IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 10, 2021, accepted April 22, 2021, date of publication April 26, 2021, date of current version May 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075686

Efficient Distributed Learning for Large-Scale Expectile Regression With Sparsity

YINGLI PAN AND ZHAN LIU
Hubei Key Laboratory of Applied Mathematics, School of Mathematics and Statistics, Hubei University, Wuhan 430062, China
Corresponding authors: Yingli Pan (panyingli220@163.com) and Zhan Liu (eleen_20040109@163.com)

This work was supported in part by the National Science Foundation of China under Grant 11901175.

ABSTRACT High-dimensional datasets often display heterogeneity due to heteroskedasticity or other forms of non-location-scale covariance effects. When the size of datasets becomes very large, it may be infeasible to store all of the high-dimensional datasets on one machine, or at least to keep the datasets in memory. In this paper, we consider penalized expectile regression using smoothly clipped absolute deviation (SCAD) and adaptive LASSO penalties, which can effectively detect the heteroskedasticity of high-dimensional data. We propose a communication-efficient approach for distributed sparsity learning, where observations are randomly partitioned across machines. By selecting the appropriate tuning parameters, we show that the proposed estimators display oracle properties. Extensive numerical experiments on both synthetic and real data validate the theoretical results and demonstrate the superior performance of our proposed method.

INDEX TERMS Expectile regression, SCAD, adaptive LASSO, communication-efficient, distributed learning.

I. INTRODUCTION

The explosive growth in the size of modern datasets has stimulated interest in distributed statistical learning [2], [4], [29]. A problem arises, for example, when the dataset is too large to store on one machine and must be stored across multiple machines. The main bottleneck in a distributed setup is communication between machines, so the primary goal of optimal design is to minimize communication complexity.

In distributed statistical learning, the most common distributed optimization method is averaging estimators locally formed by different machines [16], [29], [9]. The divide-and-conquer procedure also has applications in statistical inference [3], [15]. Both of the above approaches attempt to find a good balance between computation and communication. However, their communication complexity boundary has a poor dependence on the number of conditions [21]. To improve these methods, Jordan *et al.* [11] developed a communication-efficiency surrogate likelihood (CSL) framework for estimation and inference in regular parametric models, high-dimensional penalty regression, and Bayesian statistics. A similar method for penalized regression also appears in Wang *et al.* [26]. However, there is a problem in the optimization based on the CSL function that only the first machine solves the optimization problem, while the others only calculate the gradient; as a result, the first machine is working hard, and other machines are idling. Therefore, it is imperative to explore new distributed optimization methods.

Large-scale data are not only large in quantity but also high in dimension. High-dimensional data are often collected a wide range of research fields such as genomics, tomography, economics and finance. Heterogeneity is common in high-dimensional data due to heteroskedasticity or other forms of non-location-scale covariance. In regression applications with heterogeneous data, we often see that targeting the mean function alone is not enough to obtain the complete relationship between response variables and predictors. In this case, quantile regression based on the asymmetric L_1-norm [12] is a more appropriate tool because it allows the study of the quantile structure of the conditional distribution. Li and Zhu [13] and Wu and Liu [28] investigated the regularized quantile regression with fixed parameter dimension. While quantile regression is intuitively appealing, Newey and Powell [17] highlighted its three disadvantages: nondifferentiability, ineffectiveness of Gaussian-like error distributions, and difficulty in calculating the covariance matrix. They proposed expectile regression based on the asymmetric L_2-norm as an alternative to analyze the complete conditional distribution of the response. Expectile regression is a generalization of general mean regression and is effective

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

when typical assumptions (including homogeneity of errors) are met. It is also closely related to quantile regression, which is robust to outliers.

Since the landmark article by Newey and Powell [17], expectile regression has attracted attention in many fields [1], [10], [22]. Sobotka *et al.* [23] studied the relationship between female education and fertility in Botswana using semiparametric expectile regression. Waltrup *et al.* [24] showed that expectile regression involves fewer crossings than quantile regression and is more robust to heavy-tailed distributions. Liao *et al.* [14] discussed the pros and cons of penalized quantile and expectile regression and conducted in-depth simulation studies to compare the finite sample performance of the two methods. Pan *et al.* [18] developed a communication-efficient distributed optimization method to solve the expectile regression problem with covariates missing at random. Although expectile regression has applications in various fields, few people, to the best of our knowledge, have used the penalized version of expectile regression in a distributed environment. This paper attempts to study a distributed optimization approach for large-scale expectile regression with SCAD [6] and adaptive LASSO [31] penalties.

To address the problem of establishing the theoretical properties of parameter estimation under the distributed setting, Jordan *et al.* [11], Pan [18] and Pan *et al.* [19] indicated that CSL function can be regarded as a valid proxy of global loss function. Inspired by CSL function, we propose a more generalized proxy loss function called gradient-enhanced loss (GEL) function, which includes the CSL function as a special case. In addition, GEL function effectively avoids the disadvantages of CSL function that making the CPU work very hard while the other machines are idling. We propose a distributed estimator based on the GEL function with SCAD and adaptive LASSO penalties and then apply the ideas of Zhao and Zhang [30] and Jordan *et al.* [11] to prove the oracle properties of penalized expectile regression with independent identically distributed random error.

Another challenge stems from optimizing the penalized GEL function. The alternating direction method of multipliers (ADMM) algorithm has many successful applications in high-dimensional statistics. Boyd *et al.* [2] argued that ADMM is well suited to large-scale problems in distributed convex optimization and statistics. As an important variant of ADMM, the proximal ADMM has also been studied in various fields [5], [7]. In this paper, we propose an augmented proximal ADMM algorithm to solve the large-scale expectile regression with SCAD and adaptive LASSO penalties. Computationally, to fully exploit the computing power of the machine and accelerate convergence, all machines can optimize their corresponding GEL functions in parallel, and the results are then aggregated by the CPU. Visually, the average step requires less computation, but it helps to improve the accuracy of estimates. In terms of communication, the proposed algorithm can be proven to match the centralized method during few rounds of communication. Simulation and empirical studies show that the estimation errors (or the prediction errors) and variable selection results obtained by the proposed approach are compared with those obtained by the centralized method, and are better than the results of Pan [18] which is based on the CSL function. In addition, they also show that our proposed method can not only effectively solve the problem of data heterogeneity, but also reduce the cost of data storage and transmission.

The remainder of the paper is organized as follows. Large-scale expectile regression with SCAD and adaptive LASSO penalties are introduced in Section II. The oracle properties of the SCAD and adaptive LASSO penalized expectile estimators are presented in Section III. The augmented proximal ADMM algorithm for handling the distributed optimization problem is proposed in Section IV. Section V and Section VI present numerical results on simulations and real data, respectively. The conclusion and prospects for future work are summarized in Section VII.

II. DISTRIBUTED ESTIMATION IN LARGE-SCALE PENALIZED EXPECTILE REGRESSION

Suppose that we have a random sample $\{x_i, y_i\}_{i=1}^N$ from the following model:

$$y_i = x_i^{\mathrm{T}} \beta_0 (\tau) + \epsilon_i (\tau), \quad i = 1, 2, \cdots, N, \tag{2.1}$$

where x_i is a p-dimensional vector of covariates, $\beta_0(\tau)$ is a p-dimensional vector of parameters, and the random error $\epsilon_i(\tau)$ satisfies $\mathrm{P}(\epsilon_i(\tau) \leq 0|x_i) = \tau$ for some specified $\tau \in (0, 1)$. We drop τ from the parameters and error term for notational simplicity. We assume that the true parameter value β_0 is sparse. In other words, if we represent the support set as $\mathbb{B} = \{k : \beta_{k0} \neq 0\}$ and let $q = |\mathbb{B}|$ be the cardinality of the set $\mathbb{B}$, the sparsity assumption implies that $q \ll p$, where β_{k0} is the k-th component of β_0.

Modern large-scale datasets, where both N and p are very large, create challenges for classical approaches. In this paper, we address the challenge that observations cannot be stored in a single machine but are distributed across m machines. For simplicity, we assume that $N = nm$ and that the j-th machine has access to observations $\{x_{ji}, y_{ji}\}_{i=1}^n$. All of our results can be easily generalized for a generalized N. The regression expectile estimator proposed by Newey and Powell [17] is defined as the vector that minimizes the following global loss function

$$Q_N(\beta) = \frac{1}{nm} \sum_{j=1}^{m} \sum_{i=1}^{n} \rho_\tau \left(y_{ji} - x_{ji}^{\mathrm{T}} \beta \right), \tag{2.2}$$

where $\rho_\tau(\cdot)$ is a convex loss function of the form

$$\rho_\tau(u) = \tau u^2 I(u > 0) + (1 - \tau) u^2 I(u \leq 0)$$

with $I(\cdot)$ being the indicator function. In a large-scale dataset environment, the direct minimization of $Q_N(\beta)$ is costly. This means that we need to construct a proxy function for $Q_N(\beta)$ and change the optimization objective $Q_N(\beta)$ to the proxy function.

In a distributed environment, only p-dimensional gradient vector $\nabla Q_N(\beta)$ can communicate easily, so the linear function $Q_N^{(1)}(\beta) = Q_N(\overline{\beta}) + \langle \nabla Q_N(\overline{\beta}), \beta - \overline{\beta} \rangle$, which is the first-order Taylor expansion of $Q_N^{(1)}(\beta)$ around $\overline{\beta}$, where $\overline{\beta}$ is any initial estimator. The global loss function to be minimized can be written as

$$Q_N(\beta) = Q_N^{(1)}(\beta) + R(\beta), \tag{2.3}$$

where $R(\beta) = Q_N(\beta) - Q_N^{(1)}(\beta)$. Since the linear function $Q_N^{(1)}(\beta)$ can easily be communicated to each machine whereas $R(\beta)$ cannot, this naturally prompts us to replace $R(\beta)$ with its subsampled version at machine j:

$$R_j(\beta) = Q_j(\beta) - \left[Q_j(\overline{\beta}) + \langle \nabla Q_j(\overline{\beta}), \beta - \overline{\beta} \rangle\right], \tag{2.4}$$

where $Q_j(\beta)$ is the local loss function based on the datasets at machine j, defined by

$$Q_j(\beta) = \frac{1}{n}\sum_{i=1}^{n} \rho_\tau\left(y_{ji} - x_{ji}^{\mathrm{T}}\beta\right), \quad j = 1, 2, \cdots, m. \tag{2.5}$$

Due to this substitution, the goal of optimization becomes $Q_N^{(1)}(\beta) + R_j(\beta)$, which equals

$$Q(\beta) := Q_j(\beta) + \langle \nabla Q_N(\overline{\beta}) - \nabla Q_j(\overline{\beta}), \beta \rangle \tag{2.6}$$

up to an additive constant. We call the surrogate loss function $Q(\beta)$ the GEL function, where the function $R(\beta)$ based on the global data is replaced by the function $R_j(\beta)$ based on the local one. Note that Jordan *et al.* [11] proposed a communication-efficient surrogate likelihood method using the GEL function

$$Q_1(\beta) + \langle \nabla Q_N(\overline{\beta}) - \nabla Q_1(\overline{\beta}), \beta \rangle \tag{2.7}$$

on the first machine. Obviously, our proposed GEL function is more general, and it includes the CSL function as a special case.

Because of the high parameter dimension, the GEL function (2.6) cannot be used for the inference of β. To avoid overfitting and improve generalizability, we add penalty term to the GEL function to encourage sparsity in the coefficient estimators and propose the following penalized GEL function

$$\widetilde{Q}(\beta) = Q_j(\beta) + \langle \nabla Q_N(\overline{\beta}) - \nabla Q_j(\overline{\beta}), \beta \rangle + \sum_{k=1}^{p} p_\lambda(|\beta_k|), \tag{2.8}$$

where $p_\lambda(\cdot)$ is a penalty function with tuning parameter λ.

