Statistica Sinica **33** (2023), 1143-1167
doi:https://doi.org/10.5705/ss.202022.0095

DISTRIBUTED SPARSE COMPOSITE QUANTILE REGRESSION IN ULTRAHIGH DIMENSIONS

Canyi Chen[1], Yuwen Gu[2], Hui Zou[3] and Liping Zhu[1]

[1]*Renmin University of China,* [2]*University of Connecticut and* [3]*University of Minnesota*

Abstract: We examine distributed estimation and support recovery for ultrahigh-dimensional linear regression models under a potentially arbitrary noise distribution. The composite quantile regression is an efficient alternative to the least squares method, and provides robustness against heavy-tailed noise while maintaining reasonable efficiency in the case of light-tailed noise. The highly nonsmooth nature of the composite quantile regression loss poses challenges to both the theoretical and the computational development in an ultrahigh-dimensional distributed estimation setting. Thus, we cast the composite quantile regression into the least squares framework, and propose a distributed algorithm based on an approximate Newton method. This algorithm is efficient in terms of both computation and communication, and requires only gradient information to be communicated between the machines. We show that the resultant distributed estimator attains a near-oracle rate after a constant number of communications, and provide theoretical guarantees for its estimation and support recovery accuracy. Extensive experiments demonstrate the competitive empirical performance of our algorithm.

Key words and phrases: Composite quantile regression, distributed estimation, efficiency, heavy-tailed noise, support recovery.

1. Introduction

A fundamental task in statistics is to estimate the coefficients of a linear model. The least squares (LS) regression is routinely used for this task, and has well-established theory (Monahan (2008)). However, in the era of big data, rapid advances in information technology have raised several new challenges. The first lies in the sizes of the data sets, often measured in TBs or even PBs, making them difficult to process on a single machine. Traditional in-memory algorithms are losing power because of communication, storage, and computation restrictions (Lan, Lee and Zhou (2020)). Thus, we need distributed algorithms with theoretical guarantees. The second challenge arises from the potentially arbitrary noise.

Corresponding author: Liping Zhu, Institute of Statistics and Big Data and Center for Applied Statistics, Renmin University of China, Beijing 100872, China. E-mail: zhu.liping@ruc.edu.cn.

Under very heavy-tailed noise, where the finite variance condition is violated, the LS and Huber regressions are sub-optimal (Fan, Fan and Barut (2014); Zhou et al. (2018); Sun, Zhou and Fan (2020)). In such cases, the quantile regression (QR, Koenker (2005)) becomes an attractive alternative, because its asymptotic variance does not depend on the moments of the noise distribution. However, in terms of efficiency, a QR can be arbitrarily less efficient than an LS. For example, under the mixture normal noise $0.5\mathcal{N}(-3,1)+0.5\mathcal{N}(3,1)$, the least absolute deviation estimate is 1,272.8 times less efficient than the LS estimate (Gu and Zou (2020)). To shield the QR against potential efficiency loss, while maintaining its robust property, Zou and Yuan (2008) proposed a composite quantile regression (CQR) that combines quantile information across various quantile levels. The third challenge lies in the ultrahigh dimensionality of modern data. Here, a sparsity assumption is often adopted (Zhao and Yu (2006); Wainwright (2009); Hastie, Tibshirani and Wainwright (2015)). Despite the massive amount of literature on sparse LS under ultrahigh dimensions, few works have examined the ultrahigh-dimensional CQR; see Gu and Zou (2020). In a distributed setting, numerous studies focus on statistical estimation (Lee et al. (2017); Battey et al. (2018); Jordan, Lee and Yang (2019)). However, the *support recovery* for the CQR in a distributed setting remains largely unexplored.

We propose a new estimation procedure for an ultrahigh-dimensional CQR in the distributed setting, with theoretical guarantees on its estimation and support recovery accuracy. Specifically, we consider coefficient estimation and support recovery of the following model:

$$Y=\beta_0+\sum_{j=1}^{p}X_j\beta_j+\varepsilon, \tag{1.1}$$

where $\mathbf{x}=(X_1,\ldots,X_p)^{\mathrm{T}}$ is a p-dimensional covariate vector with mean zero, and ε is the noise. We assume ε is independent of $\mathbf{x}$ and has a density with respect to the Lebesgue measure (see, e.g., Zou and Yuan (2008); Fan, Fan and Barut (2014); Gu and Zou (2020)). Suppose β_0^* and $\boldsymbol{\beta}^*=(\beta_1^*,\ldots,\beta_p^*)^{\mathrm{T}}$ are the true coefficients that generate the N independent and identically distributed (i.i.d.) data $(\mathbf{x}_i,Y_i)_{i=1}^N$, where $\mathbf{x}_i=(X_{i1},\ldots,X_{ip})^{\mathrm{T}}$. Denote the response vector by $\mathbf{y}=(Y_1,\ldots,Y_N)^{\mathrm{T}}$, and the design matrix by $\mathbf{X}=(\mathbf{x}_1,\ldots,\mathbf{x}_N)^{\mathrm{T}}$. We assume a sparsity structure on $\boldsymbol{\beta}^*$ in the sense that only $s<p$ elements of $\boldsymbol{\beta}^*$ are nonzero.

