Hindawi
Complexity
Volume 2021, Article ID 6341707, 16 pages
https://doi.org/10.1155/2021/6341707

Research Article

Communication-Efficient Modeling with Penalized Quantile Regression for Distributed Data

Aijun Hu,[1,2] **Chujin Li**,[1] **and Jing Wu**[3]

[1]*School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China*
[2]*School of Mathematics and Economics, Hubei University of Education, Wuhan 430205, Hubei, China*
[3]*Electronic Information School, Wuhan University, Wuhan 430072, Hubei, China*

Correspondence should be addressed to Chujin Li; lichujin@hust.edu.cn

Received 24 May 2020; Revised 9 November 2020; Accepted 25 November 2020; Published 16 January 2021

Academic Editor: Roberto Natella

In order to deal with high-dimensional distributed data, this article develops a novel and communication-efficient approach for sparse and high-dimensional data with the penalized quantile regression. In each round, the proposed method only requires the master machine to deal with a sparse penalized quantile regression which could be realized fastly by proximal alternating direction method of multipliers (ADMM) algorithm and the other worker machines to compute the subgradient on local data. The advantage of the proximal ADMM algorithm is that it could make every parameter of iteration to have closed formula even in high-dimensional case, which greatly improves the speed of calculation. As for the communication efficiency, the proposed method does not sacrifice any statistical accuracy and provably improves the estimation error obtained by centralized method, provided the penalty levels are chosen properly. Moreover, the asymptotic properties of the proposed estimation and the convergence of the algorithm are convincible. Especially, it presents extensive experiments on both the numerical simulations and the HIV drug resistance data analysis, which all confirm the significant efficiency of our proposed method in quantile regression for distributed data by comparative and empirical analysis.

1. Introduction

The quick developments of modern science and technology bring us to the distributed data which are characterized as not only the multiple scales and high dimension, but also the abundant diversity and variability. Distributed data promise and often deliver big dividends and also make diverse challenges to the statistic analysis. One special challenge is that the storage and analysis of such data cannot be performed on a single machine.

Since the data are stored in multiple machines, it is natural to consider methods that split the dataset across multiple machines and conduct statistical inference in a distributed manner [1–4]. One popular consideration is the divide-and-conquer (DC) method [5–8], noting the main idea of DC is to take the simple average and only use one round of communication between different data segmentations.

However, these averaging-based one-shot communication methods tend to suffer from several drawbacks. Firstly, when the number of machines for data segmentations is large, the local sample size in each machine will be reduced, which could result in the unsatisfactory estimation of each local part, and the final estimation after average would not be accurate. Secondly, when the underlying data model is nonlinear, empirical studies show that the average estimator can only be improved slightly on accuracy as for the local estimators. In order to effectively overcome the above deficiencies, Wang et al. [9] and Jordan et al. [10] proposed the communication-efficient surrogate likelihood (CSL) procedure to solve distributed statistical learning problems. Especially, Jordan et al. [10] pointed out that CSL can replace the total likelihood function to do some statistical inferences. Therefore, it only needs to exchange a number of local data gradients which could effectively reduce the transmission

cost of the distributed data. Moreover, the estimation of the method can reach the same convergence rate as the global likelihood-based estimation.

The aforementioned studies of DC are mainly focused on linear regression, which only consider the central trend of the conditional distribution of the response variable. As an alternative model, quantile regression proposed by Koenker and Bassett [11] is to analyze the impact of regressors on the conditional distribution of response. Buchinsky [12] captured the heterogeneous impact of regressors on different parts of the distribution, and Koenker [13] exhibited great robustness to outliers by measuring the effect of the regression variable in the upper or lower tails of the distribution. Especially when the error does not follow normal distribution, the quantile regression estimator is more effective and interesting than the least square estimator. Also, quantile regression for high-dimensional data has recently been systematically studied under the sparsity assumption, i.e., only a small number of predictors correlating with the response [14–20]. The quantile regression model selection procedure has been proved to have oracle property under some appropriate penalties [15, 16, 21]. Yu and Lin [22] and Yu et al. [23] used the popular ADMM algorithm to solve the calculation problem of large-scale penalized quantile regression. Recently Chen et al. [24] studied certain high-dimensional distributed quantile regression. The idea is to transform the quantile estimate into a least squares estimate between the transformed response variable and the covariate. However, the calculation needs to estimate the density function of the error and the covariance matrix of the covariate. The main difference from our method is that we approximate the total loss function of the sample with a surrogate loss function, which greatly reduces the amount of calculation.

Here, we propose a communication-efficient method with penalized quantile regression on distributed data in this article. The observation data are supposed to be stored randomly in multiple machines. Inspired by Jordan et al. [10] at each round, we only solve the problem on the master machine and calculate the subgradient on the other local machines. After one round of communication, the estimators are passed to each local machine to update the subgradient, and then they are transferred to the master machine for the next round of solution.

Importantly, noting that the loss function of quantile regression does not have strong convexity and the subderivative does not satisfy Lipschitz-continuity, the theoretical results by Jordan et al. [10] and Wang et al. [9] cannot be directly applied. Here, under more general conditions, we prove the estimates obtained by our proposed method have oracle properties as presented by Fan and Li [25] and Zou [26]. In terms of algorithms, because the loss function of quantile regression is not smooth, the usual calculation method may not be very accurate. In addition, the traditional ADMM algorithm on iteration of some parameters in high-dimensional quantile regression has no closed-form solution, which could affect the calculation speed and efficiency. Gu et al. [27] proposed the proximal ADMM algorithm to effectively resolve the calculation problem of sparse penalized quantile regression.

Hence, we combine it with CSL and propose the proximal ADMM algorithm based on distributed computing framework, which could effectively solve the computational problems of quantile regression on distributed data. Since our method has a display solution for each parameter in each iteration and could be performed point by point, it is very suitable for parallel computing. In each round of communication, only the subgradient calculated by the local data of each machine needs to be transmitted to the host, which greatly saves the cost of data transmission and improves the calculation efficiency.

Moreover, the convergence property of the proposed algorithm is still convincible. Considering the extensive numerical simulations and the analysis on HIV Drug Resistance Database, with our high-dimensional penalized quantile regression model, the estimation errors and variable selection results obtained by our proposed approach are compared with those obtained by the centralized method which put all the data on one supercomputer. The proposed method can not only effectively solve the problem by dealing with the heterogeneity of the distributed data but also reduce the cost of storage and transmission. Most importantly, the efficiency of the communication-efficient method with penalized quantile regression on distributed data could match or transcend the centralized method.

The rest of this article is organized as follows. We propose the communication-efficient method with penalized quantile regression in the next section and present the proximal ADMM algorithm for solving the communication-efficient penalized sparse quantile regression in Section 3. In Section 4, we prove the oracle properties of the proposed estimator. We conduct simulation studies to evaluate the finite-sample performance of our proposed method in Section 5. In Section 6, we demonstrate our method with application to one real data example. Some remarks are concluded in Section 7. The proofs are outlined in Appendix.

2. Penalized Distributed Quantile Regression

Let $\{y_i, \mathbf{x}_i\}_{i=1}^N$ be N sample observations, where y is the response variable and $\mathbf{x} = (x_1, \ldots, x_p)^T$ is the p-dimensional covariate; here, p and N are both very large. Assume the data are stored across k machines, and let $\{(y_{ji}, \mathbf{x}_{ji}): i = 1, \ldots, n\}$ denote the subsample stored in the j-th machine $\mathcal{M}_j$ for $j = 1, \ldots, k$, where $N = nk$ denoted the global number of sample observations.

Given the covariate $\mathbf{x} = (x_1, \ldots, x_p)^T$, considering the linear model $Y = \mathbf{x}^T\beta + \varepsilon$, the τth $(0 < \tau < 1)$ quantile of response variable Y is a linear function of the covariate $\mathbf{x} = (x_1, \ldots, x_p)^T$, that is,

$$Q_\tau(Y \mid \mathbf{x}) = \mathbf{x}^T\beta_0(\tau), \tag{1}$$

with $P(\varepsilon \le 0|\mathbf{x}) = \tau$. Here, the conditional distributed function of Y is written as $P(Y \le y|\mathbf{x}_i) = F_Y(y|\mathbf{x}_i) = F_i(y)$, and so, $Q_\tau(Y|\mathbf{x}_i) = F_i^{-1}(\tau|\mathbf{x}_i) \equiv \xi_i(\tau)$.

Hence, the quantile regression model is to resolve the following optimization problem:

$$\widehat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} \rho_\tau\left(y_i - \mathbf{x}_i^T\beta\right), \tag{2}$$

where $\rho_\tau(t) = t(\tau - 1_{\{t\le 0\}})$ is the asymmetric absolute deviation function. For ease of the exposition, we simplify the argument τ in $\beta_0(\tau)$ and $\xi_i(\tau)$ and denote them by β_0, ξ_i, respectively.

Define the local and global quantile regression loss function as

$$\begin{aligned} \mathscr{L}_j(\beta) &= \frac{1}{n}\sum_{i=1}^{n} \rho_\tau\left(y_{ji} - x_{ji}^T\beta\right), \quad j = 1, \ldots, k, \\ \mathscr{L}_N(\beta) &= \frac{1}{k}\sum_{j=1}^{k} \mathscr{L}_j(\beta) = \frac{1}{N}\sum_{j=1}^{k}\sum_{i=1}^{n} \rho_\tau\left(y_{ji} - x_{ji}^T\beta\right), \end{aligned} \tag{3}$$

where $\mathscr{L}_j(\beta)$ is calculated at β, using the local data stored in j-th machine $\mathscr{M}_j$.

We now apply communication-efficient distributed approach to quantile regression, by proposing the surrogate loss function as

$$\widetilde{\mathscr{L}}(\beta) \coloneqq \mathscr{L}_1(\beta) - \langle \beta, \nabla\mathscr{L}_1\left(\beta^0\right) - \nabla\mathscr{L}_N\left(\beta^0\right)\rangle, \tag{4}$$

where β^0 is any initial estimator of β, $\langle\cdot,\cdot\rangle$ denotes the inner product, and $\nabla\mathscr{L}_j(\beta)$ denotes the subgradient of $\mathscr{L}_j(\beta)$ with respect to β.

For high-dimensional quantile regression, we consider the following weighted L_1-penalized surrogate estimate of quantile regression:

$$\min_{\beta} \widetilde{\mathscr{L}}(\beta) + \sum\nolimits_{j=1}^{p} p_\lambda\left(\left|\beta_j\right|\right), \tag{5}$$

as a surrogate estimate for $\widehat{\beta}(\tau)$ in (2), where $\lambda > 0$ is the regularization parameter.

In the following, we use a convex ALasso (adaptive lasso) penalty and nonconvex SCAD (smoothly clipped absolute deviation) penalty to show the significance of our model. For ALasso penalty, $p_\lambda(|\beta_j|) = w_j|\beta_j|$, where $\mathbf{w} = (w_1, \ldots, w_p)^T$ is the vector of nonnegative weight, $w_j \ge 0$, $j = 1, \ldots, p$. The typical choice of w_j is $w_j = (|\widehat{\beta}_j^{\text{lasso}}| + 1/n)^{-\nu}$, $j = 1, \ldots, p$ for some appropriately chosen $\nu > 0$, where $\widehat{\beta}^{\text{lasso}} = (\widehat{\beta}_j^{\text{lasso}}, j = 1, \ldots, p)^T$ denotes the quantile lasso estimator.

Note that (5) is a convex optimization problem, and the calculation is relatively easy. However, if the penalty is nonconvex, such as SCAD [28] case, we have

$$\begin{aligned} p_\lambda(|\beta|) &= \lambda|\beta| I(0 \le |\beta| < \lambda) + \frac{a\lambda|\beta| - \left(\beta^2 + \lambda^2\right)/2}{a-1} \\ &\cdot I(\lambda \le |\beta| < a\lambda) + \frac{(a+1)\lambda^2}{2} I(|\beta| > a\lambda), \end{aligned} \tag{6}$$

for some $a > 2$. A typical choice is $a = 3.7$ as suggested by Fan and Li [25]. Although the nonconvex penalty has a promising theoretical property, the singularity and nonconvexity of the penalty function have led us to encounter several challenges in computing. Zou and Li [29] proposed to replace the nonconvex penalized function by the local linear approximation (LLA).