According to the different penalty function, the estimation coefficient methods are divided into elastic net family penalties, including LASSO estimate, Ridge estimate and ENet estimate; Non-convex Penalty estimation, including SCAD, MCP (Minimax Concave Penalty), etc. The Lasso is a representation of the convex penalty function and is easy to calculate, and it can compress the coefficient of the independent variable to zero, so it has good robustness and is especially practical. SCAD is a representation of a nonconvex penalty function, it satisfies asymptotic unbias, but the calculation is complicated. In our paper, on the one hand, we consider a nonconvex SCAD. On the other hand, we consider a convex penalty, adaptive LASSO penalty, which can be seen as a generalization of the LASSO penalty. The main purpose of adaptive LASSO penalty is to use adaptive weights to punish the coefficients of different covariates to different degrees. Based on the two kinds penalties, we construct the SCAD-penalized GEL function and adaptive LASSO penalized GEL function in the rest of this section; we then establish the corresponding theoretical properties respectively under different regular conditions in Section III. To solve the adaptive LASSO penalty expectile regression and SCAD penalty expectile regression, we make a local linear approximation for the SCAD penalty, and we unify the SCAD-penalized GEL function and adaptive LASSO penalized GEL function, and then propose an augmented proximal ADMM algorithm in Section IV.

We first consider the SCAD penalty and define the penalty function as

$$p_\lambda(|u|) = \lambda |u| I(|u| \le \lambda) + \frac{a\lambda|u| - \left(|u|^2 + \lambda^2\right)/2}{a-1} \times I(\lambda < |u| \le a\lambda) + \frac{(a+1)\lambda^2}{2} I(|u| > a\lambda)$$

for some $a > 2$ and $\lambda > 0$. A typical choice is $a = 3.7$ as suggested by Fan and Li [6]. Combining (2.8) and the SCAD penalty, the GEL function with the SCAD penalty can be written as

$$\widetilde{Q}_{SCAD}(\beta) = Q_j(\beta) + \langle \nabla Q_N(\overline{\beta}) - \nabla Q_j(\overline{\beta}), \beta \rangle + \sum_{k=1}^{p} p_\lambda(|\beta_k|). \tag{2.9}$$

By (2.9), we propose the first estimator of β, denoted by $\widehat{\beta}^{(SCAD)}$, which is defined to solve the following SCAD penalized expectile regression optimization problem

$$\widehat{\beta}^{(SCAD)} = \arg\min_{\beta \in \mathbb{R}^p} \widetilde{Q}_{SCAD}(\beta). \tag{2.10}$$

Then we consider the adaptive LASSO penalty. Based on (2.8), the adaptive LASSO penalized expectile regression minimizes

$$\widetilde{Q}_{AL}(\beta) = Q_j(\beta) + \langle \nabla Q_N(\overline{\beta}) - \nabla Q_j(\overline{\beta}), \beta \rangle + \lambda \sum_{k=1}^{p} \overline{w}_k |\beta_k| \tag{2.11}$$

with respect to β, where $\overline{w}_k$ $(k = 1, 2, \cdots, p)$ is a prespecified weight. By (2.11), we propose the other estimator of β, denoted by $\widehat{\beta}^{(AL)}$, which is defined to solve the following adaptive LASSO penalized expectile regression optimization problem

$$\widehat{\beta}^{(AL)} = \arg\min_{\beta \in \mathbb{R}^p} \widetilde{Q}_{AL}(\beta). \tag{2.12}$$

III. ASYMPTOTIC PROPERTIES

In this section, we establish the asymptotic properties of the proposed estimators $\widehat{\beta}^{(SCAD)}$ and $\widehat{\beta}^{(AL)}$, respectively. We assume that our sample set $\{x_i, y_i\}_{i=1}^{N}$ come from the following data generation process

$$y_i = x_i^{\mathrm{T}}\beta + \epsilon_i = \left(x_i^1\right)^{\mathrm{T}} \beta_1 + \left(x_i^2\right)^{\mathrm{T}} \beta_2 + \epsilon_i, \quad i = 1, 2, \cdots, N, \qquad (3.1)$$

where $x_i = \left(\left(x_i^1\right)^{\mathrm{T}}, \left(x_i^2\right)^{\mathrm{T}}\right)^{\mathrm{T}}$, $\beta = \left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}$, $x_i^1 \in \mathbb{R}^q$, $x_i^2 \in \mathbb{R}^{p-q}$. The true regression coefficients are $\beta_1 = \beta_{10}$ with each component being nonzero, and $\beta_2 = \beta_{20} = 0$, as a result $\beta_0 = \left(\beta_{10}^{\mathrm{T}}, \beta_{20}^{\mathrm{T}}\right)^{\mathrm{T}}$.

To give the asymptotic properties, we introduce some notation. Define

$$\begin{aligned}
\epsilon_{ji} &= y_{ji} - x_{ji}^{\mathrm{T}}\beta_0, \quad \overline{\epsilon}_{ji} = y_{ji} - x_{ji}^{\mathrm{T}}\overline{\beta},\\
g(\tau) &= \tau\left[1 - F_\epsilon(0)\right] + (1-\tau) F_\epsilon(0),\\
\varphi_\tau(u) &= 2\tau u I(u > 0) + 2(1-\tau) u I(u \le 0),\\
a(\tau) &= \mathrm{Var}\left(\varphi_\tau\left(\overline{\epsilon}_{ji}\right)\right), \quad b(\tau) = \mathrm{Cov}\left(\varphi_\tau\left(\overline{\epsilon}_{ji}\right), \varphi_\tau\left(\epsilon_{ji}\right)\right),\\
c(\tau) &= \tau^2 \mathrm{E}\left[\epsilon_{ji}^2 I\left(\epsilon_{ji} > 0\right)\right] + (1-\tau)^2 \mathrm{E}\left[\epsilon_{ji}^2 I\left(\epsilon_{ji} \le 0\right)\right],\\
d(\tau) &= (m-1)\, a(\tau) + (2-2m)\, b(\tau) + 4mc(\tau),
\end{aligned}$$

where $F_\epsilon(\cdot)$ is the distribution function of ϵ_{ji}. In order to obtain our theoretical results, we enforce the following conditions throughout the paper. Note that $\|\cdot\|_2$ refers to the L_2-norm in the Euclidean space, $\xrightarrow{d}$ represents the convergence in distribution, and $\xrightarrow{\mathrm{P}}$ represents the convergence in probability.

(C1) Regression errors $\{\epsilon_i\}_{i=1}^{N}$ are independent and identically distributed with a common cumulative distribution function $F_\epsilon(\cdot)$, and given x_i, the τ-th expectile of ε_i is zero and satisfies $\mathrm{E}\left[\epsilon_i^2 \mid x_i\right] < \infty$.

(C2) There exists a positive definite matrix $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12}^{\mathrm{T}} & \Sigma_{22} \end{pmatrix}$ such that $\lim\limits_{N\to\infty}\left(\sum\limits_{i=1}^{N} x_i x_i^{\mathrm{T}}\right)/N = \Sigma$, where Σ_{11} is the q-by-q submatrix of Σ, Σ_{12} is the q-by-$(p-q)$ submatrix of Σ, Σ_{22} is the $(p-q)$-by-$(p-q)$ submatrix of Σ.

A. SCAD

We establish the $\sqrt{N}$-consistency of our proposed SCAD penalized estimator $\widehat{\beta}^{(SCAD)}$ as shown in Theorem 1 when the tuning parameter $\lambda = \lambda(N) \to 0$ as $N \to \infty$, and we also establish the oracle property of $\widehat{\beta}^{(SCAD)}$ as shown in Theorem 2 when the tuning parameter $\lambda = \lambda(N) \to 0$ and $\sqrt{N}\lambda \to \infty$ as $N \to \infty$.

Theorem 1 (Consistency): Suppose the sample set $\{x_i, y_i\}_{i=1}^{N}$ is generated according to process (3.1); under Conditions (C1) and (C2), if $\lambda = \lambda(N) \to 0$, then $\widehat{\beta}^{(SCAD)}$ converges to β_0 in probability, i.e., $\| \; \widehat{\beta}^{SCAD} - \beta_0 \; \|_2 = O_p\left(N^{-\frac{1}{2}}\right)$.

Remark 1: Theorem 1 shows that the estimator $\widehat{\beta}^{(SCAD)}$ is $\sqrt{N}$-consistent.

Theorem 2 (Oracle Property): Suppose the sample set $\{x_i, y_i\}_{i=1}^{N}$ is generated according to process (3.1); under Conditions (C1) and (C2), if $\lambda = \lambda(N) \to 0$ and $\sqrt{N}\lambda \to \infty$ as $N \to \infty$, then with probability tending to one the $\sqrt{N}$-consistent local minimizer $\widehat{\beta}^{(SCAD)} = \begin{pmatrix} \widehat{\beta}_1^{(SCAD)} \\ \widehat{\beta}_2^{(SCAD)} \end{pmatrix}$ in Theorem 1 satisfies:

(I) *Sparsity:* $\widehat{\beta}_2^{(SCAD)} = 0$;

(II) *Asymptotic Normality:* $\sqrt{N}\left(\widehat{\beta}_1^{(SCAD)} - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{d(\tau)}{4g^2(\tau)}\Sigma_{11}^{-1}\right)$.

Remark 2: For the SCAD penalized expectile regression, according to Theorem 2, if $\lambda = \lambda(N) \to 0$ and $\sqrt{N}\lambda \to \infty$ as $N \to \infty$, the penalized estimators possess the oracle property and perform as well as the expectile estimates for estimating β_1 knowing $\beta_2 = 0$.

Remark 3: If we set the initial value $\overline{\beta}$ to satisfy $\overline{\beta} = \beta_0 + O_p\left(N^{-\frac{1}{2}}\right)$, it follows that $a(\tau) = b(\tau) = 4c(\tau)$, and $d(\tau) = 4c(\tau)$, then, by Theorem 2, we have

$$\sqrt{N}\left(\widehat{\beta}_1^{(SCAD)} - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{c(\tau)}{g^2(\tau)}\Sigma_{11}^{-1}\right).$$

Remark 3 gives the same asymptotic results as those of Zhao and Zhang (2018).

B. ADAPTIVE LASSO

We establish the oracle property of our proposed adaptive LASSO penalized estimator $\widehat{\beta}^{(AL)}$ as shown in Theorem 3 when the tuning parameter $\lambda = \lambda(N)$, $\sqrt{N}\lambda \to 0$ and $N^{(r+1)/2}\lambda \to \infty$ as $N \to \infty$.

Theorem 3 (Oracle Property): Suppose the sample set $\{x_i, y_i\}_{i=1}^{N}$ is generated according to process (3.1); under Conditions (C1) and (C2), if $\lambda = \lambda(N)$, $\sqrt{N}\lambda \to 0$ and $N^{(r+1)/2}\lambda \to \infty$ as $N \to \infty$, then the adaptive LASSO expectile regression estimator $\widehat{\beta}^{(AL)} = \begin{pmatrix} \widehat{\beta}_1^{(AL)} \\ \widehat{\beta}_2^{(AL)} \end{pmatrix}$ satisfies:

(a) *Sparsity:* $\widehat{\beta}_2^{(AL)} = 0$;

(b) *Asymptotic Normality:* $\sqrt{N}\left(\widehat{\beta}_1^{(AL)} - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{d(\tau)}{4g^2(\tau)}\Sigma_{11}^{-1}\right)$.

Remark 4: Except for the results similar to Remarks 2 and 3 for the adaptive LASSO estimator $\widehat{\beta}^{(AL)}$ when $\lambda = \lambda(N)$, $\sqrt{N}\lambda \to 0$ and $N^{(r+1)/2}\lambda \to \infty$ as $N \to \infty$, Theorem 3 also shows that $\widehat{\beta}^{(AL)}$ does not need to be $\sqrt{N}$-consistent for adaptive LASSO.