We consider the CQR for estimating $\boldsymbol{\beta}^*$ in model (1.1) that is robust to heavy-tailed errors, while maintaining reasonable efficiency under light-tailed errors. Denote $F(\cdot)$ and $f(\cdot)$ as the cumulative distribution and the probability

density functions, respectively, of ε. To ensure the identifiability of β_0^*, assume $F(0) = 0.5$. Given an ordered sequence of quantile levels $\tau_1 < \tau_2 < \cdots < \tau_K \in (0,1)$, let $\alpha_k^* \stackrel{\text{def}}{=} \beta_0^* + F^{-1}(\tau_k)$ and $\boldsymbol{\alpha}^* \stackrel{\text{def}}{=} (\alpha_1^*, \ldots, \alpha_K^*)^{\mathrm{T}} \in \mathbb{R}^K$, where $F^{-1}(\tau_k) = \inf\{x\colon F(x) \geq \tau_k\}$ denotes the τ_kth quantile of ε, for $k = 1, \ldots, K$. The canonical CQR (Zou and Yuan (2008)) estimates $\boldsymbol{\beta}^*$ by minimizing

$$Q(\alpha, \beta; \mathbf{x}, Y) \stackrel{\text{def}}{=} \frac{1}{K} \sum_{k=1}^{K} \rho_{\tau_k}(Y - \alpha_k - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) \tag{1.2}$$

jointly over $\alpha = (\alpha_1, \ldots, \alpha_K)^{\mathrm{T}} \in \mathbb{R}^K$ and $\boldsymbol{\beta} \in \mathbb{R}^p$, where $\rho_{\tau_k}(u) \stackrel{\text{def}}{=} \{\tau_k - I(u < 0)\}u$ is the check loss at level τ_k, for $k = 1, \ldots, K$ (Koenker (2005)). It is easy to see that

$$(\boldsymbol{\alpha}^{*\mathrm{T}}, \boldsymbol{\beta}^{*\mathrm{T}})^{\mathrm{T}} = \operatorname*{argmin}_{\boldsymbol{\alpha}, \boldsymbol{\beta}} E\{Q(\alpha, \beta; \mathbf{x}, Y)\}.$$

Typically, we take equally spaced τ_k: $\tau_k = k/(K+1)$, for $k = 1, \ldots, K$. As $K \to \infty$, Zou and Yuan (2008) show that the asymptotic efficiency of the CQR relative to the LS has a universal lower bound of 86.4% (Kai, Li and Zou (2010)). Even with a relatively small K, such as $K = 9$, the CQR estimator achieves a substantial efficiency gain.

Although using the CQR provides robustness to heavy-tailed noise and safeguards against potential efficiency loss, the nonsmoothness of the CQR loss raises computational challenges, owing to limited computing power and memory when the sample size and dimension are both considerable. Doing so also makes theoretical developments difficult. Recently, Gu and Zou (2020) developed the theory for ultrahigh-dimensional sparse penalized CQR in a single-node setting. In addition, to consider scalability, they proposed using an alternating direction method of multipliers algorithm, rather than the linear programming algorithm considered in Zou and Yuan (2008). For ultrahigh-dimensional data stored on multiple machines, we may not be able to use existing algorithms for the sparse penalized CQR, making distributed algorithms with theoretical guarantees increasingly important. The main goals of this study are to develop a new distributed estimation approach for the ultrahigh-dimensional CQR, and to establish its *estimation* and *support recovery* theory.

There are two main data-partitioning schemes in distributed systems: "horizontal" and "vertical." In a "horizontal" distributed setting, assume N observations are scattered evenly across m local nodes, with n observations at each node. Denote by $\mathcal{H}_j$ the set of observational indices at the jth node. The divide-and-conquer strategy is popular for such data, owing to its simplicity (Zaharia et al.

(2016)); see, for example, Li, Lin and Li (2012), Zhao, Cheng and Liu (2016), Battey et al. (2018), Shi, Lu and Song (2018), Jiang et al. (2018), Fan, Guo and Wang (2021) and Fan et al. (2019). However, the final estimate, which is an average of the m local estimates, is usually no longer sparse. Moreover, although averaging reduces the variance of the local estimates, it might not remove the bias of these local estimates. Hence, restrictions on the number of nodes, for example, $m = O(N^{1/2})$, are routinely imposed to achieve the minimax convergence rate (Braverman et al. (2016)). To remove such restrictions on m, Wang et al. (2017) and Jordan, Lee and Yang (2019) developed multi-round procedures. However, their methodology and theory require second-order differentiability of the loss function, in general, and cannot be applied directly to the highly *non-smooth* CQR loss. Chen et al. (2020) studied the distributed high-dimensional QR problem, providing theoretical guarantees on its support recovery. However, their method cannot safeguard against the potential efficiency loss incurred by the QR. In a "vertical" distributed setting, each local node holds a subset of all the covariates of the data. To recover the sparsity pattern, He, Zhou and Feng (2022) proposed decorrelating the covariates before aggregating. Their work improves on previous results (Zhou et al. (2014); Song and Liang (2015)) by requiring constraints on the correlation structure of the covariates that are less strict.

Our work focuses on the "horizontal" distributed setting. We propose a new distributed procedure for estimating the coefficients of a high-dimensional linear model, with potentially arbitrary noise, using the CQR. We first show that we can estimate $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ from a pseudo-response $\widetilde{Y}_i$, instead of Y_i. This yields a pooled estimate from solving a lasso problem based on $(\mathbf{x}_i, \widetilde{Y}_i)_{i=1}^N$, without any moment conditions on the noise term. The pooled estimate is computationally much more efficient than the penalized CQR in a single-node setting. We further provide a communication-efficient distributed implementation of this pooled estimate where, at each communication, only the $(p + K)$-dimensional gradient information is communicated, instead of the $(p+K) \times (p+K)$ Hessian matrix, by modifying the approximate Newton method of Shamir, Srebro and Zhang (2014). Our results demonstrate the accuracy of our method in terms of both estimation and support recovery. We prove that, after a constant number of communications, our estimate achieves a near-oracle rate of $[s \log\{\max(p, N)\}/N]^{1/2}$ for the estimation of $\boldsymbol{\beta}^*$, and $[Ks \log\{\max(p, N)\}/N]^{1/2}$ for that of $\boldsymbol{\alpha}^*$, in terms of the ℓ_2-error (Theorem 2). These rates coincide with those of the ℓ_1-regularized CQR in a single-node setting (Gu and Zou (2020)). We also derive the "beta-min" condition for exact support recovery, which becomes weaker as the number of communications increases (Theorem 4). After a constant number of communica-

tions, our "beta-min" condition matches that of the classical setting in which all data are in a single node.