Then, the penalized procedure with both convex penalty and nonconvex penalty can be transformed as the following weighted optimization problem:

$$\min_{\beta} \widetilde{\mathscr{L}}(\beta) + \lambda\|\mathbf{w} \circ \beta\|_1, \tag{7}$$

where $\|\mathbf{w} \circ \beta\|_1 = \sum_{j=1}^{p} |w_j\beta_j| = \sum_{j=1}^{p} w_j|\beta_j|$. In the LLA case, we usually take $w_j = \lambda^{-1} p_\lambda'(|\widehat{\beta}_j^{s-1}|)$, $j = 1, \ldots, p$, where $\widehat{\beta}^{s-1} = (\widehat{\beta}_j^{s-1}, j = 1, \ldots, p)^T$ is the estimation of the $(s-1)$th step iteration.

3. Algorithm Analysis

Now, we set up to develop a proximal communication-efficient surrogate likelihood ADMM (PCA) algorithm for solving (7). We firstly solve the quantile on the master machine and calculate the subgradient of the local data on other machines. Secondly, we transfer these values back to the master machine and combine them by formula (7), then use Algorithm 1 to calculate the result, which constitutes one communication. Using Algorithm 2 and after only a few rounds of communications T, we can obtain excellent results comparable to the centralized method. The advantage of our algorithm is that each step of Algorithms 1 and 2 has a simple closed-form updata formula, which makes the calculation speed being very fast, especially for the high-dimensional case. Furthermore, Algorithm 2 combined with the CSL idea can achieve very accurate results.

Define $z = \mathbf{y} - \mathbf{X}\beta$ and $\mathbb{Q}_\tau(z) = (1/n)\sum_{i=1}^{n} \rho_\tau(z_i)$, where $\mathbf{X} = (\mathbf{x}_{11}, \ldots, \mathbf{x}_{1n})^T$, $\mathbf{y} = (y_{11}, \ldots, y_{1n})^T$, and z_i $(i = 1, \ldots, n)$ is the i-th element of z. Hence, we consider the following optimization problem:

$$\begin{aligned} &\min_{\beta,z} \mathbb{Q}_\tau(z) + \mathbf{g}^T\beta + \lambda\|\mathbf{w} \circ \beta\|_1 \\ &\text{s.t.} \mathbf{X}\beta + z = \mathbf{y}, \end{aligned} \tag{8}$$

where $\mathbf{g}$ is a given subgradient vector with $\mathbf{g} = \nabla\mathscr{L}_N(\beta^0) - \nabla\mathscr{L}_1(\beta^0)$, and β^0 is any initial estimate of β, and the subgradient of loss function at zero can take any value between two slopes.

By direct calculations, we have

$$\begin{cases} \nabla\mathscr{L}_1(\beta) = -\dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \mathbf{x}_{1i}\psi_\tau\left(y_{1i} - \mathbf{x}_{i1}^T\beta\right), \\ \nabla\mathscr{L}_N(\beta) = -\dfrac{1}{N}\displaystyle\sum_{j=1}^{k}\sum_{i=1}^{n} \mathbf{x}_{ji}\psi_\tau\left(y_{ji} - \mathbf{x}_{ji}^T\beta\right), \end{cases} \tag{9}$$

where $\psi_\tau(u) = \nabla\rho_\tau(u) = \tau I_{(u>0)} + (\tau - 1)\ I_{(u<0)} + \xi I_{(u=0)}$, $\xi \in [\tau - 1, \tau]$.

According to the idea of [10], we calculate the subgradient vectors on each local machine by (9), then calculate (7) for the master machine. Since $\widetilde{\mathscr{L}}(\beta)$ here is a nonsmooth

Input: data $\{y_i, \mathbf{x}_i\}_{i=1}^N$. Constants $\sigma > 0$ and $\tau > 0$. Set the maximum number of iterations M.
(1) **For** $m = 0, 1, 2, \ldots, M-1$ **do**
(2) Update β^{m+1} via (13).
(3) Update z^{m+1} via (14).
(4) Update θ^{m+1} via (11).
(5) **End for**
Output (β^M, z^M, θ^M).

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

Input: data $\{y_i, \mathbf{x}_i\}_{i=1}^N$ on machine $\mathscr{M}_j$, $j = 1, 2, \ldots, k$. Constants $\sigma > 0$ and $\tau > 0$. Initialize the algorithm with $(\beta^0, z^0, \theta^0) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ and $\mathbf{g} = \mathbf{0}$.
(1) Set $\mathscr{M}_1$ as the master machine and obtain the initial value $(\beta^0, z^0, \theta^0) = (\beta^M, z^M, \theta^M)$ on $\mathscr{M}_1$ using Algorithm 1.
(2) **For** $t = 0, 1, 2, \ldots, T-1$ **do**
(3) Master machine $\mathscr{M}_1$ broadcasts β^t to each worker (i.e., $\mathscr{M}_2, \ldots, \mathscr{M}_k$).
(4) Each machine $\mathscr{M}_m$ calculates subgradient $\nabla \mathscr{L}_m(\beta^t)$, $m = 1, \ldots, k$ and reduces them to master machine $\mathscr{M}_1$ to form $\mathbf{g}^t = \nabla \mathscr{L}_N(\beta^t) - \nabla \mathscr{L}_1(\beta^t)$. Update (8) by replacing $\mathbf{g}$ with $\mathbf{g}^t$.
(5) Master machine $\mathscr{M}_1$ updates (β^t, z^t, θ^t) via Algorithm 1.
(6) **End for**
Output: β^T.

Algorithm 2: (PCA)-proximal communication-efficient surrogate likelihood ADMM algorithm for solving the weighted L_1-penalized quantile regression.

function and the direct calculation efficiency is very low, we use ADMM [30] to approximate $\widetilde{\mathscr{L}}(\beta)$ as follows. For fixed $\sigma > 0$, the augmented Lagrangian function of (8) is defined as

$$L_\sigma(\beta, z, \boldsymbol{\theta}) = \mathbb{Q}_\tau(z) + \mathbf{g}^T\beta + \lambda\|\mathbf{w} \circ \beta\|_1 - \langle \boldsymbol{\theta}, \mathbf{X}\beta + z - y\rangle + \frac{\sigma}{2}\|\mathbf{X}\beta + z - y\|_2^2, \tag{10}$$

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. Denoting (β^m, z^m, θ^m) as the mth iteration of the algorithm for $m \geq 0$, the standard ADMM algorithm is given by

$$\begin{aligned}
\beta \text{ step: } \beta^{m+1} &= \arg\min_\beta L_\sigma(\beta, z^m, \boldsymbol{\theta}^m) \\
&= \arg\min_\beta \mathbf{g}^T\beta + \lambda\|\mathbf{w} \circ \beta\|_1 - \langle \boldsymbol{\theta}^m, \mathbf{X}\beta\rangle \\
&\quad + \frac{\sigma}{2}\|\mathbf{X}\beta + z^m - \mathbf{y}\|_2^2, \\
z \text{ step: } z^{m+1} &= \arg\min_z L_\sigma\left(\beta^{m+1}, z, \boldsymbol{\theta}^m\right) \\
&= \arg\min_z \mathbb{Q}_\tau(z) - \langle \boldsymbol{\theta}^m, z\rangle + \frac{\sigma}{2}\|\mathbf{X}\beta^{m+1} + z - \mathbf{y}\|_2^2, \\
\boldsymbol{\theta} \text{ step: } \boldsymbol{\theta}^{m+1} &= \boldsymbol{\theta}^m - \gamma\sigma\left(\mathbf{X}\beta^{m+1} + z^{m+1} - \mathbf{y}\right),
\end{aligned} \tag{11}$$

where γ is a constant controlling the step length for the θ step.

Note that β step does not have closed-form formula with a general design matrix $\mathbf{X}$, usually just a numerical solution. Gu et al. [27] proposed a proximal ADMM algorithm for computing the sparse penalized quantile regression. We use this idea to improve our algorithm so that the iteration of each parameter has a closed-form expression, which speeds up the calculation obviously, and the details are as follows. We consider to adding a proximal term to the objective function in the β step and modify the β step in (11) with the following augmented β step.

Augmented β step:

$$\begin{aligned}
\beta^{m+1} \coloneqq \arg\min_\beta &\ \mathbf{g}^T\beta + \lambda\|\mathbf{w} \circ \beta\|_1 - \langle \boldsymbol{\theta}^m, \mathbf{X}\beta\rangle \\
&+ \frac{\sigma}{2}\|\mathbf{X}\beta + z^m - \mathbf{y}\|_2^2 + \frac{1}{2}\|\beta - \beta^m\|_{\mathbf{S}}^2,
\end{aligned} \tag{12}$$

where $\mathbf{S}$ is a positive semidefinite matrix.

Let $\mathbf{S} = \sigma(\eta\mathbf{I}_p - \mathbf{X}^T\mathbf{X})$ with $\eta \geq \Lambda_{\max}(\mathbf{X}^T\mathbf{X})$, where $\Lambda_{\max}(\cdot)$ denotes the largest eigenvalue of a real symmetric matrix. Here, $\|v\|_{\mathbf{S}}^2 \coloneqq \langle v, \mathbf{S}v\rangle$ is the seminorm induced by the semiinner product via $\mathbf{S}$. In the augmented β step, the update of β can also be carried out component-wisely,

$$\begin{aligned}
\beta^{m+1} &\coloneqq \arg\min_\beta \mathbf{g}^T\beta + \lambda\|\mathbf{w} \circ \beta\|_1 - \langle \boldsymbol{\theta}^m, \mathbf{X}\beta\rangle \\
&\quad + \frac{\sigma\eta}{2}\left\|\beta - \frac{\sigma\eta\beta^m - \mathbf{g} + \mathbf{X}^T\left(\boldsymbol{\theta}^m + \sigma\mathbf{y} - \sigma\mathbf{X}\beta^m - \sigma z^m\right)}{\sigma\eta}\right\|_2^2 \\
&= \left(\text{Shrink}\left[\beta_j^m - \frac{g_j}{\sigma\eta} + \frac{1}{\sigma\eta}\Delta_j, \frac{\lambda w_j}{\sigma\eta}\right]\right)_{1 \leq j \leq p},
\end{aligned} \tag{13}$$

where $\Delta_j = \mathbf{X}_j^T(\theta^m + \sigma\mathbf{y} - \sigma\mathbf{X}\beta^m - \sigma z^m)$, $\text{Shink}[u, \alpha] = \text{sgn}(u)\max(|u| - \alpha, 0)$ denotes the soft shrinkage operator with

$\operatorname{sgn}(\cdot)$ being the sign function and $\mathbf{X}_j$ denotes the jth column of $\mathbf{X}$, $j = 1, \ldots, p$, β_j^m, and g_j ($j = 1, \ldots, p$) are the jth components of β^m and $\mathbf{g}$, respectively.

The update of z^k has a closed-form solution and can be implemented component-wisely too. Namely, for $i = 1, \ldots, n$, we have

$$\begin{aligned} z_i^{m+1} &\coloneqq \arg\min_{z_i} \frac{1}{n}\rho_\tau(z_i) - \theta_i^m z_i + \frac{\sigma}{2}\left(\mathbf{x}_{1i}^T\beta^{m+1} + z_i - y_{1i}\right)^2 \\ &= \arg\min_{z_i} \rho_\tau(z_i) + \frac{n\sigma}{2}\left[z_i - \left(y_{1i} - \mathbf{x}_{1i}^T\beta^{m+1} + \frac{1}{\sigma}\theta_i^m\right)\right]^2 \\ &= \operatorname{Prox}_{\rho_\tau}\left(y_{1i} - \mathbf{x}_{1i}^T\beta^{m+1} + \frac{1}{\sigma}\theta_i^m, n\sigma\right), \end{aligned} \tag{14}$$

where the proximal mapping operator $\operatorname{Prox}_{\rho_\tau}(\cdot,\cdot)$ is given by Gu et al. [27] as follows:

$$\begin{aligned} \operatorname{Prox}_{\rho_\tau}(\xi, \alpha) &\coloneqq \arg\min_{u\in\mathbb{R}} \rho_\tau(u) + \frac{\alpha}{2}(u-\xi)^2 \\ &= \max\left(\xi - \frac{\tau}{a}, 0\right) - \max\left(-\xi - \frac{1-\tau}{a}, 0\right). \end{aligned} \tag{15}$$

In the following, we present the algorithms. Algorithm 1 is to solve the problem on the master machine and the final communication-efficient distributed estimation can be obtained by Algorithm 2.

