ADMM for High-Dimensional Sparse Penalized Quantile Regression

YUWEN GU[*], JUN FAN[†], LINGCHEN KONG[†], SHIQIAN MA[‡] AND HUI ZOU[§]

Abstract

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 non-smoothness of the quantile regression loss function. We introduce fast alternating direction method of multipliers (ADMM) algorithms for computing the 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. Supplementary materials for this article are available online.

Keywords: Alternating direction method of multipliers; Lasso; Quantile regression; Nonconvex penalty; Variable selection.

[*]Department of Statistics, University of Connecticut, Storrs, CT, USA. Yuwen Gu and Jun Fan are joint first authors.
[†]Department of Applied Mathematics, Beijing Jiaotong University, Beijing, China.
[‡]Department of Mathematics, University of California, Davis, CA, USA.
[§]School of Statistics, University of Minnesota, Minneapolis, MN, USA. Hui Zou is the corresponding author. Email: zouxx019@umn.edu.

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 sophisticated 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 regression 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 article, 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 article is organized as follows. In Section 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. The numerical and statistical efficiency of the proposed algorithms is demonstrated by simulation studies and real data analysis in Section 4. Technical proofs can be found in the online supplementary file.

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 handle 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 $Q_Y(\tau|\mathbf{x}) = \inf\{y\colon F_Y(y|\mathbf{x}) \geq \tau\}$ be the τth conditional quantile for $\tau \in (0,1)$. The linear quantile regression model assumes $Q_Y(\tau|\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\beta(\tau)$ for some unknown coefficient vector $\beta(\tau)$. Given observations $(\mathbf{x}_i, y_i)_{i=1}^n$, the quantile regression estimator of $\beta(\tau)$ is obtained through minimization of the empirical loss function $\sum_{i=1}^n \rho_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\beta)$ over $\beta \in \mathbb{R}^p$, 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

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

over $\beta \in \mathbb{R}^p$. Here, $p_\lambda(\cdot)$, $\lambda > 0$ 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 $\beta(\tau)$, the lasso penalized regression estimator is shown to be consistent at near-oracle rate $\mathscr{O}(\sqrt{s\log p/n})$ by Belloni and Chernozhukov (2011), where $s = \|\beta(\tau)\|_0 = \sum_{j=1}^{p} I(\beta_j(\tau) \neq 0)$. 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 non-smoothness of the penalty functions, let alone the issues arising from nonconvex optimization when folded concave penalties are used. In this article, 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

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

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 $\|\mathbf{w}\circ\beta\|_1 = \sum_{j=1}^{p}|w_j\beta_j| = \sum_{j=1}^{p} w_j|\beta_j|$ with $\circ$ denoting the Hadamard product. We note that in formulation (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 (1) is general, note that for the lasso penalized quantile regression, one can choose $\mathbf{w} = \mathbf{1}_p$, a p-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 $\widehat{\beta}^{\text{lasso}} = (\hat{\beta}_j^{\text{lasso}}, j = 1,\ldots,p)^{\mathrm{T}}$ denotes the quantile lasso estimator.

Once the problem in (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 (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,

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

via the following iterations:

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

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

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

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

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 $\widehat{\beta}^0 = \widehat{\beta}^{\text{lasso}}$ and run the LLA algorithm for two iterations to obtain the quantile SCAD estimator.

3 Alternating Direction Algorithm

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, 2015) 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 (1) is equivalent to the linear program

$$\begin{aligned} \text{minimize} \quad & \tau \mathbf{1}_n^{\mathrm{T}}\mathbf{u} + (1-\tau)\mathbf{1}_n^{\mathrm{T}}\mathbf{v} + (n\lambda)\mathbf{w}^{\mathrm{T}}\beta^{+} + (n\lambda)\mathbf{w}^{\mathrm{T}}\beta^{-} \\ \text{subject to} \quad & \mathbf{u} - \mathbf{v} + \mathbf{X}\beta^{+} - \mathbf{X}\beta^{-} = \mathbf{y} \\ & \mathbf{u}, \mathbf{v} \in \mathbb{R}_{+}^{n}, \ \beta^{+}, \beta^{-} \in \mathbb{R}_{+}^{p}, \end{aligned} \tag{2}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^{\mathrm{T}}$. Problem (2) is often solved with the interior point method (Koenker and Ng, 2005) in its dual domain