The rest of this paper is organized as follows. We describe the distributed algorithm for the penalized CQR in Section 2, and derive the estimation error bounds and the support recovery results for the distributed estimator in Section 3. Extensive simulations in Section 4 provide empirical evidence for our theoretical findings. Section 5 concludes the paper. All technical proofs are relegated to the online Supplementary Material.

We use the following notation. We denote $C, C_0, C_1, \ldots, c, c_0, c_1, \ldots$ as generic constants that may vary at each appearance. We also use the standard asymptotic notation. Given two sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n = O(b_n)$ if there exists a constant $C < \infty$ such that $a_n \le Cb_n$, and $a_n = o(b_n)$ if $a_n/b_n \to 0$. For two sets of random variables $\{X_n\}$ and $\{Y_n\}$, we write $X_n = O_p(Y_n)$ if for any $\epsilon > 0$, there exists a finite $M > 0$ and a finite $n_0 > 0$ such that $\mathrm{pr}(|X_n/Y_n| > M) < \epsilon$, for any $n > n_0$. For a vector $\mathbf{v} = (v_1, \ldots, v_p)^{\mathrm{T}}$, we denote its support by $\mathrm{supp}(\mathbf{v}) \stackrel{\mathrm{def}}{=} \{j \in \mathbb{N}\colon v_j \neq 0\}$. We further define $|\mathbf{v}|_1 \stackrel{\mathrm{def}}{=} \sum_{i=1}^p |v_i|$, $|\mathbf{v}|_2 \stackrel{\mathrm{def}}{=} \big(\sum_{i=1}^p v_i^2\big)^{1/2}$, and $\mathbf{v}^{\min} \stackrel{\mathrm{def}}{=} \min_{i \in \mathrm{supp}(\mathbf{v})} |v_i|$. For $\mathcal{S} \subseteq \{1, \ldots, p\}$ with length $|\mathcal{S}|$, let $\mathbf{v}_{\mathcal{S}} \stackrel{\mathrm{def}}{=} (v_i, i \in \mathcal{S}) \in \mathbb{R}^{|\mathcal{S}|}$. For a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{p \times q}$, we define $|\mathbf{A}|_\infty \stackrel{\mathrm{def}}{=} \max_{1 \le i \le p, 1 \le j \le q} |a_{ij}|$, $\|\mathbf{A}\|_\infty \stackrel{\mathrm{def}}{=} \max_{1 \le i \le p} \sum_{1 \le j \le q} |a_{ij}|$, and $\|\mathbf{A}\|_{\mathrm{op}} \stackrel{\mathrm{def}}{=} \max_{|\mathbf{v}|_2 = 1} |\mathbf{A}\mathbf{v}|_2$. For two subsets $\mathcal{S}_1 \subseteq \{1, \ldots, p\}$ and $\mathcal{S}_2 \subseteq \{1, \ldots, q\}$, we let $\mathbf{A}_{\mathcal{S}_1 \times \mathcal{S}_2} = (a_{ij}, i \in \mathcal{S}_1, j \in \mathcal{S}_2)$. Finally, denote the largest and smallest singular values of $\mathbf{A}$ by $\Lambda_{\max}(\mathbf{A})$ and $\Lambda_{\min}(\mathbf{A})$, respectively.

2. Distributed Sparse CQR

2.1. The Newton update and a surrogate loss

Motivated by the Newton–Raphson method, we cast the CQR as an LS problem. Let $\boldsymbol{\phi} \stackrel{\mathrm{def}}{=} (\boldsymbol{\alpha}^{\mathrm{T}}, \boldsymbol{\beta}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{p+K}$ and $\boldsymbol{g}(\boldsymbol{\phi}; \mathbf{x}, Y) \stackrel{\mathrm{def}}{=} \{\boldsymbol{g}_{\boldsymbol{\alpha}}(\boldsymbol{\phi}; \mathbf{x}, Y)^{\mathrm{T}}, \boldsymbol{g}_{\boldsymbol{\beta}}(\boldsymbol{\phi}; \mathbf{x}, Y)^{\mathrm{T}}\}^{\mathrm{T}}$, where $\boldsymbol{g}_{\boldsymbol{\alpha}}(\boldsymbol{\phi}; \mathbf{x}, Y) \in \mathbb{R}^K$ and $\boldsymbol{g}_{\boldsymbol{\beta}}(\boldsymbol{\phi}; \mathbf{x}, Y) \in \mathbb{R}^p$ are the subgradients of the loss function $Q(\boldsymbol{\phi}; \mathbf{x}, Y)$ with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. Moreover, let $\boldsymbol{H}(\boldsymbol{\phi}) \stackrel{\mathrm{def}}{=} \partial E\{\boldsymbol{g}(\boldsymbol{\phi}; \mathbf{x}, Y)\}/\partial \boldsymbol{\phi}^{\mathrm{T}} \in \mathbb{R}^{(p+K) \times (p+K)}$ denote the population Hessian matrix of $E\{Q(\boldsymbol{\phi}; \mathbf{x}, Y)\}$. Given any initial solution $\boldsymbol{\phi}_0 \stackrel{\mathrm{def}}{=} (\boldsymbol{\alpha}_0^{\mathrm{T}}, \boldsymbol{\beta}_0^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{p+K}$, the population version of the Newton–Raphson iteration has the form

$$\boldsymbol{\phi}_1 \stackrel{\mathrm{def}}{=} (\boldsymbol{\alpha}_1^{\mathrm{T}}, \boldsymbol{\beta}_1^{\mathrm{T}})^{\mathrm{T}} = \boldsymbol{\phi}_0 - \boldsymbol{H}(\boldsymbol{\phi}_0)^{-1} E\{\boldsymbol{g}(\boldsymbol{\phi}_0; \mathbf{x}, Y)\}. \tag{2.1}$$