IV. AUGMENTED PROXIMAL ADMM ALGORITHM

To propose an algorithm for calculating the SCAD and adaptive LASSO penalized estimators, we first set $\overline{\beta}$ to be the current t-th iteration value $\beta^{(t)}$ and unify the SCAD-penalized GEL function $\widetilde{Q}_{SCAD}$ and adaptive LASSO penalized GEL function $\widetilde{Q}_{AL}$ as follows:

$$\widetilde{Q}^{(t)}(\beta) = Q_j(\beta) + \left\langle \nabla F\left(\beta^{(t)}\right), \beta\right\rangle + \lambda \left\|w \circ \beta\right\|_1 \qquad (4.1)$$

where $\nabla F\left(\beta^{(t)}\right) = \nabla Q_N\left(\beta^{(t)}\right) - \nabla Q_j\left(\beta^{(t)}\right)$, $w = \left(w_1, \cdots, w_p\right)^{\mathrm{T}}$ is the vector of nonnegative weights, and $\|w \circ \beta\|_1 = \sum_{k=1}^{p} w_k |\beta_k|$ with $\circ$ denoting the Hadamard product.

As discussed in Zou and Li [32], the SCAD penalty function can be local linear approximation as

$$p_\lambda(|\beta_k|) \approx \nabla p_\lambda\left(\left|\beta_k^{(t)}\right|\right)|\beta_k|,$$

where $\nabla p_\lambda(\cdot)$ is the first-order derivative of $p_\lambda(\cdot)$, defined by

$$\nabla p_\lambda(u) = \lambda\left[I(u \leq \lambda) + \frac{(a\lambda - u)_+}{(a-1)\lambda} I(u > \lambda)\right]$$

for some $a > 2$ and $u > 0$, and the notation e_+ denotes the positive of e; that is, e_+ is e if $e > 0$, zero otherwise. Thus, for the SCAD penalized expectile regression, $w_k\ (k = 1, 2, \cdots, p)$ in (4.1) can be chosen as $w_k = \lambda^{-1}\nabla p_\lambda\left(\left|\beta_k^{(t)}\right|\right)$. For the adaptive LASSO penalized expectile regression, we choose $w_k = \left(\widehat{\beta}_k^{(L)} + 1/n\right)^{-1}$ for $k = 1, 2, \cdots, p$, where $\widehat{\beta}_k^{(L)}$ denotes the expectile LASSO estimator for β_k. We see that the problem of solving the above penalized GEL function can be transformed into a problem of solving a convex optimization.

The implementation of optimizing penalized GEL function (4.1) is difficult and complicated in practice, owing to the fact that the data volume is large and the dimension is high. The ADMM algorithm has become popular recently owing to its capability of solving large-scale optimization problems. In the following, we propose an augmented proximal ADMM algorithm as an efficient tool for solving large-scale expectile regression with sparsity. Denote $G_\tau(z) = n^{-1}\sum_{i=1}^{n}\rho_\tau(z_i)$ for $z = (z_1, z_2, \cdots, z_n)^{\mathrm{T}}$, where $z = y - x\beta$, $y = \left(y_{j1}, y_{j2}, \cdots, y_{jn}\right)^{\mathrm{T}}$ is an $(n \times 1)$-dimensional vector and $x = \left(x_{j1}, x_{j2}, \cdots, x_{jn}\right)^{\mathrm{T}}$ is a matrix with dimension $n \times p$. The problem of optimizing the penalized GEL function (4.1) can be transformed into the following optimization problem:

$$\begin{cases} \min\limits_{\beta \in \mathbb{R}^p, z \in \mathbb{R}^n} & G_\tau(z) + \left\langle \nabla F\left(\beta^{(t)}\right), \beta\right\rangle + \lambda\|w \circ \beta\|_1, \\ \text{s.t.} & x\beta + z = y. \end{cases} \tag{4.2}$$

The ADMM solves problem (4.2) and can be rewritten as in the following equivalent form:

$$\begin{cases} \min\limits_{\beta \in \mathbb{R}^p, z \in \mathbb{R}^n} & [G_\tau(z) + \left\langle \nabla F\left(\beta^{(t)}\right), \beta\right\rangle + \lambda\|w \circ \beta\|_1 \\ & + \frac{\gamma}{2}\|x\beta + z - y\|_2^2], \\ \text{s.t.} & x\beta + z = y, \end{cases} \tag{4.3}$$

where the last term $\frac{\gamma}{2}\|x\beta + z - y\|_2^2$ is called the augmentation term and added to achieve better convergence properties. $\gamma > 0$ is a tuning augmentation parameter. According to the convex optimization literature, the problem (4.3) has the following Lagrangian function:

$$\begin{aligned} L_\gamma(\beta, z, \theta) = G_\tau(z) + \left\langle \nabla F\left(\beta^{(t)}\right), \beta\right\rangle + \lambda\|w \circ \beta\|_1 \\ - \langle \theta, x\beta + z - y\rangle + \frac{\gamma}{2}\|x\beta + z - y\|_2^2, \end{aligned}$$

where $\theta = (\theta_1, \theta_2, \cdots, \theta_n)^{\mathrm{T}}$ is the Lagrangian multiplier.

The standard ADMM algorithm alternately minimizes the Lagrangian function in β and z, and maximizes θ in the dual direction, which results in the update as follows:

$$\begin{cases} \beta^{(t+1),j} := \arg\min\limits_{\beta \in \mathbb{R}^p} L_r\left(\beta, z^{(t)}, \theta^{(t)}\right), \\ z^{(t+1),j} := \arg\min\limits_{z \in \mathbb{R}^n} L_r\left(\beta^{(t+1),j}, z, \theta^{(t)}\right), \\ \theta^{(t+1),j} := \theta^{(t)} - \gamma\left(x\beta^{(t+1),j} + z^{(t+1),j} - y\right), \\ \quad j = 1, 2, \cdots, m, \end{cases} \tag{4.4}$$

where $\left(\beta^{(t)}, z^{(t)}, \theta^{(t)}\right)$ denotes the t-th iteration of the algorithm for $t \geq 0$. Discard the constant term irrelevant to the corresponding parameter, the update (4.4) can be rewritten as:

$$\begin{cases} \beta^{(t+1),j} := \arg\min\limits_{\beta \in \mathbb{R}^p} [\left\langle \nabla F\left(\beta^{(t)}\right), \beta\right\rangle + \lambda\|w \circ \beta\|_1 \\ \qquad - \left\langle \theta^{(t)}, x\beta\right\rangle + \frac{\gamma}{2}\left\|x\beta + z^{(t)} - y\right\|_2^2], \\ z^{(t+1),j} := \arg\min\limits_{z \in \mathbb{R}^n}\left[G_\tau(z) - \left\langle \theta^{(t)}, z\right\rangle\right. \\ \qquad \left. + \frac{\gamma}{2}\left\|x\beta^{(t+1),j} + z - y\right\|_2^2\right], \\ \theta^{(t+1),j} := \theta^{(t)} - \gamma\left(x\beta^{(t+1),j} + z^{(t+1),j} - y\right), \\ \qquad j = 1, 2, \cdots, m. \end{cases} \tag{4.5}$$

The z-update in (4.5) has a closed-form solution, which can be executed component-wisely. That is, for $i = 1, 2, \cdots, n$, we have

$$\begin{aligned} & z_i^{(t+1),j} : \\ & = \arg\min_{z_i \in \mathbb{R}} \frac{1}{n}\rho_\tau(z_i) - \theta_i^{(t)} z_i + \frac{\gamma}{2}\left(z_i + x_{ji}^{\mathrm{T}}\beta^{(t+1),j} - y_{ji}\right)^2 \\ & = \arg\min_{z_i \in \mathbb{R}} \rho_\tau(z_i) + \frac{n\gamma}{2}\left[z_i - \left(y_{ji} - x_{ji}^{\mathrm{T}}\beta^{(t+1),j} + \frac{\theta_i^{(t)}}{\gamma}\right)\right]^2 \\ & = \mathrm{Prox}_{\rho_\tau}\left(y_{ji} - x_{ji}^{\mathrm{T}}\beta^{(t+1),j} + \gamma^{-1}\theta_i^{(t)}, n\gamma\right), \\ & \quad j = 1, 2, \cdots, m, \end{aligned} \tag{4.6}$$

where for a given $\tau \in (0, 1)$ and $\alpha > 0$, the proximal mapping $\mathrm{Prox}_{\rho_\tau}[\xi, \alpha]$ can be written as

$$\begin{aligned} \mathrm{Prox}_{\rho_\tau}[\xi, \alpha] & := \arg\min_{u \in \mathbb{R}} \rho_\tau(u) + \frac{\alpha}{2}(u - \xi)^2 \\ & = \begin{cases} \dfrac{\alpha\xi}{2\tau + \alpha}, & \xi > 0, \\ \dfrac{\alpha\xi}{2(1-\tau) + \alpha}, & \xi \leq 0. \end{cases} \end{aligned}$$

Performing the average of $z_i^{(t+1),j}$ in (4.6) between m machines, we get

$$z_i^{(t+1)} = \frac{1}{m}\sum_{j=1}^{m} z_i^{(t+1),j}, \quad i = 1, 2, \cdots, n. \tag{4.7}$$

The computational difficulty mainly lies in the β-update in (4.5). Unlike the z-update, the β-update does not have a simple closed-form formula for the generic design matrix x. However, if we update the formula for β with a simple closed-form as well, the algorithm will be more transparent and easy to code. To do this, we use a widely used technique called "linearization". Specifically, we consider adding a proximal term to the objection function in the β-update and replacing the β-update in (4.5) with the following augmented β-update:

$$\beta^{(t+1)} := \arg\min_{\beta\in\mathbb{R}^p} [\langle \nabla F\left(\beta^{(t)}\right), \beta\rangle + \lambda \left\| w \circ \beta \right\|_1 - \langle \theta^{(t)}, x\beta\rangle + \frac{\gamma}{2}\left\| x\beta + z^{(t)} - y \right\|_2^2 + \frac{1}{2}\left\| \beta - \beta^{(t)} \right\|_V^2],$$

where V is a positive semidefinite matrix. We choose $V = \gamma\left(\eta I_{p\times p} - x^{\mathrm{T}}x\right)$ with $\eta \geq \Lambda_{\max}\left(x^{\mathrm{T}}x\right)$, where $I_{p\times p}$ is the $(p\times p)$-dimensional unit matrix, $\Lambda_{\max}$ () denotes the largest eigenvalue of a real symmetric matrix and $\|S\|_V^2 := \langle S, VS\rangle$ is the seminorm.

For $j = 1, 2, \cdots, m$, the augmented β-update can be performed component-wisely as, $\beta^{(t+1),j}$, shown at the bottom of the page.

Define a soft shrinkage operator Shrink $[u, \alpha] = \mathrm{sgn}(u)(|u| - a)_+$, where sgn $(\cdot)$ is a sign function, and then by performing some simple calculation, we have, (4.8), as shown at the bottom of the page, where $x_{(k)}$ denotes the k-th column of x for $k = 1, 2, \cdots, p$. By performing the average of $\beta^{(t+1),j}$ in (4.8) between m machines, we get

$$\beta^{(t+1)} = m^{-1}\sum_{j=1}^{m}\beta^{(t+1),j}. \tag{4.9}$$

Similarly, we average the θ-update $\theta^{(t+1),j}$ in (4.5) between m machines and get

$$\theta^{(t+1)} = m^{-1}\sum_{j=1}^{m}\theta^{(t+1),j}. \tag{4.10}$$

Based on (4.6), (4.7), (4.8), (4.9) and (4.10), we summarize the augmented proximal ADMM algorithm for large-scale expectile regression with sparsity in Algorithm 1.