4. Asymptotic Properties

In this section, for establishing the theoretical properties of the penalized communication-efficient quantile regression, we assume the data $\{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$ consist of N observations with the quantile regression model

$$y_i = \mathbf{x}_i^T\beta_0 + \varepsilon_i = \mathbf{x}_{i1}^T\beta_{10} + \mathbf{x}_{i2}^T\beta_{20} + \varepsilon_i, \quad i = 1, \ldots, N, \tag{16}$$

with $P(\varepsilon_i \le 0 \mid \mathbf{x}) = \tau$. Here, $\mathbf{x}_i = (\mathbf{x}_{i1}^T, \mathbf{x}_{i2}^T)^T, \beta_0 = (\beta_{10}^T, \beta_{20}^T)^T$, $\mathbf{x}_{i1} \in \mathbb{R}^s, \mathbf{x}_{i2} \in \mathbb{R}^{p-s}$. The model is sparse, and the true regression coefficient is β_{10} with each component being nonzero, and $\beta_{20} = \mathbf{0}$ (as a result $\beta_0 = (\beta_{10}^T, \mathbf{0}^T)^T$). This means the first s regressors are important while the remaining $p - s$ regressors are noise variables.

In the following, we impose the conditions as follows:

C1. The distributed functions $\{F_i\}$ are absolutely continuous, with continuous densities $f_i(\xi_i)$ uniformly bounded away from 0 and ∞ at the points $\xi_i, i = 1, 2, \ldots$.

C2. For $j = 1, \ldots, k$, there exists positive definite matrices Σ_0 and $\Sigma_1(\tau)$ such that

(i) $\lim_{n\longrightarrow\infty} n^{-1}\sum_{i=1}^n \mathbf{x}_{ji}\mathbf{x}_{ji}^T = \Sigma_0$,

(ii) $n^{-1}\sum_{i=1}^n f_i(\xi_i)\mathbf{x}_{ji}\mathbf{x}_{ji}^T = \Sigma_1(\tau)$,

(iii) $\max_{i=1,\ldots,n}\|\mathbf{x}_{ji}\|_2/\sqrt{n} \longrightarrow 0$.

C3. $\lambda_n \longrightarrow 0$ and $\sqrt{n}\lambda_n \longrightarrow \infty$ as $n \longrightarrow \infty$ (SCAD penalty condition).

C4. $\sqrt{n}\lambda_n \longrightarrow 0$ and $n^{(\nu+1)/2}\lambda_n \longrightarrow \infty$ for some appropriately chosen $\nu > 0$ (ALasso penalty condition).

For ease of the exposition, we simplify the argument τ in $\Sigma_1(\tau)$ and denote it by Σ_1.

Remark 1. The condition C1 here is a common condition in general quantile regression. The condition C2 is the technical condition being used in [31]. The conditions C3 and C4 here are the classic conditions for tuning parameters in the SCAD and ALasso penalties, respectively.

Now, we present the asymptotic properties of the communication-efficient distributed quantile regression for the convex penalty (ALasso) and the nonconvex penalty (SCAD).

Theorem 1 (consistency). *Consider a sample $\{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$ from model (16). For the SCAD penalty with conditions C1, C2, and C3, or for the ALasso penalty with conditions C1, C2, and C4, there exists a local minimizer $\widetilde{\beta}$ such that $\|\widetilde{\beta} - \beta_0\| = O_p(n^{-1/2})$.*

The following theorem gives the asymptotic oracle properties of the estimate.

Theorem 2 (oracle). *Consider a sample $\{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$ from model (16). For the SCAD penalty if conditions C1, C2, and C3 are satisfied, or for the ALasso penalty if conditions C1, C2 and C4 are satisfied, then we have*

(a) Sparsity: $\widetilde{\beta}_2 = \mathbf{0}$;

(b) Asymptotic normality:

$$\sqrt{N}\left(\widetilde{\beta}_1 - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, D\Sigma_1^{-1}\Sigma_{01}\Sigma_1^{-1}\right), \tag{17}$$

where $\longrightarrow^d$ means convergence in terms of distribution and Σ_{01} is defined as the top-left s-by-s submatrix of Σ_0, $D = k\tau(1-\tau) + (k-1)\{Var(\psi_\tau(\overline{\varepsilon})) - 2Cov(\psi_\tau(\overline{\varepsilon}), \psi_\tau(\varepsilon))\}$, $\overline{\varepsilon} = y - \mathbf{x}^T\beta^0$, $\varepsilon = y - \mathbf{x}^T\beta_0$.

Note the initial value β^0 satisfies $\|\beta^0 - \beta_0\|_2 = O_p(n^{-1/2})$ and the regression error $\{\varepsilon_i\}$ is independently and identically distributed, with the τth quantile zero and a continuous density $f(\cdot)$ in a neighborhood of zero.

We could obtain by Theorem 2 that $D = \tau(1-\tau)$, and $\sqrt{N}(\widetilde{\beta}_1 - \beta_{10}) \longrightarrow^d \mathcal{N}(\mathbf{0}, \tau(1-\tau)\Sigma_{01}^{-1}/f^2(0))$. In this way, the covariance of the proposed method is exactly same as that of the centralized method [16]. Commonly, since the initial value β^0 can be chosen from the first machine satisfying $\|\beta^0 - \beta_0\|_2 = O_p(n^{-1/2})$, this means that our proposed estimate is unbiased and applicable.

In the following, we establish the convergence of Algorithm 1 and Algorithm 2. Note that $\mathbf{g}$ is a constant; by Theorem 1 of Gu et al. [27], the convergence of Algorithm 1 is direct. Also, by choosing the appropriate step length γ,

PCA Algorithm 2 can own a sequence $\{(\beta^t, z^t), t = 1, 2, \ldots\}$ that converges to the global minimizer of problem (7).

Theorem 3. *Given* $\lambda > 0, \sigma > 0, 0 < \tau < 1, 0 < \gamma < (\sqrt{5}+1)/2$ *and a nonnegative weight vector* $\mathbf{w}$, *let* (β^t, z^t, θ^t) *be generated by PCA Algorithm 2. Then, the sequence* $\{(\beta^t, z^t), t = 0, 1, \ldots\}$ *converges to an optimal solution* (β^*, z^*) *to (8), and* $\{\theta^t, t = 0, 1, \ldots\}$ *converges to an optimal solution* θ^* *to the dual problem of (8). Equivalently,* $\{\beta^t, t = 0, 1, \ldots\}$ *converges to a global minimizer of problem (7). Moreover, when* $\gamma = 1$, *the sequence of norms* $\{\|\beta^t - \beta^*\|_{\mathbf{S}}^2 + \sigma\|z^t - z^*\|_2^2 + \sigma^{-1}\|\theta^t - \theta^*\|_2^2, t \geq 0\}$ *is nonincreasing and satisfies* $\{\|\beta^t - \beta^*\|_{\mathbf{S}}^2 + \sigma\|z^t - z^*\|_2^2 + \sigma^{-1}\|\theta^t - \theta^*\|_2^2 = O_p(1/t)\}$ *as* $t \longrightarrow \infty$.

5. Numerical Analysis

In this section, we demonstrate simulation studies to evaluate the finite-sample performance of the communication-efficient distributed quantile regression approach. Since least squares regression has no advantage in dealing with heterogeneous data compared with quantile regression, we only focus on distributed quantile regression model in simulation comparison.

Now, we consider the heteroscedastic location-scale model as in Wang et al. [18] for comparing the proposed method with the centralized method which is supposed to store all the data together in one supercomputer. Denote the proposed method and the centralized method with ALasso and SCAD penalties by *E*-ALasso, *E*-SCAD, *C*-ALasso, and *C*-SCAD, respectively.

Let the random vectors $(Z_1, Z_2, \ldots, Z_p)^T$ generate by the multivariate normal distribution $N_p(0, \Sigma)$, with $\Sigma = (\sigma_{ij})_{p\times p}$ and $\sigma_{ij} = 0.5^{|i-j|}$, and $X_1 = \Phi(Z_1)$ and $X_j = Z_j$ for $j = 2, 3, \ldots, p$. The scalar response is generated according to the following heteroscedastic model:

$$Y = X_6 + X_{12} + X_{15} + X_{20} + 0.7X_1\varepsilon, \tag{18}$$

where ε is distributed as the standard normal distribution $N(0, 1)$, the student's distribution $t(2)$, and the Laplace distribution $L(0, 1)$ with the density being $(1/2)e^{-|x|}$, respectively. In that, these three distributions represent the light or heavy tail characteristic.

Here, we set $n = 100$, $p = 300$, the machine number $k = 10$ or 50, and the total sample size $N = nk$. We randomly divide all the data S into k subsets and denote the data stored in the j-th machine as $S^{(j)}$, $j = 1, \ldots, k$. We use the data $S^{(j)}$ in one machine as the test set and the data $S - S^{(j)}$ of the remaining $(k-1)$ machines as the training set to estimate the parameter $\widetilde{\beta}^{(j)}(\lambda)$. By selecting the tuning parameter λ, the k-fold cross-validation is used to minimize the estimate prediction error in our communication-efficient distributed model. As for each λ, the k-fold cross-validation standard is

$$\mathrm{CV}(\lambda) = \sum_{j=1}^{k} \sum_{(\mathbf{x}_j, y_j)\in S^{(j)}} \rho_\tau\left(y_j - \mathbf{x}_j^T\widetilde{\beta}^{(j)}(\lambda)\right), \tag{19}$$

then it is to resolve $\widehat{\lambda} = \mathrm{argmin}_\lambda \mathrm{CV}(\lambda)$. In the centralized case, we select the regularization parameters according to the usual 5-fold cross-validation.

In each case, we present 100 repeated simulations and compare the results obtained by the two methods mentioned above in accordance with the following four criteria:

Size: the average number containing nonzero regression coefficients: $\widehat{\beta}_j \neq 0$, $j = 1, 2, \ldots, p$.

P_a: the proportion including all true important regression variables, namely, $\widehat{\beta}_j \neq 0$, for any $j \geq 1$, satisfying $\beta_j \neq 0$. This means the percentage of time including $X_6, X_{12}, X_{15}, X_{20}$, and X_1 at $\tau = 0.3$ and $\tau = 0.7$, while X_1 does not have to be included at $\tau = 0.5$, if the error distribution is symmetric about the zero.

P_1: the proportion of simulation running if X_1 is selected.

AE: the average of absolute estimation error defined by $\sum_{j=1}^{p} |\widehat{\beta}_j - \beta_j|$.

Time: the average running time of the CPU in 100 runs.

For the two cases of $k = 10$ and $k = 50$, Tables 1 and 2 , respectively, depict the simulation results of the above-mentioned five criteria. From Tables 1 and 2, we can see that for three different types of error distributions and two different numbers of storage machines, our proposed method has a very good performance, and the results could match or transcend the centralized situation. Specifically, the results of the two methods are basically the same on the first three criteria. As for the five criteria, the proposed method is even smaller in the majority of the average absolute error than the centralized method, but the standard deviation does not change much. This can be explained as our method is to reuse the data information of each machine in communication, and similar results were also obtained by Wang et al. [9]. For the error distribution of three different tails and the two different machine numbers, the calculation results of the simulated AE are also very close, which explain the advantages of our method in dealing with variable selection and estimation of distributed data.