For the CQR loss $Q(\boldsymbol{\phi}; \mathbf{x}, Y)$ in (1.2), we consider a subgradient of the form $\boldsymbol{g}_{\boldsymbol{\alpha}}(\boldsymbol{\phi}; \mathbf{x}, Y) = \{I(Y - \alpha_1 - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} \le 0) - \tau_1, \ldots, I(Y - \alpha_K - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} \le 0) - \tau_K\}^{\mathrm{T}}/K$

and $\boldsymbol{g}_{\boldsymbol{\beta}}(\boldsymbol{\phi};\mathbf{x},Y)=\sum_{k=1}^{K}\mathbf{x}\{I(Y-\alpha_k-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}\le 0)-\tau_k\}/K$. Note that the Hessian matrix takes the form

$$\boldsymbol{H}(\boldsymbol{\phi})=\begin{pmatrix} \frac{\partial E\{\boldsymbol{g}_{\boldsymbol{\alpha}}(\boldsymbol{\phi};\mathbf{x},Y)\}}{\partial\boldsymbol{\alpha}^{\mathrm{T}}} & \frac{\partial E\{\boldsymbol{g}_{\boldsymbol{\alpha}}(\boldsymbol{\phi};\mathbf{x},Y)\}}{\partial\boldsymbol{\beta}^{\mathrm{T}}} \\ \frac{\partial E\{\boldsymbol{g}_{\boldsymbol{\beta}}(\boldsymbol{\phi};\mathbf{x},Y)\}}{\partial\boldsymbol{\alpha}^{\mathrm{T}}} & \frac{\partial E\{\boldsymbol{g}_{\boldsymbol{\beta}}(\boldsymbol{\phi};\mathbf{x},Y)\}}{\partial\boldsymbol{\beta}^{\mathrm{T}}} \end{pmatrix}_{(p+K)\times(p+K)},$$

where

$$\begin{aligned}
\frac{\partial E\{\boldsymbol{g}_{\boldsymbol{\alpha}}(\boldsymbol{\phi};\mathbf{x},Y)\}}{\partial\alpha^{\mathrm{T}}} &= \frac{1}{K}\mathrm{diag}(f\{\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)+\alpha_1\},\ldots,f\{\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)+\alpha_K\}),\\
\frac{\partial E\{\boldsymbol{g}_{\boldsymbol{\beta}}(\boldsymbol{\phi};\mathbf{x},Y)\}}{\partial\beta^{\mathrm{T}}} &= \frac{1}{K}\sum_{k=1}^{K}E[\mathbf{x}\mathbf{x}^{\mathrm{T}}f\{\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)+\alpha_k\}],\ \text{and}\\
\frac{\partial E\{\boldsymbol{g}_{\boldsymbol{\alpha}}(\boldsymbol{\phi};\mathbf{x},Y)\}}{\partial\beta^{\mathrm{T}}} &= \frac{1}{K}\{E(\mathbf{x}f\{\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)+\alpha_1\}),\ldots,\\
&\quad E(\mathbf{x}f\{\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta}-\boldsymbol{\beta}^*)+\alpha_K\})\}^{\mathrm{T}}.
\end{aligned}$$

When the initial estimate $\boldsymbol{\phi}_0$ is close to the true parameter $\boldsymbol{\phi}^*\overset{\text{def}}{=}(\boldsymbol{\alpha}^{*\mathrm{T}},\boldsymbol{\beta}^{*\mathrm{T}})^{\mathrm{T}}$, $\boldsymbol{H}(\boldsymbol{\phi}_0)$ is approximately

$$\boldsymbol{H}(\boldsymbol{\phi}^*)=\frac{1}{K}\left(\begin{array}{ccc|c} f(\alpha_1^*) & & & \\ & \ddots & & \\ & & f(\alpha_K^*) & \\ \hline & & & \sum_{k=1}^{K}f(\alpha_k^*)\Sigma \end{array}\right),$$

where $\Sigma=E(\mathbf{x}\mathbf{x}^{\mathrm{T}})$, and the zero entries are left blank. Replacing $\boldsymbol{H}(\boldsymbol{\phi}_0)$ with $\boldsymbol{H}(\boldsymbol{\phi}^*)$ in (2.1) results in the following iteration:

$$\boldsymbol{\phi}_1=\boldsymbol{\phi}_0-\boldsymbol{H}(\boldsymbol{\phi}^*)^{-1}E\{\boldsymbol{g}(\boldsymbol{\phi}_0;\mathbf{x},Y)\}. \tag{2.2}$$

This, together with the Taylor expansion of $E\{g(\boldsymbol{\phi}_0;\mathbf{x},Y)\}$ around $\boldsymbol{\phi}^*$,

$$E\{g(\boldsymbol{\phi}_0;\mathbf{x},Y)\}=\boldsymbol{H}(\boldsymbol{\phi}^*)(\boldsymbol{\phi}_0-\boldsymbol{\phi}^*)+O(|\boldsymbol{\phi}_0-\boldsymbol{\phi}^*|_2^2),$$

ensures an improved convergence rate of $\boldsymbol{\phi}_1$ in the ℓ_2-norm; that is,

$$\begin{aligned}
|\boldsymbol{\phi}_1-\boldsymbol{\phi}^*|_2 &= |\boldsymbol{\phi}_0-\boldsymbol{H}(\boldsymbol{\phi}^*)^{-1}\{\boldsymbol{H}(\boldsymbol{\phi}^*)(\boldsymbol{\phi}_0-\boldsymbol{\phi}^*)+O(|\boldsymbol{\phi}_0-\boldsymbol{\phi}^*|_2^2)\}-\boldsymbol{\phi}^*|_2\\
&= O(|\boldsymbol{\phi}_0-\boldsymbol{\phi}^*|_2^2).
\end{aligned}$$

Therefore, by refining a consistent estimate $\boldsymbol{\phi}_0$ using the Newton–Raphson iteration (2.2), we obtain an improved estimate $\boldsymbol{\phi}_1$.

