By the mean value theorem, it can be seen that

$$\begin{aligned}&\max_{j\in\mathscr{A}^{\mathrm{c}}}\sup_{(\boldsymbol{\delta},\,\boldsymbol{\Delta})\in B_{\mathscr{A}}(r_*)}\Big|\mathbb{E}\big[S_j^n(\boldsymbol{\alpha}^*+\boldsymbol{\delta},\boldsymbol{\beta}^*+\boldsymbol{\Delta})-S_j^n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)\Big|\\&\le\frac{1}{nK}\bar{f}\,M_{\mathscr{A}^{\mathrm{c}}}\sup_{(\boldsymbol{\delta},\,\boldsymbol{\Delta})\in B_{\mathscr{A}}(r_*)}\sum_{i=1}^{n}\sum_{k=1}^{K}|\delta_k+\mathbf{x}_{i\mathscr{A}}^{\mathrm{T}}\boldsymbol{\Delta}_{\mathscr{A}}|\le\bar{f}\,M_{\mathscr{A}^{\mathrm{c}}}\overline{\mu}^{1/2}r_*:=B_2.\end{aligned}$$

Note that $2B=a_1\lambda-B_1-B_2$. It follows that

$$\begin{aligned}\Pr(\mathcal{E}_2^{\mathrm{c}})&\le\Pr\big((\widehat{\boldsymbol{\delta}}^{\mathrm{o}},\widehat{\boldsymbol{\Delta}}^{\mathrm{o}})\notin B_{\mathscr{A}}(r_*)\big)+\Pr\Big(\max_{i\in\mathscr{A}^{\mathrm{c}}}\gamma_j(r_*)\ge B\Big)\\&\quad+\Pr\Big(\max_{j\in\mathscr{A}^{\mathrm{c}}}|S_j^n(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)|\ge B\Big).\end{aligned}$$

Note that $r_*\sqrt{(s+1)M_{\mathscr{A}}}\le\mathcal{U}_0$. By similar arguments in (ii), it can be shown that

$$\Pr\big((\widehat{\boldsymbol{\delta}}^{\mathrm{o}},\widehat{\boldsymbol{\Delta}}^{\mathrm{o}})\notin B_{\mathscr{A}}(r_*)\big)\le\exp\Big(-\frac{nt_*^2}{32\overline{\mu}r_*^2}\Big),$$

where $t_* = \underline{f}\,\mu r_*^2/(4K) - 2r_*\sqrt{(s+1)M_{\mathscr{A}}/n}$. Applying Hoeffding's inequality, we obtain

$$\Pr\Big(\max_{j\in\mathscr{A}^{\mathrm{c}}} |S_j^n(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)| \geq B\Big) \leq 2(p-s)\exp\bigg(-\frac{2nB^2}{M_0}\bigg).$$

Lastly, we apply Lemma 4.7 to obtain the bound on $\Pr\Big(\max_{j\in\mathscr{A}^{\mathrm{c}}} \gamma_j(r_*) \geq B\Big)$. Let $\psi = 4r_*/n^2$. It can be shown that the ψ-covering number of $B_{\mathscr{A}}(r_*)$ satisfies

$$\Big(\frac{r_*}{\psi}\Big)^{K+s} \leq N_\psi \leq \Big(\frac{2r_*+\psi}{\psi}\Big)^{K+s} \leq n^{2(K+s)},\ n \geq 2.$$

By Lemma 4.7, we have

$$\begin{aligned}\Pr\Big(\max_{j\in\mathscr{A}^{\mathrm{c}}} \gamma_j(r_*) \geq B\Big) &\leq 2(p-s)N_\psi \exp\bigg(-\frac{nB^2}{8\bar{f}M_{\mathscr{A}^{\mathrm{c}}}^2\overline{\mu}^{1/2}r_* + \frac{8}{3}M_{\mathscr{A}^{\mathrm{c}}}B}\bigg)\\ &\quad + 2(p-s)N_\psi \exp\bigg(-\frac{nB_0^2}{2\bar{f}M_{\mathscr{A}^{\mathrm{c}}}^2\big((s+1)M_{\mathscr{A}}\big)^{1/2}\psi + \frac{4}{3}M_{\mathscr{A}^{\mathrm{c}}}B_0}\bigg),\end{aligned}$$

where $B_0 = \big[B/2 - 2\bar{f}M_{\mathscr{A}^{\mathrm{c}}}\big((s+1)M_{\mathscr{A}}\big)^{1/2}\psi\big]_+$. This completes the proof. □

Chapter 5

Conclusion

5.1 Discussion

In this dissertation, we considered three types of unconventional sparse penalized regressions, that is, the (coupled) sparse asymmetric least squares, the penalized quantile regression and the penalized composite quantile regression, and studied both the theoretical and numerical properties of those methods. The proposed methods can be readily applied to analyze high-dimensional data that exhibit heteroscedasticity or heavy-tailedness.

In Chapter 2, we systematically extended the asymmetric least squares regression to high dimensions. Through lasso and folded concave penalties, the penalized asymmetric least squares was shown to enjoy the strong oracle property. We also proposed efficient coordinate descent algorithm for solving penalized asymmetric least squares under a unified framework. We then applied the methodology to analyze heteroscedasticity in high-dimensional data. A more calibrated method called the coupled sparse asymmetric least least squares was also studied to analyze heteroscedasticity. The advantage of our proposed methods is that they are computationally very efficient, while are still capable of handling heteroscedasticity.

In Chapter 3, we proposed pADMM and scdADMM to solve the high-dimensional sparse penalized quantile regression. The computational efficiency of our algorithms have been tested with extensive numerical experiments. We note that both pADMM and scdADMM

algorithms can be readily modified to solve the elastic net penalized quantile regression. Our R package `FHDQR` includes functions for solving the weighted elastic net penalized quantile regression. We present the algorithmic details of the weighted elastic net penalized quantile regression in Appendix B.

Computational burden is a real issue that prevents the data analyst from using the high-dimensional quantile regression as frequently as the sparse penalized least squares. Our algorithms and R package drastically alleviate this burden and hence make sparse quantile regression a part of the standard toolbox for data analysts.

In Chapter 4, we studied the sparse penalized CQR under various regularization. In particular, we established the estimation consistency of the CQR lasso estimator. Through the LLA algorithm, we showed that the CQR oracle estimator could be achieved via folded concave penalized CQR. Our theoretical analysis remains valid even when the dimensionality is ultrahigh, that is, $p = \mathcal{O}(n^{\gamma})$ with $0 < \gamma < 1$.

We also developed a fast sparse coordinate descent ADMM (scdADMM) algorithm for solving the weighted L_1-penalized CQR. Numerical studies proved the efficiency of the algorithm. We note that our algorithm can be readily modified to solve adaptive elastic net penalized CQR. Specifically, a similar algorithm can be devised to solve the weighted elastic net penalized CQR:

$$\min_{\boldsymbol{\beta},\alpha_1,\ldots,\alpha_K} \frac{1}{nK} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k}(y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \lambda_1 \sum_{j=1}^{p} d_j |\beta_j| + \frac{\lambda_2}{2} \sum_{j=1}^{p} h_j \beta_j^2, \qquad (5.1)$$

where $\lambda_1,\ \lambda_2 \geq 0$, $d_j \geq 0$ and $h_j \geq 0$, $j = 1, \ldots, p$. For the sake of brevity, we relegate the optimization algorithm for solving (5.1) to Appendix C. We also provide an R package called `FHDCQR` (fast high-dimensional composite quantile regression) that implements the above algorithm.

5.2 Future Work

This dissertation provides an aspiring start for the methodological and numerical research on unconventional sparse penalized regressions for high-dimensional data. On one hand, the methods can be extended to work with censored and longitudinal data. Longitudinal studies are often conducted in observational studies and clinical trials in a variety of fields such as medicine, biology, epidemiology, sociology and economics, in which censored observations can be very common. Examples include the HIV study (Hughes, 1999), virologic study (Thompson et al., 2010), and many others. A typical approach to analyzing such data is through mixed models. However, the maximum likelihood estimators may fail to be consistent when the random-effects density is misspecified (Vock et al., 2012). Unconventional regressions can provide new perspectives for handling such data (Wang and Fygenson, 2009). When gene markers are used to augment the studies, the problem can become really high-dimensional. Our methodologies have the potential to deal with such censored longitudinal data with very minimal modifications.

On the other hand, from the optimization point of view, the ADMM algorithms have the potential to deal with very large-scale data. When n is moderate and p is large, we have shown that the ADMM algorithms can be really efficient using a single machine. However, when n is very big, it may not even be possible to store the data on a single machine. To that end, we point out that the ADMM algorithm can be carried out in a distributed manner, where the data are split into smaller subsets and stored on multiple slave machines. For the subset data chunks, we fit them with slave machines simultaneously using our proposed algorithms for the penalized regression. The problem only has small to moderate n for each slave machine because of the split and thus can be solved very efficiently.

References

Bai, Z. and Wu, Y. (1994). Limiting behavior of M-estimators of regression coefficients in high dimensional linear models I. scale dependent case. *Journal of Multivariate Analysis*, 51(2):211–239.

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

Belloni, A. and Chernozhukov, V. (2011). ℓ_1-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39(1):82–130.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009). Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732.

Bien, J., Taylor, J., and Tibshirani, R. (2013). A lasso for hierarchical interactions. *The Annals of Statistics*, 41(3):1111–1141.

Bogdan, M., Berg, E. v. d., Su, W., and Candes, E. (2013). Statistical estimation and testing via the sorted ℓ_1 norm. *arXiv preprint arXiv:1310.1969v2*.

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

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer.

Candes, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351.

Chambers, R. and Tzavidis, N. (2006). M-quantile models for small area estimation. *Biometrika*, 93(2):255–268.

Chen, C. (2007). A finite smoothing algorithm for quantile regression. *Journal of Computational and Graphical Statistics*, 16(1):136–164.

Chen, X., Bai, Z., and Zao, L. (1990). Asymptotic normality of minimum L_1-norm estimates in linear models. *Science in China Series A: Mathematics, Physics, Astronomy*, 33(11):1311–1328.

Chiang, A. P., Beck, J. S., Yen, H.-J., Tayeh, M. K., Scheetz, T. E., Swiderski, R. E., Nishimura, D. Y., Braun, T. A., Kim, K.-Y. A., Huang, J., et al. (2006). Homozygosity mapping with SNP arrays identifies TRIM32, an E3 ubiquitin ligase, as a Bardet–Biedl syndrome gene (BBS11). *Proceedings of the National Academy of Sciences*, 103(16):6287–6292.

Daye, Z. J., Chen, J., and Li, H. (2012). High-dimensional heteroscedastic regression with an application to eQTL data analysis. *Biometrics*, 68(1):316–326.

Donoho, D. L. et al. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, 1:1–32.

Eckstein, J. (1994). Some saddle-function splitting methods for convex programming. *Optimization Methods and Software*, 4(1):75–83.

Efron, B. (1991). Regression percentiles using asymmetric squared loss. *Statistica Sinica*, 55(1):93–125.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Eilers, P. H. and Boelens, H. F. (2005). Baseline correction with asymmetric least squares smoothing. *Leiden University Medical Centre Report*.

Fan, J., Fan, Y., and Barut, E. (2014a). Adaptive robust variable selection. *The Annals of Statistics*, 42(1):324–351.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fan, J. and Lv, J. (2011). Nonconcave penalized likelihood with NP-dimensionality. *IEEE Transactions on Information Theory*, 57(8):5467–5484.

Fan, J., Xue, L., and Zou, H. (2014b). Strong oracle optimality of folded concave penalized estimation. *The Annals of Statistics*, 42(3):819–849.

Fazel, M., Pong, T. K., Sun, D., and Tseng, P. (2013). Hankel matrix rank minimization with applications to system identification and realization. *SIAM Journal on Matrix Analysis and Applications*, 34(3):946–977.

Frank, L. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Goldstein, T. and Osher, S. (2009). The split Bregman method for L1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343.

Haagerup, U. (1981). The best constants in the Khintchine inequality. *Studia Mathematica*, 3(70):231–283.

He, B., Liao, L.-Z., Han, D., and Yang, H. (2002). A new inexact alternating directions method for monotone variational inequalities. *Mathematical Programming*, 92(1):103–118.

He, B. and Yuan, X. (2015). On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577.

He, X. and Shao, Q.-M. (2000). On parameters of increasing dimensions. *Journal of Multivariate Analysis*, 73(1):120–135.

Hong, M., Wang, X., Razaviyayn, M., and Luo, Z.-Q. (2013). Iteration complexity analysis of block coordinate descent methods. *arXiv preprint arXiv:1310.6957*.