$$\begin{aligned} \text{minimize} \quad & (-\mathbf{y}^{\mathrm{T}}, \ \mathbf{0}_p^{\mathrm{T}})\mathbf{d} \\ \text{subject to} \quad & [\mathbf{X}^{\mathrm{T}} \ \ (2n\lambda)\mathrm{diag}(\mathbf{w})]\mathbf{d} = (1-\tau)\mathbf{X}^{\mathrm{T}}\mathbf{1}_n + (n\lambda)\mathbf{w} \\ & 0 \le d_k \le 1, \ k = 1, \ldots, n+p, \end{aligned} \tag{3}$$

where $\mathrm{diag}(\mathbf{w})$ denotes the diagonal matrix with the components of $\mathbf{w}$ on its diagonals. It can be seen that the dual problem (3) involves p equality constraints. We note that the interior point algorithm is the state-of-the-art method for fitting penalized quantile regression in low to moderate dimensions, but it fails to scale well with high dimensions. For numerical evidence, see Sections 4 and 5. This observation motivates us to consider an efficient alternative for fitting the high-dimensional quantile regression.

During the revision, one reviewer called our attention to 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 `hqreg` offers an approximate solution.

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 non-smoothness of the check loss, we introduce new variables $\mathbf{z} = \mathbf{y} - \mathbf{X}\beta$. By convexity, problem (1) is equivalent to

$$\begin{aligned} \min_{\beta,\mathbf{z}} \quad & \mathbb{Q}_\tau(\mathbf{z}) + \lambda \|\mathbf{w} \circ \beta\|_1 \\ \text{subject to} \quad & \mathbf{X}\beta + \mathbf{z} = \mathbf{y}. \end{aligned} \tag{4}$$

Fix $\sigma > 0$ and the augmented Lagrangian function of (4) is

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

where $\theta \in \mathbb{R}^n$ 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

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

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

$$
\begin{aligned}
\beta \text{ step}: \quad & \beta^{k+1} := \arg\min_{\beta} \lambda \|\mathbf{w} \circ \beta\|_1 - \langle \theta^k, \mathbf{X}\beta \rangle + \frac{\sigma}{2} \|\mathbf{X}\beta + \mathbf{z}^k - \mathbf{y}\|_2^2 \\
\mathbf{z} \text{ step}: \quad & \mathbf{z}^{k+1} := \arg\min_{\mathbf{z}} \mathbb{Q}_{\tau}(\mathbf{z}) - \langle \theta^k, \mathbf{z} \rangle + \frac{\sigma}{2} \|\mathbf{z} + \mathbf{X}\beta^{k+1} - \mathbf{y}\|_2^2 \\
\theta \text{ step}: \quad & \theta^{k+1} := \theta^k - \sigma(\mathbf{X}\beta^{k+1} + \mathbf{z}^{k+1} - \mathbf{y}).
\end{aligned} \tag{5}
$$

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 non-smoothness 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

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

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

$$
\mathrm{Prox}_{\rho_{\tau}}[\xi, \alpha] := \arg\min_{u \in \mathbb{R}} \rho_{\tau}(u) + \frac{\alpha}{2} (u - \xi)^2, \tag{6}
$$

whose solution is given in the following lemma. The proof of the lemma can be found in the online supplementary materials.