Next, we demonstrate how to cast the Newton–Raphson iteration of the CQR problem as an LS problem. Let $f(\boldsymbol{\alpha}^*) \stackrel{\text{def}}{=} \sum_{k=1}^K f(\alpha_k^*)/K$. Because $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are decoupled in the Newton–Raphson iteration (2.2), $\boldsymbol{\alpha}_1 = (\alpha_{1,1}, \ldots, \alpha_{1,K})^{\mathrm{T}}$ and $\boldsymbol{\beta}_1$ admit the explicit forms

$$\alpha_{1,k} = \alpha_{0,k} - f^{-1}(\alpha_k^*) E\{I(Y - \alpha_{0,k} - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 0) - \tau_k\}, \quad k = 1, \ldots, K, \tag{2.3}$$

and

$$\begin{aligned}\boldsymbol{\beta}_1 &= \boldsymbol{\beta}_0 - \Sigma^{-1} f^{-1}(\boldsymbol{\alpha}^*) E\{\boldsymbol{g}_{\boldsymbol{\beta}}(\boldsymbol{\phi}_0; \mathbf{x}, Y)\} \\ &= \Sigma^{-1} E\left\{\mathbf{x}\left(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 - f^{-1}(\boldsymbol{\alpha}^*)\left[\frac{1}{K}\sum_{k=1}^K \{I(Y - \alpha_{0,k} - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 0) - \tau_k\}\right]\right)\right\}.\end{aligned}$$

Define a pseudo response

$$\widetilde{Y} = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 - f^{-1}(\boldsymbol{\alpha}^*)\left[\frac{1}{K}\sum_{k=1}^K \{I(Y - \alpha_{0,k} - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \le 0) - \tau_k\}\right].$$

Then, $\boldsymbol{\beta}_1 = \Sigma^{-1} E(\mathbf{x}\widetilde{Y}) = \operatorname{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} E(\widetilde{Y} - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})^2$ is the LS regression coefficient of $\widetilde{Y}$ on $\mathbf{x}$. To encourage sparsity in the coefficient vector, we consider the following penalized LS problem:

$$\boldsymbol{\beta}_{1,\,\mathrm{pen}} = \operatorname*{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{2} E(\widetilde{Y} - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})^2 + P(\boldsymbol{\beta}, \lambda), \tag{2.4}$$

where $P(\cdot, \cdot)$ is a sparsity-inducing penalty. Examples of penalties include the ℓ_1-norm penalty (LASSO, Tibshirani (1996)), smoothly clipped absolute deviation penalty (SCAD, Fan and Li (2001)), and minimax concave penalty (MCP, Zhang (2010)); see Hastie, Tibshirani and Wainwright (2015) and the references therein for comprehensive reviews of recent developments. In the present context, we adopt the ℓ_1-norm penalty $P(\boldsymbol{\beta}, \lambda) \stackrel{\text{def}}{=} \lambda|\boldsymbol{\beta}|_1$, for ease of exposition. Then, (2.4) becomes

$$\boldsymbol{\beta}_{1,\,\ell_1} = \operatorname*{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{2} E(\widetilde{Y} - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})^2 + \lambda|\boldsymbol{\beta}|_1. \tag{2.5}$$

At the population level, if we have a consistent estimate $(\boldsymbol{\alpha}_0^{\mathrm{T}}, \boldsymbol{\beta}_0^{\mathrm{T}})^{\mathrm{T}}$ of $(\boldsymbol{\alpha}^{*\mathrm{T}}, \boldsymbol{\beta}^{*\mathrm{T}})^{\mathrm{T}}$, then we can estimate $\boldsymbol{\alpha}^*$ and the ultrahigh-dimensional sparse $\boldsymbol{\beta}^*$ by solving a simple iteration (2.3) and a penalized LS problem (2.5), rather than solving the original penalized CQR.

Now, we define the empirical version of $\boldsymbol{\beta}_{1,\,\ell_1}$ in a single-node setting. Let $\widehat{\boldsymbol{\alpha}}^{(0)}$ and $\widehat{\boldsymbol{\beta}}^{(0)}$ be the initial estimates of $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$, respectively, and let $\widehat{f}(\boldsymbol{\alpha}^*)$

be an estimate of $f(\boldsymbol{\alpha}^*)$. For $i = 1, \ldots, N$, define the pseudo responses

$$\widetilde{Y}_i = \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(0)} - \widehat{f}^{-1}(\boldsymbol{\alpha}^*) \left[\frac{1}{K} \sum_{k=1}^{K} \{ I(Y_i - \widehat{\alpha}_k^{(0)} - \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(0)} \leq 0) - \tau_k \} \right].$$

We estimate $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ using the empirical versions of (2.3) and (2.5), respectively:

$$\widehat{\alpha}_k^{(1)} = \widehat{\alpha}_k^{(0)} - \widehat{f}^{-1}(\alpha_k^*) \cdot \frac{1}{N} \sum_{i=1}^{N} \Big\{ I(Y_i - \widehat{\alpha}_k^{(0)} - \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(0)} \leq 0) - \tau_k \Big\}, \tag{2.6}$$

and

$$\widehat{\boldsymbol{\beta}}_{\ell_1}^{(1)} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2N} \sum_{i=1}^{N} (\widetilde{Y}_i - \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta})^2 + \lambda_N |\boldsymbol{\beta}|_1. \tag{2.7}$$

In a single-node setting, problems (2.6) and (2.7) correspond to a simple LS problem and an LS lasso problem, respectively, which are computationally much easier than an ℓ_1-regularized CQR problem.

We take $\widehat{f}(\boldsymbol{\alpha}^*) = \sum_{k=1}^{K} \widehat{f}(\alpha_k^*)/K$ as the average of the K kernel density estimates

$$\widehat{f}(\alpha_k^*) = (Nh)^{-1} \sum_{i=1}^{N} \mathcal{K} \left\{ \frac{(Y_i - \widehat{\alpha}_k^{(0)} - \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(0)})}{h} \right\}, \quad k = 1, \ldots, K. \tag{2.8}$$

Here, $\mathcal{K}(\cdot)$ is a kernel function fulfilling Condition (C3) of Section 3, and $h > 0$ is the bandwidth. In Sections 3 and 4, we discuss the selection of the bandwidth.