Huang, J., Ma, S., and Zhang, C.-H. (2008). Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18(4):1603–1618.

Huang, J. and Zhang, C.-H. (2012). Estimation and selection via absolute penalized convex minimization and its multistage adaptive applications. *The Journal of Machine Learning Research*, 13(1):1839–1864.

Hughes, J. P. (1999). Mixed effects models with censored data with application to HIV RNA levels. *Biometrics*, 55(2):625–629.

Hunter, D. R. and Lange, K. (2000). Quantile regression via an mm algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60–77.

Kadkhodaie, M., Sanjabi, M., and Luo, Z.-Q. (2014). On the linear convergence of the approximate proximal splitting method for non-smooth convex optimization. *Journal of the Operations Research Society of China*, 2(2):123–141.

Knight, K. (1998). Limiting distributions for L_1 regression estimators under general conditions. *The Annals of Statistics*, 26(2):755–770.

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press, Cambridge, United Kingdom.

Koenker, R. (2016). *quantreg: Quantile Regression*. R package version 5.29.

Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica: Journal of the Econometric Society*, 46(1):33–50.

Koenker, R. and Bassett, G. (1982). Robust tests for heteroscedasticity based on regression quantiles. *Econometrica: Journal of the Econometric Society*, 50(1):43–61.

Koenker, R. and Ng, P. (2005). A Frisch-Newton algorithm for sparse quantile regression. *Acta Mathematicae Applicatae Sinica*, 21(2):225–236.

Koenker, R. and Zhao, Q. (1994). L-estimatton for linear heteroscedastic models. *Journal of Nonparametric Statistics*, 3(3-4):223–235.

Kuan, C.-M., Yeh, J.-H., and Hsu, Y.-C. (2009). Assessing value at risk with CARE, the conditional autoregressive expectile models. *Journal of Econometrics*, 150(2):261–270.

Ledoux, M. L. M. and Talagrand, M. (1991). *Probability in Banach Spaces, Isoperimetry and Processes*. Springer-Verlag Berlin Heidelberg, 1 edition.

Li, Y. and Zhu, J. (2008). L_1-norm quantile regression. *Journal of Computational and Graphical Statistics*, 17(1):163–185.

Luo, Z.-Q. and Tseng, P. (1992). On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization*, 30(2):408–425.

Luo, Z.-Q. and Tseng, P. (1993). Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178.

Lv, J. and Fan, Y. (2009). A unified approach to model selection and sparse recovery using regularized least squares. *The Annals of Statistics*, 37(6A):3498–3528.

Lv, S., He, X., and Wang, J. (2016). A unified penalized method for sparse additive quantile models: an RKHS approach. *Annals of the Institute of Statistical Mathematics*, pages 1–27.

Meier, L., Van de Geer, S., Bühlmann, P., et al. (2009). High-dimensional additive modeling. *The Annals of Statistics*, 37(6B):3779–3821.

Negahban, S. N., Ravikumar, P., Wainwright, M. J., and Yu, B. (2012). A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557.

Newey, W. K. and Powell, J. L. (1987). Asymmetric least squares estimation and testing. *Econometrica*, 55(4):819–847.

O'Donoghue, B., Stathopoulos, G., and Boyd, S. (2013). A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 21(6):2432–2442.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231.

Peng, B. (2016). *QICD: Estimate the Coefficients for Non-Convex Penalized Quantile Regression Model by using QICD Algorithm*. R package version 1.1.1.

Peng, B. and Wang, L. (2015). An iterative coordinate descent algorithm for high-dimensional nonconvex penalized quantile regression. *Journal of Computational and Graphical Statistics*, 24(3):676–694.

Pollard, D. (1990). Empirical processes: Theory and applications. *NSF-CBMS Regional Conference Series in Probability and Statistics*, 2:i–86.

Pollard, D. (1991). Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, 7(02):186–199.

Rudelson, M. and Vershynin, R. (2013). Hanson-Wright inequality and sub-gaussian concentration. *Electron. Commun. Probab.*, 18(82):1–9.

Salvati, N., Tzavidis, N., Pratesi, M., and Chambers, R. (2012). Small area estimation via M-quantile geographically weighted regression. *Test*, 21(1):1–28.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., Philp, A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., et al. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434.

Taylor, J. W. (2008). Estimating value at risk and expected shortfall using expectiles. *Journal of Financial Econometrics*, 6(2):231–252.

Thompson, A. J., Muir, A. J., Sulkowski, M. S., Ge, D., Fellay, J., Shianna, K. V., Urban, T., Afdhal, N. H., Jacobson, I. M., Esteban, R., et al. (2010). Interleukin-28B polymorphism improves viral kinetics and is the strongest pretreatment predictor of sustained virologic response in genotype 1 hepatitis C virus. *Gastroenterology*, 139(1):120–129.

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

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494.

van der Vaart, A. and Wellner, J. (1996). *Weak convergence and empirical processes*. Springer, New York.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027v7*.

Vock, D. M., Davidian, M., Tsiatis, A. A., and Muir, A. J. (2012). Mixed model analysis of censored longitudinal data with flexible random-effects density. *Biostatistics*, 13(1):61.

Wang, H. J. and Fygenson, M. (2009). Inference for censored quantile regression models in longitudinal studies. *The Annals of Statistics*, 37(2):756–781.

Wang, L. (2013). The L_1 penalized LAD estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135 – 151.

Wang, L., Kim, Y., Li, R., et al. (2013). Calibrating nonconvex penalized regression in ultra-high dimension. *The Annals of Statistics*, 41(5):2505–2536.

Wang, L., Wu, Y., and Li, R. (2012). Quantile regression for analyzing heterogeneity in ultra-high dimension. *Journal of the American Statistical Association*, 107(497):214–222.

Welsh, A. (1989). On M-processes and M-estimation. *The Annals of Statistics*, 17(1):337–361.

Wu, T. T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244.

Wu, Y. and Liu, Y. (2009). Variable selection in quantile regression. *Statistica Sinica*, 19:801–817.

Xie, S., Zhou, Y., and Wan, A. T. (2014). A varying-coefficient expectile model for estimating Value at Risk. *Journal of Business & Economic Statistics*, 32(4):576–592.

Xue, L., Ma, S., and Zou, H. (2012). Positive-definite ℓ_1-penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 107(500):1480–1491.

Yang, J., Meng, X., and Mahoney, M. W. (2013). Quantile regression for large-scale applications. *arXiv preprint arXiv:1305.0087v3*.

Yang, Y. and Zou, H. (2013). An efficient algorithm for computing the hhsvm and its generalizations. *Journal of Computational and Graphical Statistics*, 22(2):396–415.

Ye, F. and Zhang, C.-H. (2010). Rate minimaxity of the Lasso and Dantzig selector for the ℓ_q loss in ℓ_r balls. *The Journal of Machine Learning Research*, 11:3519–3540.

Yi, C. (2016). *hqreg: Regularization Paths for Huber Loss Regression and Quantile Regression Penalized by Lasso or Elastic-Net*. R package version 1.3.

Yi, C. and Huang, J. (2016). Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression. *Journal of Computational and Graphical Statistics*, (just-accepted).

Yin, W., Osher, S., Goldfarb, D., and Darbon, J. (2008). Bregman iterative algorithms for l1-minimization with applications to compressed sensing. *SIAM J. Imaging Sci*, 1(1):143–168.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942.

Zhang, H., Jiang, J., and Luo, Z.-Q. (2013). On the linear convergence of a proximal gradient method for a class of nonsmooth convex minimization problems. *Journal of the Operations Research Society of China*, 1(2):163–186.

Zhang, T., Zou, H., et al. (2014). Sparse precision matrix estimation via lasso penalized D-trace loss. *Biometrika*, 101(1):103–120.

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

Zhao, Z. and Xiao, Z. (2014). Efficient regressions via optimally combining quantile information. *Econometric theory*, 30(6):1272–1314.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4):1509–1533.

Zou, H. and Yuan, M. (2008). Composite quantile regression and the oracle model selection theory. *The Annals of Statistics*, 36(3):1108–1126.

Appendix A

Iteration complexity analysis of the SALES algorithm

Notation. For a vector $\mathbf{v} = (v_1, \ldots, v_d)^{\mathrm{T}} \in \mathbb{R}^d$ and a univariate function $u(\cdot)$, we write $u(\mathbf{v}) = (u(v_1), \ldots, u(v_d))^{\mathrm{T}}$. Also, denote the subvector of $\mathbf{v}$ with its kth component removed by $\mathbf{v}_{-k} = (v_1, \ldots, v_{k-1}, v_{k+1}, \ldots, v_d)^{\mathrm{T}}$ and recover $\mathbf{v}$ from $\mathbf{v}_{-k}$ by $\mathbf{v} = [v_k, \mathbf{v}_{-k}]$. The column vector of all ones will be denoted by $\mathbf{1}_n \in \mathbb{R}^n$. The L_2-norm of $\mathbf{v}$ is denoted by $\|\mathbf{v}\| = \left(\sum_{k=1}^d v_k^2\right)^{1/2}$. We also let ∂h be the sub-differential of a nonsmooth convex function h (see e.g., Bertsekas, 1999).

Iteration Complexity Analysis. For ease of exposition, let us rewrite (2.5) as the following unconstrained optimization problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) + \sum_{k=1}^{p} h_k(\beta_k) \tag{A.1}$$

where $g(\boldsymbol{\beta}) = n^{-1}\mathbf{1}_n^{\mathrm{T}}\Psi_\tau(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ is smooth convex in $\boldsymbol{\beta} \in \mathbb{R}^p$, while $h_k(\beta_k) = w_k|\beta_k|$ is nonsmooth convex in β_k for each $k = 1, \ldots, p$. For ease of exposition, also let $h(\boldsymbol{\beta}) = \sum_{k=1}^p h_k(\beta_k)$. Note that $\nabla g(\boldsymbol{\beta}) = -n^{-1}\mathbf{X}^{\mathrm{T}}\Psi'_\tau(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ with

$$\nabla_k g(\boldsymbol{\beta}) = -n^{-1}\sum_{i=1}^{n} \Psi'_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})x_{ik} = -n^{-1}X_k^{\mathrm{T}}\Psi'_\tau(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}),\ k = 1, \ldots, p.$$

Let $\rho_{\max} = \lambda_{\max}(n^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{X}) = \lambda_{\max}(n^{-1}\mathbf{X}\mathbf{X}^{T})$. It follows that

$$\begin{aligned}\|\nabla g(\boldsymbol{\beta}) - \nabla g(\boldsymbol{\beta}')\| &= n^{-1}\|\mathbf{X}^{\mathrm{T}}(\Psi_\tau'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) - \Psi_\tau'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}'))\| \\ &\le (\rho_{\max}/n)^{1/2}\|\Psi_\tau'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) - \Psi_\tau'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}')\| \\ &\le 2\bar{c}(\rho_{\max}/n)^{1/2}\|\mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}')\| \\ &\le 2\bar{c}\rho_{\max}\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|,\end{aligned}$$

which implies that the gradient of $g(\cdot)$ is uniformly Lipschitz continuous with Lipschitz constant $L = 2\bar{c}\rho_{\max}$. When restricted to each coordinate, we have

$$|\nabla_k g([\beta_k, \boldsymbol{\beta}_{-k}]) - \nabla_k g([\beta_k', \boldsymbol{\beta}_{-k}])| \le 2n^{-1}\bar{c}\|X_k\|^2|\beta_k - \beta_k'|,\ k = 1, \ldots, p,$$

which implies that the gradient of $g(\cdot)$ is coordinate-wise uniformly Lipschitz continuous with Lipschitz constants $L_k = 2n^{-1}\bar{c}\|X_k\|^2$, $k = 1, \ldots, p$.