Lemma 1. *Given $\tau \in (0,1)$ and $\alpha > 0$, the proximal mapping $Prox_{\rho_\tau}[\xi,\alpha]$ in (6) has explicit expression: $Prox_{\rho_\tau}[\xi,\alpha] = \xi - \max\left((\tau-1)/\alpha, \min(\xi,\tau/\alpha)\right)$, or equivalently,*

$$
Prox_{\rho_\tau}[\xi,\alpha] = \begin{cases} \xi - \frac{\tau}{\alpha}, & \textit{if } \xi > \frac{\tau}{\alpha} \\ 0, & \textit{if } \frac{\tau-1}{\alpha} \le \xi \le \frac{\tau}{\alpha} \\ \xi - \frac{\tau-1}{\alpha}, & \textit{if } \xi < \frac{\tau-1}{\alpha}. \end{cases}
$$

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

$$
z_i^{k+1} = \mathrm{Prox}_{\rho_\tau}\left[y_i - \mathbf{x}_i^{\mathrm{T}}\beta + \frac{1}{\sigma}\theta_i^k, n\sigma\right], i = 1,\ldots,n. \tag{7}
$$

Unlike the $\mathbf{z}$ step, the β 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 β 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 β step and replace the β step in the standard ADMM (5) with the following augmented β step:

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

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 $\eta \ge \Lambda_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})$, 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 β step, the update of β can be also carried out component-wisely,

$$
\begin{aligned}
\beta^{k+1} &= \arg\min_{\beta} \lambda\|\mathbf{w}\circ\beta\|_1 + \frac{\sigma\eta}{2}\left\|\beta - \frac{\sigma\eta\beta^k + \mathbf{X}^{\mathrm{T}}(\theta^k + \sigma\mathbf{y} - \sigma\mathbf{X}\beta^k - \sigma\mathbf{z}^k)}{\sigma\eta}\right\|_2^2 \\
&= \left(\mathrm{Shrink}\left[\beta_j^k + \frac{1}{\sigma\eta}X_j^{\mathrm{T}}(\theta^k + \sigma\mathbf{y} - \sigma\mathbf{X}\beta^k - \sigma\mathbf{z}^k), \frac{\lambda w_j}{\sigma\eta}\right]\right)_{1\le j\le p},
\end{aligned} \tag{8}
$$

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

Based on (7) and (8), we present the proximal ADMM (pADMM) algorithm for solving the penalized quantile regression

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

where γ is a constant controlling the step length for the θ step. We summarize the proximal ADMM algorithm in Algorithm 1.

Note that the β 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 β 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 β step in (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 β 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 2.

Algorithm 1: pADMM – Proximal ADMM algorithm for solving the weighted L_1-penalized quantile regression.

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

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

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

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

Algorithm 2: scdADMM – Sparse coordinate descent ADMM algorithm for solving the weighted L_1-penalized quantile regression with coordinate descent steps.

1. Initialize the algorithm with $(\beta^0, \mathbf{z}^0, \theta^0)$.
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 $\beta^{k,0} = \beta^k$.

   (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 $\beta^{k+1} \leftarrow \beta^{k,m+1}$.

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

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

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 2) can be directly obtained from Boyd et al. (2011). Therefore, we only establish the convergence result for the pADMM algorithm (Algorithm 1). We show that with proper choice of the step length γ, the pADMM algorithm yields a sequence $\{(\beta^k, \mathbf{z}^k), k = 1, 2, \ldots\}$ that converges to a global minimizer of problem (4).

Theorem 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 $\{(\beta^k, \mathbf{z}^k, \theta^k)\}$ be generated by the pADMM algorithm as described in Algorithm 1. Then, the sequence $\{(\beta^k, \mathbf{z}^k), k = 0, 1, 2, \ldots\}$ converges to an optimal solution $(\beta^*, \mathbf{z}^*)$ to* (4) *and $\{\theta^k, k = 0, 1, 2, \ldots\}$ converges to an optimal solution θ^* to the dual problem of* (4)*. Equivalently, $\{\beta^k, k = 0, 1, 2, \ldots\}$ converges to a global minimizer of problem* (1)*. Moreover, when $\gamma = 1$, the sequence of norms $\{\|\beta^k - \beta^*\|_{\mathbf{S}}^2 + \sigma\|\mathbf{z}^k - \mathbf{z}^*\|_2^2 + \sigma^{-1}\|\theta^k - \theta^*\|_2^2, k \geq 0\}$ is non-increasing and satisfies $\|\beta^k - \beta^*\|_{\mathbf{S}}^2 + \sigma\|\mathbf{z}^k - \mathbf{z}^*\|_2^2 + \sigma^{-1}\|\theta^k - \theta^*\|_2^2 = \mathscr{O}(1/k)$ as $k \to \infty$.*

The proof of the theorem can be found in the online supplementary materials. 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 1 can be naturally applied to scdADMM.