To minimize problems (2.6) and (2.7), we may first pool all the data into a single central node, which we then optimize. However, this may require substantial memory and storage for large amounts of data. Distributed systems require computationally efficient algorithms with very low communication costs (Jordan, Lee and Yang (2019); Fan et al. (2019); Lan, Lee and Zhou (2020)). In this paper, we introduce a distributed algorithm that robustly and efficiently estimates $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ at near-oracle rates.

2.2. Distributed estimation

Here, we develop a distributed communication-efficient algorithm to compute $\widehat{\boldsymbol{\alpha}}^{(1)}$ and $\widehat{\boldsymbol{\beta}}_{\ell_1}^{(1)}$ in (2.6) and (2.7) under the "horizontal" distributed setting. Because

$$\widehat{\alpha}_k^{(1)} = \widehat{\alpha}_k^{(0)} - \widehat{f}^{-1}(\alpha_k^*)\frac{1}{N}\sum_{j=1}^{m}\sum_{i\in\mathcal{H}_j}\{I(Y_i - \widehat{\alpha}_k^{(0)} - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{(0)} \le 0) - \tau_k\}, \;\; k = 1, \ldots, K,$$

the communication cost to obtain $\widehat{\boldsymbol{\alpha}}^{(1)}$ is $O(mK)$, which is communication-efficient. For $\widehat{\boldsymbol{\beta}}_{\ell_1}^{(1)}$, we solve problem (2.7) using an approximate Newton method (Wang et al. (2017); Jordan, Lee and Yang (2019); Fan, Guo and Wang (2021)), that has a communication cost $O(mp)$. For ease of notation, let

$$\mathbf{z}_{n,j} \overset{\text{def}}{=} \frac{1}{n}\sum_{i\in\mathcal{H}_j}\mathbf{x}_i\widetilde{Y}_i, \; \mathbf{z}_N \overset{\text{def}}{=} \frac{1}{m}\sum_{j=1}^{m}\mathbf{z}_{n,j}, \; \widehat{\Sigma}_j \overset{\text{def}}{=} \frac{1}{n}\sum_{i\in\mathcal{H}_j}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}, \text{ and } \widehat{\Sigma} \overset{\text{def}}{=} \frac{1}{m}\sum_{j=1}^{m}\widehat{\Sigma}_j.$$

We further define the pseudo local and global loss functions, respectively, as

$$\mathcal{L}_j(\boldsymbol{\beta}) \overset{\text{def}}{=} \frac{1}{2n}\sum_{i\in\mathcal{H}_j}(\widetilde{Y}_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})^2 \text{ and } \mathcal{L}_N(\boldsymbol{\beta}) \overset{\text{def}}{=} \frac{1}{m}\sum_{j=1}^{m}\mathcal{L}_j(\boldsymbol{\beta}).$$

Denote the gradient of $\mathcal{L}_N(\boldsymbol{\beta})$ by $\nabla\mathcal{L}_N(\boldsymbol{\beta})$, which is $\widehat{\Sigma}\boldsymbol{\beta} - \mathbf{z}_N$. Given an initial estimate $\widehat{\boldsymbol{\beta}}^{(0)}$, we have

$$\begin{aligned}\mathcal{L}_N(\boldsymbol{\beta}) = &\mathcal{L}_N(\widehat{\boldsymbol{\beta}}^{(0)}) + \{\nabla\mathcal{L}_N(\widehat{\boldsymbol{\beta}}^{(0)})\}^{\mathrm{T}}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)}) \\ &+ \frac{1}{2}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)})^{\mathrm{T}}\widehat{\Sigma}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)}).\end{aligned} \tag{2.9}$$

Recall that in a "horizontal" distributed system, the data are scattered across m nodes. Transmitting the Hessian matrix $\widehat{\Sigma}$ requires a communication cost $O(p^2)$, which is typically expensive in a high-dimensional setting. To reduce the communication cost, we approximate the Hessian matrix $\widehat{\Sigma}$ by $\widehat{\Sigma}_1$, which leads to the surrogate loss

$$\begin{aligned}\widetilde{\mathcal{L}}^*(\boldsymbol{\beta}) \overset{\text{def}}{=} &\mathcal{L}_N(\widehat{\boldsymbol{\beta}}^{(0)}) + \{\nabla\mathcal{L}_N(\widehat{\boldsymbol{\beta}}^{(0)})\}^{\mathrm{T}}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)}) \\ &+ \frac{1}{2}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)})^{\mathrm{T}}\widehat{\Sigma}_1(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)}).\end{aligned} \tag{2.10}$$

Comparing (2.10) with (2.9), we obtain the approximation error of the surrogate loss

$$\begin{aligned}\widetilde{\mathcal{L}}^*(\boldsymbol{\beta}) - \mathcal{L}_N(\boldsymbol{\beta}) &= (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)})^{\mathrm{T}}(\widehat{\Sigma} - \widehat{\Sigma}_1)(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)}) \\ &= O_p\big\{\|\widehat{\Sigma} - \widehat{\Sigma}_1\|_{\mathrm{op}} \cdot |\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)}|_2^2\big\},\end{aligned}$$

where the last equality follows from the Cauchy–Schwarz inequality. The approximation error is negligible if either $\|\widehat{\Sigma} - \widehat{\Sigma}_1\|_{\mathrm{op}}$ or $|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(0)}|_2$ is $o_p(1)$, which

is possible if p is much smaller than n or if a sparsity structure exists in the coefficient vector when p is much greater than n.