Tables 1 and 2 show the comparison of CPU running time for 10 rounds of communication. It is clear that when the number of machines is 10, the running times of the two methods are similar. However, when the number of machines is increased to 50, the running time of our method is significantly shorter than that of the oracle method, which fully demonstrates the effective communication superiority of our method for dealing with big data.

Figure 1 plots how the estimation error $\|\beta^t - \beta^*\|_2$ varies for the proposed method with the ALasso penalty when $k = 50$, but the estimation error of the centralized method looks as a horizontal line. In addition, the results of $k = 10$ and SCAD penalty are similar to Figure 1, which we will omit here. Moreover, by Figure 1, the estimation error obtained by our method decreases to be truly competitive with the centralized method within very few rounds of communications. Usually, it just requires less than 5 communications and could achieve smaller prediction error than the centralized method. This phenomenon also appeared in [9], which can be explained as our method is to use data information repeatedly.

Table 1: Simulation results for $k = 10$, $n = 100$, $p = 300$, and $N = nk$.

ε	τ	Method	Size	P_a%	P_1%	AE	Time
$N(0,1)$	0.3	*E*-ALasso	5 (0)	100	100	0.38 (0.03)	0.18
		C-ALasso	5 (0)	100	100	0.40 (0.03)	0.21
		E-SCAD	5 (0)	100	100	0.22 (0.05)	0.18
		C-SCAD	5 (0)	100	100	0.33 (0.03)	0.21
	0.5	*E*-ALasso	4 (0)	100	0	0.03 (0.01)	0.18
		C-ALasso	4 (0)	100	0	0.04 (0.02)	0.21
		E-SCAD	4 (0)	100	0	0.03 (0.01)	0.18
		C-SCAD	4 (0)	100	0	0.07 (0.02)	0.21
	0.7	*E*-ALasso	5 (0)	100	100	0.35 (0.03)	0.18
		C-ALasso	5 (0)	100	100	0.40 (0.04)	0.21
		E-SCAD	5.01 (0.10)	100	100	0.22 (0.04)	0.18
		C-SCAD	5 (0)	100	100	0.31 (0.03)	0.21
$t(2)$	0.3	*E*-ALasso	5 (0)	100	100	0.45 (0.04)	0.19
		C-ALasso	5 (0)	100	100	0.47 (0.04)	0.21
		E-SCAD	5.02 (0.14)	100	100	0.31 (0.06)	0.19
		C-SCAD	5 (0)	100	100	0.34 (0.04)	0.21
	0.5	*E*-ALasso	4 (0)	100	0	0.03 (0.01)	0.18
		C-ALasso	4 (0)	100	0	0.04 (0.01)	0.21
		E-SCAD	4 (0)	100	0	0.05 (0.02)	0.19
		C-SCAD	4 (0)	100	0	0.04 (0.01)	0.21
	0.7	*E*-ALasso	5 (0)	100	100	0.44 (0.04)	0.19
		C-ALasso	5 (0)	100	100	0.46 (0.04)	0.23
		E-SCAD	5 (0)	100	100	0.31 (0.06)	0.19
		C-SCAD	5 (0)	100	100	0.35 (0.04)	0.21
$L(0,1)$	0.3	*E*-ALasso	5 (0)	100	100	0.37 (0.04)	0.19
		C-ALasso	5 (0)	100	100	0.39 (0.04)	0.21
		E-SCAD	5.07 (0.26)	100	100	0.27 (0.06)	0.18
		C-SCAD	5 (0)	100	100	0.32 (0.04)	0.21
	0.5	*E*-ALasso	4 (0)	100	0	0.03 (0.01)	0.17
		C-ALasso	4 (0)	100	0	0.04 (0.01)	0.19
		E-SCAD	4 (0)	100	0	0.04 (0.02)	0.18
		C-SCAD	4 (0)	100	0	0.05 (0.02)	0.21
	0.7	*E*-ALasso	5 (0)	100	100	0.37 (0.04)	0.18
		C-ALasso	5 (0)	100	100	0.40 (0.04)	0.21
		E-SCAD	5 (0)	100	100	0.19 (0.04)	0.18
		C-SCAD	5 (0)	100	100	0.30 (0.04)	0.21

Size, the average number containing nonzero regression coefficients; P_a, the proportion including all true important regression variables; P_1, the proportion of simulation run X_1 is selected; AE, the average of absolute estimation error, the numbers in parentheses are the corresponding standard deviations; Time, CPU average running time. $L(0,1)$, Laplace$(0,1)$ distribution.

Table 2: Simulation results for $k = 50$, $n = 100$, $p = 300$, and $N = nk$.

ε	τ	Method	Size	P_a%	P_1%	AE	Time
$N(0,1)$	0.3	*E*-ALasso	5 (0)	100	100	0.36 (0.02)	0.37
		C-ALasso	5 (0)	100	100	0.37 (0.02)	1.24
		E-SCAD	5.05 (0.26)	100	100	0.27 (0.05)	0.37
		C-SCAD	5 (0)	100	100	0.32 (0.02)	1.24
	0.5	*E*-ALasso	4 (0)	100	0	0.03 (0.01)	0.36
		C-ALasso	4 (0)	100	0	0.03 (0.01)	1.24
		E-SCAD	4 (0)	100	0	0.03 (0.01)	0.37
		C-SCAD	4 (0)	100	0	0.01 (0.00)	1.24
	0.7	*E*-ALasso	5 (0)	100	100	0.35 (0.02)	0.36
		C-ALasso	5 (0)	100	100	0.38 (0.02)	1.24
		E-SCAD	5 (0)	100	100	0.27 (0.04)	0.36
		C-SCAD	5 (0)	100	100	0.32 (0.02)	1.23

Table 2: Continued.

ε	τ	Method	Size	P_a%	P_1%	AE	Time
$t(2)$	0.3	*E*-ALasso	5 (0)	100	100	0.43 (0.02)	0.36
		C-ALasso	5 (0)	100	100	0.46 (0.02)	1.21
		E-SCAD	5.03 (0.17)	100	100	0.39 (0.05)	0.36
		C-SCAD	5 (0)	100	100	0.39 (0.02)	1.21
	0.5	*E*-ALasso	4 (0)	100	0	0.02 (0.01)	0.36
		C-ALasso	4 (0)	100	0	0.04 (0.01)	1.21
		E-SCAD	4 (0)	100	0	0.03 (0.01)	0.36
		C-SCAD	4 (0)	100	0	0.02 (0.01)	1.21
	0.7	*E*-ALasso	5 (0)	100	100	0.43 (0.02)	0.36
		C-ALasso	5 (0)	100	100	0.45 (0.03)	1.19
		E-SCAD	5 (0)	100	100	0.29 (0.04)	0.35
		C-SCAD	5 (0)	100	100	0.38 (0.03)	1.19
$L(0,1)$	0.3	*E*-ALasso	5 (0)	100	100	0.34 (0.02)	0.34
		C-ALasso	5 (0)	100	100	0.37 (0.02)	1.13
		E-SCAD	5.02 (0.14)	100	100	0.25 (0.05)	0.36
		C-SCAD	5 (0)	100	100	0.30 (0.02)	1.25
	0.5	*E*-ALasso	4 (0)	100	0	0.03 (0.02)	0.33
		C-ALasso	4 (0)	100	0	0.03 (0.01)	1.10
		E-SCAD	4 (0)	100	0	0.03 (0.02)	0.33
		C-SCAD	4 (0)	100	0	0.01 (0.00)	1.11
	0.7	*E*-ALasso	5 (0)	100	100	0.36 (0.02)	0.34
		C-ALasso	5 (0)	100	100	0.36 (0.02)	1.11
		E-SCAD	5.01 (0.10)	100	100	0.25 (0.04)	0.37
		C-SCAD	5 (0)	100	100	0.29 (0.02)	1.25

Size, the average number containing nonzero regression coefficients; P_a, the proportion including all true important regression variables; P_1, the proportion of simulation run X_1 is selected; AE, the average of absolute estimation error, the numbers in parentheses are the corresponding standard deviations; Time, CPU average running time. $L(0,1)$, Laplace$(0,1)$ distribution.

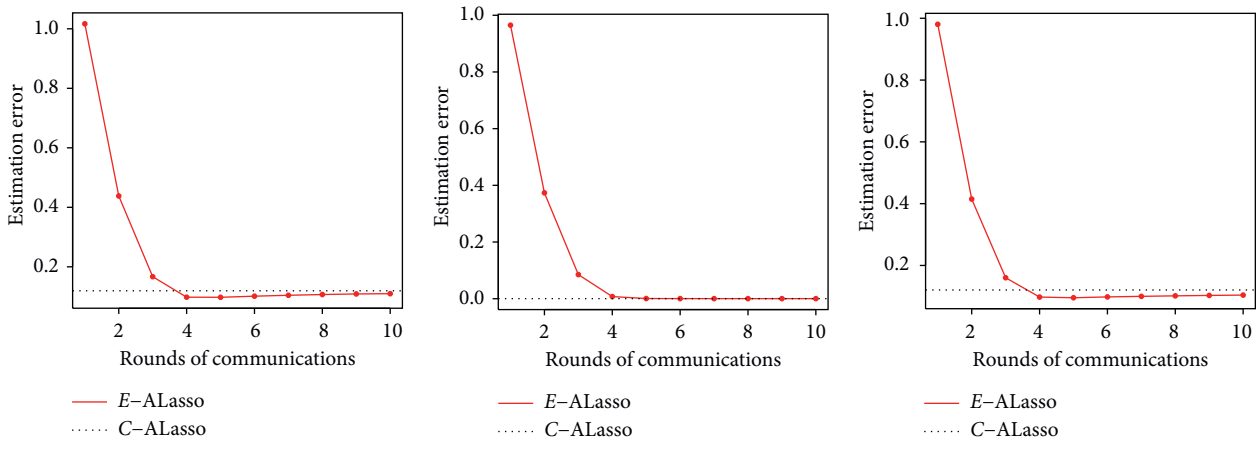

Figure 1: Continued.