3.4 Implementation Details

We implement Algorithms 1–2 in an R package called `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, $\widehat{\beta}$ is an optimal solution to problem (1) if

$$0 \in -\frac{1}{n}\sum_{i=1}^{n} \partial\rho_\tau(y_i - \mathbf{x}_i^{\mathrm{T}}\widehat{\beta})x_{ij} + \lambda w_j \partial|\hat{\beta}_j| \tag{9}$$

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 (9) that the choice

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

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

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), Section 3.3.1. Specifically, the algorithm is terminated either when the sequence $\{(\beta^k, \mathbf{z}^k, \theta^k)\}$ meets the following criterion

$$\|\mathbf{X}\beta^k + \mathbf{z}^k - \mathbf{y}\|_2 \le \sqrt{n}\varepsilon_1 + \varepsilon_2 \max\{\|\mathbf{X}\beta^k\|_2, \|\mathbf{z}^k\|_2, \|\mathbf{y}\|_2\},$$
$$\sigma\|\mathbf{X}^{\mathrm{T}}(\mathbf{z}^k - \mathbf{z}^{k-1})\|_2 \le \sqrt{p}\varepsilon_1 + \varepsilon_2\|\mathbf{X}^{\mathrm{T}}\theta^k\|_2,$$

where typical choices are $\varepsilon_1 = 10^{-3}$ and $\varepsilon_2 = 10^{-3}$, or when the number of ADMM iterations exceeds a certain number, say 10^5, at each λ value along the sequence.

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.

4.1 Timing Comparisons

We have conducted extensive timing comparisons under various scenarios of high-dimensional models. Through these timing comparisons, we demonstrate that `FHDQR` compares favorably with `quantreg` and `hqreg`. For the timing comparison, we only consider the lasso penalty for demonstration purposes. 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

$$Y = \sum_{j=1}^{p} X_j\beta_j + k\cdot\varepsilon, \tag{10}$$

where $(X_1, \ldots, X_p)^{\mathrm{T}} \sim N(\mathbf{0}, \Sigma)$ with $\Sigma = (\alpha + (1-\alpha)I(i=j))_{p\times p}$, $\beta_j = (-1)^j \exp(-(2j-1)/20)$, $\varepsilon \sim N(0,1)$, 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 demonstration purposes, three different τ values, 0.25, 0.50 and 0.75, are considered.

The average timings over three runs are reported in Tables 1–2. 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 the objective function values of problem (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 the online supplementary materials 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

$$y_i = \mathbf{x}_i^{\mathrm{T}}\beta^* + \varepsilon_i, \quad \mathbf{x}_i \sim N(\mathbf{0}, \Sigma_{\mathbf{x}}), \quad i = 1, \ldots, n, \tag{11}$$

where the true coefficient vector is fixed at

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

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 $\Sigma_{\mathbf{x}}$, several scenarios are also considered, from the independence structure $\Sigma_{\mathbf{x}} = \mathbf{I}_p$, to the autoregressive structures $\Sigma_{\mathbf{x}} = (0.5^{|i-j|})$ and $(0.8^{|i-j|})$, denoted by $\mathrm{AR}_{0.5}$ and $\mathrm{AR}_{0.8}$ respectively, to the compound symmetric structures $\Sigma_{\mathbf{x}} = (\alpha + (1-\alpha)I(i=j))$ with $\alpha = 0.5$ and 0.8, denoted by $\mathrm{CS}_{0.5}$ and $\mathrm{CS}_{0.8}$ 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. We report results at levels $\tau = 0.50$ and $\tau = 0.75$ for demonstration purposes. Several observations can be readily made from this timing comparison. First of all, it is clear from Table 3 that the ADMM algorithms are very fast. Secondly, pADMM works fairly well for covariance structures $\mathbf{I}$, $\mathrm{AR}_{0.5}$ and $\mathrm{AR}_{0.8}$ and becomes slower for $\mathrm{CS}_{0.5}$ and $\mathrm{CS}_{0.8}$. 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 1: Timings (in seconds) for running lasso penalized quantile regression ($\tau = 0.25$, 0.5 and 0.75) on model (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