In the cyclic coordinate descent algorithm, let $\boldsymbol{\beta}^r$ be the update of $\boldsymbol{\beta}$ after the rth cycle, $r \ge 0$. When updating β_k in the $(r+1)$th cycle using the proximal gradient method, let $\beta_k^{r,s}$ be the update after the sth iteration, $s \ge 0$. Note that since the proximal gradient algorithm for updating β_k in the $(r+1)$th cycle is initialized by β_k^r, we have $\beta_k^{r,0} = \beta_k^r$. Suppose the proximal gradient algorithm converges (namely, the convergence criterion for the proximal gradient algorithm is satisfied) after S_k^r iterations. For some large and fixed integer $B \in \mathbb{Z}^+$, if $S_k^r \le B$, we let $\beta_k^{r+1} := \beta_k^{r,S_k^r}$. Otherwise, let $\beta_k^{r+1} := \beta_k^{r,B}$, which means that the proximal gradient algorithm will be terminated after at most B iterations. For ease of notation, denote

$$\begin{aligned}\mathbf{b}_k^{r+1} &= (\beta_1^{r+1}, \ldots, \beta_{k-1}^{r+1}, \beta_k^r, \beta_{k+1}^r, \ldots, \beta_p^r)^{\mathrm{T}},\ k = 1, \ldots, p, \\ \mathbf{b}_{-k}^{r+1} &= (\beta_1^{r+1}, \ldots, \beta_{k-1}^{r+1}, \beta_{k+1}^r, \ldots, \beta_p^r)^{\mathrm{T}},\ k = 1, \ldots, p.\end{aligned}$$

Clearly we have $\mathbf{b}_1^{r+1} = \boldsymbol{\beta}^r$ and $\mathbf{b}_{p+1}^{r+1} = \boldsymbol{\beta}^{r+1}$. Note that in the proximal gradient update,

$$\beta_k^{r,s+1} := \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^{r,s} - L_k^{-1}\nabla_k g([\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]))$$

is equivalent to

$$\beta_k^{r,s+1} := \arg\min_{\beta_k} u_k(\beta_k; [\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k),$$

where the proximity operator **prox** does the soft-thresholding (Parikh and Boyd, 2013) and

$$u_k(\beta_k; [\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]) = g([\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]) + \nabla_k g([\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}])(\beta_k - \beta_k^{r,s}) + \frac{L_k}{2}(\beta_k - \beta_k^{r,s})^2$$

is a quadratic majorization function of $\hat{g}(\beta_k; \mathbf{b}_{-k}^{r+1}) := g([\beta_k, \mathbf{b}_{-k}^{r+1}])$ at $\beta_k^{r,s}$. It is easy to see that $u_k(\beta_k; [\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}])$ is strongly convex in β_k. By the optimality of $\beta_k^{r,s+1}$, there exists $\zeta_k^{r,s+1} \in \partial h_k(\beta_k^{r,s+1})$ such that

$$(\nabla u_k(\beta_k^{r,s+1}; [\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]) + \zeta_k^{r,s+1})(\beta_k - \beta_k^{r,s+1}) \geq 0, \ \forall \beta_k. \tag{A.2}$$

Our analysis will be divided into three parts: the sufficient descent step, the cost-to-go estimate step, and the local error bound step. Similar techniques can be found in Luo and Tseng (1992), Luo and Tseng (1993), Zhang et al. (2013) and Hong et al. (2013).

Sufficient Descent. Consider the proximal gradient method applied to solving the following problem

$$\min_{\beta_k \in \mathbb{R}} f([\beta_k, \mathbf{b}_{-k}^{r+1}]) = g([\beta_k, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k),$$

we have by (A.2)

$$
\begin{aligned}
f(\mathbf{b}_k^{r+1}) - f(\mathbf{b}_{k+1}^{r+1}) &= f([\beta_k^r, \mathbf{b}_{-k}^{r+1}]) - f([\beta_k^{r+1}, \mathbf{b}_{-k}^{r+1}]) \\
&= \sum_{s=0}^{\min(S_k^r, B)-1} \left[f([\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]) - f([\beta_k^{r,s+1}, \mathbf{b}_{-k}^{r+1}]) \right] \\
&\geq \sum_{s=0}^{\min(S_k^r, B)-1} \Big[u_k(\beta_k^{r,s}; [\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]) - u_k(\beta_k^{r,s+1}; [\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]) \\
&\qquad + h_k(\beta_k^{r,s}) - h_k(\beta_k^{r,s+1}) \Big] \\
&= \sum_{s=0}^{\min(S_k^r, B)-1} \Big[\nabla_k u_k(\beta_k^{r,s+1}; [\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}])(\beta_k^{r,s} - \beta_k^{r,s+1}) \\
&\qquad + h_k(\beta_k^{r,s}) - h_k(\beta_k^{r,s+1}) + \frac{L_k}{2}(\beta_k^{r,s} - \beta_k^{r,s+1})^2 \Big] \\
&\geq \sum_{s=0}^{\min(S_k^r, B)-1} \Big[(\nabla_k u_k(\beta_k^{r,s+1}; [\beta_k^{r,s}, \mathbf{b}_{-k}^{r+1}]) + \zeta_k^{r,s+1})(\beta_k^{r,s} - \beta_k^{r,s+1}) \\
&\qquad + \frac{L_k}{2}(\beta_k^{r,s} - \beta_k^{r,s+1})^2 \Big] \\
&\geq \sum_{s=0}^{\min(S_k^r, B)-1} \frac{L_k}{2}(\beta_k^{r,s} - \beta_k^{r,s+1})^2 \geq \frac{L_k}{2B}\big(\beta_k^{r,0} - \beta_k^{r,\min(S_k^r, B)}\big)^2 \\
&= \frac{L_k}{2B}(\beta_k^r - \beta_k^{r+1})^2.
\end{aligned}
$$

It follows that

$$
f(\boldsymbol{\beta}^r) - f(\boldsymbol{\beta}^{r+1}) = \sum_{k=1}^{p} \left[f(\mathbf{b}_k^{r+1}) - f(\mathbf{b}_{k+1}^{r+1}) \right] \geq \underline{L}(2B)^{-1} \|\boldsymbol{\beta}^r - \boldsymbol{\beta}^{r+1}\|^2, \quad \text{(A.3)}
$$

where $\underline{L} = \min_{1 \leq k \leq p} L_k = 2n^{-1}\bar{c} \min_{1 \leq k \leq p} \|X_k\|^2$.

Cost-to-go Estimate. For notational convenience, denote $\mathscr{C}_k^r = \min(S_k^r, B)$. Let $\mathscr{X}^* := \{\boldsymbol{\beta}^* | f(\boldsymbol{\beta}^*) = \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})\}$ be the optimal solution set of problem (A.1). Let $\bar{\boldsymbol{\beta}}^r \in \mathscr{X}^*$ be the point in $\mathscr{X}^*$ such that $\mathrm{d}_{\mathscr{X}^*}(\boldsymbol{\beta}^r) := \min_{\boldsymbol{\beta} \in \mathscr{X}^*} \|\boldsymbol{\beta} - \boldsymbol{\beta}^r\| = \|\bar{\boldsymbol{\beta}}^r - \boldsymbol{\beta}^r\|$. By the optimality

of

$$\beta_k^{r+1} := \beta_k^{r,\mathscr{C}_k^r} = \arg\min_{\beta_k \in \mathbb{R}} u_k(\beta_k; [\beta_k^{r,\mathscr{C}_k^r - 1}, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k),$$

one has

$$\begin{aligned} &h(\beta_k^{r+1}) - h(\bar{\beta}_k^r) + \nabla_k g([\beta_k^{r,\mathscr{C}_k^r - 1}, \mathbf{b}_{-k}^{r+1}])(\beta_k^{r+1} - \bar{\beta}_k^r) \\ &\leq \frac{L_k}{2}(\bar{\beta}_k^r - \beta_k^{r,\mathscr{C}_k^r - 1})^2 \leq L_k\big[(\bar{\beta}_k^r - \beta_k^r)^2 + (\beta_k^r - \beta_k^{r,\mathscr{C}_k^r - 1})^2\big]. \end{aligned}$$

By the mean value theorem, there exists $\lambda \in [0, 1]$ and $\boldsymbol{\xi}^r = \lambda \boldsymbol{\beta}^{r+1} + (1 - \lambda)\bar{\boldsymbol{\beta}}^r$ such that

$$g(\boldsymbol{\beta}^{r+1}) - g(\bar{\boldsymbol{\beta}}^r) = \langle \nabla g(\boldsymbol{\xi}^r), \boldsymbol{\beta}^{r+1} - \bar{\boldsymbol{\beta}}^r \rangle.$$

It follows that

$$\begin{aligned} &f(\boldsymbol{\beta}^{r+1}) - f(\bar{\boldsymbol{\beta}}^r) = g(\boldsymbol{\beta}^{r+1}) - g(\bar{\boldsymbol{\beta}}^r) + \sum_{k=1}^p \big[h_k(\beta_k^{r+1}) - h_k(\bar{\beta}_k^r)\big] \\ &= \sum_{k=1}^p \big[\nabla_k g(\boldsymbol{\xi}^r)(\beta_k^{r+1} - \bar{\beta}_k^r) + h_k(\beta_k^{r+1}) - h_k(\bar{\beta}_k^r)\big] \\ &= \sum_{k=1}^p \big[\nabla_k g([\beta_k^{r,\mathscr{C}_k^r - 1}, \mathbf{b}_{-k}^{r+1}])(\beta_k^{r+1} - \bar{\beta}_k^r) + h_k(\beta_k^{r+1}) - h_k(\bar{\beta}_k^r) \\ &\qquad + \big(\nabla_k g(\boldsymbol{\xi}^r) - \nabla_k g([\beta_k^{r,\mathscr{C}_k^r - 1}, \mathbf{b}_{-k}^{r+1}])\big)(\beta_k^{r+1} - \bar{\beta}_k^r)\big] \\ &\leq \sum_{k=1}^p \big[L_k(\bar{\beta}_k^r - \beta_k^r)^2 + L_k(\beta_k^r - \beta_k^{r,\mathscr{C}_k^r - 1})^2 \\ &\qquad + \big(\nabla_k g(\boldsymbol{\xi}^r) - \nabla_k g([\beta_k^{r,\mathscr{C}_k^r - 1}, \mathbf{b}_{-k}^{r+1}])\big)(\beta_k^{r+1} - \bar{\beta}_k^r)\big]. \end{aligned}$$

For each coordinate β_k, by the sufficient descent property of the proximal gradient algorithm, it can be shown that $|\beta_k^{r,s+1} - \beta_k^{r,s}| \to 0$ as $s \to \infty$. Therefore, for sufficiently large $\mathscr{C}_k^r$, we have

$$|\beta_k^{r,\mathscr{C}_k^r} - \beta_k^{r,\mathscr{C}_k^r - 1}| \leq |\beta_k^{r,\mathscr{C}_k^r} - \beta_k^{r,0}| = |\beta_k^{r+1} - \beta_k^r|. \tag{A.4}$$

Together with the fact that $\nabla g(\cdot)$ is Lipschitz continuous, this implies

$$
\begin{aligned}
&\Big(\sum_{k=1}^{p}\big(\nabla_k g(\boldsymbol{\xi}^r) - \nabla_k g([\beta_k^{r,\mathscr{C}_k^r-1}, \mathbf{b}_{-k}^{r+1}])\big)(\beta_k^{r+1} - \bar{\beta}_k^r)\Big)^2 \\
&\le \Big(\sum_{k=1}^{p}\|\nabla g(\boldsymbol{\xi}^r) - \nabla g([\beta_k^{r,\mathscr{C}_k^r-1}, \mathbf{b}_{-k}^{r+1}])\|^2\Big)\Big(\sum_{k=1}^{p}(\beta_k^{r+1} - \bar{\beta}_k^r)^2\Big) \\
&\le \Big(\sum_{k=1}^{p} L^2\|\boldsymbol{\xi}^r - [\beta_k^{r,\mathscr{C}_k^r-1}, \mathbf{b}_{-k}^{r+1}]\|^2\Big)\|\boldsymbol{\beta}^{r+1} - \bar{\boldsymbol{\beta}}^r\|^2 \\
&= \Big(\sum_{k=1}^{p} L^2\|\lambda(\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r) + (1-\lambda)(\bar{\boldsymbol{\beta}}^r - \boldsymbol{\beta}^r) + \boldsymbol{\beta}^r - [\beta_k^{r,\mathscr{C}_k^r-1}, \mathbf{b}_{-k}^{r+1}]\|^2\Big) \\
&\qquad \cdot 2(\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 + \|\boldsymbol{\beta}^r - \bar{\boldsymbol{\beta}}^r\|^2) \\
&\le 16(p+1)L^2\big[\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 + \|\boldsymbol{\beta}^r - \bar{\boldsymbol{\beta}}^r\|^2\big]^2 \\
&= 16(p+1)L^2\big[\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 + \mathrm{d}^2_{\mathscr{X}^*}(\boldsymbol{\beta}^r)\big]^2.
\end{aligned}
$$

It follows that

$$
f(\boldsymbol{\beta}^{r+1}) - f(\bar{\boldsymbol{\beta}}^r) \le (4L\sqrt{p+1} + 2\bar{L})\big[\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 + \mathrm{d}^2_{\mathscr{X}^*}(\boldsymbol{\beta}^r)\big], \tag{A.5}
$$

where $\bar{L} = \max_{1\le k\le p} L_k = 2n^{-1}\bar{c}\max_{1\le k\le p}\|X_k\|^2$.