Algorithm 1 Augmented Proximal ADMM Algorithm for Solving Large-Scale Expectile Regression With Sparsity

```
Initialize $\beta^{(0)}, z^{(0)}, \theta^{(0)}$;
for $l = 0, 1, \ldots, L-1$ do
    for $j = 1, 2, \ldots, m$ do
        Each machine evaluates $\nabla Q_j\left(\beta^{(l)}\right)$ and sends it to the CPU;
        The CPU computes
        $\nabla Q_N\left(\beta^{(l)}\right) = m^{-1}\sum_{j=1}^{m}\nabla Q_j\left(\beta^{(l)}\right)$
        and broadcasts to machines;
        Do the following iterates in each machine and send to the CPU:
        for $t = 0, 1, \ldots, T-1$ do
            $\beta^{(t+1),j} = \left[\mathrm{Shrink}\left(\beta_k^{(t)} - \frac{\nabla F(\beta_k^{(t)}) - x_{(k)}^{\mathrm{T}}(\gamma y - \gamma x\beta^{(t)} - \gamma z^{(t)} + \theta^{(t)})}{\gamma\eta}, \frac{\lambda w_k}{\gamma\eta}\right)\right]_{1\leq k\leq p}$;
            $z^{(t+1),j} = \left[\mathrm{Prox}_{\rho_\tau}\left(y_{ji} - x_{ji}^{\mathrm{T}}\beta^{(t+1),j} + \gamma^{-1}\theta_i^{(t)}, n\gamma\right)\right]_{1\leq i\leq n}$;
            $\theta^{(t+1),j} = \theta^{(t)} - \gamma(x\beta^{(t+1),j} + z^{(t+1),j} - y)$;
        end for
    end for
    The CPU computes
        $\beta^{(l+1)} = m^{-1}\sum_{j=1}^{m}\beta^{(T),j}$;
        $z^{(l+1)} = m^{-1}\sum_{j=1}^{m} z^{(T),j}$;
        $\theta^{(l+1)} = m^{-1}\sum_{j=1}^{m}\theta^{(T),j}$
    and broadcasts $\beta^{(l+1)}, z^{(l+1)}, \theta^{(l+1)}$ to each machine;
end for
Return $\widehat{\beta}^{SCAD}(\widehat{\beta}^{AL}) = \beta^{(L)}$.
```

Remark 5: If we optimize directly the objective function $Q_N(\beta) + \lambda\|w \circ \beta\|_1$, Algorithm 1 can be degenerated into the following Algorithm 2. However, $x = (x_1, x_2, \cdots, x_N)^T$ is a $(N \times p)$-dimensional matrix and $z = (z_1, z_2, \cdots, z_N)^T$ is a N-dimensional vector defined by $z = y - x\beta$, where $y = (y_1, y_2, \cdots, y_N)^T$.

$$\beta^{(t+1),j} := \arg\min_{\beta\in\mathbb{R}^p}\left[\lambda\left\| w\circ\beta\right\|_1 + \frac{\gamma\eta}{2}\left\| \beta - \frac{-\nabla F\left(\beta^{(t)}\right) + \gamma\eta\beta^{(t)} + x^{\mathrm{T}}\left(\gamma y - \gamma x\beta^{(t)} - \gamma z^{(t)} + \theta^{(t)}\right)}{\gamma\eta}\right\|_2^2\right]$$

$$\beta^{(t+1),j} := \left[\mathrm{Shrink}\left(\beta_k^{(t)} - \frac{\nabla F\left(\beta_k^{(t)}\right) - x_{(k)}^{\mathrm{T}}\left(\gamma y - \gamma x\beta^{(t)} - \gamma z^{(t)} + \theta^{(t)}\right)}{\gamma\eta}, \frac{\lambda w_k}{\gamma\eta}\right)\right]_{1\leq k\leq p} \tag{4.8}$$

TABLE 1. Simulation results for SCAD penalized expectile regression model.

ϵ	τ	Method	Size	P1	P2	ER
$N(0,1)$	0.5	Psingle	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0004(0.0003)
		Centralize	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0013(0.0007)
		Paverage	**4.00(0.00)**	**1.00(0.00)**	**0.00(0.00)**	**0.0004(0.0003)**
	0.3	Psingle	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0232(0.0079)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0389(0.0093)
		Paverage	**4.99(0.10)**	**1.00(0.00)**	**0.99(0.10)**	**0.0149(0.0063)**
	0.7	Psingle	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0243(0.0080)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0365(0.0094)
		Paverage	**5.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.0159(0.0066)**
$t(3)$	0.5	Psingle	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0013(0.0012)
		Centralize	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0044(0.0019)
		Paverage	**4.00(0.00)**	**1.00(0.00)**	**0.00(0.00)**	**0.0017(0.0015)**
	0.3	Psingle	5.06(0.31)	1.00(0.00)	1.00(0.00)	0.0384(0.0149)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0603(0.0181)
		Paverage	**5.02(0.25)**	**1.00(0.00)**	**0.98(0.14)**	**0.0221(0.0109)**
	0.7	Psingle	5.02(0.20)	1.00(0.00)	1.00(0.00)	0.0452(0.0154)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0526(0.0154)
		Paverage	**5.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.0197(0.0097)**
Laplace(0, 1)	0.5	Psingle	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0007(0.0005)
		Centralize	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0031(0.0013)
		Paverage	**4.00(0.00)**	**1.00(0.00)**	**0.00(0.00)**	**0.0008(0.0006)**
	0.3	Psingle	5.02(0.14)	1.00(0.00)	1.00(0.00)	0.0407(0.0128)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0545(0.0147)
		Paverage	**4.98(0.14)**	**1.00(0.00)**	**0.98(0.14)**	**0.0163(0.0081)**
	0.7	Psingle	5.04(0.24)	1.00(0.00)	1.00(0.00)	0.0469(0.0154)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0549(0.0148)
		Paverage	**5.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.0199(0.0092)**

V. SIMULATION STUDIES

We conduct simulation studies to investigate the variable selection and parameter estimation accuracy of our proposed Algorithm 1 (Paverage for short). We compare it with the distributed optimization method (Psingle for short) and the optimal global method (Centralize for short). For the Psingle method, we use the augmented ADMM algorithm to solve the CSL function (2.7) with penalty term $\lambda \|w \circ \beta\|_1$ (Pan [18]). For the Centralize method, we use Algorithm 2 in Remark 5 to optimize the global loss function (2.2) with penalty term $\lambda \|w \circ \beta\|_1$.

The scalar response is generated according to the following heteroskedastic model [27]:

$$y = x_6 + x_{12} + x_{15} + x_{20} + 0.7 x_1 \epsilon, \tag{4.11}$$

where the predictors $x_1, x_2, \cdots, x_p$ are generated by $x_1 = \Phi(\widetilde{x}_1)$ and $x_j = \widetilde{x}_j$ for $j = 2, 3, \cdots, p$ with $\Phi(\cdot)$ being the cumulative distribution function of the standard normal distribution and $\left(\widetilde{x}_1, \widetilde{x}_2, \cdots, \widetilde{x}_p\right)^{\mathrm{T}} \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = \left(0.5^{|i-j|}\right)_{p \times p}$ for $i, j = 1, 2, \cdots, p$. Note that x_1 plays an important role in the conditional distribution of y given the predictors but does not directly influence on the mean of the conditional distribution.

For a comprehensive comparison, we consider the standard normal $\mathcal{N}(0, 1)$ and two heavy-tailed distributions $t(3)$ and Laplace $(0, 1)$ for the error term ϵ. We consider the sample size $N = 2000$ and divide the data evenly and randomly on $m = 20$ machines with the sample size of each machine $n = 100$. We fix the parameter dimension $p = 300$. Similar to Wang *et al.* [27], we consider three different expectiles $\tau = 0.3, 0.5$ and 0.7. We choose tuning parameter λ by

TABLE 2. Simulation results for adaptive LASSO penalized expectile regression model.

ϵ	τ	Method	Size	P1	P2	ER
$N(0,1)$	0.5	Psingle	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0003(0.0002)
		Centralize	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0003(0.0002)
		Paverage	**4.00(0.00)**	**1.00(0.00)**	**0.00(0.00)**	**0.0003(0.0002)**
	0.3	Psingle	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0516(0.0109)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0540(0.0111)
		Paverage	**5.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.0329(0.0088)**
	0.7	Psingle	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0554(0.0116)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0563(0.0117)
		Paverage	**5.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.0371(0.0097)**
$t(3)$	0.5	Psingle	4.01(0.10)	1.00(0.00)	0.00(0.00)	0.0009(0.0008)
		Centralize	4.02(0.14)	1.00(0.00)	0.00(0.00)	0.0010(0.0009)
		Paverage	**4.00(0.00)**	**1.00(0.00)**	**0.00(0.00)**	**0.0009(0.0007)**
	0.3	Psingle	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.1020(0.0241)
		Centralize	5.06(0.28)	1.00(0.00)	1.00(0.00)	0.1046(0.0245)
		Paverage	**5.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.0665(0.0188)**
	0.7	Psingle	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0968(0.0223)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0991(0.0226)
		Paverage	**5.01(0.10)**	**1.00(0.00)**	**1.00(0.00)**	**0.0734(0.0193)**
Laplace(0, 1)	0.5	Psingle	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0006(0.0005)
		Centralize	4.00(0.00)	1.00(0.00)	0.00(0.00)	0.0007(0.0005)
		Paverage	**4.00(0.00)**	**1.00(0.00)**	**0.00(0.00)**	**0.0006(0.0005)**
	0.3	Psingle	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0865(0.0192)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0892(0.0198)
		Paverage	**5.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.0545(0.0153)**
	0.7	Psingle	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0951(0.0205)
		Centralize	5.00(0.00)	1.00(0.00)	1.00(0.00)	0.0971(0.0209)
		Paverage	**5.00(0.00)**	**1.00(0.00)**	**1.00(0.00)**	**0.0646(0.0171)**

minimizing the test error with a 20-fold cross-validation method, i.e., $\min\limits_{\lambda\in\mathbb{R}} \sum_{i\in\text{test}} \rho_\tau(y_i - x_i^{\mathrm{T}}\widehat{\beta}_{\text{train}})$ with $\widehat{\beta}_{\text{train}}$ being the parameter estimator based on the training set.

Based on the simulation of 100 repetitions, we compare the performance of the aforementioned three methods in terms of the following criteria. The average of nonzero regression coefficients $\widehat{\beta}_k \neq 0$ for $k = 1, 2, \cdots, p$ (Size for short). The proportion of simulation runs including all true crucial predictors (P1 for short), that is, $\widehat{\beta}_k \neq 0$ for any $k \geq 1$ satisfying the true coefficient $\beta_{k0} \neq 0$. Note that when $\tau = 0.5$, this implies the percentage of times that we include $x_6, x_{12}, x_{15}, x_{20}$, and when $\tau = 0.3$ and 0.7, x_1 should also be included. The proportion of simulation runs x_1 is selected (P2 for short). The estimated error is defined by $\|\widehat{\beta} - \beta\|_2^2$ (ER for short). The numbers in parentheses in the columns labeled Size, P1, P2 and ER are the corresponding sample standard deviations.