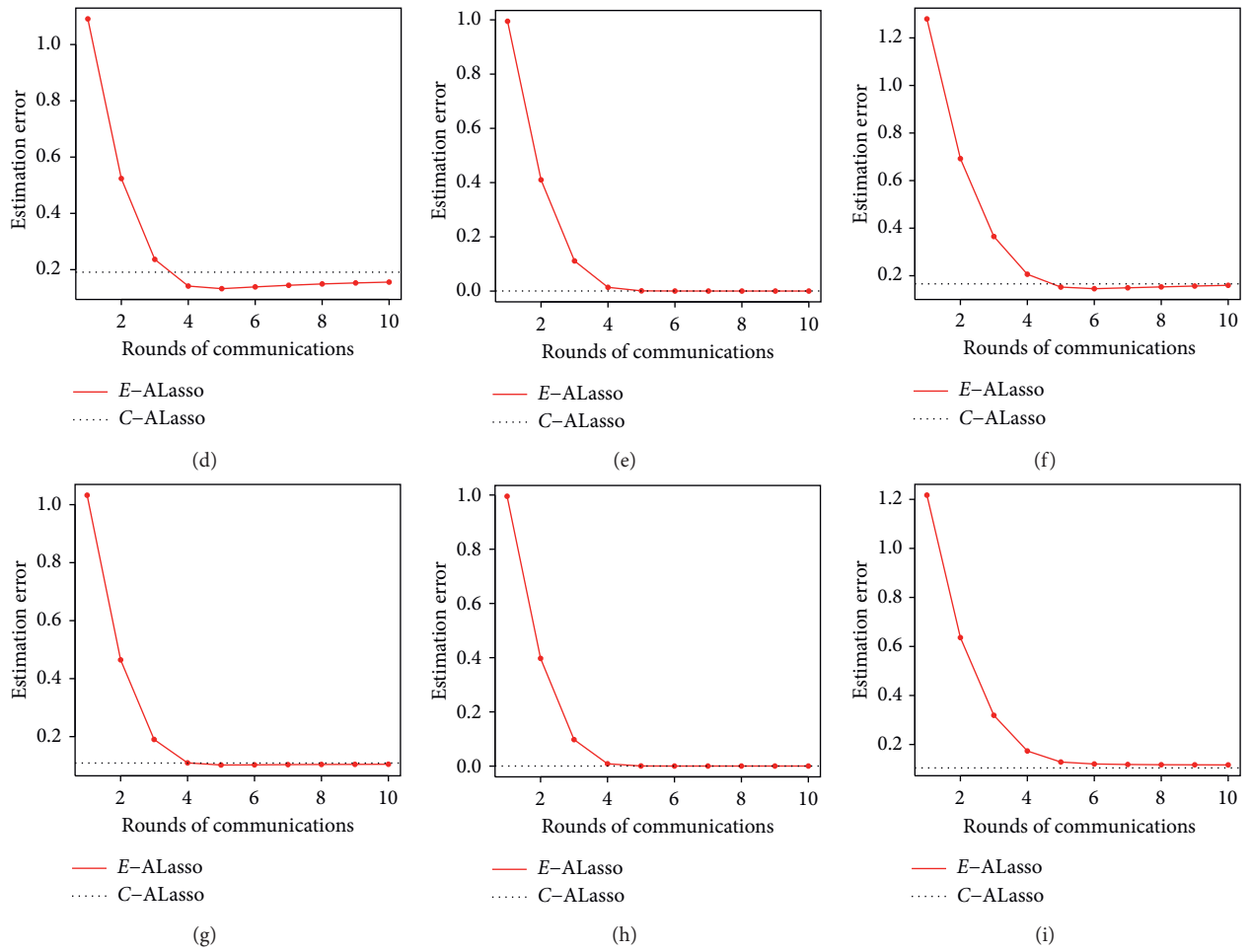

FIGURE 1: Comparison of two methods for the estimation error based on $k = 50$; the sparse ALasso penalized quantile regression. $L(0, 1)$ denote the Laplace$(0, 1)$ distribution. (a) $N(0, 1), \tau = 0.3$. (b) $N(0, 1), \tau = 0.5$. (c) $N(0, 1), \tau = 0.7$. (d) $t(2), \tau = 0.3$. (e) $t(2), \tau = 0.5$. (f) $t(2), \tau = 0.7$. (g) $L(0, 1), \tau = 0.3$. (h) $L(0, 1), \tau = 0.5$. (i) $L(0, 1), \tau = 0.7$.

6. Study on HIV Drug Resistance

As an illustration, we apply the communication-efficient penalized quantile regression model on a human immunodeficiency virus (HIV) drug susceptibility dataset from the HIV Drug Resistance Database (http://hivdb.stanford.edu), a public resource for the study of sequence variation and mutations in the molecular targets of HIV drugs [32]. The HIV data are also available from Wang et al. [33]. When a patient begins antiretroviral therapy, infected HIV may form new mutations. Some mutations may not respond to existing drugs, a feature known as drug resistance or reduced drug sensitivity which means that the drug's role in preventing viral reproduction is diminished, and researchers have estimated that in an untreated HIV-infected person, every possible single-point mutation occurs 10^4 to 10^5 times a day [34]. Drug resistance has become a major obstacle to the treatment of HIV. Therefore, understanding the impact of mutations on drug resistance is an important research topic.

We analyze the susceptibility data for the drug efavirenz (EFV). After excluding some rare mutations, the dataset includes 1472 HIV isolates and 197 locations of mutations. Although the sample size is not too big, the data background traits conform to the distributed effect extremely. The susceptibility of an HIV sample is defined as the fold decrease in susceptibility of a single virus isolate compared with the susceptibility of a wild type control isolate, that is, the virus that has never been challenged by drugs. We focus on predicting the $\log_{10}$-susceptibility, denoted by Y, related to the EFV based on $X_k, k = 1, \ldots, p = 197$, where X_k indicates the presence of a mutation of interest in the k-th viral sequence position. It is noted that the susceptibility data are often stored in different locations and highly nonnormal, even after logarithmic transformation. Therefore, communication-efficient quantile regression could provide a valuable method for analyzing these data. In practice, the location of virus mutation is usually very scarce, so we would get a sparse solution by adding penalties. Note the analysis of the upper quantile of sensitivity is particularly important, which is related to stronger drug resistance. Therefore, we consider two quantile levels: $\tau = 0.5$ and 0.75.

In our analysis, we use the ALasso and SCAD penalized quantile regression model with the proposed approach and

the centralized approach to select those mutations that are interested. We conduct 20 random partitions. For each partition, we randomly select 1200 HIV isolates as the training data and the other 272 as the testing data. We set 1200 training data on 12 machines and store 100 data on each machine ($n = 100, k = 12$). A twelve-fold cross-validation is applied to the training data to select the tuning parameters. But, we use the usual 5-fold cross-validation to select it, while using the centralized method. Table 3 records the average number of nonzero regression coefficients (ave # nonzero) and evaluates the average prediction error using the quantile loss function with $\tau = 0.5$ and $\tau = 0.75$, where the prediction error is defined as $(1/272)\sum_{i=1}^{272}\rho_\tau(y_i - \widehat{y}_i)$ and the numbers in the parentheses are the corresponding standard errors across 20 random partitions. Table 4 lists the frequency of the important variables selected by the various methods for 20 random partitions. Figure 2 plots how the prediction error varies for the proposed method with the rounds of communication, but a horizontal line for centralized method. Moreover, from Figure 2, we can see that as long as a certain number of communications, the prediction error of our method can be very close to even smaller results than the centralized method, which shows our method can match the performance of the centralized method. This phenomenon is consistent with the results of simulated data.

From Table 3, we can see the average number of variables selected is very close to 25 when $\tau = 0.5$; however, when $\tau = 0.75$, it is close to 30. This shows that the dataset has a certain heteroscedasticity. From the perspective of the prediction error, the proposed method yields a smaller prediction error, which further demonstrates the good performance of our method in the HIV data.

As we can see from Table 4, the variables selected using our method are very consistent with those selected using the centralized method. Similar phenomena also appear for the case $\tau = 0.75$ in Table 4. But, the ALasso penalized communication-efficient distributed method performs poor in variables X.172K, X.65R, and X.67G; for other variables, the abovementioned two methods are very perfect. However, the variables selected by the SCAD penalized communication-efficient distributed method are highly consistent with the variables selected by the centralized method. These results show that our method performs well in variable selection in the HIV drug susceptibility dataset.

Table 3: Analysis of HIV drug susceptibility dataset. The numbers in parentheses are the corresponding standard deviations.

Method	$\tau = 0.5$		$\tau = 0.75$	
	Anon	PE	Anon	PE
E-ALasso	24.60 (2.19)	0.16 (0.01)	28.65 (2.43)	0.15 (0.01)
C-ALasso	24.25 (1.83)	0.20 (0.01)	27.25 (2.47)	0.17 (0.01)
E-SCAD	27.05 (2.54)	0.17 (0.01)	30.25 (2.27)	0.17 (0.01)
C-SCAD	25.15 (3.20)	0.20 (0.01)	29.65 (3.23)	0.17 (0.01)

Anon and PE represent the number of average nonzero variables and prediction error, respectively.

Table 4: Frequency table for the HIV data.

Variable	$\tau = 0.5$				Variable	$\tau = 0.75$			
	CAL	EAL	CS	ES		CAL	EAL	ES	CS
X.230L	20	20	20	20	X.230L	20	20	20	20
X.227L	20	20	20	18	X.227L	20	20	20	20
X.225H	20	20	20	20	X.225H	20	20	20	20
X.190A	20	20	20	20	X.190A	20	20	20	20
X.190S	20	20	20	20	X.190S	20	20	20	20
X.188L	20	20	20	20	X.188L	20	20	20	20
X.179D	20	20	20	20	X.179D	20	20	20	20
X.108I	20	20	20	20	X.108I	20	20	20	20
X.103N	20	20	20	20	X.103N	20	20	20	20
X.101H	20	20	20	19	X.102Q	20	20	20	20
X.101Q	20	20	20	20	X.101Q	20	20	20	20
X.101P	20	20	20	20	X.101P	20	20	20	20
X.101E	20	20	20	20	X.101E	20	20	20	20
X.100I	20	20	20	20	X.100I	20	20	20	20
X.90I	20	20	20	20	X.90I	20	19	20	20
X.219N	19	17	18	17	X.181C	19	19	18	20
X.98G.1	19	20	19	20	X.221Y	18	18	14	18
X.98G	19	20	19	20	X.101H	18	20	19	18
X.221Y	17	19	17	20	X.219N	16	13	18	17
X.215Y	16	15	14	12	X.219E	16	17	15	12
X.138A	13	14	13	11	X.103R	15	20	18	17
X.181C	11	20	13	20	X.98G.1	15	16	17	18
X.219E	10	5	10	14	X.98G	15	16	17	18
X.135L	10	4	13	9	X.74V	14	5	17	19
X.172K	7	11	10	12	X.65R	14	0	12	10
					X.172K	11	2	13	6
					X.67G	10	2	11	12
					X.189I	9	13	16	17
					X.102R	8	17	12	12
					X.106I	7	6	9	2

Here, CAL, EAL, CS, and ES represent *C*-ALasso, *E*-ALasso, *C*-SCAD, and *E*-SCAD, respectively.

7. Discussion

We propose a new communication-efficient distributed method to solve sparse penalized quantile regression on massive data, with the parameter estimation obtained by the proposed method having oracle properties. In terms of computation, with a proximal ADMM algorithm under the distributed framework, which makes every parameter of iteration have closed formulas, we also establish the convergence of the algorithm which owns nice efficiency and accuracy. The computational efficiency and accuracy of the proposed method are verified by extensive numerical simulation. The simulation results show that our method usually takes only a few communications to achieve priority performance compared with the centralized approach even having the heteroscedasticity, and the real data example demonstrates that our proposed method has fine performance in parameter estimation and variable selection in high-dimensional quantile regression, especially for multiple and distributed data with heterogeneity and outliers.

It is noted that our method and the PCA algorithm can be modified to the quantile regression with the elastic net