Ignoring the additive terms in (2.10) irrelevant to $\boldsymbol{\beta}$, the surrogate loss can be simplified to

$$\widetilde{\mathcal{L}}(\boldsymbol{\beta}) \stackrel{\text{def}}{=} \frac{1}{2n} \sum_{i \in \mathcal{H}_1} (\mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta})^2 - \boldsymbol{\beta}^{\mathrm{T}} \{ \mathbf{z}_N + (\widehat{\Sigma}_1 - \widehat{\Sigma}) \widehat{\boldsymbol{\beta}}^{(0)} \}.$$

Here, rather than working with the pseudo global loss $\mathcal{L}_N$ in (2.7), we work with $\widetilde{\mathcal{L}}(\boldsymbol{\beta})$ to reduce the communication cost. Specifically, we define

$$\widehat{\boldsymbol{\beta}}^{(1)} \stackrel{\text{def}}{=} \operatorname*{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} \widetilde{\mathcal{L}}(\boldsymbol{\beta}) + \lambda_N |\boldsymbol{\beta}|_1. \tag{2.11}$$

Note that we can calculate $\widehat{\Sigma} \widehat{\boldsymbol{\beta}}^{(0)}$ and $\mathbf{z}_N$ very efficiently in a distributed manner with a communication cost $O(mp)$, because $\widehat{\Sigma}_j \widehat{\boldsymbol{\beta}}^{(0)}$ and $\mathbf{z}_{n,j}$, which form $\widehat{\Sigma} \widehat{\boldsymbol{\beta}}^{(0)}$ and $\mathbf{z}_N$, respectively, are both p-dimensional vectors (see Algorithm 1). There is no need to communicate the $p \times p$ covariance matrix $\widehat{\Sigma}_j$.

In practice, when feasible, we recommend using the ℓ_1-penalized CQR estimate (Gu and Zou (2020); Pietrosanu et al. (2021)), fitted from the data collected only at the first node as the initial estimate:

$$\{\widehat{\boldsymbol{\alpha}}^{(0)}, \widehat{\boldsymbol{\beta}}^{(0)}\} \stackrel{\text{def}}{=} \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{R}^K, \boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{2nK} \sum_{i \in \mathcal{H}_1} \sum_{k=1}^{K} \rho_{\tau_k}(Y_i - \alpha_k - \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda_n |\boldsymbol{\beta}|_1. \tag{2.12}$$

Our distributed estimation procedure then proceeds iteratively from the initial estimate. Specifically, for any $t \geq 1$, let $\widehat{\boldsymbol{\beta}}^{(t-1)}$ be the distributed estimate in the $(t-1)$th communication, and let

$$\widehat{f}^{(t)}(\boldsymbol{\alpha}^*) \stackrel{\text{def}}{=} (NKh^{(t)})^{-1} \sum_{k=1}^{K} \sum_{i=1}^{N} \mathcal{K}\left\{ \frac{Y_i - \widehat{\alpha}_k^{(t-1)} - \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(t-1)}}{h^{(t)}} \right\}$$

be the estimate of $f(\boldsymbol{\alpha}^*)$ and $h^{(t)}$ be the associated bandwidth (specified in Theorem 2) in the tth communication. Define

$$\widetilde{Y}_i^{(t)} = \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(t-1)} - \widehat{f}^{-1}(\boldsymbol{\alpha}^*) \left[\frac{1}{K} \sum_{k=1}^{K} \{ I(Y_i - \widehat{\alpha}_k^{(t-1)} - \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(t-1)} \leq 0) - \tau_k \} \right],$$

for $i = 1, \ldots, N$, and $\mathbf{z}_N^{(t)} = N^{-1} \sum_{i=1}^{N} \mathbf{x}_i \widetilde{Y}_i^{(t)}$. The distributed estimate in the tth

communication takes the form

$$\widehat{\alpha}_k^{(t)} \stackrel{\text{def}}{=} \widehat{\alpha}_k^{(t-1)} - \widehat{f}^{-1}(\alpha_k^*)\frac{1}{N}\sum_{i=1}^{N}\{I(Y_i - \widehat{\alpha}_k^{(t-1)} - \mathbf{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^{(t-1)} \le 0) - \tau_k\} \tag{2.13}$$

and

$$\widehat{\boldsymbol{\beta}}^{(t)} \stackrel{\text{def}}{=} \operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2n}\sum_{i\in\mathcal{H}_1}(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})^2 - \boldsymbol{\beta}^{\mathrm{T}}\big\{\mathbf{z}_N^{(t)} + (\widehat{\Sigma}_1 - \widehat{\Sigma})\widehat{\boldsymbol{\beta}}^{(t-1)}\big\} + \lambda_{N,t}|\boldsymbol{\beta}|_1. \tag{2.14}$$

Problem (2.14) is an ℓ_1-regularized quadratic program, which can be solved using a first-order method (Combettes and Pesquet (2011); Bach et al. (2012); Tropp and Wright (2010)), a Newton-type algorithm (Fountoulakis, Gondzio and Zhlobich (2014); Dassios, Fountoulakis and Gondzio (2015)), or the coordinate descent algorithm (Friedman, Hastie and Tibshirani (2010)). In our implementation, we use the primal dual active set (PDAS, Fan, Jiao and Lu (2014)) method, which is essentially a generalized Newton-type method. It converges after one iteration if the initial value is good enough. To select the regularization parameter, because $\widehat{\boldsymbol{\beta}}^{(t)}$ is a piecewise linear function of $\lambda_{N,t}$ (Osborne, Presnell and Turlach (2000)), we use a continuation procedure in order to fully exploit the fast convergence of the PDAS method. Specifically, we use the solution from the previous step as the initial value for the current step. When the continuation procedure completes, we have a solution path for (2.14), from which we choose the best regularization parameter based on maximum voting. In particular, we set $\lambda_1 = |\mathbf{z}_N^{(t)} + (\widehat{\Sigma}_1 - \widehat{\Sigma})\widehat{\boldsymbol{\beta}}^{(t-1)}|_\infty$, which has a solution to (2.14) that is exactly zero, by the Karush–Kuhn–Tucker conditions. Let $\lambda_\ell = \lambda_1\rho^{\ell-1}$, with $\rho \in (0,1)$, for $\ell \ge 1$. For some pre-fixed threshold $s_0 \in \mathbb{N}_+$, we apply the PDAS method to compute a solution path $\{\widehat{\boldsymbol{\beta}}^{(t),\lambda_1}, \ldots, \widehat{\boldsymbol{\beta}}^{(t),\lambda_L}\}$ until $|\widehat{\boldsymbol{\beta}}^{(t),\lambda_L}|_0 > s_0$ for a smallest possible L. Let $\mathcal{S}_v = \{\lambda_\ell \colon |\widehat{\boldsymbol{\beta}}^{(t),\lambda_\ell}|_0 = v, \ell = 1, \ldots, L\}$ be the set of regularization parameters at which the solution to (2.14) has v nonzero elements, where $v = 1, \ldots, s_0$. We determine $\lambda_{N,t}$ by maximum voting, that is,