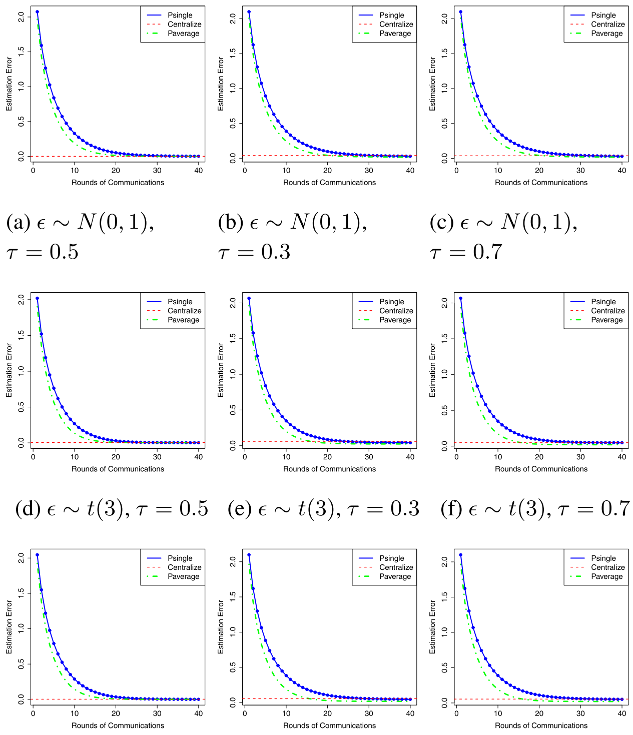
Table 1 reports the parameter estimation and variable selection results for SCAD penalized expectile regression model, and Table 2 reports the results for adaptive LASSO penalized expectile regression model. In terms of parameter estimation, the Paverage and Psingle methods generate estimates that can compete with the Centralize method. The Paverage method exhibits smaller estimation errors than the Psingle method. In terms of parameter estimation, the Paverage and Psingle methods generate estimates that can compete with the Centralize method. In terms of variable selection, from Table 1, when $\tau = 0.3$ and 0.7, all methods successfully select the five variables in the mean (x_1, x_6, x_{12}, x_{15} and x_{20}) with high probabilities (P1). For x_1, except for $\tau = 0.5$, all

Algorithm 2 ADMM Algorithm

```
Initialize $\beta^{(0)}, z^{(0)}, \theta^{(0)}$;
1: for $t = 0, 1, \ldots, T-1$ do
2:     $\beta^{(t+1)} = \left[\text{Shrink}\left(\beta_k^{(t)} - \frac{-x_{(k)}^{\mathrm{T}}(\gamma y - \gamma x\beta^{(t)} - \gamma z^{(t)} + \theta^{(t)})}{\gamma\eta}, \frac{\lambda w_k}{\gamma\eta}\right)\right]_{1\le k\le p}$;
3:     $z^{(t+1)} = \left[\text{Prox}_{\rho_\tau}\left(y_i - x_i^{\mathrm{T}}\beta^{(t+1)} + \gamma^{-1}\theta_i^{(t)}, N\gamma\right)\right]_{1\le i\le N}$;
4:     $\theta^{(t+1)} = \theta^{(t)} - \gamma(x\beta^{(t+1)} + z^{(t+1)} - y)$;
5: end for
Return $\widehat{\beta}^{SCAD}(\widehat{\beta}^{AL}) = \beta^{T}$.
```

methods show high probabilities (P2). Similar conclusions can be obtained from Table 2.

To prove that our proposed method is communication-efficient, we plot how the estimation error varies with the number of rounds of communication. Figure 1 reports the results for the SCAD penalized expectile regression model, and Figure 2 reports the results for the adaptive LASSO penalized expectile regression model. The results show that for our proposed method, the estimation error declines to that of the Centralize method within a few rounds of communication. In addition, compared with the Psingle method, the estimation error of the Paverage method converges rapidly to that of the Centralize method in all scenarios, which shows that average steps lead to better performance.



(a) $\epsilon \sim N(0,1)$, $\tau = 0.5$ (b) $\epsilon \sim N(0,1)$, $\tau = 0.3$ (c) $\epsilon \sim N(0,1)$, $\tau = 0.7$

(d) $\epsilon \sim t(3)$, $\tau = 0.5$ (e) $\epsilon \sim t(3)$, $\tau = 0.3$ (f) $\epsilon \sim t(3)$, $\tau = 0.7$

(g) $\epsilon \sim \text{Laplace}(0,1)$, $\tau = 0.5$ (h) $\epsilon \sim \text{Laplace}(0,1)$, $\tau = 0.3$ (i) $\epsilon \sim \text{Laplace}(0,1)$, $\tau = 0.7$

FIGURE 1. Graphs for the SCAD penalized estimation error $\|\widehat{\beta} - \beta_0\|_2^2$ versus rounds of communications.

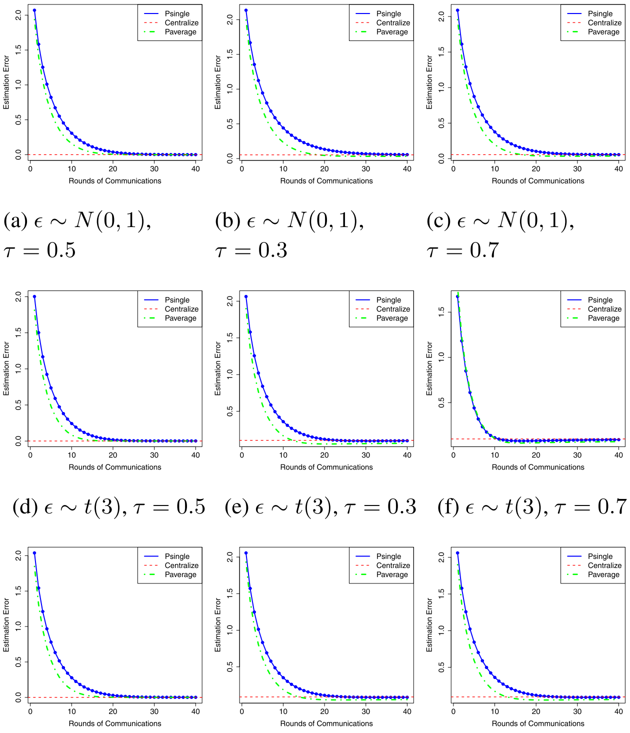
VI. REAL DATA ANALYSIS

We apply the proposed procedures to a dataset from the Communities and Crime study. The dataset combine socioeconomic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR, which contain 1994 observations and 128 variables. The dataset can be obtained from the website http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime. We remove the variables with excessive missing information and use the remaining 101 variables as covariates and the total number of violent crimes per 100000 population as the response variable. Figure 3 reports the histogram and density function graph of the response variable, from which we can see that the response variable is heavy-tailed. Therefore, it is necessary to use the expectile regression method to study the effect of the 101 covariates on the response variable.

We use Paverage, Centralize, and Psingle to analyze these data and compare their performance. We consider three different expectiles $\tau = 0.3, 0.5$ and 0.7 and randomly divide the dataset into two parts: a training dataset and a test dataset with sample sizes of 1400 and 594, respectively. A 14-fold cross-validation method is performed on the 1400-observation training dataset to determine the tuning parameter λ. The training dataset is randomly segmented into 14 machines, which means that each machine stores 100 samples at random. An estimate of the regression coefficient is obtained based on our proposed distributed method, and then a cross-validation method is used to evaluate the prediction error (PE) defined by PE $= \frac{1}{594}\sum_{i\in\text{test}}\rho_\tau(y_i - x_i^{\mathrm{T}}\widehat{\beta})$. where $\widehat{\beta}$ is the parameter estimation obtained by the training set. We repeat the above procedure 100 times, and the results are summarized in Table 3 and include the number of selected important explanatory variables (Size) and prediction error (PE). The numbers in parentheses in the columns labeled Size and PE are the corresponding sample standard deviations.

TABLE 3. Results for the analysis of the Communities and Crime study data set.

		SCAD		ALasso	
τ	Method	Size	PE	Size	PE
0.3	Psingle	9.92(2.60)	0.0224(0.0023)	10.68(2.38)	0.0222(0.0022)
	Centralize	14.52(1.11)	0.0215(0.0024)	15.42(0.98)	0.0170(0.0014)
	Paverage	**15.89(1.12)**	**0.0200(0.0019)**	**16.23(1.07)**	**0.0199(0.0019)**
0.5	Psingle	20.06(3.30)	0.0281(0.0032)	21.02(3.34)	0.0277(0.0032)
	Centralize	19.63(1.75)	0.0295(0.0038)	19.43(1.55)	0.0213(0.0021)
	Paverage	**26.16(1.26)**	**0.0251(0.0025)**	**26.44(1.32)**	**0.0250(0.0026)**
0.7	Psingle	26.84(3.41)	0.0304(0.0036)	27.96(3.57)	0.0300(0.0036)
	Centralize	22.12(1.59)	0.0313(0.0038)	21.04(1.33)	0.0238(0.0027)
	Paverage	**34.35(2.36)**	**0.0269(0.0030)**	**34.83(2.55)**	**0.0268(0.0030)**



(a) $\epsilon \sim N(0,1)$, $\tau = 0.5$

(b) $\epsilon \sim N(0,1)$, $\tau = 0.3$

(c) $\epsilon \sim N(0,1)$, $\tau = 0.7$

(d) $\epsilon \sim t(3)$, $\tau = 0.5$

(e) $\epsilon \sim t(3)$, $\tau = 0.3$

(f) $\epsilon \sim t(3)$, $\tau = 0.7$

(g) $\epsilon \sim \text{Laplace}(0,1)$, $\tau = 0.5$

(h) $\epsilon \sim \text{Laplace}(0,1)$, $\tau = 0.3$

(i) $\epsilon \sim \text{Laplace}(0,1)$, $\tau = 0.7$

FIGURE 2. Graphs for the adaptive LASSO penalized estimation error $\|\widehat{\beta} - \beta_0\|_2^2$ versus rounds of communications.

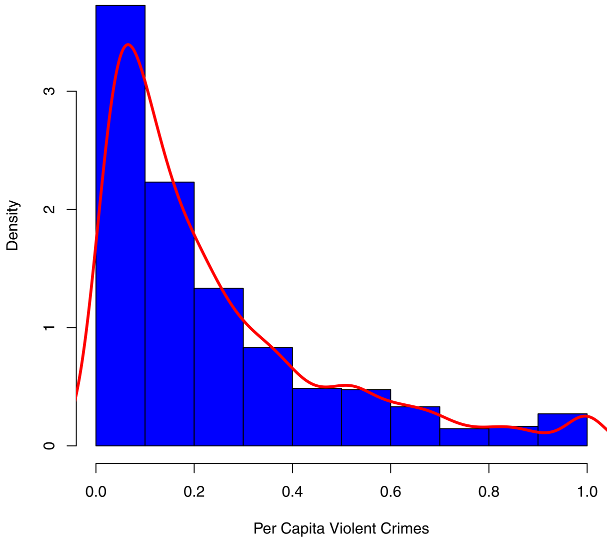


FIGURE 3. Histogram and density function graph of response variable "Per Capita Violent Crimes".

From Table 3, it can be concluded that the number of selected important variables and the standard deviation of prediction errors derived by the Paverage method is correspondingly close to those derived by the Centralize method. The results from the real example again verify that our proposed distributed method is an effective method to solve distributed sparse expectile regression and can produce results that are highly competitive with the Centralize method.

VII. CONCLUSION AND DISCUSSION

Expectile regression is a popular alternative when working with heterogeneous data and studies the overall conditional distribution of response to a given predictor in a heterogeneous environment. High-dimensional data are often collected across a wide range of research areas and often display heterogeneity. In addition, the sheer size of the data often makes it impossible to store all of them on one machine. Therefore, it is necessary to store data in a distributed way and develop a new distributed learning method with sparsity in expectile regression. In a distributed environment, we propose penalized large-scale expectile regression using SCAD and adaptive LASSO penalties and demonstrate the oracle properties introduced in Fan and Li [6] and Zou [31].

To generate a statistically optimal estimator with low communication complexity, we construct a GEL function that provides a communication-efficient surrogate for the global loss. The GEL function can be used in high-dimensional regularized estimation. To optimize the penalized GEL function, we propose an augmented proximal ADMM algorithm. Simulation results show that the proposed method has good performance with finite sample size, and the estimation error can be improved after a few rounds of communication until it matches that of the Centralize method. A real data example demonstrates that the implementation of the proposed method is easy in practice.

Several new directions are worth exploring in the future. First, here we examine a scenario where multiple node machines are connected to CPU, and all updates are done simultaneously. It would be interesting to extend this algorithm to decentralized and asynchronous settings. Second, the communication-efficient versions of the confidence regions and hypothesis tests of sparse expectile regression are of great significance in making statistical inference on the distribution. Finally, the future works may also explore the ideas presented to improve the computational cost of communication-efficient distributed multi-task learning with shared support [25].

APPENDIX

PROOF OF THEOREMS

To prove three theorems in Section 3, we first introduce and prove five lemmas.

Lemma 1: Let $\{h_n(u) : u \in U\}$ be a sequence of random convex functions defined on convex, open subset U of $\mathbb{R}^p$. Suppose $h(u)$ is a real-valued function on U for which $h_n(u) \to h(u)$ in probability, for each $u \in U$. Then for each compact subset $\overline{U}$ of U,

$$\sup_{u\in\overline{U}} |h_n(u) - h(u)| \xrightarrow{P} 0,$$

and the function $h(\cdot)$ is necessarily convex on U.

The proof of Lemma 1 can be found in Pollard [20].