In practice, (A.4) can be easily satisfied if we monitor the proximal gradient step in a way such that the algorithm is terminated after s_k^r iterations if we have $|\beta_k^{r,s_k^r} - \beta_k^{r,s_k^r-1}| \le |\beta_k^{r,s_k^r} - \beta_k^{r,0}|$. This is possible since we can always take $s_k^r = 1$, which corresponds to the one-step update scheme in Yang and Zou (2013).

Local error bound. Let $\mathbf{d}_{\mathscr{X}^*}(\boldsymbol{\beta}) \equiv \min_{\boldsymbol{\beta}^*\in\mathscr{X}^*}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}\|$. Note that the function $p(\mathbf{z}) = n^{-1}\mathbf{1}_n^{\mathrm{T}}\Psi_\tau(\mathbf{y}-\mathbf{z})$ is strongly convex in $\mathbf{z}\in\mathbb{R}^n$. We can see that $g(\boldsymbol{\beta}) = p(\mathbf{X}\boldsymbol{\beta})$. It follows from Zhang et al. (2013) that for any $\xi \ge \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$, there exist $\kappa, \varepsilon > 0$ such that

$$
\mathrm{d}_{\mathscr{X}^*}(\boldsymbol{\beta}) \le \kappa\|\boldsymbol{\beta} - \mathbf{prox}_h(\boldsymbol{\beta} - \nabla g(\boldsymbol{\beta}))\|, \tag{A.6}
$$

for all $\boldsymbol{\beta}$ such that $\|\boldsymbol{\beta} - \mathbf{prox}_h(\boldsymbol{\beta} - \nabla g(\boldsymbol{\beta}))\| \le \varepsilon$ and $f(\boldsymbol{\beta}) \le \xi$.

As a summary, we show in the following theorem that the SALES algorithm converges at least linearly.

Theorem A.1
The SALES algorithm (Algorithm 1) converges at least linearly to a solution in $\mathscr{X}^*$. □

Proof A.1
We first show that there exists some $\sigma > 0$ such that

$$\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\| \leq \sigma \|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|, \ \forall r \geq 1. \tag{A.7}$$

For any $r \geq 1$ and any $1 \leq k \leq p$, by the optimality of

$$\beta_k^{r+1} := \beta_k^{r,\mathscr{C}_k^r} = \arg\min_{\beta_k} u_k(\beta_k; [\beta_k^{r,\mathscr{C}^r-1}, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k),$$

we have

$$\beta_k^{r+1} = \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^{r+1} - L_k^{-1}\nabla u_k(\beta_k^{r+1}; [\beta_k^{r,\mathscr{C}^r-1}, \mathbf{b}_{-k}^{r+1}])).$$

Let $\hat{L}_k = \max(1, L_k)$ and $\tilde{L}_k = \max(1, L_k^{-1})$. It follows from Lemma 4.3 of Kadkhodaie et al. (2014) that

$$\begin{aligned}
&|\beta_k^r - \mathbf{prox}_{h_k}(\beta_k^r - \nabla_k g(\boldsymbol{\beta}^r))| \leq \hat{L}_k |\beta_k^r - \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^r - L_k^{-1}\nabla_k g(\boldsymbol{\beta}^r))| \\
&\leq \hat{L}_k \big[|\beta_k^{r+1} - \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^r - L_k^{-1}\nabla_k g(\boldsymbol{\beta}^r))| + |\beta_k^{r+1} - \beta_k^r|\big] \\
&\leq \hat{L}_k \big[|\mathbf{prox}_{L_k^{-1}h_k}(\beta_k^{r+1} - L_k^{-1}\nabla u_k(\beta_k^{r+1}; [\beta_k^{r,\mathscr{C}^r-1}, \mathbf{b}_{-k}^{r+1}])) \\
&\qquad - \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^r - L_k^{-1}\nabla_k g(\boldsymbol{\beta}^r))| + |\beta_k^{r+1} - \beta_k^r|\big] \\
&\leq 2\hat{L}_k |\beta_k^{r+1} - \beta_k^r| + \hat{L}_k L_k^{-1} |\nabla u_k(\beta_k^{r+1}; [\beta_k^{r,\mathscr{C}^r-1}, \mathbf{b}_{-k}^{r+1}]) - \nabla_k g(\boldsymbol{\beta}^r)| \\
&\leq 3\hat{L}_k |\beta_k^{r+1} - \beta_k^r| + \tilde{L}_k \|\nabla g([\beta_k^{r,\mathscr{C}^r-1}, \mathbf{b}_{-k}^{r+1}]) - \nabla g(\boldsymbol{\beta}^r)\| \\
&\leq (3\hat{L}_k + L\tilde{L}_k) \|\boldsymbol{\beta}_k^{r+1} - \boldsymbol{\beta}_k^r\|.
\end{aligned}$$

It follows that

$$\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\| \leq (3\hat{L} + L\tilde{L})\sqrt{p} \|\boldsymbol{\beta}_k^{r+1} - \boldsymbol{\beta}_k^r\|,$$

where $\hat{L} = \max(1, \bar{L})$ and $\tilde{L} = \max(1, \underline{L}^{-1})$. Therefore, when we take $\sigma = (3\hat{L} + L\tilde{L})\sqrt{p}$, we get the desired result in (A.7). Note that the sufficient descent property (A.3) implies that $\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\| \to 0$ as $r \to \infty$. It follows from (A.7) that $\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r -$

$\nabla g(\boldsymbol{\beta}^r))\| \to 0$ as $r \to \infty$. Thus, by (A.6) we have $\mathrm{d}_{\mathscr{X}^*}(\boldsymbol{\beta}^r) \to 0$ as $r \to \infty$. Consequently, from (A.5) it implies that $f(\boldsymbol{\beta}^r) \to f^* := \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$, which shows that the SALES algorithm converges to the global minimum.

Now let $c_1 = \underline{L}(2B)^{-1}$, $c_2 = 4L\sqrt{p+1} + 2\bar{L}$, and $\Delta^r = f(\boldsymbol{\beta}^r) - f^*$. By the local error bound (A.6) and the cost-to-go estimate (A.5), we obtain

$$
\begin{aligned}
\Delta^{r+1} &\le c_2\big[\mathrm{d}^2_{\mathscr{X}^*}(\boldsymbol{\beta}^r) + \|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2\big] \\
&\le c_2\kappa^2\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\|^2 + c_2\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 \\
&\le (c_2\kappa^2\sigma^2 + c_2)\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 \\
&\le (c_2\kappa^2\sigma^2 + c_2)c_1^{-1}[f(\boldsymbol{\beta}^r) - f(\boldsymbol{\beta}^{r+1})] \\
&= (c_2\kappa^2\sigma^2 + c_2)c_1^{-1}(\Delta^r - \Delta^{r+1}),
\end{aligned}
$$

which implies that

$$
\Delta^{r+1} \le \frac{c_3}{1 + c_3}\Delta^r, \tag{A.8}
$$

where $c_3 = (c_2\kappa^2\sigma^2 + c_2)c_1^{-1}$. We can see from (A.8) that $f(\boldsymbol{\beta}^r)$ approaches f^* with at least linear rate of convergence. From (A.3) again, this further implies that the sequence $\{\boldsymbol{\beta}^r\}$ converges at least linearly. □

Appendix B

Computational Issues of Penalized Quantile Regression

Proof of Lemma 3.1

Proof B.1 (PROOF OF LEMMA 3.1)
Note that $\rho_\tau(u) = u\{\tau - I(u < 0)\} = (1/2)|u| + (\tau - 1/2)u$. It follows that

$$\begin{aligned}
&\rho_\tau(u) + \frac{\alpha}{2}(u-\xi)^2 = \frac{1}{2}|u| + \Big(\tau - \frac{1}{2}\Big)u + \frac{\alpha}{2}(u-\xi)^2 \\
&= \alpha\Big\{\frac{1}{2}\Big[u - \Big(\xi + \frac{1-2\tau}{2\alpha}\Big)\Big]^2 + \frac{1}{2\alpha}|u|\Big\} + \frac{\alpha}{2}\xi^2 - \frac{(1-2\tau)^2}{8\alpha}.
\end{aligned}$$

Therefore, by definition

$$\begin{aligned}
\mathrm{Prox}_{\rho_\tau}[\xi, \alpha] &= \arg\min_{u\in\mathbb{R}} \rho_\tau(u) + \frac{\alpha}{2}(u-\xi)^2 \\
&= \arg\min_{u\in\mathbb{R}} \frac{1}{2}\Big[u - \Big(\xi + \frac{1-2\tau}{2\alpha}\Big)\Big]^2 + \frac{1}{2\alpha}|u|.
\end{aligned}$$

This implies that the proximal operator of ρ_τ is equivalent to a soft thresholding operator. To be specific, we have

$$\begin{aligned}
&\mathrm{Prox}_{\rho_\tau}[\xi, \alpha] = \mathrm{Shrink}\Big(\xi + \frac{1-2\tau}{2\alpha}, \frac{1}{2\alpha}\Big) \\
&= \mathrm{sgn}\Big(\xi + \frac{1-2\tau}{2\alpha}\Big) \max\Big(\Big|\xi + \frac{1-2\tau}{2\alpha}\Big| - \frac{1}{2\alpha}, 0\Big).
\end{aligned}$$

Consider the three cases where $\xi \in (-\infty, (\tau-1)/\alpha)$, $\xi \in [(\tau-1)/\alpha, \tau/\alpha]$, and $\xi \in (\tau/\alpha, \infty)$ separately, we get the desired proximal operator for ρ_τ as shown in Lemma 1. □

Proof of Theorem 3.1

Proof B.2 (PROOF OF THEOREM 3.1)
For ease of notation, let $f(\boldsymbol{\beta}) = \lambda\|\mathbf{w} \circ \boldsymbol{\beta}\|_1$ and $g(\mathbf{z}) = \mathbb{Q}_\tau(\mathbf{z})$. Also define $\mathbf{r}^k = \mathbf{X}\boldsymbol{\beta}^k + \mathbf{z}^k - \mathbf{y}$ for $k \geq 0$. Note that both f and g are convex. Therefore, by convexity of the constrained form of the weighted L_1-penalized quantile regression (3.4), the KKT condition implies that $(\boldsymbol{\beta}^*, \mathbf{z}^*)$ is an optimal solution if and only if there exists a Lagrangian multiplier $\boldsymbol{\theta}^*$ such that

$$\mathbf{X}^\top\boldsymbol{\theta}^* \in \partial f(\boldsymbol{\beta}^*),\ \boldsymbol{\theta}^* \in \partial g(\mathbf{z}^*),\ \text{and } \mathbf{X}\boldsymbol{\beta}^* + \mathbf{z}^* - \mathbf{y} = \mathbf{0}, \tag{B.1}$$

where ∂f and ∂g are the sub-differentials of f and g, respectively. In the proximal ADMM iterations, the optimality conditions of $\boldsymbol{\beta}^{k+1}$ and $\mathbf{z}^{k+1}$ reveal that

$$\begin{aligned}&\mathbf{0} \in \partial f(\boldsymbol{\beta}^{k+1}) - \mathbf{X}^\top[\boldsymbol{\theta}^k - \sigma\mathbf{r}^{k+1} + \sigma(\mathbf{z}^{k+1} - \mathbf{z}^k)] + \mathbf{S}(\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k),\ \text{and}\\ &\mathbf{0} \in \partial g(\mathbf{z}^{k+1}) - (\boldsymbol{\theta}^k - \sigma\mathbf{r}^{k+1}).\end{aligned}$$

Together with the fact that $\boldsymbol{\theta}^k = \gamma\sigma\mathbf{r}^{k+1} + \boldsymbol{\theta}^{k+1}$, we obtain

$$\begin{aligned}&\mathbf{X}^\top[\boldsymbol{\theta}^{k+1} + \sigma(\gamma-1)\mathbf{r}^{k+1} + \sigma(\mathbf{z}^{k+1} - \mathbf{z}^k)] - \mathbf{S}(\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k) \in \partial f(\boldsymbol{\beta}^{k+1}),\ \text{and}\\ &\boldsymbol{\theta}^{k+1} + \sigma(\gamma-1)\mathbf{r}^{k+1} \in \partial g(\mathbf{z}^{k+1}).\end{aligned} \tag{B.2}$$