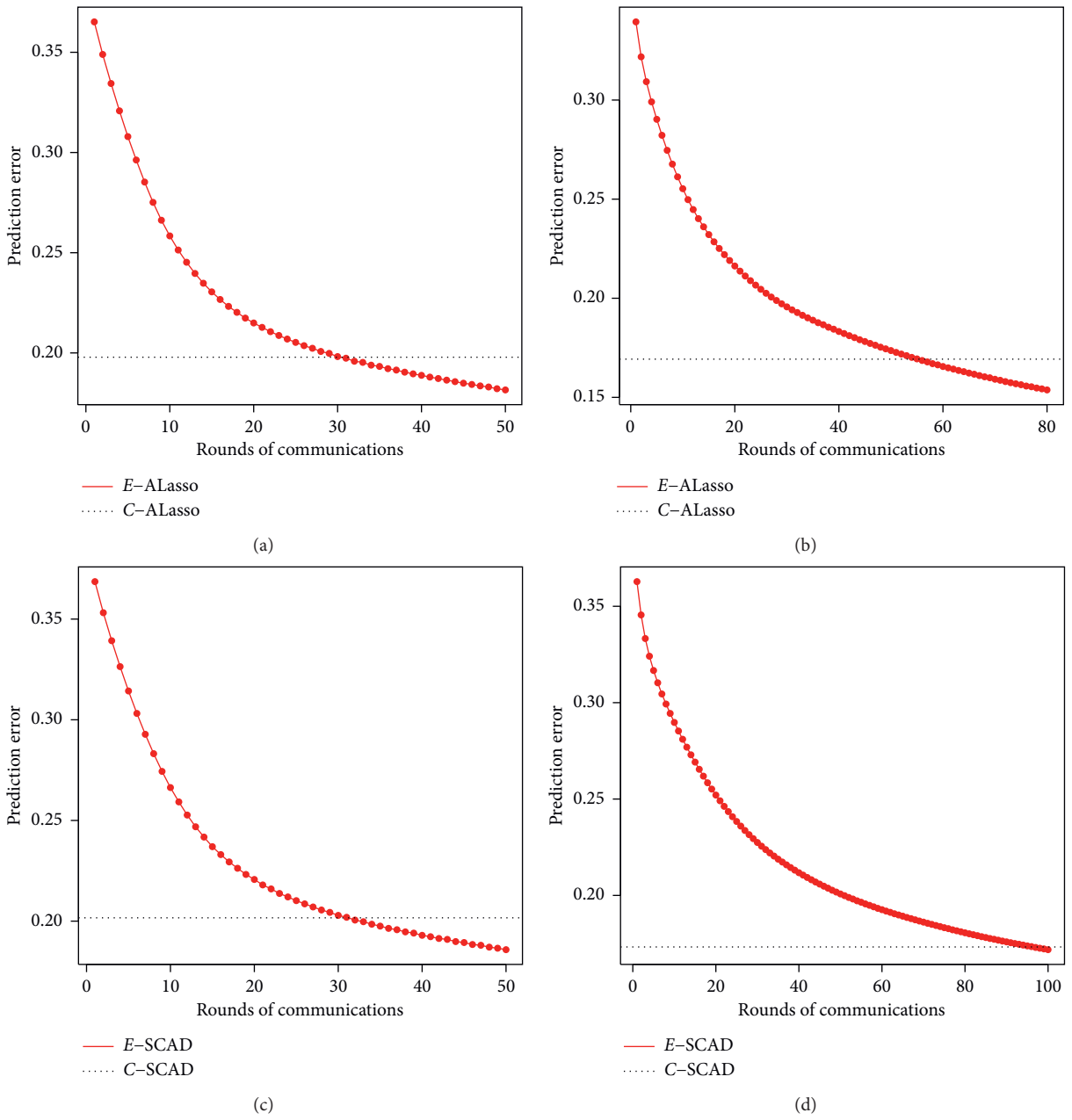

Figure 2: Comparison of two methods for the prediction error with the HIV data. (a) ALasso, $\tau = 0.5$. (b) ALasso, $\tau = 0.75$. (c) Scad, $\tau = 0.5$. (d) Scad, $\tau = 0.75$.

penalty, and the actual implementation of our proposed PCA algorithm can be further improved to make it more efficient in communication. For example, in each iteration of (13), (14), and (11), we have to update all the data on each machine to complete the iteration before it is done. In addition, due to the synchronization in (13), (14), and (11), the total computation speed must be limited by the slowest computing machine. Zhang and Kwok [35] proposed an asynchronous ADMM algorithm to resolve the problem. Moreover, if the distributed data are ultrahigh-dimensional, one necessary choice is to divide the data along the p direction, which may be of further research interest.

Appendix

The penalized communication-efficient distributed quantile regression solves the optimal problem $\min_{\beta\in\mathbb{R}^p} Q(\beta)$, where $Q(\beta) = n\widetilde{\mathscr{L}}(\beta) + n\sum_{j=1}^{p} p_\lambda(|\beta_j|)$.

Lemma A.1. *For model (16) with true parameter β_0, denoting*

$$G_n(\mathbf{u}) = \sum_{i=1}^{n} n\left[\widetilde{\mathscr{L}}\left(\frac{\beta_0 + \mathbf{u}}{\sqrt{n}}\right) - \widetilde{\mathscr{L}}(\beta_0)\right], \tag{A.1}$$

with conditions C1 and C2, we have, for any fixed $\mathbf{u}$,

$$\begin{aligned} G_n(\mathbf{u}) &= \frac{1}{2n}\sum_{i=1}^{n} f_i(\xi_i)\mathbf{u}^T\mathbf{x}_{1i}\mathbf{x}_{1i}^T\mathbf{u} - \frac{1}{\sqrt{n}}\sum_{j=1}^{n}\mathbf{x}_{1i}^T\mathbf{u}\psi_\tau(\varepsilon_{1i}) \\ &+ \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}^T\mathbf{u}\psi_\tau(\overline{\varepsilon}_{1i}) - \frac{1}{\sqrt{k}}\frac{1}{\sqrt{N}}\sum_{j=1}^{k}\sum_{i=1}^{n}\mathbf{x}_{ji}^T\mathbf{u}\psi_\tau\left(\overline{\varepsilon}_{ji}\right) \\ &+ o_p(1), \end{aligned} \tag{A.2}$$

where $\varepsilon_{1i} = y_{1i} - \mathbf{x}_{1i}^T\beta_0$, $\overline{\varepsilon}_{ji} = y_{ji} - \mathbf{x}_{ji}^T\beta^0$ $(1 \le i \le n, 1 \le j \le k)$.

Proof of Lemma A.1. Denote

$$\begin{aligned} W_n &= -\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(\varepsilon_{1i}), \\ \overline{W}_n &= -\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(\overline{\varepsilon}_{1i}), \\ \overline{W}_N &= -\frac{1}{\sqrt{N}}\sum_{j=1}^{k}\sum_{i=1}^{n}\mathbf{x}_{ji}\psi_\tau\left(\overline{\varepsilon}_{ji}\right), \end{aligned} \tag{A.3}$$

respectively.

We can obtain

$$\begin{aligned} G_n(u) &= \sum_{i=1}^{n} n\left[\widetilde{\mathscr{L}}\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right)t - n\widetilde{\mathscr{L}}q(\beta_0)\right] \\ &= \sum_{i=1}^{n}\left[\rho_\tau\left(\frac{\varepsilon_{1i} - \mathbf{x}_{1i}^T\mathbf{u}}{\sqrt{n}}\right) - \rho_\tau(\varepsilon_{1i})\right] \\ &\quad - \langle\sqrt{n}\,\mathbf{u}, \nabla\mathscr{L}_1\left(\beta^0\right) - \nabla\mathscr{L}_N\left(\beta^0\right)\rangle. \end{aligned} \tag{A.4}$$

Using Taylor's theorem and Knight's identity, $\rho_\tau(u-v) - \rho_\tau(u) = -v\psi_\tau(u) + \int_0^v (I(u \le s) - I(u \le 0))\mathrm{d}s$, we get $\sum_{i=1}^{n}[\rho_\tau(\varepsilon_{1i} - x_{1i}^T\mathbf{u}/\sqrt{n}) - \rho_\tau(\varepsilon_{1i})] = G_{1n}(\mathbf{u}) + G_{2n}(\mathbf{u}) + o_p(1)$, where

$$\begin{aligned} G_{1n}(\mathbf{u}) &= \frac{1}{2n}\sum_{i=1}^{n} f_i(\xi_i)\mathbf{u}^T\mathbf{x}_{1i}\mathbf{x}_{1i}^T\mathbf{u}, \\ G_{2n}(\mathbf{u}) &= -\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}^T\mathbf{u}\psi_\tau(\varepsilon_{1i}). \end{aligned} \tag{A.5}$$

Merge with the second item to get $G_n(\mathbf{u}) = G_{1n}(\mathbf{u}) + (W_n - \overline{W}_n + (1/\sqrt{k})\overline{W}_N)^T\mathbf{u} + o_p(1)$, which completes the proof.

Proof of Theorem 1. we adopt the method by Fan and Li [25] or Wu and Liu [16]. To prove Theorem 1, it only needs to be proved that for any given $\delta > 0$, there exists a large constant C such that

$$P\left\{\inf_{\|\mathbf{u}\|=C} Q\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right) > Q(\beta_0)\right\} \ge 1 - \delta. \tag{A.6}$$

(1) For the SCAD penalty, note that

$$\begin{aligned} &Q\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right) - Q(\beta_0) \\ &= G_n(\mathbf{u}) + n\sum_{j=1}^{p}\left[p_{\lambda_n}\left(\left|\frac{\beta_{j0}+u_j}{\sqrt{n}}\right|\right) - p_{\lambda_n}\left(\left|\beta_{j0}\right|\right)\right] \\ &\ge G_n(\mathbf{u}) + n\sum_{j=1}^{s}\left[p_{\lambda_n}\left(\left|\frac{\beta_{j0}+u_j}{\sqrt{n}}\right|\right) - p_{\lambda_n}\left(\left|\beta_{j0}\right|\right)\right], \end{aligned} \tag{A.7}$$

where s is the number of components in β_{10}, and β_{j0} denotes the j-th components of β_{10}.

Note that, for large n,

$$n\sum_{j=1}^{s}\left[p_{\lambda_n}\left(\left|\frac{\beta_{j0}+u_j}{\sqrt{n}}\right|\right) - p_{\lambda_n}\left(\left|\beta_{j0}\right|\right)\right] = 0, \tag{A.8}$$

uniformly in any compact set of $\mathbb{R}^p$ due to $\beta_{j0} > 0$, for $j = 1, 2, \ldots, s$, while the SCAD penalty is flat for coefficient of magnitude larger than $a\lambda_n$ when $\lambda_n \longrightarrow 0$. Since

$$\begin{aligned} G_n(\mathbf{u}) &= \frac{1}{2n}\sum_{i=1}^{n} f_i(\xi_i)\mathbf{u}^T\mathbf{x}_{1i}\mathbf{x}_{1i}^T\mathbf{u} \\ &+ \left(W_n - \overline{W}_n + \frac{1}{\sqrt{k}}\overline{W}_N\right)^T\mathbf{u} + o_p(1), \end{aligned} \tag{A.9}$$

and let $h_n(\mathbf{u}) = (W_n - \overline{W}_n + (1/\sqrt{k})\overline{W}_N)^T\mathbf{u}$ with $\mathrm{E}h_n(\mathbf{u}) = 0$. By condition C2 and Lemma 2 of Wu and Liu [16], we can obtain $(1/2n)\sum_{i=1}^{n} f_i(\xi_i)\mathbf{u}^T\mathbf{x}_{1i}\mathbf{x}_{1i}^T\mathbf{u}$ uniformly convergent to $(1/2)\mathbf{u}^T\sum_1\mathbf{u}$ on any compact subsect of $\mathbb{R}^p$ which implies (A.6) holding, then the conclusion of the theorem is established for the SCAD penalty case.

(2) For the ALasso penalty case, note that

$$\begin{aligned} &Q\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right) - Q(\beta_0) \\ &= \sum_{i=1}^{n}\left[\rho_\tau\left(y_{1i} - \mathbf{x}_{1i}^T\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right)\right) - \rho_\tau\left(y_{1i} - \mathbf{x}_{1i}^T\beta_0\right)\right] - \langle\sqrt{n}\rangle\mathbf{u}, \\ &\overline{W}_n - \overline{W}_N\rangle + n\lambda_n\sum_{j=1}^{p}\left[\widetilde{w}_j\left|\frac{\beta_{j0}+u_j}{\sqrt{n}}\right| - \widetilde{w}_j\left|\beta_{j0}\right|\right], \end{aligned} \tag{A.10}$$

and the first terms are exactly the same as in (A.7). A proof similar to Theorem 3 [16] is that we have $n\lambda_n(\widetilde{w}_j|\beta_{j0} + u_j/\sqrt{n}| - \widetilde{w}_j|\beta_{j0}|)$ which converges to

∞ in probability. So, similar to the proof of the first part, (A.6) holds and completes the proof. □

Lemma A.2. (sparsity) *Consider a sample* $\{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$ *from model (16). For the SCAD penalty, if conditions C1, C2, and C3 are satisfied, or for the ALasso penalty, if conditions C1, C2, and C4 are satisfied, then with probability tending to one, for any given* β_1 *satisfying* $\|\widetilde{\beta} - \beta_0\| = O_p(n^{-1/2})$ *and any constant C, we have*

$$Q\left(\left(\beta_1^T, \mathbf{0}^T\right)^T\right) = \min_{\|\beta_2\| \le Cn^{-1/2}} Q\left(\left(\beta_1^T, \beta_2^T\right)^T\right). \tag{A.11}$$

Proof of Lemma A.2. noting $\beta_1 - \beta_{10} = O_p(n^{-1/2})$, and $0 < \|\beta_2\| \le Cn^{-1/2}$.