Lemma 2: Let V be a symmetric and positive definite matrix, W be a random variable and $A_n(u)$ be a convex objective function with arg min α_n, if

$$A_n(u) = \frac{1}{2}u^{\mathrm{T}}Vu + W^{\mathrm{T}}u + o_p(1)$$

then $\alpha_n \xrightarrow{d} -V^{-1}W$.

The proof of Lemma 2 can be found in Hjort and Pollard (1993).

Lemma 3: Suppose the sample set $\{x_i, y_i\}_{i=1}^{N}$ is generated according to process (3.1); under Conditions (C1) and (C2), denote

$$H_n(u) = \sum_{i=1}^{n}\left[\rho_\tau\left(\epsilon_{ji} - x_{ji}^{\mathrm{T}}u/\sqrt{n}\right) - \rho_\tau\left(\epsilon_{ji}\right)\right] + n\left[\langle \nabla Q_N\left(\overline{\beta}\right) - Q_j\left(\overline{\beta}\right), \frac{u}{\sqrt{n}}\rangle\right], \qquad \text{(A.1)}$$

then we have

$$H_n(u) = g(\tau)\,u^{\mathrm{T}}\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]u + W_n^{\mathrm{T}}u + o_p(1),$$

where $g(\tau) = \tau[1 - F_\epsilon(0)] + (1-\tau)F_\epsilon(0)$, $W_n = \frac{1}{\sqrt{n}}\sum\limits_{i=1}^{n} D_i$, and $D_i = \xi_i - \frac{\eta_i}{m} - \zeta_i, i = 1, 2, \cdots, n$ with $\xi_i = \varphi_\tau\left(\overline{\epsilon}_{ji}\right)x_{ji}$, $\eta_i = \sum\limits_{j=1}^{m}\varphi_\tau\left(\overline{\epsilon}_{ji}\right)x_{ji}$, $\zeta_i = \varphi_\tau\left(\epsilon_{ji}\right)x_{ji}$, $\overline{\epsilon}_{ji} = y_{ji} - x_{ji}^{\mathrm{T}}\overline{\beta}$, $\epsilon_{ji} = y_{ji} - x_{ji}^{\mathrm{T}}\beta_0$.

Proof of Lemma 3: Under Conditions (C1) and (C2), similarly to the arguments of Zhao and Zhang [30], we obtain

$$\sum_{i=1}^{n}\left[\rho_\tau\left(\epsilon_{ji} - x_{ji}^{\mathrm{T}}u/\sqrt{n}\right) - \rho_\tau\left(\epsilon_{ji}\right)\right] = g(\tau)\,u^{\mathrm{T}}\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]u - \left[\frac{\sum_{i=1}^{n}\varphi_\tau\left(\epsilon_{ji}\right)x_{ji}}{\sqrt{n}}\right]^{\mathrm{T}}u + o_p(1). \qquad \text{(A.2)}$$

By performing a simple calculation, we have

$$\nabla Q_N\left(\overline{\beta}\right) - \nabla Q_j\left(\overline{\beta}\right) = \frac{1}{n}\sum_{i=1}^{n}\left[\varphi_\tau\left(\overline{\epsilon}_{ji}\right)x_{ji} - \frac{1}{m}\sum_{j=1}^{m}\varphi_\tau\left(\overline{\epsilon}_{ji}\right)x_{ji}\right], \qquad \text{(A.3)}$$

where $\varphi_\tau(u) = 2\tau u I(u > 0) + 2(1-\tau)uI(u \le 0)$.

Combining (A.1), (A.2), and (A.3), we obtain

$$\begin{aligned} H_n(u) &= g(\tau)\,u^{\mathrm{T}}\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]u \\ &\quad + \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left[\varphi_\tau\left(\overline{\epsilon}_{ji}\right)x_{ji} - \frac{1}{m}\sum_{j=1}^{m}\varphi_\tau\left(\overline{\epsilon}_{ji}\right)x_{ji} - \varphi_\tau\left(\epsilon_{ji}\right)x_{ji}\right]^{\mathrm{T}} \\ &\quad \times u + o_p(1). \end{aligned} \qquad \text{(A.4)}$$

If we denote $\xi_i = \varphi_\tau\left(\overline{\epsilon}_{ji}\right)x_{ji}$, $\eta_i = \sum\limits_{j=1}^{m}\varphi_\tau\left(\overline{\epsilon}_{ji}\right)x_{ji}$, $\zeta_i = \varphi_\tau\left(\epsilon_{ji}\right)x_{ji}$, $D_i = \xi_i - \frac{\eta_i}{m} - \zeta_i$, $W_n = \frac{1}{\sqrt{n}}\sum\limits_{i=1}^{n} D_i$, then we have

$$H_n(u) = g(\tau)\,u^{\mathrm{T}}\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]u + W_n^{\mathrm{T}}u + o_p(1). \qquad \text{(A.5)}$$

This completes the proof.

Lemma 4: Under Conditions (C1) and (C2), W_n is defined by Lemma 3, then we obtain

$$W_n = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} D_i \xrightarrow{d} \mathcal{N}\left(0, m^{-1}d(\tau)\,\Sigma\right).$$

Proof of Lemma 4: The Condition (C1) that the τ-expectile of the error term is zero, implies $\mathrm{E}\left[\varphi_\tau\left(\overline{\epsilon}_{ji}\right)|x_{ji}\right] = \mathrm{E}\left[\varphi_\tau\left(\epsilon_{ji}\right)|x_{ji}\right] = 0$, and

$$\begin{aligned} &\mathrm{Cov}(\xi_i, \xi_i) = a(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, \quad \mathrm{Cov}(\eta_i, \eta_i) = a(\tau)\sum_{j=1}^{m} x_{ji}x_{ji}^{\mathrm{T}}, \\ &\mathrm{Cov}(\zeta_i, \zeta_i) = 4c(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, \quad \mathrm{Cov}(\xi_i, \eta_i) = a(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, \\ &\mathrm{Cov}(\xi_i, \zeta_i) = b(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, \quad \mathrm{Cov}(\eta_i, \zeta_i) = b(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, \end{aligned}$$

where

$$\begin{aligned} &a(\tau) = \mathrm{Var}\left(\varphi_\tau\left(\overline{\epsilon}_{ji}\right)\right) \quad b(\tau) = \mathrm{Cov}\left(\varphi_\tau\left(\overline{\epsilon}_{ji}\right), \varphi_\tau\left(\epsilon_{ji}\right)\right) \\ &c(\tau) = \tau^2\mathrm{E}\left[\epsilon_{ji}^2 I\left(\epsilon_{ji} > 0\right)\right] + (1-\tau)^2\,\mathrm{E}\left[\epsilon_{ji}^2 I\left(\epsilon_{ji} \le 0\right)\right]. \end{aligned}$$

By routine calculation, we get

$$W \triangleq \mathrm{Var}\begin{pmatrix}\xi_i \\ \eta_i \\ \zeta_i\end{pmatrix} = \begin{pmatrix} a(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, & a(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, & b(\tau)\,x_{ji}x_{ji}^{\mathrm{T}} \\ a(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, & a(\tau)\sum\limits_{j=1}^{m} x_{ji}x_{ji}^{\mathrm{T}}, & b(\tau)\,x_{ji}x_{ji}^{\mathrm{T}} \\ b(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, & b(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}, & 4c(\tau)\,x_{ji}x_{ji}^{\mathrm{T}} \end{pmatrix}. \qquad \text{(A.6)}$$

By the fact that D_i is independent and identically distributed zero-mean random vectors, Condition (C2) and

$$D_i = \left(I_{p\times p}, -\frac{I_{p\times p}}{m}, -I_{p\times p}\right)_{p\times 3p}\begin{pmatrix}\xi_i \\ \eta_i \\ \zeta_i\end{pmatrix}_{3p\times 1}, \quad i = 1, 2, \cdots, n,$$

we have

$$\begin{aligned} \mathrm{Var}(W_n) &= \frac{1}{n}\sum_{i=1}^{n}\left(I_{p\times p}, \frac{-I_{p\times p}}{m}, -I_{p\times p}\right)W\begin{pmatrix} I_{p\times p} \\ \frac{-I_{p\times p}}{m} \\ -I_{p\times p}\end{pmatrix} \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{m-2}{m}a(\tau)\,x_{ji}x_{ji}^{\mathrm{T}} + \frac{2-2m}{m}b(\tau)\,x_{ji}x_{ji}^{\mathrm{T}}\right. \\ &\qquad \left. + 4c(\tau)\,x_{ji}x_{ji}^{\mathrm{T}} + \frac{a(\tau)}{m^2}\sum_{j=1}^{m} x_{ji}x_{ji}^{\mathrm{T}}\right] \\ &\xrightarrow{\mathrm{P}} m^{-1}d(\tau)\,\Sigma, \quad as \quad n \to \infty, \end{aligned} \qquad \text{(A.7)}$$

where $d(\tau) = (m-1)a(\tau) + (2-2m)b(\tau) + 4mc(\tau)$. By Central limit theorem, we obtain that

$$W_n = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} D_i \xrightarrow{d} \mathcal{N}\left(0, m^{-1}d(\tau)\Sigma\right). \tag{A.8}$$

This completes the proof.

Lemma 5: Suppose the sample set $\{x_i, y_i\}_{i=1}^{N}$ is generated according to process (3.1); under Conditions (C1) and (C2), if $\lambda = \lambda(n) \to 0$ and $\sqrt{n}\lambda \to \infty$ as $n \to \infty$, then with probability tending to one, for any given β_1 satisfying $\|\beta_1 - \beta_{10}\|_2 = O_p\left(n^{-\frac{1}{2}}\right)$ and any constant C, we obtain

$$\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}} = \arg \min_{\|\beta_2\|_2 \le Cn^{-\frac{1}{2}}} \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right).$$

i.e., for any $\delta > 0$

$$\mathrm{P}\left[\inf_{\|\beta_2\|_2 \le Cn^{-\frac{1}{2}}} \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right) > \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right)\right] \ge 1 - \delta.$$

Proof of Lemma 5: For any $\|\beta_1 - \beta_{10}\|_2 = O_p\left(n^{-\frac{1}{2}}\right)$, $\|\beta_2\|_2 \le Cn^{-\frac{1}{2}}$, and by Lemma 3, we have

$$\begin{aligned}
&n\left[\widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) - \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right)\right] \\
&= n\left[\widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) - \widetilde{Q}_{SCAD}\left(\left(\beta_{10}^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right)\right] \\
&\quad - n\left[\widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right) - \widetilde{Q}_{SCAD}\left(\left(\beta_{10}^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right)\right] \\
&= H_n\left(\sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) \\
&\quad - H_n\left(\sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right) - n\sum_{k=q+1}^{p} p_\lambda(|\beta_k|) \\
&= g(\tau)\sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, 0^{\mathrm{T}}\right)\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]\sqrt{n} \\
&\quad \times \left((\beta_1 - \beta_{10})^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}} + \sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, 0^{\mathrm{T}}\right)W_n \\
&\quad - g(\tau)\sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]\sqrt{n} \\
&\quad \times \left((\beta_1 - \beta_{10})^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}} \\
&\quad - \sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)W_n - n\sum_{k=q+1}^{p} p_\lambda(|\beta_k|) + o_p(1).
\end{aligned} \tag{A.9}$$

The conditions $\|\beta_1 - \beta_{10}\|_2 = O_p\left(n^{-\frac{1}{2}}\right)$ and $\|\beta_2\|_2 \le Cn^{-\frac{1}{2}}$ imply that

$$\begin{aligned}
&g(\tau)\sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, 0^{\mathrm{T}}\right)\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]\sqrt{n} \\
&\quad \times \left((\beta_1 - \beta_{10})^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}} = O_p(1), \\
&g(\tau)\sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]\sqrt{n} \\
&\quad \times \left((\beta_1 - \beta_{10})^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}} = O_p(1),
\end{aligned} \tag{A.10}$$

and

$$\begin{aligned}
&\sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, 0^{\mathrm{T}}\right)W_n - \sqrt{n}\left((\beta_1 - \beta_{10})^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)W_n \\
&= -\sqrt{n}\left(0^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)W_n \\
&= O_p\left(\sqrt{nm^{-1}d(\tau)\beta_2^{\mathrm{T}}\Sigma_{22}\beta_2}\right) = O_p(1),
\end{aligned} \tag{A.11}$$

where Σ_{22} is the right-bottom $(p-q)$-by-$(p-q)$ submatrix of Σ.