In Section 4.2, we will compare the finite-sample statistical performance of the penalized quantile regression to that of the penalized least squares and show that the penalized quantile

Table 2: Timings (in seconds) for running lasso penalized quantile regression ($\tau = 0.25$, 0.5 and 0.75) on model (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

regression has better performance under some error distributions. We note that the penalized least squares is faster in timing due to the smooth quadratic error loss it uses. Specifically, we run lasso penalized least squares for model (11) along a sequence of pre-chosen λ sequence using the R package `glmnet` (Friedman et al., 2010). For each setting of model (11), this takes `glmnet` 0.04–0.12 seconds. Compared to Table 3, we can clearly get the message that the penalized least squares is faster, but our algorithm makes the penalized quantile regression a worthy competitor in terms of timing.

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.

Table 3: Timings (in seconds) for running lasso penalized quantile regression ($\tau = 0.5$ and 0.75) on model (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

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 4–8. It is clear that the penalized quantile regression performs better than the penalized least squares under heavy-tailed error distributions, such as Cauchy.

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 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 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

Table 4: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (11) with independent covariates $\Sigma = \mathbf{I}$. 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.

		$\Sigma = \mathbf{I}$		
		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 5: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (11) with covariance matrix $\Sigma = (0.5^{|i-j|})$. 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.

		$\Sigma = (0.5^{|i-j|})$		
		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 6: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (11) with covariance matrix $\Sigma = (0.8^{|i-j|})$. 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.

		$\Sigma = (0.8^{\lvert i-j \rvert})$		
		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 7: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (11) with covariance matrix $\Sigma = (0.5 + 0.5I(i = j))$. 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.

		$\Sigma = (0.5 + 0.5I(i = j))$		
		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 8: Estimation and selection performance of the penalized least squares and penalized quantile regression (with $\tau = 0.5$ and 0.75) for model (11) with covariance matrix $\Sigma = (0.8 + 0.2I(i = j))$. 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.

		$\Sigma = (0.8 + 0.2I(i = j))$		
		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)

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 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 $(1/40)\sum_{i\in\text{validation}} \rho_\tau(y_i - \hat{\beta}_0 - \mathbf{x}_i^{\mathrm{T}}\widehat{\beta})$. The average number of selected genes and prediction errors over the 50 partitions are reported in the third and fourth columns of Table 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 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 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)

5 Discussion

In this article, 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 the online supplementary materials.

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.

Supplementary Materials

Proofs, algorithms, and illustration: This supplementary file provides the technical proofs of Lemma 1 and Theorem 1, the pADMM and scdADMM algorithms for solving the elastic net penalized quantile regression, and an illustrative comparison of the obtained optimal objective function values by respectively `quantreg`, scdADMM, pADMM and `hqreg`. (PDF file)

Acknowledgment

The authors thank the editor, an associate editor and two anonymous reviewers for their helpful comments that led to a much better presentation of this article. This work is supported in part by NSF grant DMS-1505111, the 111 Project of China (B16002) and National Science Foundation of China grant 11431002.

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**, 211–239.

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

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

BIEN, J., TAYLOR, J. and TIBSHIRANI, R. (2013). A lasso for hierarchical interactions. *Annals of statistics* **41**, 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**, 1–122.

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**, 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**, 6287–6292.

ECKSTEIN, J. (1994). Some saddle-function splitting methods for convex programming. *Optimization Methods and Software* **4**, 75–83.

EFRON, B., HASTIE, T., JOHNSTONE, I., TIBSHIRANI, R. et al. (2004). Least angle regression. *The Annals of Statistics* **32**, 407–499.