Denote $\boldsymbol{d}_{\boldsymbol{\beta}}^k = \boldsymbol{\beta}^k - \boldsymbol{\beta}^*$, $\boldsymbol{d}_{\mathbf{z}}^k = \mathbf{z}^k - \mathbf{z}^*$, and $\boldsymbol{d}_{\boldsymbol{\theta}}^k = \boldsymbol{\theta}^k - \boldsymbol{\theta}^*$, $k \geq 1$ for an optimal solution $(\boldsymbol{\beta}^*, \mathbf{z}^*, \boldsymbol{\theta}^*)$. Now by the optimality conditions (B.1) of $\boldsymbol{\beta}^*$ and the optimality conditions (B.2) of $\boldsymbol{\beta}^{k+1}$, we have

$$\begin{aligned}\langle\mathbf{X}^\top[\boldsymbol{\theta}^{k+1} + \sigma(\gamma-1)\mathbf{r}^{k+1} + \sigma(\mathbf{z}^{k+1} - \mathbf{z}^k)] - \mathbf{S}(\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k) - \mathbf{X}^\top\boldsymbol{\theta}^*,\\ \boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^*\rangle \geq 0,\end{aligned}$$

which simplifies to

$$\langle\boldsymbol{d}_{\boldsymbol{\theta}}^{k+1} + \sigma(\gamma-1)\mathbf{r}^{k+1} + \sigma(\mathbf{z}^{k+1} - \mathbf{z}^k), \mathbf{X}\boldsymbol{d}_{\boldsymbol{\beta}}^{k+1}\rangle - \langle\mathbf{S}(\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k), \boldsymbol{d}_{\boldsymbol{\beta}}^{k+1}\rangle \geq 0. \tag{B.3}$$

Similarly, it can be shown from the optimality conditions (B.1) of $\mathbf{z}^*$ and the optimality conditions (B.2) of $\mathbf{z}^{k+1}$ that

$$\langle \boldsymbol{d}_{\boldsymbol{\theta}}^{k+1} + \sigma(\gamma-1)\mathbf{r}^{k+1}, \boldsymbol{d}_{\mathbf{z}}^{k+1} \rangle \geq 0. \tag{B.4}$$

Adding up (B.3) and (B.4) and applying the identity $\mathbf{X}\boldsymbol{d}_{\boldsymbol{\beta}}^{k+1} + \boldsymbol{d}_{\mathbf{z}}^{k+1} = \mathbf{r}^{k+1} = (\gamma\sigma)^{-1}(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k+1})$, we obtain

$$\begin{aligned} &(\gamma\sigma)^{-1}\langle \boldsymbol{d}_{\boldsymbol{\theta}}^{k+1}, \boldsymbol{d}_{\boldsymbol{\theta}}^{k} - \boldsymbol{d}_{\boldsymbol{\theta}}^{k+1} \rangle + \sigma(\gamma-1)\|\mathbf{r}^{k+1}\|_2^2 - \sigma\langle \boldsymbol{d}_{\mathbf{z}}^{k+1} - \boldsymbol{d}_{\mathbf{z}}^{k}, \boldsymbol{d}_{\mathbf{z}}^{k+1} \rangle \\ &- \langle \mathbf{S}(\boldsymbol{d}_{\boldsymbol{\beta}}^{k+1} - \boldsymbol{d}_{\boldsymbol{\beta}}^{k}), \boldsymbol{d}_{\boldsymbol{\beta}}^{k+1} \rangle + \sigma\langle \mathbf{z}^{k+1} - \mathbf{z}^{k}, \mathbf{r}^{k+1} \rangle \geq 0. \end{aligned} \tag{B.5}$$

Note that $\boldsymbol{\theta}^{k+1} + \sigma(\gamma-1)\mathbf{r}^{k+1} \in \partial g(\mathbf{z}^{k+1})$ and $\boldsymbol{\theta}^{k} + \sigma(\gamma-1)\mathbf{r}^{k} \in \partial g(\mathbf{z}^{k})$. By optimality of $\mathbf{z}^{k+1}$ and $\mathbf{z}^k$ and the identity $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \gamma\sigma\mathbf{r}^{k+1}$ again, we have

$$\begin{aligned} 0 &\leq \langle \boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^{k} + \gamma\sigma\mathbf{r}^{k+1} - \sigma\mathbf{r}^{k+1} - \sigma(\gamma-1)\mathbf{r}^{k}, \mathbf{z}^{k+1} - \mathbf{z}^{k} \rangle \\ &= \sigma(1-\gamma)\langle \mathbf{z}^{k+1} - \mathbf{z}^{k}, \mathbf{r}^{k} \rangle - \sigma\langle \mathbf{z}^{k+1} - \mathbf{z}^{k}, \mathbf{r}^{k+1} \rangle, \end{aligned}$$

which implies that $\sigma\langle \mathbf{z}^{k+1} - \mathbf{z}^{k}, \mathbf{r}^{k+1} \rangle \leq \sigma(1-\gamma)\langle \mathbf{z}^{k+1} - \mathbf{z}^{k}, \mathbf{r}^{k} \rangle$. Note that for both $\mathbf{M} = \mathbf{I}$ and $\mathbf{M} = \mathbf{S}$, the identity $2\langle \mathbf{M}\mathbf{u}, \mathbf{v} \rangle = \|\mathbf{u}+\mathbf{v}\|_{\mathbf{M}}^2 - \|\mathbf{u}\|_{\mathbf{M}}^2 - \|\mathbf{v}\|_{\mathbf{M}}^2$ holds. Now applying this identity, we can infer from (B.5) and the fact $\boldsymbol{d}_{\boldsymbol{\theta}}^{k+1} - \boldsymbol{d}_{\boldsymbol{\theta}}^{k} = \boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^{k} = -\gamma\sigma\mathbf{r}^{k+1}$ that

$$\begin{aligned} 0 \leq\ & (\gamma\sigma)^{-1}\big(\|\boldsymbol{d}_{\boldsymbol{\theta}}^{k}\|_2^2 - \|\boldsymbol{d}_{\boldsymbol{\theta}}^{k+1}\|_2^2\big) + \sigma(\gamma-2)\|\mathbf{r}^{k+1}\|_2^2 \\ &+ \sigma\big(\|\boldsymbol{d}_{\mathbf{z}}^{k}\|_2^2 - \|\boldsymbol{d}_{\mathbf{z}}^{k+1}\|_2^2 - \|\mathbf{z}^{k+1} - \mathbf{z}^{k}\|_2^2\big) \\ &+ \big(\|\boldsymbol{d}_{\boldsymbol{\beta}}^{k}\|_{\mathbf{S}}^2 - \|\boldsymbol{d}_{\boldsymbol{\beta}}^{k+1}\|_{\mathbf{S}}^2 - \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^{k}\|_{\mathbf{S}}^2\big) \\ &+ 2\sigma(1-\gamma)\langle \mathbf{z}^{k+1} - \mathbf{z}^{k}, \mathbf{r}^{k} \rangle. \end{aligned} \tag{B.6}$$

In what follows, we consider two different scenarios: $0 < \gamma \leq 1$ and $1 < \gamma < (\sqrt{5}+1)/2$. We show that the algorithm converges under both scenarios, although the techniques will differ.

(A). $0 < \gamma \leq 1$. Note that $1 - \gamma \geq 0$. By Cauchy–Schwarz inequality we have

$$2\sigma(1-\gamma)\langle \mathbf{z}^{k+1} - \mathbf{z}^{k}, \mathbf{r}^{k} \rangle \leq \sigma(1-\gamma)\big[\|\mathbf{z}^{k+1} - \mathbf{z}^{k}\|_2^2 + \|\mathbf{r}^{k}\|_2^2\big].$$

It follows from (B.6) that

$$\begin{aligned}&\big[(\gamma\sigma)^{-1}\|\boldsymbol{d}_{\boldsymbol{\theta}}^{k+1}\|_2^2+\|\boldsymbol{d}_{\boldsymbol{\beta}}^{k+1}\|_{\mathbf{S}}^2+\sigma\|\boldsymbol{d}_{\mathbf{z}}^{k+1}\|_2^2+\sigma(1-\gamma)\|\mathbf{r}^{k+1}\|_2^2\big]\\&-\big[(\gamma\sigma)^{-1}\|\boldsymbol{d}_{\boldsymbol{\theta}}^{k}\|_2^2+\|\boldsymbol{d}_{\boldsymbol{\beta}}^{k}\|_{\mathbf{S}}^2+\sigma\|\boldsymbol{d}_{\mathbf{z}}^{k}\|_2^2+\sigma(1-\gamma)\|\mathbf{r}^{k}\|_2^2\big]\\&\le-\gamma\sigma\|\mathbf{z}^{k+1}-\mathbf{z}^k\|_2^2-\|\boldsymbol{\beta}^{k+1}-\boldsymbol{\beta}^k\|_{\mathbf{S}}^2-\sigma\|\mathbf{r}^{k+1}\|_2^2.\end{aligned}$$

It is convenient to denote $u_k=(\gamma\sigma)^{-1}\|\boldsymbol{d}_{\boldsymbol{\theta}}^{k}\|_2^2+\|\boldsymbol{d}_{\boldsymbol{\beta}}^{k}\|_{\mathbf{S}}^2+\sigma\|\boldsymbol{d}_{\mathbf{z}}^{k}\|_2^2+\sigma(1-\gamma)\|\mathbf{r}^{k}\|_2^2$, then we can see that

$$u_{k+1}-u_k\le-\gamma\sigma\|\mathbf{z}^{k+1}-\mathbf{z}^k\|_2^2-\|\boldsymbol{\beta}^{k+1}-\boldsymbol{\beta}^k\|_{\mathbf{S}}^2-\sigma\|\mathbf{r}^{k+1}\|_2^2. \tag{B.7}$$

This implies that $\{u_k,\ k\ge1\}$ is a nonincreasing sequence bounded by $u_0<\infty$. It follows immediately that $\{\|\boldsymbol{\theta}^k\|_2,\ k\ge1\}$, $\{\|\mathbf{z}^k\|_2,\ k\ge1\}$, and $\{\|\boldsymbol{\beta}^k\|_{\mathbf{S}}^2,\ k\ge1\}$ are all bounded sequences. Moreover, we have

$$\begin{aligned}0&\le\lim_{t\to\infty}u_t=u_0+\lim_{t\to\infty}\sum_{k=0}^{t-1}(u_{k+1}-u_k)\\&\le u_0+\lim_{t\to\infty}\sum_{k=0}^{t-1}\big[-\gamma\sigma\|\mathbf{z}^{k+1}-\mathbf{z}^k\|_2^2-\|\boldsymbol{\beta}^{k+1}-\boldsymbol{\beta}^k\|_{\mathbf{S}}^2-\sigma\|\mathbf{r}^{k+1}\|_2^2\big],\end{aligned}$$

from which we obtain that

$$\lim_{k\to\infty}\|\mathbf{z}^{k+1}-\mathbf{z}^k\|_2^2=0,\ \lim_{k\to\infty}\|\boldsymbol{\beta}^{k+1}-\boldsymbol{\beta}^k\|_{\mathbf{S}}^2=0,\ \text{and}\ \lim_{k\to\infty}\|\mathbf{r}^k\|_2^2=0. \tag{B.8}$$

Now observe that $\|\mathbf{X}\boldsymbol{\beta}^k\|_2=\|\mathbf{r}^k-\mathbf{z}^k+\mathbf{y}\|_2\le\|\mathbf{r}^k\|_2+\|\mathbf{z}^k\|_2+\|\mathbf{y}\|_2$. One can then see that $\{\|\mathbf{X}\boldsymbol{\beta}^k\|_2,\ k\ge1\}$ is bounded. Consequently, we know that $\{\|\boldsymbol{\beta}^k\|_2,\ k\ge1\}$ is also bounded since $\|\boldsymbol{\beta}^k\|_2^2=(\sigma\eta)^{-1}\big[\sigma\|\mathbf{X}\boldsymbol{\beta}^k\|_2^2+\|\boldsymbol{\beta}^k\|_{\mathbf{S}}^2\big]$. Therefore, the updates $\{(\boldsymbol{\beta}^k,\mathbf{z}^k,\boldsymbol{\theta}^k),\ k\ge1\}$ are bounded and we can hence find a subsequence $\{(\boldsymbol{\beta}^{k_m},\mathbf{z}^{k_m},\boldsymbol{\theta}^{k_m}),\ m\ge1\}$ that converges to a cluster point which we denote by $(\boldsymbol{\beta}^\infty,\mathbf{z}^\infty,\boldsymbol{\theta}^\infty)$. Observe that by (B.8) and the identity $\mathbf{r}^k=\mathbf{X}\boldsymbol{\beta}^k+\mathbf{z}^k-\mathbf{y}$, we have

$$\begin{aligned}\|\mathbf{X}(\boldsymbol{\beta}^{k+1}-\boldsymbol{\beta}^k)\|_2&=\|(\mathbf{r}^{k+1}-\mathbf{r}^k)-(\mathbf{z}^{k+1}-\mathbf{z}^k)\|_2\\&\le\|\mathbf{r}^{k+1}\|_2+\|\mathbf{r}^k\|_2+\|\mathbf{z}^{k+1}-\mathbf{z}^k\|_2\to0\ \text{as}\ n\to\infty.\end{aligned}$$