Based on the fact that $\varliminf_{\lambda\to 0}\varliminf_{\theta\to 0^+} \frac{p_\lambda'(\theta)}{\lambda} = 1$, we have

$$\begin{aligned}
&n\sum_{k=q+1}^{p} p_\lambda(|\beta_k|) \\
&\ge n\lambda\left[\varliminf_{\lambda\to 0}\varliminf_{\theta\to 0^+}\frac{p_\lambda'(\theta)}{\lambda}\right]\left[\sum_{k=q+1}^{p}|\beta_k|(1+o(1))\right] \\
&= n\lambda\left(\sum_{k=q+1}^{p}|\beta_k|\right)(1+o(1)).
\end{aligned}$$

Then the condition $\sqrt{n}\lambda \to \infty$ implies that $n\lambda = \sqrt{n}\left(\sqrt{n}\lambda\right)$ is of higher order than $\sqrt{n}$, which implies that, the last term of Eq. (A.9) dominates in magnitude, that is, $\widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) - \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right) < 0$ for large n. Then

$$\inf_{\|\beta_2\|_2 \le Cn^{-\frac{1}{2}}} \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right) > \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right)$$

with probability tending to one. This completes the proof.

Proof of Theorem 1: As discussed in Fan and Li [6], to prove Theorem 1, it is enough to show that for any given $\delta > 0$, there exists a large enough constant C, such that

$$\mathrm{P}\left[\inf_{\|u\|_2 = C} \widetilde{Q}_{SCAD}\left(\beta_0 + u/\sqrt{n}\right) > \widetilde{Q}_{SCAD}(\beta_0)\right] \ge 1 - \delta, \tag{A.12}$$

which implies that there exists a local minimum in the ball $\left\{\beta_0 + u/\sqrt{n} : \|u\|_2 \le C\right\}$ with probability at least $1-\delta$. This is in turn implies that there exists a local minimizer such that $\left\|\widehat{\beta}^{(SCAD)} - \beta_0\right\|_2 = O_p(n^{-\frac{1}{2}})$. By performing a simple calculation and Lemma 3, we get

$$\begin{aligned}
&n\left[\widetilde{Q}_{SCAD}\left(\beta_0 + u/\sqrt{n}\right) - \widetilde{Q}_{SCAD}(\beta_0)\right] \\
&= n\left[Q_j\left(\beta_0 + u/\sqrt{n}\right) - Q_j(\beta_0)\right] \\
&\quad + n\left[\left\langle\nabla Q_N\left(\bar{\beta}\right) - \nabla Q_j\left(\bar{\beta}\right), u/\sqrt{n}\right\rangle\right] \\
&\quad + n\left[\sum_{k=1}^{p} p_\lambda\left(\left|\beta_{k0} + u_k/\sqrt{n}\right|\right) - \sum_{k=1}^{p} p_\lambda\left(|\beta_{k0}|\right)\right]
\end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{n}\left[\rho_\tau\left(\epsilon_{ji} - x_{ji}^{\mathrm{T}}u/\sqrt{n}\right) - \rho_\tau(\epsilon_{ji})\right] \\ &\quad + n\left[\langle \nabla Q_N\left(\overline{\beta}\right) - \nabla Q_j\left(\overline{\beta}\right), u/\sqrt{n}\rangle\right] \\ &\quad + n\left[\sum_{k=1}^{p} p_\lambda\left(\left|\beta_{k0} + u_k/\sqrt{n}\right|\right) - \sum_{k=1}^{p} p_\lambda\left(|\beta_{k0}|\right)\right] \\ &\geq H_n(u) + I_n(u), \end{aligned} \tag{A.13}$$

where

$$\begin{aligned} H_n(u) &= g(\tau)u^{\mathrm{T}}\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]u + W_n^{\mathrm{T}}u + o_p(1), \\ I_n(u) &= n\sum_{k=1}^{q}\left[p_\lambda\left(\left|\beta_{k0} + u_k/\sqrt{n}\right|\right) - p_\lambda\left(|\beta_{k0}|\right)\right]. \end{aligned}$$

Due to Lemma 4, we have $W_n \xrightarrow{d} \mathcal{N}\left(0, m^{-1}d(\tau)\Sigma\right)$, therefore, $W_n^{\mathrm{T}}u$ is bounded in probability, i.e.,

$$W_n^{\mathrm{T}}u = O_p\left(\sqrt{m^{-1}d(\tau)u^{\mathrm{T}}\Sigma u}\right). \tag{A.14}$$

By applying Lemma 1 to $G_n(u) = H_n(u) - W_n^{\mathrm{T}}u$, we can strengthens this point-wise convergence to uniform convergence on compact subset of $\mathbb{R}^p$. Note that, for large n,

$$n\sum_{k=1}^{q}\left[p_\lambda\left(\left|\beta_{k0} + u_k/\sqrt{n}\right|\right) - p_\lambda\left(|\beta_{k0}|\right)\right] = 0 \tag{A.15}$$

uniformly in any compact set of $\mathbb{R}^p$ due to the fact that for the SCAD penalty $p_\lambda(\theta)$, we have $\nabla p_\lambda(\theta) = 0$ for $\theta \in [a\lambda, +\infty)$, that is, $p_\lambda(\theta)$ is flat for coefficient of magnitude larger than $a\lambda$, and $\lambda \to 0$. By Condition (C2) and Eqs. (A.13), (A.14) and (A.15), $n\left[\widetilde{Q}_{SCAD}\left(\beta_0 + u/\sqrt{n}\right) - \widetilde{Q}_{SCAD}(\beta_0)\right]$ is dominated by the term $g(\tau)u^{\mathrm{T}}\Sigma u$ for $\|u\|_2$ equal to sufficiently large C. Thus, we have Eq. (A.12). It in turn implies that $\left\|\widehat{\beta}^{(SCAD)} - \beta_0\right\|_2 = O_p(n^{-\frac{1}{2}})$ as $n \to \infty$. Owing to $N = nm$, if $\lambda = \lambda(N) \to 0$, we have $\left\|\widehat{\beta}^{(SCAD)} - \beta_0\right\|_2 = O_p(N^{-\frac{1}{2}})$ as $N \to \infty$. This completes the proof.

Proof of Theorem 2: (a) As discussed in Fan and Li [6] and by Lemma 5, as $N = nm$, it is easy to show that, $\lambda = \lambda(N) \to 0$ and $\sqrt{N}\lambda \to \infty$ as $N \to \infty$, we have $\widehat{\beta}_2^{(SCAD)} = 0$. (b) Next we prove the asymptotic normality of $\widehat{\beta}_1^{(SCAD)}$. By Theorem 1, we can demonstrate that there exists a $\sqrt{N}$-consistent minimizer $\widehat{\beta}_1^{(SCAD)}$ of $\widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right)$ as a function of β_1.

The proof of Theorem 1 implies that $\sqrt{n}\left(\widehat{\beta}_1^{(SCAD)} - \beta_{10}\right)$ minimizes

$$H_n\left(\left(\theta^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) + n\sum_{k=1}^{q} p_\lambda\left(\left|\beta_{k0} + \theta_k/\sqrt{n}\right|\right) \tag{A.16}$$

with respect to θ, where $\theta = \left(\theta_1, \theta_2, \cdots, \theta_q\right)^{\mathrm{T}} \in \mathbb{R}^q$. By the convexity Lemma 1 and Lemma 3, we get

$$\begin{aligned} H_n\left(\left(\theta^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) &= g(\tau)\left(\theta^{\mathrm{T}}, 0^{\mathrm{T}}\right)\left[\frac{\sum_{i=1}^{n} x_{ji}x_{ji}^{\mathrm{T}}}{n}\right]\left(\theta^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}} \\ &\quad + \left(\theta^{\mathrm{T}}, 0^{\mathrm{T}}\right)W_n + o_p(1) \\ &= g(\tau)\theta^{\mathrm{T}}\left[\frac{\sum_{i=1}^{n} x_{ji}^1\left(x_{ji}^1\right)^{\mathrm{T}}}{n}\right]\theta \\ &\quad + \left(W_n^1\right)^{\mathrm{T}}\theta + o_p(1) \end{aligned} \tag{A.17}$$

uniformly in any compact subset of $\mathbb{R}^q$, where $W_n^1 = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} D_i^1$, and $D_i^1 = \xi_i^1 - \eta_i^1/m - \zeta_i^1$ with $\xi_i^1 = \varphi_\tau\left(\overline{\varepsilon}_{ji}\right)x_{ji}^1$, $\eta_i = \sum_{j=1}^{m}\varphi_\tau\left(\overline{\varepsilon}_{ji}\right)x_{ji}^1$, $\zeta_i = \varphi_\tau\left(\varepsilon_{ji}\right)x_{ji}^1$, and similar proof of Lemma 4, we have

$$W_n^1 \xrightarrow{d} \mathcal{N}\left(0, m^{-1}d(\tau)\Sigma_{11}\right). \tag{A.18}$$

Notice that, for large n, under condition $\lambda = \lambda(n) \to 0$, we have

$$n\sum_{k=1}^{q} p_\lambda\left(\left|\beta_{k0} + \theta_k/\sqrt{n}\right|\right) = n\sum_{k=1}^{q} p_\lambda(|\beta_{k0}|) \tag{A.19}$$

uniformly in any compact set of $\mathbb{R}^q$, and this term does not depend on the parameter θ. By Eqs. (A.16), (A.17) and (A.19) and Condition (C2), we get

$$\begin{aligned} &H_n\left(\left(\theta^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) + n\sum_{k=1}^{q} p_\lambda\left(\left|\beta_{k0} + \theta_k/\sqrt{n}\right|\right) \\ &\quad = g(\tau)\theta^{\mathrm{T}}\Sigma_{11}\theta + \left(W_n^1\right)^{\mathrm{T}}\theta + n\sum_{k=1}^{q} p_\lambda(|\beta_{k0}|). \end{aligned} \tag{A.20}$$

By Lemma 2, and Eq. (A.18), it can obtain that

$$\sqrt{n}\left(\widehat{\beta}_1^{(SCAD)} - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{d(\tau)}{4mg^2(\tau)}\Sigma_{11}^{-1}\right). \tag{A.21}$$

Due to $N = nm$, Eq (A.21) and Slutsky's theorem, we have

$$\sqrt{N}\left(\widehat{\beta}_1^{(SCAD)} - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{d(\tau)}{4g^2(\tau)}\Sigma_{11}^{-1}\right). \tag{A.22}$$

This completes the proof.

Proof of Theorem 3: (a) For any $\|\beta_1 - \beta_{10}\|_2 = O_p\left(n^{-\frac{1}{2}}\right)$, $\|\beta_2\| \leq Cn^{-\frac{1}{2}}$, similarly to the derivative process of Eq. (A.9), it obtain that

$$\begin{aligned} &n\left[\widetilde{Q}_{AL}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) - \widetilde{Q}_{AL}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right)\right] \\ &= H_n\left(\sqrt{n}\left(\left(\beta_1 - \beta_{10}\right)^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) \\ &\quad - H_n\left(\sqrt{n}\left(\left(\beta_1 - \beta_{10}\right)^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right) \\ &\quad - n\lambda\sum_{k=q+1}^{p}\overline{w}_k\left|\beta_k\right|. \end{aligned} \tag{A.23}$$

Note here the first two terms of Eq. (A.23) are exactly the same as in Eq. (A.9) and hence can be bounded similarly.