(1) For the SCAD penalty, note that

$$\begin{aligned}
&Q\left(\left(\beta_1^T, \mathbf{0}^T\right)^T\right) - Q\left(\left(\beta_1^T, \beta_2^T\right)^T\right) \\
&= \left[Q\left(\left(\beta_1^T, \mathbf{0}^T\right)^T\right) - Q\left(\left(\beta_{10}^T, \mathbf{0}^T\right)^T\right)\right] - \left[Q\left(\left(\beta_1^T, \beta_2^T\right)^T\right) - Q\left(\left(\beta_{10}^T, \mathbf{0}^T\right)^T\right)\right] \\
&= G_n\left(\sqrt{n}\left((\beta_1 - \beta_{10})^T, \mathbf{0}^T\right)^T\right) - G_n\left(\sqrt{n}\left((\beta_1 - \beta_{10})^T, \beta_2^T\right)^T\right) \\
&\quad - n\sum_{j=s+1}^{p} p_{\lambda_n}\left(\left|\beta_j\right|\right), \\
&= \frac{\sqrt{n}}{2}\left((\beta_1 - \beta_{10})^T, \mathbf{0}^T\right) n^{-1}\sum_{i=1}^{n} f_i(\xi_i)\mathbf{x}_{1i}\mathbf{x}_{1i}^T\sqrt{n}\left((\beta_1 - \beta_{10})^T, \mathbf{0}^T\right)^T \\
&\quad + \sqrt{n}\left((\beta_1 - \beta_{10})^T, \mathbf{0}^T\right)\left(W_n - \overline{W}_n + \frac{1}{\sqrt{k}}\overline{W}_N\right) \\
&\quad - \frac{\sqrt{n}}{2}\left((\beta_1 - \beta_{10})^T, \beta_2^T\right) n^{-1}\sum_{i=1}^{n} f_i(\xi_i)\mathbf{x}_{1i}\mathbf{x}_{1i}^T\sqrt{n}\left((\beta_1 - \beta_{10})^T, \beta_2^T\right)^T \\
&\quad - \sqrt{n}\left((\beta_1 - \beta_{10})^T, \beta_2^T\right)\left(W_n - \overline{W}_n + \frac{1}{\sqrt{k}}\overline{W}_N\right) - n\sum_{j=s+1}^{p} p_{\lambda_n}\left(\left|\beta_j\right|\right) \\
&\quad + o(1) + o_p(1).
\end{aligned} \tag{A.12}$$

The conditions $\beta_1 - \beta_{10} = O_p(n^{-1/2})$ and $0 < \|\beta_2\| \le Cn^{-1/2}$ imply that the first and third items mentioned above are $O_p(1)$.

In addition,

$$\begin{aligned}
&\sqrt{n}\left((\beta_1 - \beta_{10})^T, \mathbf{0}^T\right)\left(W_n - \overline{W}_n + \frac{1}{\sqrt{k}}\overline{W}_N\right) \\
&\quad - \sqrt{n}\left((\beta_1 - \beta_{10})^T,\right) \\
&\beta_2^T\left(W_n - \overline{W}_n + \frac{1}{\sqrt{k}}\overline{W}_N\right) \\
&= -\sqrt{n}\left(\mathbf{0}^T, \beta_2^T\right)\left(W_n - \overline{W}_n + \frac{1}{\sqrt{k}}\overline{W}_N\right) \\
&= -\sqrt{n}\left(\mathbf{0}^T, \beta_2^T\right)\widetilde{W}_n,
\end{aligned} \tag{A.13}$$

where $\widetilde{W}_n = W_n - \overline{W}_n + (1/\sqrt{k})\overline{W}_N$.

Under conditions C1 and C2, it follows from the Lindeberg-Feller central limit theorem that

$$\left(W_n^T, \overline{W}_n^T, \overline{W}_N^T\right)^T \xrightarrow{d} \mathcal{N}(0, \Omega), \tag{A.14}$$

where

$$\begin{aligned}
&\Omega = \begin{pmatrix} \Omega_{11} & \Omega_{12} & \Omega_{13} \\ \Omega_{21} & \Omega_{22} & \Omega_{23} \\ \Omega_{31} & \Omega_{32} & \Omega_{33} \end{pmatrix}, \\
&\Omega_{ij} = \Omega_{ji}, \Omega_{11} = \tau(1-\tau)\Sigma_0, \\
&\Omega_{22} = \Omega_{33} = \mathrm{Var}\left(\psi_\tau(\overline{\varepsilon})\right)\Sigma_0, \\
&\quad \mathrm{Cov}(W_n, \overline{W}_n) \longrightarrow \mathrm{Cov}\left(\psi_\tau(\overline{\varepsilon}), \psi_\tau(\varepsilon)\right)\Sigma_0 = \Omega_{21}, \\
&\quad \mathrm{Cov}(\overline{W}_N, W_n) \longrightarrow \frac{1}{\sqrt{k}}\mathrm{Cov}\left(\psi_\tau(\overline{\varepsilon}), \psi_\tau(\varepsilon)\right)\Sigma_0 \\
&\quad \mathrm{Cov}(\overline{W}_N, \overline{W}_n) \longrightarrow \frac{1}{\sqrt{k}}\mathrm{Var}\left(\psi_\tau(\overline{\varepsilon})\right)\Sigma_0 = \Omega_{32}.
\end{aligned} \tag{A.15}$$

Hence, we have

$$W_n - \overline{W}_n + \frac{1}{\sqrt{k}}\overline{W}_N = \left(I_{p\times p}, -I_{p\times p}, \frac{1}{\sqrt{k}}I_{p\times p}\right)\left(W_n^T, \overline{W}_n^T, \overline{W}_N^T\right)^T \xrightarrow{d} \mathcal{N}\left(0, \alpha_k \Omega \alpha_k^T\right), \tag{A.16}$$

where $\alpha_k = (I_{p\times p}, -I_{p\times p}, (1/\sqrt{k})I_{p\times p})$, and

$$\widetilde{W}_n \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, \frac{D\Sigma_0}{k}\right), \sqrt{n}\left(\mathbf{0}^T, \beta_2^T\right)\widetilde{W}_n = \sqrt{\frac{nD}{k}}\sqrt{\beta_2^T \Sigma_{02} \beta_2}\left(1 + o_p(1)\right). \tag{A.17}$$

Note that

$$\begin{aligned} & n\sum_{j=s+1}^{p} p_{\lambda_n}\left(\left|\beta_j\right|\right) \\ & \geq n\lambda_n\left(\liminf_{\lambda\longrightarrow 0}\liminf_{\theta\longrightarrow 0+}\frac{p_\lambda'(\theta)}{\lambda}\right)\left(\sum_{j=s+1}^{p}\left|\beta_j\right|\right)(1 + o(1)) \\ & = n\lambda_n\left(\sum_{j=s+1}^{p}\left|\beta_j\right|\right)(1 + o(1)), \end{aligned} \tag{A.18}$$

where the last step follows based on the fact that $\liminf_{\lambda\longrightarrow 0}\liminf_{\theta\longrightarrow 0+}(p_\lambda'(\theta)/\lambda) = 1$.

Then, $\sqrt{n}\lambda_n \longrightarrow \infty$ implies that $n\lambda_n = \sqrt{n}(\sqrt{n}\,\lambda_n)$ is of higher order than $\sqrt{n}$. So, we can obtain

$$Q\left(\left(\beta_1^T, \mathbf{0}^T\right)^T\right) - Q\left(\left(\beta_1^T, \beta_2^T\right)^T\right) < 0, \tag{A.19}$$

for large n. This completes the proof of the SCAD penalty case.

(2) For the ALasso penalty, note that

$$\begin{aligned} Q\left(\left(\beta_1^T, \mathbf{0}^T\right)^T\right) - Q\left(\left(\beta_1^T, \beta_2^T\right)^T\right) &= \left[Q\left(\left(\beta_1^T, \mathbf{0}^T\right)^T\right) - Q\left(\left(\beta_{10}^T, \mathbf{0}^T\right)^T\right)\right] - \left[Q\left(\left(\beta_1^T, \beta_2^T\right)^T\right) - Q\left(\left(\beta_{10}^T, \mathbf{0}^T\right)^T\right)\right] \\ &= G_n\left(\sqrt{n}\left(\left(\beta_1 - \beta_{10}\right)^T, \mathbf{0}^T\right)^T\right) - G_n\left(\sqrt{n}\left(\left(\beta_1 - \beta_{10}\right)^T, \beta_2^T\right)^T\right) \\ &\quad - n\lambda_n\sum_{j=s+1}^{p}\widetilde{w}_j\left|\beta_j\right|. \end{aligned} \tag{A.20}$$

Also, note that the fore terms are exactly the same as in (A.12) and here it can be bounded similarly. by Theorem 3 in Wu and Liu [16], the proof is completed.

Proof of Theorem 2. part (a) can be established by Lemma A.2. We only need to prove part (b).

(1) For the SCAD penalty, we prove that there exists a root-n consistent minimizer $\widetilde{\beta}_1$ of $Q((\beta_1^T, \mathbf{0}^T)^T)$.

We can deduce from the proof of Theorem 1 that

$$\sqrt{n}\left(\widetilde{\beta}_1 - \beta_{10}\right), \tag{A.21}$$

minimizes $G_n((\theta^T, \mathbf{0}^T)^T) + n\sum_{j=1}^{s} p_{\lambda_n}(|\beta_{j0} + \theta_j/\sqrt{n}|)$, where $\theta = (\theta_1, \theta_2, \ldots, \theta_s)^T \in \mathbb{R}^s$.

From the proof of Lemma A.2, Theorem 1, and Lemma 2 [16], we have

$$\begin{aligned} G_n\left(\left(\boldsymbol{\theta}^T, \mathbf{0}^T\right)^T\right) &= \frac{1}{2n}\theta^T\sum_{i=1}^{n} f_i(\xi_i)\mathbf{x}_{i1}\mathbf{x}_{i1}^T\boldsymbol{\theta} \\ &\quad + \left(W_{n,11} - \overline{W}_{n,11} + \frac{1}{\sqrt{k}}\overline{W}_{N,11}\right)^T\boldsymbol{\theta} \\ &\quad + o_p(1), \end{aligned} \tag{A.22}$$

uniformly in any compact subset of $\mathbb{R}^s$. Here, $W_{n,11} = -(1/\sqrt{n})\sum_{i=1}^{n}\mathbf{x}_{i1}\psi_\tau(\varepsilon_i)$, $\overline{W}_{n,11} = -(1/\sqrt{n})\sum_{i=1}^{n}\mathbf{x}_{i1}\psi_\tau(\overline{\varepsilon}_i)$ and $\overline{W}_{N,11} = -(1/\sqrt{N})\sum_{i=1}^{N}\mathbf{x}_{i1}\psi_\tau(\overline{\varepsilon}_i)$.