This together with the fact $\|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_{\mathrm{S}} \to 0$ implies that

$$\begin{aligned}\|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_2^2 &= (\sigma\eta)^{-1}\big[\|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_{\mathrm{S}}^2 + \sigma\|\mathbf{X}(\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k)\|_2^2\big] \\ &\to 0 \text{ as } n \to \infty.\end{aligned}$$

Note that ∂f and ∂g are nonempty and closed by convexity of f and g. Now take the limits on both sides of (B.2) along the sequence $\{k_m,\ m \geq 1\}$ to obtain

$$\mathbf{X}^{\mathrm{T}}\boldsymbol{\theta}^{\infty} \in \partial f(\boldsymbol{\beta}^{\infty}) \quad \text{and} \quad \boldsymbol{\theta}^{\infty} \in \partial g(\mathbf{z}^{\infty}),$$

which by (B.1) essentially means that $(\boldsymbol{\beta}^{\infty}, \mathbf{z}^{\infty})$ is an optimal solution with corresponding Lagrangian multiplier $\boldsymbol{\theta}^{\infty}$ since by the fact that $\mathbf{r}^k \to 0$ we also have $\mathbf{X}\boldsymbol{\beta}^{\infty} + \mathbf{z}^{\infty} = \mathbf{y}$.

Now to show that $(\boldsymbol{\beta}^{\infty}, \mathbf{z}^{\infty}, \boldsymbol{\theta}^{\infty})$ is the unique limit of $\{(\boldsymbol{\beta}^k, \mathbf{z}^k, \boldsymbol{\theta}^k),\ k \geq 1\}$, let us take $(\boldsymbol{\beta}^*, \mathbf{z}^*, \boldsymbol{\theta}^*) = (\boldsymbol{\beta}^{\infty}, \mathbf{z}^{\infty}, \boldsymbol{\theta}^{\infty})$ as in $\boldsymbol{d}_{\boldsymbol{\beta}}^k$, $\boldsymbol{d}_{\mathbf{z}}^k$, and $\boldsymbol{d}_{\boldsymbol{\theta}}^k$. Then it follows from (B.8) that $\lim_{m\to\infty} u_{k_m} = 0$. Since $\{u_k,\ k \geq 1\}$ is a nonincreasing sequence bounded below by zero, every subsequence of $\{u_k,\ k \geq 1\}$ should converge to zero. This implies that $\lim_{k\to\infty} u_k = 0$. Therefore, we have $\lim_{k\to\infty} \boldsymbol{\theta}^k = \boldsymbol{\theta}^{\infty}$, $\lim_{k\to\infty} \mathbf{z}^k = \mathbf{z}^{\infty}$, and $\lim_{k\to\infty} \|\boldsymbol{d}_{\boldsymbol{\beta}}^k\|_{\mathrm{S}}^2 = 0$. Since we also have $\lim_{k\to\infty} \|\mathbf{X}\boldsymbol{d}_{\boldsymbol{\beta}}^k\|_2 \leq \lim_{k\to\infty}(\|\mathbf{r}^k\|_2 + \|\boldsymbol{d}_{\mathbf{z}}^k\|_2) = 0$, it follows immediately that

$$\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^{\infty}\|_2^2 = (\sigma\eta)^{-1}\big[\sigma\|\mathbf{X}\boldsymbol{d}_{\boldsymbol{\beta}}^k\|_2^2 + \|\boldsymbol{d}_{\boldsymbol{\beta}}^k\|_{\mathrm{S}}^2\big] \to 0 \text{ as } n \to \infty.$$

Therefore, we have shown that $\lim_{k\to\infty}(\boldsymbol{\beta}^k, \mathbf{z}^k, \boldsymbol{\theta}^k) = (\boldsymbol{\beta}^{\infty}, \mathbf{z}^{\infty}, \boldsymbol{\theta}^{\infty})$.

(B). $1 < \gamma < (\sqrt{5}+1)/2$. By the extended Cauchy-Schwarz inequality we have

$$2\sigma(1-\gamma)\langle \mathbf{z}^{k+1} - \mathbf{z}^k, \mathbf{r}^k\rangle \leq \sigma(\gamma - 1)\big[\gamma\|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2 + \gamma^{-1}\|\mathbf{r}^k\|_2^2\big].$$

It follows again from (B.6) that

$$\begin{aligned}&\big[(\gamma\sigma)^{-1}\|\boldsymbol{d}_{\boldsymbol{\theta}}^{k+1}\|_2^2 + \|\boldsymbol{d}_{\boldsymbol{\beta}}^{k+1}\|_2^2 + \sigma\|\boldsymbol{d}_{\mathbf{z}}^{k+1}\|_2^2 + \sigma(1-\gamma^{-1})\|\mathbf{r}^{k+1}\|_2^2\big] \\ &- \big[(\gamma\sigma)^{-1}\|\boldsymbol{d}_{\boldsymbol{\theta}}^{k}\|_2^2 + \|\boldsymbol{d}_{\boldsymbol{\beta}}^{k}\|_2^2 + \sigma\|\boldsymbol{d}_{\mathbf{z}}^{k}\|_2^2 + \sigma(1-\gamma^{-1})\|\mathbf{r}^{k}\|_2^2\big] \\ &\leq -\sigma(\gamma + 1 - \gamma^2)\|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2 - \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_{\mathrm{S}}^2 - \sigma\gamma^{-1}(\gamma + 1 - \gamma^2)\|\mathbf{r}^{k+1}\|_2^2.\end{aligned}$$

Denote $v_k = (\gamma\sigma)^{-1}\|\boldsymbol{d}_{\boldsymbol{\theta}}^k\|_2^2 + \|\boldsymbol{d}_{\boldsymbol{\beta}}^k\|_2^2 + \sigma\|\boldsymbol{d}_{\mathbf{z}}^k\|_2^2 + \sigma(1-\gamma^{-1})\|\mathbf{r}^k\|_2^2$. It follows that

$$v_{k+1} - v_k \leq -\sigma(\gamma + 1 - \gamma^2)\|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2 - \|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_{\mathbf{S}}^2 - \sigma\gamma^{-1}(\gamma + 1 - \gamma^2)\|\mathbf{r}^{k+1}\|_2^2.$$

Note that $\gamma + 1 - \gamma^2 > 0$ when $\gamma \in (1, (\sqrt{5}+1)/2)$. Following the same line of arguments as in the $\gamma \in (0, 1]$ case, we can also show that $\lim_{k\to\infty}(\boldsymbol{\beta}^k, \mathbf{z}^k, \boldsymbol{\theta}^k) = (\boldsymbol{\beta}^\infty, \mathbf{z}^\infty, \boldsymbol{\theta}^\infty)$, which is the unique limit. This completes the first part of the proof.

Now let us assume $\gamma = 1$. For ease of notation, define the vector $\mathbf{v} = (\boldsymbol{\beta}^{\mathrm{T}}, \mathbf{z}^{\mathrm{T}}, \boldsymbol{\theta}^{\mathrm{T}})^{\mathrm{T}}$ and the matrix $\mathbf{H} = \mathrm{diag}(\mathbf{S}, \sigma\mathbf{I}_n, \sigma^{-1}\mathbf{I}_n)$. From (B.7), noting that since $\mathbf{r}^{k+1} = \sigma^{-1}(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k+1})$, we have

$$\|\mathbf{v}^{k+1} - \mathbf{v}^*\|_{\mathbf{H}}^2 \leq \|\mathbf{v}^k - \mathbf{v}^*\|_{\mathbf{H}}^2 - \|\mathbf{v}^k - \mathbf{v}^{k+1}\|_{\mathbf{H}}^2. \tag{B.9}$$

By Theorem 5.1 of He and Yuan (2015), it can be shown that $\{\|\mathbf{v}^k - \mathbf{v}^{k+1}\|_{\mathbf{H}}^2,\ k \geq 0\}$ is a non-increasing sequence, that is,

$$\|\mathbf{v}^k - \mathbf{v}^{k+1}\|_{\mathbf{H}}^2 \leq \|\mathbf{v}^{k-1} - \mathbf{v}^k\|_{\mathbf{H}}^2,\ \forall k \geq 1. \tag{B.10}$$

It then follows from (B.9) and (B.10) that

$$(k+1)\|\mathbf{v}^k - \mathbf{v}^{k+1}\|_{\mathbf{H}}^2 \leq \sum_{t=1}^{\infty}\|\mathbf{v}^t - \mathbf{v}^{t+1}\|_{\mathbf{H}}^2 \leq \|\mathbf{v}^0 - \mathbf{v}^*\|_{\mathbf{H}}^2,$$

which implies that $\|\mathbf{v}^k - \mathbf{v}^{k+1}\|_{\mathbf{H}}^2 = \mathcal{O}(1/k)$ as $k \to \infty$. This completes the proof. □

ADMM algorithms for the weighted elastic net penalized quantile regression

Let us consider the following weighted elastic net penalized quantile regression

$$\min_{\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n}\rho_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \lambda_1\sum_{j=1}^{p} w_j|\beta_j| + \frac{\lambda_2}{2}\sum_{j=1}^{n} v_j\beta_j^2,$$

where $\lambda_1,\ \lambda_2 \geq 0$ are the regularization parameters and $w_j,\ v_j \geq 0,\ j = 1, \ldots, p$ are the weights. Again, let $\eta \geq \Lambda_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})$. The proximal ADMM (pADMM) algorithm for solving this problem is shown in Algorithm 9 and the sparse coordinate descent ADMM (scdADMM) algorithm is displayed in Algorithm 10.

Algorithm 9: pADMM – Proximal ADMM algorithm for solving the weighted elastic net penalized quantile regression.

---

1. Initialize the algorithm with $(\boldsymbol{\beta}^0, \mathbf{z}^0, \boldsymbol{\theta}^0)$.
2. For $k = 0, 1, 2, \ldots$, repeat steps (2.1) – (2.3) until the convergence criterion is met.

   (2.1) Update

   $$\boldsymbol{\beta}^{k+1} \leftarrow \left((\sigma\eta + \lambda_2 v_j)^{-1}\,\text{Shrink}\left[(\sigma\eta)\beta_j^k + X_j^{\mathrm{T}}(\boldsymbol{\theta}^k + \sigma\mathbf{y} - \sigma\mathbf{X}\boldsymbol{\beta}^k - \sigma\mathbf{z}^k),\ \lambda_1 w_j\right]\right)_{1\leq j\leq p}.$$

   (2.2) Update $\mathbf{z}^{k+1} \leftarrow \left(\text{Prox}_{\rho_\tau}\left[y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{k+1} + \sigma^{-1}\theta_i^k, n\sigma\right]\right)_{1\leq i\leq n}$.

   (2.3) Update $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k - \gamma\sigma(\mathbf{X}\boldsymbol{\beta}^{k+1} + \mathbf{z}^{k+1} - \mathbf{y})$.

Algorithm 10: scdADMM - Sparse coordinate descent ADMM algorithm for solving the weighted elastic net penalized quantile regression with coordinate descent steps.

---

1. Initialize the algorithm with $(\boldsymbol{\beta}^0, \mathbf{z}^0, \boldsymbol{\theta}^0)$.
2. For $k = 0, 1, 2, \ldots$, repeat steps (2.1) – (2.3) until the convergence criterion is met.

   (2.1) Carry out the coordinate descent steps (2.1.1) – (2.1.3).

   (2.1.1) Initialize $\boldsymbol{\beta}^{k,0} = \boldsymbol{\beta}^k$.

   (2.1.2) For $m = 0, 1, 2, \ldots$, repeat step (2.1.2.1) until convergence.