FAN, J., FAN, Y. and BARUT, E. (2014a). Adaptive robust variable selection. *The Annals of Statistics* **42**, 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**, 1348–1360.

FAN, J. and LV, J. (2011). Nonconcave penalized likelihood with np-dimensionality. *IEEE Transactions on Information Theory* **57**, 5467–5484.

FAN, J., XUE, L. and ZOU, H. (2014b). Strong oracle optimality of folded concave penalized estimation. *The Annals of Statistics* **42**, 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**, 946–977.

FRANK, L. E. and FRIEDMAN, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics* **35**, 109–135.

FRIEDMAN, J., HASTIE, T., HÖFLING, H. and TIBSHIRANI, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics* **1**, 302–332.

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1–22.

GOLDSTEIN, T. and OSHER, S. (2009). The split bregman method for l1-regularized problems. *SIAM journal on imaging sciences* **2**, 323–343.

HE, B., LIAO, L.-Z., HAN, D. and YANG, H. (2002). A new inexact alternating directions method for monotone variational inequalities. *Mathematical Programming* **92**, 103–118.

HE, X. and SHAO, Q.-M. (2000). On parameters of increasing dimensions. *Journal of Multivariate Analysis* **73**, 120–135.

HUANG, J., MA, S. and ZHANG, C.-H. (2008). Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica* , 1603–1618.

HUNTER, D. R. and LANGE, K. (2000). Quantile regression via an mm algorithm. *Journal of Computational and Graphical Statistics* **9**, 60–77.

KOENKER, R. (2005). *Quantile Regression*. Cambridge, United Kingdom: Cambridge University Press.

KOENKER, R. (2015). *quantreg: Quantile Regression*. R package version 5.19.

KOENKER, R. and BASSETT, G. (1978). Regression quantiles. *Econometrica: Journal of the Econometric Society* **46**, 33–50.

KOENKER, R. and BASSETT, G. (1982). Robust tests for heteroscedasticity based on regression quantiles. *Econometrica: Journal of the Econometric Society* **50**, 43–61.

KOENKER, R. and NG, P. (2005). A frisch-newton algorithm for sparse quantile regression. *Acta Mathematicae Applicatae Sinica* **21**, 225–236.

LI, Y. and ZHU, J. (2008). L1-norm quantile regression. *Journal of Computational and Graphical Statistics* **17**, 163–185.

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* , 1–27.

O'DONOGHUE, B., STATHOPOULOS, G. and BOYD, S. (2013). A splitting method for optimal control. *IEEE Transactions on Control Systems Technology* **21**, 2432–2442.

PARIKH, N. and BOYD, S. (2013). Proximal algorithms. *Foundations and Trends in optimization* **1**, 123–231.

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**, 676–694.

POLLARD, D. (1991). Asymptotics for least absolute deviation regression estimators. *Econometric Theory* **7**, 186–199.

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**, 14429–14434.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B* **58**, 267–288.

WANG, L. (2013). The L_1 penalized LAD estimator for high dimensional linear regression. *Journal of Multivariate Analysis* **120**, 135 – 151.

WANG, L., WU, Y. and LI, R. (2012). Quantile regression for analyzing heterogeneity in ultra-high dimension. *Journal of the American Statistical Association* **107**, 214–222.

WELSH, A. (1989). On M-processes and M-estimation. *The Annals of Statistics* **17**, 337–361.

WU, T. T. and LANGE, K. (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics* **2**, 224–244.

WU, Y. and LIU, Y. (2009). Variable selection in quantile regression. *Statistica Sinica* **19**, 801–817.

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**, 1480–1491.

YANG, J., MENG, X. and MAHONEY, M. W. (2013). Quantile regression for large-scale applications. *arXiv preprint arXiv:1305.0087v3* .

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* .

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**, 143–168.

ZHANG, T., ZOU, H. et al. (2014). Sparse precision matrix estimation via lasso penalized D-trace loss. *Biometrika* **101**, 103–120.

ZOU, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* **101**, 1418–1429.

ZOU, H. and HASTIE, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B* **67**, 301–320.

ZOU, H. and LI, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics* **36**, 1509.