As a result of (A.8), we have

$$\begin{aligned} & G_n\left(\left(\boldsymbol{\theta}^T, \mathbf{0}^T\right)^T\right) + n\sum_{j=1}^{s} p_{\lambda_n}\left(\left|\frac{\beta_{j0}+\theta_j}{\sqrt{n}}\right|\right) \\ &= \frac{1}{2}\boldsymbol{\theta}^T n^{-1}\sum_{i=1}^{n} f_i(\xi_i)\mathbf{x}_{i1}\mathbf{x}_{i1}^T\theta + \left(W_{n,11} - \overline{W}_{n,11} + \frac{1}{\sqrt{k}}\overline{W}_{N,11}\right)^T\boldsymbol{\theta} \\ &\quad + n\sum_{j=1}^{s} p_{\lambda_n}\left(\left|\beta_{j0}\right|\right) + o_p(1) \\ &= \frac{1}{2}\boldsymbol{\theta}^T n^{-1}\sum_{i=1}^{n} f(\xi_i)\mathbf{x}_{i1}\mathbf{x}_{i1}^T\theta + \widetilde{W}_{n,11}^T\boldsymbol{\theta}_i + n\sum_{j=1}^{s} p_{\lambda_n}\left(\left|\beta_{j0}\right|\right) + o_p(1). \end{aligned} \tag{A.23}$$

Notice that the term $n\sum_{j=1}^{s} p_{\lambda_n}(|\beta_{j0}|)$ does not depend on θ; by derivation of the upper function, we have

$$\widehat{\boldsymbol{\theta}} = \left(n^{-1}\sum_{i=1}^{n} f_i(\xi_i)\mathbf{x}_{i1}\mathbf{x}_{i1}^T\right)^{-1}\left(W_{n,11} - \overline{W}_{n,11} + \frac{1}{\sqrt{k}}\overline{W}_{N,11}\right). \tag{A.24}$$

Similar to Lemma A.2, we can get

$$W_{n,11} - \overline{W}_{n,11} + \frac{1}{\sqrt{k}}\overline{W}_{N,11} \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, \frac{D\Sigma_{01}}{k}\right). \tag{A.25}$$

Applying Slutsky's theorem [36–38], we have

$$\sqrt{N}\left(\widetilde{\beta}_1 - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(\mathbf{0}, D\Sigma_{11}^{-1}\Sigma_{01}\Sigma_{11}^{-1}\right). \tag{A.26}$$

(2) For the ALasso penalty, note that

$$\begin{aligned} & Q\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right) - Q(\beta_0) \\ &= \sum_{i=1}^{n}\left[\rho_\tau\left(y_{1i} - x_{1i}^T\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right)\right) - \rho_\tau\left(y_{1i} - \mathbf{x}_{1i}^T\beta_0\right)\right] \\ &\quad - \langle\sqrt{n}\,\mathbf{u}, \overline{W}_n - \overline{W}_N\rangle \\ &\quad + n\lambda_n\sum_{j=1}^{p}\left(\widetilde{\omega}_j\left|\frac{\beta_{j0}+u_j}{\sqrt{n}}\right| - \widetilde{\omega}_j\left|\beta_{j0}\right|\right). \end{aligned} \tag{A.27}$$

Similarly, we have $n\lambda_n(\widetilde{\omega}_j|\beta_{j0} + u_j/\sqrt{n}| - \widetilde{\omega}_j|\beta_{j0}|)$ which converges to ∞ in probability when $u_j \neq 0$ or it converges to 0, otherwise, for large n.

As a result of Lemma A.1, we obtain

$$Q_1\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right) - Q_1(\beta_0) \xrightarrow{d} V(\mathbf{u}) = \begin{cases} \frac{1}{2}\mathbf{u}_1^T\Sigma_{11}\mathbf{u}_1 + \widetilde{W}_{n,11}^T\mathbf{u}_1, & \text{when } u_j = 0 \text{ for } j \geq s+1, \\ \infty, & \text{otherwise,} \end{cases} \tag{A.28}$$

where $\mathbf{u}_1 = (u_1, u_2, \ldots, u_s)^T$ and $\widetilde{W}_{n,11} = W_{n,11} - \overline{W}_{n,11} + (1/\sqrt{k})\overline{W}_{N,11}$.

The epi-convergence results of Geyer [39] imply that

$$\arg\min Q\left(\frac{\beta_0+\mathbf{u}}{\sqrt{n}}\right) = \sqrt{n}\left(\widetilde{\beta} - t\beta_0\right) \xrightarrow{d} \arg\min V(\mathbf{u}). \tag{A.29}$$

Similarly, this proves the asymptotic normality part and completes the proof.

Proof of Theorem 3. in Algorithm 2, denote $\mathbf{g}^0 = \nabla\mathscr{L}_N(\beta^0) - \nabla\mathscr{L}_1(\beta^0)$, $\mathbf{g}^t = \nabla\mathscr{L}_N(\beta^t) - \nabla\mathscr{L}_1(\beta^t)$, $H(\beta) = \lambda\|\mathbf{w}\circ\beta\|_1 + \langle g^0, \beta\rangle$, and $G(z) = \mathbb{Q}_\tau(z)$.

Replace $f(\beta)$ and $g(z)$ in Theorem 1 of Gu et al. [27] with $H(\beta)$ and $G(z)$, respectively.

Similar to the derivation of Theorem 1 in Gu et al. [27], formula (A.6) in the proof of Theorem 1 in Gu et al. [27] becomes

$$\begin{aligned} 0 \leq\ & (\gamma\sigma)^{-1}\left(\left\|\mathbf{d}_{\boldsymbol{\theta}}^t\right\|_2^2 - \left\|\mathbf{d}_{\boldsymbol{\theta}}^{t+1}\right\|_2^2\right) + \sigma(\gamma-2)\left\|\mathbf{r}^{t+1}\right\|_2^2 \\ &+ \sigma\left(\left\|\mathbf{d}_z^t\right\|_2^2 - \left\|\mathbf{d}_z^{t+1}\right\|_2^2 - \left\|z^{t+1} - z^t\right\|_2^2\right) \\ &+ \left(\left\|\mathbf{d}_\beta^t\right\|_S^2 - \left\|\mathbf{d}_\beta^{t+1}\right\|_{\mathbf{S}}^2 - \left\|\beta^{t+1} - \beta^t\right\|_{\mathbf{S}}^2\right) \\ &+ 2\sigma(1-\gamma)\langle z^{t+1} - z^t, \mathbf{r}^t\rangle + 2\langle\mathbf{g}^0 - \mathbf{g}^t, \mathbf{d}_\beta^{t+1}\rangle. \end{aligned} \tag{A.30}$$

Note that according to the law of strong numbers, $\mathbf{g}^0$ and $\mathbf{g}^t$ converge almost everywhere to zero when n is sufficiently large; therefore, if the last term of the above formula is removed, the above inequality is still true.

Hence, the proof of Theorem 3 is finished.

Data Availability

The NNRTI-EFV.csv data used to support the findings of this study have been deposited in [33] (DOI: 10.1111/rssb.12258) and also included within the article; since it is a public open dataset, itcould be downloaded freely. Requests for the data or the algorithmic routine, 6 months after publication of this article, will be considered by the corresponding author. The NNRTI-EFV.csv data may be released upon application to the HIV Drug Resistance Database, who can be contacted via the website http://hivdb.stanford.edu or http://blogs.gwu.edu/judywang/software/QMET/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Defense Foundation Research Project JCKY2018207C121.

References

[1] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis

and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[2] R. Kannan, S. Vempala, and D. Woodruff, "Principal component analysis and higher correlations for distributed data," in *Proceedings of the Conference on Learning Theory*, Berlin, Germany, 2014.

[3] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate Newton-type method," in *Proceedings of the International Conference on Machine Learning*, Berlin, Germany, 2014.

[4] J. D. Lee, Y. Sun, Q. Liu, and J. E. Taylor, "Communication-efficient sparse regression: a one-shot approach," *Machine Learning*, vol. 34, 2015.

[5] R. Mcdonald, M. Mohri, N. Silberman, D. Walker, and G. S. Mann, "Efficient large-scale distributed training of conditional maximum entropy models," *Advances in Neural Information Processing Systems*, vol. 3, 2009.

[6] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3321–3363, 2013.

[7] Y. Zhang and X. Lin, "Disco: distributed optimization for self-concordant empirical loss," in *Proceedings of the International Conference on Machine Learning*, New York, NY, USA, 2015.

[8] G. Xu, Z. Shang, and G. Cheng, "Optimal tuning for divide-and-conquer kernel ridge regression with massive data," in *Proceedings of the International Conference on Machine Learning*, New York, NY, USA, 2018.

[9] J. Wang, M. Kolar, N. Srebro, and T. Zhang, "Efficient distributed learning with sparsity," in *Proceedings of the 34th International Conference on Machine Learning*, London, UK, 2017.

[10] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *Journal of the American Statistical Association*, vol. 114, no. 526, pp. 668–681, 2019.

[11] R. Koenker and G. Bassett, "Regression quantiles," *Econometrica*, vol. 46, no. 1, pp. 33–50, 1978.

[12] M. Buchinsky, "Changes in the U.S. Wage structure 1963-1987: application of quantile regression," *Econometrica*, vol. 62, no. 2, pp. 405–458, 1994.

[13] R. Koenker, *Quantile Regression*, Cambridge University Press, Cambridge, UK, 2005.

[14] Y. Li and J. Zhu, "*L*1-norm quantile regression," *Journal of Computational and Graphical Statistics*, vol. 17, no. 1, pp. 163–185, 2008.

[15] H. Zou and M. Yuan, "Composite quantile regression and the oracle model selection theory," *The Annals of Statistics*, vol. 36, no. 3, pp. 1108–1126, 2008.

[16] Y. Wu and Y. Liu, "Variable selection in quantile regression," *Statistica Sinica*, vol. 19, no. 2, pp. 801–817, 2009.

[17] A. Belloni and V. Chernozhukov, "ℓ1-penalized quantile regression in high-dimensional sparse models," *The Annals of Statistics*, vol. 39, no. 1, pp. 82–130, 2011.

[18] L. Wang, Y. Wu, and R. Li, "Quantile regression for analyzing heterogeneity in ultra-high dimension," *Journal of the American Statistical Association*, vol. 107, no. 497, pp. 214–222, 2012.

[19] B. Peng and L. Wang, "An iterative coordinate descent algorithm for high-dimensional nonconvex penalized quantile regression," *Journal of Computational and Graphical Statistics*, vol. 24, no. 3, pp. 676–694, 2015.

[20] C. Yi and J. Huang, "Semismooth Newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression," *Journal of Computational and Graphical Statistics*, vol. 26, no. 3, pp. 547–557, 2017.

[21] W. Xiong and M. Tian, "A new model selection procedure based on dynamic quantile regression," *Journal of Applied Statistics*, vol. 41, no. 10, pp. 2240–2256, 2014.

[22] L. Yu and N. Lin, "Admm for penalized quantile regression in big data," *International Statistical Review*, vol. 85, no. 3, pp. 494–518, 2017.

[23] L. Yu, N. Lin, and L. Wang, "A parallel algorithm for large-scale nonconvex penalized quantile regression," *Journal of Computational and Graphical Statistics*, vol. 26, no. 4, pp. 935–939, 2017.

[24] X. Chen, W. D. Liu, X. J. Mao, and Z. Y. Yang, "Distributed high-dimensional regression under a quantile loss function," *Journal of Machine Learning Research*, vol. 21, no. 182, 2020.

[25] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[26] H. Zou, "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[27] Y. Gu, J. Fan, L. Kong, S. Ma, and H. Zou, "Admm for high-dimensional sparse penalized quantile regression," *Technometrics*, vol. 60, no. 3, pp. 319–331, 2018.

[28] J. Fan and J. Lv, "Nonconcave penalized likelihood with np-dimensionality," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5467–5484, 2011.

[29] H. Zou and R. Li, "One-step sparse estimates in nonconcave penalized likelihood models," *The Annals of Statistics*, vol. 36, no. 4, pp. 1509–1533, 2008.

[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[31] Q. Xu, C. Cai, C. Jiang, F. Sun, and X. Huang, "Block average quantile regression for massive dataset," *Statistical Papers*, vol. 34, pp. 1–25, 2017.

[32] S.-Y. Rhee, M. J. Gonzales, R. Kantor, B. J. Betts, J. Ravela, and R. W. Shafer, "Human immunodeficiency virus reverse transcriptase and protease sequence database," *Nucleic Acids Research*, vol. 31, no. 1, pp. 298–303, 2003.

[33] H. J. Wang, I. W. McKeague, and M. Qian, "Testing for marginal linear effects in quantile regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 80, no. 2, pp. 433–452, 2018.

[34] J. Coffin, "Hiv population dynamics in vivo: implications for genetic variation, pathogenesis, and therapy," *Science*, vol. 267, no. 5197, pp. 483–489, 1995.

[35] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *Proceedings of the International Conference on Machine Learning*, Berlin, Germany, 2014.

[36] D. Pollard, "Asymptotics for least absolute deviation regression estimators," *Econometric Theory*, vol. 7, no. 2, pp. 186–199, 1991.

[37] N. L. Hjort and D. Pollard, "Asymptotics for minimisers of convex processes," *Statistics Theory*, vol. 7, 1993.

[38] K. Knight, "Limiting distributions for $L\sb 1$ regression estimators under general conditions," *The Annals of Statistics*, vol. 26, no. 2, pp. 755–770, 1998.

[39] C. J. Geyer, "On the asymptotics of constrained *m*-estimation," *The Annals of Statistics*, vol. 22, no. 4, 1994.