   (2.1.2.1) For $j = 1, \ldots, p$, update

   $$\beta_j^{k,m+1} \leftarrow \frac{\text{Shrink}\left[\sum_{i=1}^n x_{ij}\left\{\theta_i^k + \sigma\left(y_i - z_i^k - \sum_{t\neq j} x_{it}\beta_t^{k,m+I(t<j)}\right)\right\},\ \lambda_1 w_j\right]}{\sigma\|X_j\|_2^2 + \lambda_2 v_j}.$$

   (2.1.3) Set $\boldsymbol{\beta}^{k+1} \leftarrow \boldsymbol{\beta}^{k,m+1}$.

   (2.2) Update $\mathbf{z}^{k+1} \leftarrow \left(\text{Prox}_{\rho_\tau}\left[y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{k+1} + \sigma^{-1}\theta_i^k, n\sigma\right]\right)_{1\leq i\leq n}$.

   (2.3) Update $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k - \sigma(\mathbf{X}\boldsymbol{\beta}^{k+1} + \mathbf{z}^{k+1} - \mathbf{y})$.

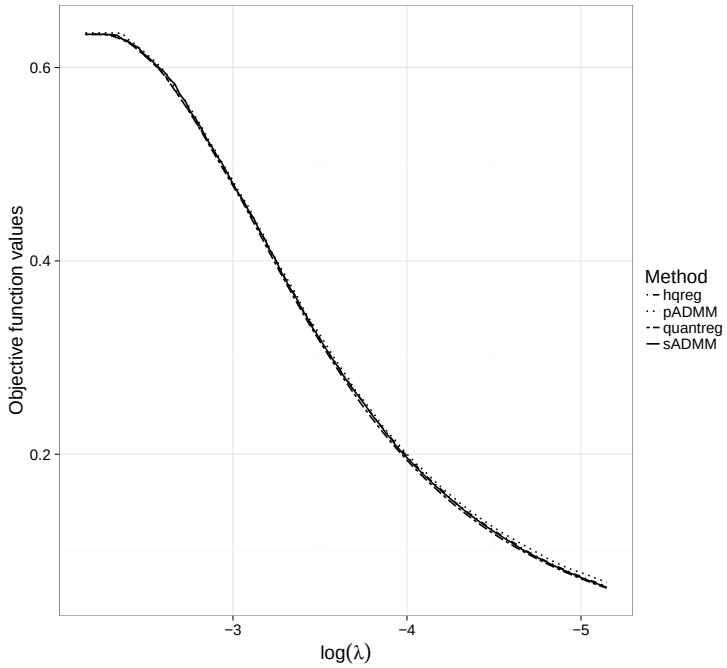

Figure B.1: Objective function values of lasso penalized quantile regression ($\tau = 0.75$) fitted on model (3.10) with $\alpha = 0.5$, $n = 100$, and $p = 5000$ at the optimal solutions computed by `quantreg`, scdADMM, pADMM and `hqreg` along a sequence of one hundred pre-chosen λ values.

Obtained objective function values

In Section 3.4, in order to draw meaningful timing comparison conclusion, we strive to match the objective function values of the lasso penalized quantile regression at the optimal solutions computed by the different algorithms. As a demonstration, shown in Figure B.1 are the objective function values of the lasso penalized quantile regression with $\tau = 0.75$ fitted on model (3.10) having $\alpha = 0.5$ at the optimal solutions computed by respectively `quantreg`, scdADMM, pADMM and `hqreg` for a sequence of one hundred pre-chosen λ values.

Alternative algorithms

The MM principle is widely adopted to solve many statistical problems, including the quantile regression. According to Hunter and Lange (2000), the MM technique is applied to

a perturbed (smoothed) version of the quantile check loss

$$\rho_\tau^\epsilon(r) = \rho_\tau(r) - \frac{\epsilon}{2}\log(\epsilon + |r|) \tag{B.11}$$

for some small ϵ, such as $\epsilon = 10^{-6}$. Therefore, in Hunter and Lange (2000), an approximate solution to the quantile regression can be obtained by

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \rho_\tau^\epsilon(r_i), \text{ where } r_i = y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta},$$

which can be solved iteratively by applying the MM technique:

1. Initialize $\boldsymbol{\beta}$ with β^0 and calculate $r_i^0 = y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^0$, $i = 1, \ldots, n$.
2. For $k = 1, \ldots, M$, minimize the majorization problem

$$\boldsymbol{\beta}^k = \min_{\boldsymbol{\beta}} \sum_{i=1}^n \zeta_\tau^\epsilon(r_i|r_i^{k-1})$$

and update $r_i^k = y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^k$, $i = 1, \ldots, n$, where

$$\zeta_\tau^\epsilon(r|r^k) = \frac{1}{4}\left[\frac{r^2}{\epsilon + |r^k|} + (4\tau - 2)r + c\right]$$

and c is a constant chosen so that $\zeta_\tau^\epsilon(r^k|r^k) = \rho_\tau^\epsilon(r^k)$.

Following Hunter and Lange (2000), for the penalized quantile regression, we approximate its solution by solving

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{n}\rho_\tau^\epsilon(r_i) + \lambda\sum_{j=1}^p w_j|\beta_j|, \text{ where } r_i = y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}, \tag{B.12}$$

The same idea was used in Lv et al. (2016) for kernel quantile regression with smoothness-sparsity constraint. Applying the MM technique, the above problem can be solved iteratively by

1. Initialize $\boldsymbol{\beta}$ with $\boldsymbol{\beta}^0$ and calculate $r_i^0 = y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^0$, $i = 1, \ldots, n$.

2. For $k = 1, \ldots, M$, minimize the majorization problem

$$\boldsymbol{\beta}^k = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{4} \left[\frac{r_i^2}{\epsilon + |r_i^{k-1}|} + (4\tau - 2) r_i + c_i \right] + \lambda \sum_{j=1}^{p} w_j |\beta_j| \quad \text{(B.13)}$$

and update $r_i^k = y_i - \mathbf{x}_i^\top \boldsymbol{\beta}^k$, $i = 1, \ldots, n$.

Note that problem (B.13) can be cast as a penalized (weighted) least squares problem

$$\boldsymbol{\beta}^k = \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \tilde{w}_i \left(\tilde{y}_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j|$$

with observational weights $\tilde{w}_i = 0.25(\epsilon + |r_i^{k-1}|)^{-1}$ and auxiliary responses $\tilde{y}_i = y_i + (2\tau - 1)(\epsilon + |r_i^{k-1}|)$, $i = 1, \ldots, n$. Therefore, one can use the `glmnet` or `gcdnet` package to solve it very efficiently.

To compare with this MM algorithm, we did a numerical study using the simulation model from our first study in Section 3.4. Specifically, we take $n = 100$, $p = 200$ and $\alpha = 0.5$. During our simulation, we found that when $\epsilon = 10^{-5}$, the MM algorithm often takes more than 10000 iterations to converge. Therefore, we used $\epsilon = 10^{-4}$ instead in the MM algorithm for faster computation. This usually takes the MM algorithm around 1000 iterations to converge. Note that the perturbed quantile function (B.11) has worse approximation with larger ϵ. We plot the objective function values evaluated at the optimal solutions returned by the MM procedure in Figure B.2. Also included are those from our ADMM algorithm and `quantreg`. It can be seen that the MM algorithm is quite unstable and does not provide accurate solutions. The timings for running MM, `quantreg` and ADMM are given in Table B.1. We see that MM takes a longer time than `quantreg` and our algorithm. To achieve better accuracy with the MM algorithm (using $\epsilon = 10^{-5}$ for example), one needs to run a even longer time.

Our result is consistent with the findings from Chen (2007). Specifically, it was noted in Chen (2007) that the MM algorithm implemented according to Hunter and Lange (2000) is significantly inferior to the smoothing and interior point algorithms for linear quantile regression.

Both Peng and Wang (2015) and Yi and Huang (2016) applied the idea of coordinate descent for solving quantile lasso. Specifically, in Peng and Wang (2015), each coordinate descent step involves solving a weighted median regression, which requires iterative

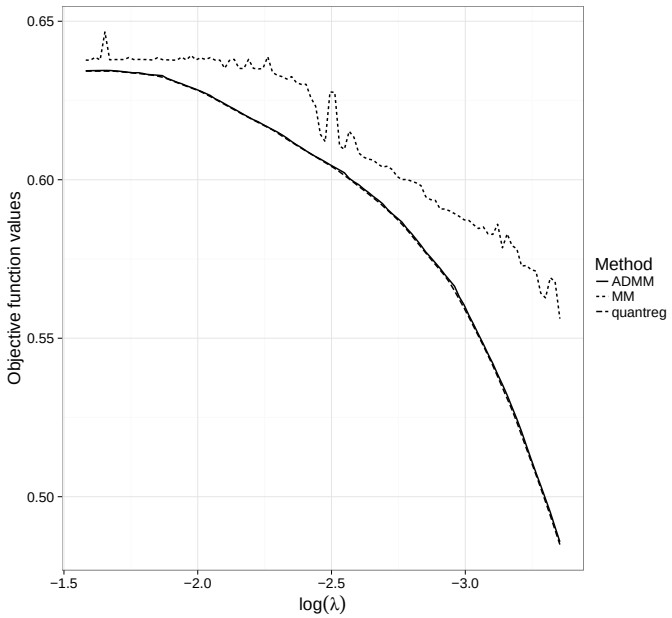

Figure B.2: Objective function values of lasso penalized quantile regression ($\tau = 0.25$) fitted on Friedman's model ($\alpha = 0.5$, $n = 100$ and $p = 200$) with MM, `quantreg` and ADMM along a sequence of pre-chosen λ values.

Table B.1: Timings (in seconds) for running lasso penalized quantile regression ($\tau = 0.25$) on Friedman's model ($\alpha = 0.5$, $n = 100$ and $p = 200$) over a sequence of pre-chosen λ values by `MM`, `quantreg` and `ADMM`

`MM`	`quantreg`	`ADMM`
22.86	3.51	0.20

sorting. While in Yi and Huang (2016), a Huber smoothing step is first applied, followed by coordinate descent for the penalized Huber regression. Therefore, their algorithm only approximately solves the penalized quantile regression.

We compare our algorithm with those by Peng and Wang (2015) and Yi and Huang (2016) in terms of both timing and accuracy. The respective R packages implementing their algorithms are `QICD` (Peng, 2016) and `hqreg` (Yi, 2016). For demonstration purpose, we consider the simulation setup from our first study (3.10) and set $\alpha = 0.5$, $n = 100$ and $p = 400$. Based on our numerical study, we observe that, in terms of computational accuracy measured through objective function values, the `QICD` algorithm presents high instability and does not seem to always converge to the correct solutions. Specifically, the objective function values for the optimal solutions at a sequence of λ values returned by ADMM, `quantreg`, `QICD`, and `hqreg` are presented in Figure B.3. We see that ADMM, `quantreg` and `hqreg` are closely comparable, while `QICD` is far less accurate. The timings by the four algorithms are reported in Table B.2. Due to the convergence issue with `QICD`, we only report timing comparisons between our algorithms and `hqreg` in Chapter 3.4.

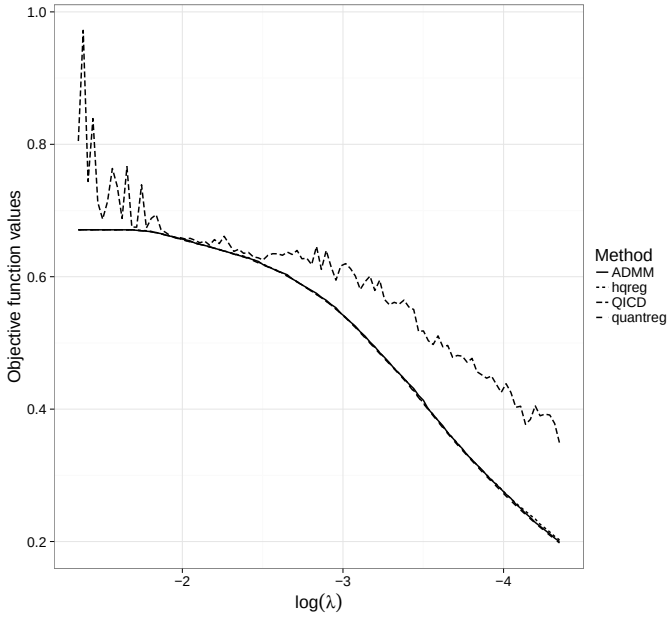

Figure B.3: Objective function values of lasso penalized quantile regression ($\tau = 0.75$) fitted on Friedman's model ($\alpha = 0.5$, $n = 100$ and $p = 400$) with `ADMM`, `quantreg`, `QICD`, and `hqreg` along a sequence of pre-chosen λ values.