However the third term

$$-n\lambda \sum_{k=q+1}^{p} \overline{w}_k \left|\beta_k\right| = -\left[n^{(1+r)/2}\lambda\right]\sqrt{n} \times \left[\sum_{k=q+1}^{p}\left|\left(\sqrt{n}\left|\widehat{\beta}_k\right|\right)^{-r}\right|\left|\beta_k\right|\right] \to -\infty$$

owing to $n^{(1+r)/2}\lambda \to \infty$ and $\sqrt{n}\widehat{\beta}_k = O_p\left(1\right)$.

These facts in turn implies that

$$n\left[\widetilde{Q}_{AL}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}\right) - \widetilde{Q}_{AL}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right)\right] < 0$$

for $n \to \infty$. Then we have

$$\mathrm{P}\left[\inf_{\|\beta_2\|_2 = Cn^{-\frac{1}{2}}} \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, \beta_2^{\mathrm{T}}\right)^{\mathrm{T}}\right) > \widetilde{Q}_{SCAD}\left(\left(\beta_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)\right)^{\mathrm{T}}\right] \geq 1 - \delta.$$

Owing to $N = nm$ and the conditions $\lambda = \lambda\left(N\right)$, $\sqrt{N}\lambda \to 0$ and $N^{(r+1)/2} \to \infty$ as $N \to \infty$, it can obtain that $\widehat{\beta}_2^{(AL)} = 0$.

(b) Similarly to Eq. (A.13), by performing some simple calculation, we obtain that

$$\begin{aligned} n&\left[\widetilde{Q}_{AL}\left(\beta_0 + u/\sqrt{n}\right) - \widetilde{Q}_{AL}(\beta_0)\right] \\ &= \sum_{i=1}^{n}\left[\rho_\tau\left(\epsilon_{ji} - x_{ji}^{\mathrm{T}}u/\sqrt{n}\right) - \rho_\tau(\epsilon_{ji})\right] \\ &\quad + n\left[\langle \nabla Q_N\left(\overline{\beta}\right) - \nabla Q_j\left(\overline{\beta}\right), u/\sqrt{n}\rangle\right] \\ &\quad + n\lambda\sum_{k=1}^{p}\left[\overline{w}_k\left|\beta_{k0} + u_k/\sqrt{n}\right| - \overline{w}_k\left|\beta_{k0}\right|\right]. \quad \text{(A.24)} \end{aligned}$$

We consider the third term of Eq. (A.24) first. For $k = 1, 2, \cdots, q$, we have $\beta_{k0} \neq 0$; as a result, $\overline{w}_k \xrightarrow{\mathrm{P}} \left|\beta_{k0}\right|^{-r}$, hence

$$n\lambda\left[\overline{w}_k\left|\beta_{k0} + u_k/\sqrt{n}\right| - \overline{w}_k\left|\beta_{k0}\right|\right] \xrightarrow{\mathrm{P}} 0 \qquad \text{(A.25)}$$

as $\sqrt{n}\lambda \to 0$ and $\sqrt{n}\left(\left|\beta_{k0} + u_k/\sqrt{n}\right| - \left|\beta_{k0}\right|\right) \to u_k \mathrm{sgn}\left|\beta_{k0}\right|$. For $k = q+1, q+2, \cdots, p$, the true coefficient $\beta_{k0} = 0$; so

$$n\lambda\left[\overline{w}_k\left|\beta_{k0}+u_k/\sqrt{n}\right| - \overline{w}_k\left|\beta_{k0}\right|\right] = \sqrt{n}\lambda\left[\overline{w}_k\left|u_k\right|\right] \to \infty, \qquad \text{(A.26)}$$

where $u_k \neq 0$ and $= 0$ otherwise due to $\sqrt{n}\lambda\overline{w}_k = n^{(1+r)/2}\lambda\left(\sqrt{n}\left|\widehat{\beta}_k\right|\right)^{-r}$ with $\sqrt{n}\widehat{\beta}_k = O_p\left(1\right)$, $n^{(1+r)/2}\lambda \to \infty$. By Eqs. (A.24), (A.25), (A.26) and Lemma 3 and Condition (C2), we have

$$\begin{aligned} n&\left[\widetilde{Q}_{AL}\left(\beta_0 + u/\sqrt{n}\right) - \widetilde{Q}_{AL}(\beta_0)\right] \\ &\xrightarrow{d} V\left(u\right) = \begin{cases} g(\tau)u^1\Sigma_{11}u^1 + \left(W_n^1\right)^{\mathrm{T}}u^1, & \\ \qquad \text{when } u_k = 0 \quad \text{for } k = q+1, \cdots, p, & \\ \infty, \quad \text{otherwise,} & \end{cases} \qquad \text{(A.27)} \end{aligned}$$

where $u^1 = \left(u_1, u_2, \cdots, u_q\right)^{\mathrm{T}}$. Notice that $n[\widetilde{Q}_{AL}(\beta_0 + u/\sqrt{n}) - \widetilde{Q}_{AL}(\beta_0)]$ is convex in u and $V\left(u\right)$ has a unique minimizer, then we have

$$\begin{aligned} &\arg\min_{u\in\mathbb{R}^q} n\left[\widetilde{Q}_{AL}\left(\beta_0 + u/\sqrt{n}\right) - \widetilde{Q}_{AL}(\beta_0)\right] \\ &\quad = \sqrt{n}\left(\widehat{\beta}^{(AL)} - \beta_0\right) \xrightarrow{d} \arg\min_{u\in\mathbb{R}^q} V\left(u\right). \qquad \text{(A.28)} \end{aligned}$$

By Eqs. (A.18), (A.27), (A.28) and Lemma 2, we obtain

$$\sqrt{n}\left(\widehat{\beta}_1^{(AL)} - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{d\left(\tau\right)}{4mg^2\left(\tau\right)}\Sigma_{11}^{-1}\right). \qquad \text{(A.29)}$$

By (A.28), when $\sqrt{N}\lambda \to 0$, and $N^{(r+1)/2} \to \infty$ as $N \to \infty$, we have

$$\sqrt{N}\left(\widehat{\beta}_1^{(AL)} - \beta_{10}\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{d\left(\tau\right)}{4g^2\left(\tau\right)}\Sigma_{11}^{-1}\right). \qquad \text{(A.30)}$$

This completes the proof.

REFERENCES

[1] D. J. Aigner, T. Amemiya, and D. J. Poirier, "On the estimation of production frontiers: Maximum likelihood estimation of the parameters of a discontinuous density function," *Int. Econ. Rev.*, vol. 17, no. 2, pp. 377–396, 1976.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[3] G. Cheng and Z. Shang, "Computational limits of divide- and-conquer method," 2015, *arxiv:1512.09226*. [Online]. Available: https://arxiv.org/abs/1512.09226

[4] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[5] J. Eckstein, "Some saddle-function splitting methods for convex programming," *Optim. Methods Softw.*, vol. 4, no. 1, pp. 75–83, Jan. 1994.

[6] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Stat. Assoc.*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[7] B. He, L.-Z. Liao, D. Han, and H. Yang, "A new inexact alternating directions method for monotone variational inequalities," *Math. Program.*, vol. 92, no. 1, pp. 103–118, 2002.

[8] N. Hjort and D. Pollard, "Asymtotics for minimisers of convex process," *Statist. Theory*, vol. 7, no. 1, 1993.

[9] C. Huang and X. Huo, "A distributed one-step estimator," 2015, *arXiv:1511.01443*. [Online]. Available: http://arxiv.org/abs/1511.01443

[10] M. C. Jones, "Expectiles and M-quantiles are quantiles," *Statist. Probab. Lett.*, vol. 20, no. 2, pp. 149–153, May 1994.

[11] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical learning," *J. Amer. Stat. Assoc.*, vol. 114, no. 526, pp. 668–681, 2019.

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

[13] Y. Li and J. Zhu, "L1-norm quantile regression," *J. Comput. Graph. Statist.*, vol. 17, no. 1, pp. 163–185, Mar. 2008.

[14] L. Liao, C. Park, and H. Choi, "Penalized expectile regression: An alternative to penalized quantile regression," *Ann. Inst. Stat. Math.*, vol. 71, no. 2, pp. 409–438, Apr. 2019.

[15] J. Lu, G. Cheng, and H. Liu, "Nonparametric heterogeneity testing for massive data," 2016, *arXiv:1601.06212*. [Online]. Available: http://arxiv.org/abs/1601.06212

[16] R. Mcdonald, M. Mohri, N. Silberman, D. D. Walker, and G. S. Mann, "Efficient large-scale distributed training of conditional maximum entropy models," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 1231–1239.

[17] W. K. Newey and J. L. Powell, "Asymmetric least squares estimation and testing," *Econometrica*, vol. 55, no. 4, pp. 819–847, 1987.

[18] Y. Pan, "Distributed optimization and statistical learning for large-scale penalized expectile regression," *J. Korean Stat. Soc.*, vol. 50, no. 1, pp. 290–314, Mar. 2021.

[19] Y. Pan, Z. Liu, and W. Cai, "Large-scale expectile regression with covariates missing at random," *IEEE Access*, vol. 8, pp. 36502–36513, 2020.

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

[21] J. D. Rosenblatt and B. Nadler, "On the optimality of averaging in distributed statistical learning," *Inf. Inference*, vol. 5, no. 4, pp. 379–404, Dec. 2016.

[22] F. Sobotka and T. Kneib, "Geoadditive expectile regression," *Comput. Statist. Data Anal.*, vol. 56, no. 4, pp. 755–767, Apr. 2012.

[23] F. Sobotka, R. Radice, G. Marra, and T. Kneib, "Estimating the relationship between women's education and fertility in Botswana by using an instrumental variable approach to semiparametric expectile regression," *J. Roy. Stat. Soc., Ser. C (Appl. Statist.)*, vol. 62, no. 1, pp. 25–45, Jan. 2013.

[24] L. S. Waltrup, F. Sobotka, T. Kneib, and G. Kauermann, "Expectile and quantile regression—David and goliath?" *Stat. Model., Int. J.*, vol. 15, no. 5, pp. 433–456, Oct. 2015.

[25] J. Wang, M. Kolar, and N. Srerbo, "Distributed multi-task learning," in *Proc. Conf. Artif. Intell. Statictics*, Cadiz, Spain, 2016, pp. 751–760.

[26] J. Wang, M. Kolar, N. Srebro, and T. Zhang, "Efficient distributed learning with sparsity," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3636–3645.

[27] L. Wang, Y. Wu, and R. Li, "Quantile regression for analyzing heterogeneity in ultra-high dimension," *J. Amer. Stat. Assoc.*, vol. 107, no. 497, pp. 214–222, Mar. 2012.

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

[29] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 3321–3363, 2013.

[30] J. Zhao and Y. Zhang, "Variable selection in expectile regression," *Commun. Statist.-Theory Methods*, vol. 47, no. 7, pp. 1731–1746, Apr. 2018.

[31] H. Zou, "The adaptive lasso and its oracle properties," *J. Amer. Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, Dec. 2006.

[32] H. Zou and R. Li, "One-step sparse estimates in nonconcave penalized likelihood models," *Ann. Statist.*, vol. 36, no. 4, p. 1509, Aug. 2008.

YINGLI PAN received the B.S. degree in mathematics from Henan Normal University, Henan, China, in 2011, the M.S. degree in mathematics from Central China Normal University, Wuhan, China, in 2014, and the Ph.D. degree in statistics from the Huazhong University of Science and Technology, Wuhan, in 2018.

Since July 2018, he has been working with the School of Mathematics and Statistics, Hubei University. His research interests include survival analysis, data analysis and statistical calculation, and distributed optimization method.

ZHAN LIU received the B.S. and M.S. degrees in mathematics from Hubei University, Wuhan, China, in 2004 and 2007 respectively, and the Ph.D. degree in statistics from the Renmin University of China, Beijing, China, in 2017.

From August 2016 to March 2017, she was a Visiting Ph.D. Student with the JPSM, University of Maryland, MD, USA. Since July 2017, she has been working as an Associate Professor with the School of Mathematics and Statistics, Hubei University. From May 2018 to August 2018 and December 2018 to February 2019, she worked as a Research Associate with the Department of Statistics, The Chinese University of Hong Kong, Hong Kong. Her research interests include sampling inference, missing data, and distributed optimization method.