Table B.2: Timings (in seconds) for running lasso penalized quantile regression ($\tau = 0.75$) on Friedman's model ($\alpha = 0.5$, $n = 100$ and $p = 400$) over a sequence of pre-chosen λ values by `ADMM`, `hqreg`, `QICD` and `quantreg`. The question mark implies inaccurate result

`ADMM`	`hqreg`	`QICD`	`quantreg`
1.16	7.01	1.91 (?)	27.66

Appendix C

Computational Issues of Penalized Composite Quantile Regression

In this appendix, we provide the sparse coordinate descent ADMM algorithm for solving the weighted elastic net penalized composite quantile regression. We call this algorithm CQR-scdADMM for short. We also investigate the uniqueness and numerical properties of the oracle solution to the composite quantile regression.

ADMM algorithm for the weighted elastic net penalized composite quantile regression

Recall that the problem to solve is

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \frac{1}{nK}\sum_{i=1}^{n}\sum_{k=1}^{K}\rho_{\tau_k}(y_i-\alpha_k-\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})+\lambda_1\sum_{j=1}^{p}d_j|\beta_j|+\frac{\lambda_2}{2}\sum_{j=1}^{p}v_j\beta_j^2,$$

where $\lambda_1, \lambda_2 \geq 0$ and the weights $d_j, v_j \geq 0$ for $j = 1, \ldots, p$. The CQR-scdADMM algorithm is summarized in Algorithm 11.

Algorithm 11: CQR-scdADMM – Sparse coordinate descent ADMM algorithm for solving the weighted elastic net penalized composite quantile regression.

1. Initialize the algorithm with $(\boldsymbol{\alpha}^0, \boldsymbol{\beta}^0, \mathbf{Z}^0, \boldsymbol{\Theta}^0)$.

2. For $r = 0, 1, 2, \ldots$, repeat steps (2.1) – (2.3) until the convergence criterion is met.

   (2.1) Carry out the coordinate descent steps (2.1.1) – (2.1.3).

   (2.1.1) Initialize $\boldsymbol{\alpha}^{r,0} = \boldsymbol{\alpha}^r$ and $\boldsymbol{\beta}^{r,0} = \boldsymbol{\beta}^r$.

   (2.1.2) For $m = 0, 1, 2, \ldots$, repeat steps (2.1.2.1) – (2.1.2.2) until convergence.

   (2.1.2.1) For $k = 1, \ldots, K$, update

$$\alpha_k^{r,m+1} \leftarrow (n\sigma)^{-1}\mathbf{1}^{\mathrm{T}}[\Theta_k^r + \sigma(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{r,m} - Z_k^r)].$$

   (2.1.2.2) For $j = 1, \ldots, p$, update

$$\beta_j^{r,m+1} \leftarrow \frac{\text{Shrink}\left[X_j^{\mathrm{T}} \sum_{k=1}^K \left\{\sigma\left(\mathbf{y} - \alpha_k^{r,m+1}\mathbf{1} - \sum_{t\neq j} X_t \beta_t^{r,m+I(t<j)} - Z_k^r\right) + \Theta_k^r\right\}, \lambda_1 w_j\right]}{K\sigma\|X_j\|_2^2 + \lambda_2 v_j}.$$

   (2.1.3) Set $\boldsymbol{\alpha}^{r+1} \leftarrow \boldsymbol{\alpha}^{r,m+1}$ and $\boldsymbol{\beta}^{r+1} \leftarrow \boldsymbol{\beta}^{r,m+1}$.

   (2.2) Update $\mathbf{Z}^{r+1} \leftarrow \left(\text{Prox}_{\rho_{\tau_k}}\left(y_i - \alpha_k^{r+1} - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{r+1} + \frac{\theta_{ik}^r}{\sigma}, nK\sigma\right)\right)_{1\le i\le n,\, 1\le k\le K}$.

   (2.3) Update $\boldsymbol{\Theta}^{r+1} \leftarrow \left(\theta_{ik}^r - \sigma[\alpha_k^{r+1} + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{r+1} - y_i + z_{ik}^{r+1}]\right)_{1\le i\le n,\, 1\le k\le K}$.

Numerical properties of the CQR oracle solution

The oracle knows the set of true variables, so the oracle estimator for the composite quantile regression is obtained through regression on the true set of variables

$$(\widehat{\boldsymbol{\alpha}}^{\mathrm{o}}, \widehat{\boldsymbol{\beta}}^{\mathrm{o}}) := \underset{(\boldsymbol{\alpha}, \boldsymbol{\beta}):\boldsymbol{\beta}_{\mathscr{A}^{\mathrm{c}}}=\mathbf{0}}{\arg\min} \sum_{k=1}^{K} w_k \sum_{i=1}^{n} \rho_{\tau_k}(y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}).$$

For ease of exposition, we will restrict the scope of variables under consideration to those in $\mathscr{A}$. Specifically, let $\mathbf{a} = \boldsymbol{\alpha}$, $\mathbf{b} = \boldsymbol{\beta}_{\mathscr{A}} \in \mathbb{R}^s$ and $\mathbf{z}_i = \mathbf{x}_{i\mathscr{A}}$, $i = 1, \ldots, n$. The oracle solution can be equivalently obtained through the following minimization problem

$$(\widehat{\mathbf{a}}, \widehat{\mathbf{b}}) := \underset{\mathbf{a},\mathbf{b}}{\arg\min} \sum_{k=1}^{K} w_k \sum_{i=1}^{n} \rho_{\tau_k}(y_i - a_k - \mathbf{z}_i^{\mathrm{T}}\mathbf{b}).$$

Now let $\mathbf{u}_k = \big(\mathbf{y} - a_k\mathbf{1}_n - \mathbf{Z}\mathbf{b}\big)_+$ and $\mathbf{v}_k = \big(\mathbf{y} - a_k\mathbf{1}_n - \mathbf{Z}\mathbf{b}\big)_-$, $k = 1, \ldots, K$, where the positive and negative parts are taken componentwisely. Also, let $\mathbf{u} = (\mathbf{u}_1^{\mathrm{T}}, \ldots, \mathbf{u}_K^{\mathrm{T}})^{\mathrm{T}}$ and $\mathbf{v} = (\mathbf{v}_1^{\mathrm{T}}, \ldots, \mathbf{v}_K^{\mathrm{T}})^{\mathrm{T}}$. Then the above regression problem can be cast into the following linear program of standard form

$$\begin{array}{ll} \text{minimize} & c^{\mathrm{T}}x \\ \text{subject to} & Ax = b \\ & x \succeq 0, \end{array}$$

where $b = \mathbf{1}_K \otimes \mathbf{y}$, and

$$x = \big(\mathbf{a}_+^{\mathrm{T}}, \mathbf{a}_-^{\mathrm{T}}, \mathbf{b}_+^{\mathrm{T}}, \mathbf{b}_-^{\mathrm{T}}, \mathbf{u}^{\mathrm{T}}, \mathbf{v}^{\mathrm{T}}\big),$$
$$c = \big(\mathbf{0}_K^{\mathrm{T}}, \mathbf{0}_K^{\mathrm{T}}, \mathbf{0}_p^{\mathrm{T}}, \mathbf{0}_p^{\mathrm{T}}, w_1\tau_1\mathbf{1}_n^{\mathrm{T}}, \ldots, w_K\tau_K\mathbf{1}_n^{\mathrm{T}}, w_1(1-\tau_1)\mathbf{1}_n^{\mathrm{T}}, \ldots, w_K(1-\tau_K)\mathbf{1}^{\mathrm{T}}\big)^{\mathrm{T}},$$
$$A = \begin{pmatrix} \mathbf{1}_n & \cdots & \mathbf{0} & -\mathbf{1}_n & \cdots & \mathbf{0} & \mathbf{Z} & -\mathbf{Z} & \mathbf{I}_n & -\mathbf{I}_n \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{1}_n & \mathbf{0} & \cdots & -\mathbf{1}_n & \mathbf{Z} & -\mathbf{Z} & \mathbf{I}_n & -\mathbf{I}_n \end{pmatrix}_{(nK)\times(2K+2s+2n)}.$$

Without loss of generality, assume that $\mathbf{1}_n \notin \text{Span}(\mathbf{Z})$, where $\text{Span}(\mathbf{Z})$ denotes the column span of $\mathbf{Z}$. Write

$$D = \begin{pmatrix} \mathbf{1}_n & \cdots & \mathbf{0} & \mathbf{Z} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{1}_n & \mathbf{Z} \end{pmatrix}.$$

The rows of D will be denoted by d_i^{T}, $i = 1, \ldots, n$. Let $\mathcal{H}$ be the collection of $(K+s)$-element subsets of $\{1, \ldots, n\}$. For $h \in \mathcal{H}$, let $D(h)$ denote the submatrix of D with rows $\{d_i^{\mathrm{T}}, i \in h\}$ and $b(h)$ be the $(K+s)$-vector with coordinates $\{b_i, i \in h\}$. We also let $\bar{h} = \{1, \ldots, n\} \backslash h$ for $h \in \mathcal{H}$. Let $H = \{h \in \mathcal{H} : |D(h)| \neq 0\}$. By similar arguments as in Section 6.2 of Koenker (2005), one can verify that the vertices of the polyhedron $\{x : Ax = b,\ x \succeq 0\}$ are given by

$$\begin{aligned} &(\mathbf{a}(h)^{\mathrm{T}}, \mathbf{b}(h)^{\mathrm{T}})^{\mathrm{T}} = \big[D(h)\big]^{-1} b(h) \\ &\mathbf{u}(h) = \mathbf{v}(h) = \mathbf{0} \\ &\mathbf{u}(\bar{h}) = \left[b(\bar{h}) - D(\bar{h}) \begin{pmatrix} \mathbf{a}(h) \\ \mathbf{b}(h) \end{pmatrix} \right]_+ \\ &\mathbf{v}(\bar{h}) = \left[b(\bar{h}) - D(\bar{h}) \begin{pmatrix} \mathbf{a}(h) \\ \mathbf{b}(h) \end{pmatrix} \right]_- \end{aligned}$$

for all $h \in H$. According to the simplex algorithm (see, for example, Bertsimas and Tsitsiklis, 1997, Chapter 3), the optimal solution to this linear program is among the above set of vertices. Assume that y has a density with respect to the Lebesgue measure. Then it can be shown that with probability one, there are at most $K(K+s)$ zero residuals for which $y_i - \hat{a}_k - \mathbf{z}_i^{\mathrm{T}} \widehat{\mathbf{b}} = 0$, $1 \leq i \leq n$, $1 \leq k \leq K$, given each optimal solution $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}})$. Otherwise, suppose that there exist $h \in H$ and $i \in \bar{h}$ such that $\mathbf{u}(i) = \mathbf{v}(i) = 0$. Then since $D(h)$ is non-singular, it follows that

$$b_i = d_i^{\mathrm{T}} \begin{pmatrix} \mathbf{a}(h) \\ \mathbf{b}(h) \end{pmatrix} = d_i^{\mathrm{T}} \big[D(h)\big]^{-1} b(h),$$

which implies that b_i is a linear combination of $b(h)$. By the assumption that y has a density and the structure of b, this occurs with probability zero unless $b_i = b_j$ for some $j \in h$.

However, there are at most $(K-1)$ such i's for each $j \in h$. This means with probability one, at each vertex, there are at most $K(K+s)$ indices i for which $\mathbf{u}(i) = \mathbf{v}(i) = 0$.

Timing of the ADMM algorithm for penalized CQR

The following table (Table C.1) lists the timings for running lasso penalized CQR over a sequence of one hundred λ values on simulated data from model (4.10) using the ADMM algorithm .

Table C.1: Timings (in seconds) for running lasso penalized CQR over a sequence of one hundred λ values on simulated data from model (4.10) using the ADMM algorithm

$\mathbf{\Sigma}$	Method	$N(0,3)$	MN	MDG	t_3	Cauchy
		$n = 100, p = 600$				
$(0.5^{\|i-j\|})$	LS-lasso	0.016	0.016	0.019	0.018	0.017
	CQR-lasso	8.316	8.837	8.873	7.778	8.865
$(0.8^{\|i-j\|})$	LS-lasso	0.016	0.016	0.016	0.019	0.031
	CQR-lasso	10.969	11.938	11.336	10.291	11.145
		$n = 200, p = 1200$				
$(0.5^{\|i-j\|})$	LS-lasso	0.044	0.045	0.047	0.043	0.081
	CQR-lasso	33.629	32.836	34.477	32.461	15.377
$(0.8^{\|i-j\|})$	LS-lasso	0.044	0.060	0.042	0.063	0.108
	CQR-lasso	43.928	42.434	45.550	43.918	24.193