$$\lambda_{N,t} = \max\{\mathcal{S}_{\overline{v}}\} \text{ and } \overline{v} = \operatorname*{argmax}_v |\mathcal{S}_v|,$$

where $|\mathcal{S}_v|$ is the cardinality of the set $\mathcal{S}_v$. Our parameter selection rule is seamlessly integrated with the continuation procedure without incurring extra communication and computation costs. Classical cross-validation approaches can be used in the distributed setting (Yu, Chao and Cheng (2021)) as well. We summarize our distributed algorithm in Algorithm 1.

Algorithm 1 Distributed algorithm for sparse CQR.

Input: Data $\{(\mathbf{x}_i, Y_i)_{i\in\mathcal{H}_j}\}$, for $j = 1, \ldots, m$, the number of iterations t, the quantile levels τ_k, for $k = 1, \ldots, K$, a sequence of bandwidths $h^{(g)}$, for $g = 1, \ldots, t$, the regularization parameters λ_n and $\lambda_N^{(g)}$, for $g = 1, \ldots, t$.

1: Compute the initial estimates $\widehat{\boldsymbol{\alpha}}^{(0)}$ and $\widehat{\boldsymbol{\beta}}^{(0)}$ using (2.12), based on $\{(\mathbf{x}_i, Y_i)_{i\in\mathcal{H}_1}\}$.
2: **for** $g = 1, \ldots, t$ **do**
3: Transmit $\widehat{\boldsymbol{\alpha}}^{(g-1)}$ and $\widehat{\boldsymbol{\beta}}^{(g-1)}$ from the first node to the local ones labeled with $2, \ldots, m$.
4: **for** $j = 1, \ldots, m$ **do**
5: Calculate

$$\widehat{f}^{(g,j)}(\boldsymbol{\alpha}^*) = (nKh^{(g)})^{-1} \sum_{k=1}^{K} \sum_{i\in\mathcal{H}_j} \mathcal{K}\left\{ \frac{Y_i - \widehat{\alpha}_k^{(g-1)} - \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}}^{(g-1)}}{h^{(g)}} \right\}$$

at the jth node and send it back to the first node.
6: **end for**
7: The first node computes $\widehat{f}^{(g)}(\boldsymbol{\alpha}^*) = m^{-1} \sum_{j=1}^{m} \widehat{f}^{(g,j)}(\boldsymbol{\alpha}^*)$ and transmits it to the local nodes labeled with $2, \ldots, m$.
8: **for** $j = 1, \ldots, m$ **do**
9: Calculate $\widehat{\Sigma}_j \widehat{\boldsymbol{\beta}}^{(g-1)}$ and $\mathbf{z}_{n,j}^{(g)} = n^{-1} \sum_{i\in\mathcal{H}_j} \mathbf{x}_i \widetilde{Y}_i^{(g)}$ at the jth node and send them back to the first node.
10: **end for**
11: Calculate $\widehat{\boldsymbol{\alpha}}^{(g)}$ and $\widehat{\boldsymbol{\beta}}^{(g)}$ on the first node, based on (2.13) and (2.14).
12: **end for**

Output: The final estimates $\widehat{\boldsymbol{\alpha}}^{(t)}$ and $\widehat{\boldsymbol{\beta}}^{(t)}$ obtained from the first node.

3. Theoretical Results

In this section, we show the estimation and support recovery accuracy of the distributed CQR estimate. Denote $\mathcal{S} \overset{\text{def}}{=} \operatorname{supp}(\boldsymbol{\beta}^*)$ as the support of $\boldsymbol{\beta}^*$, and let $s \overset{\text{def}}{=} |\mathcal{S}|$. Following Wainwright (2019), we say a random vector $\mathbf{x} \in \mathbb{R}^p$ is sub-Gaussian if it satisfies $\sup_{|\alpha|_2=1} E \exp\{t(\alpha^{\mathrm{T}}\mathbf{x})^2\} \le C$, for some $t > 0$ and $C > 0$. We assume the following conditions:

(C1) The density f is bounded and Lipschitz continuous, that is, $|f(x) - f(y)| \le C_L|x-y|$, for any $x, y \in \mathbb{R}$ and some constant $C_L > 0$. Moreover, we assume $f(\alpha_k^*) \ge \underline{f} > 0$, for all $k = 1, \ldots, K$.

(C2) There exists a constant $c_0 > 0$ such that $c_0^{-1} \le \Lambda_{\min}(\Sigma) \le \Lambda_{\max}(\Sigma) \le c_0$. Furthermore, we assume $\|\Sigma_{\mathcal{S}^c\times\mathcal{S}}\Sigma_{\mathcal{S}\times\mathcal{S}}^{-1}\|_\infty \le 1 - \alpha$, for some $0 < \alpha < 1$.

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

(C4) The covariate vector $\mathbf{x}$ is sub-Gaussian. The dimension p satisfies $p = O(N^{\nu})$, for some $\nu > 0$. The sample size n at each local node satisfies $n \geq N^{\omega}$, for some $0 < \omega < 1$, the sparsity level s satisfies $s = O(n^{r})$, for some $0 \leq r < 1/3$, and the number of quantile levels K satisfies $K = O(n^{r})$, for some $0 \leq r < 1/3$.

(C5) The initial estimates $\widehat{\boldsymbol{\alpha}}^{(0)}$ and $\widehat{\boldsymbol{\beta}}^{(0)}$ satisfy $\mathrm{pr}(\mathrm{supp}(\widehat{\boldsymbol{\beta}}^{(0)}) \subseteq \mathcal{S}) \to 1$ and $|\widehat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^{*}|_2 = O_p(a_n)$ and $|\widehat{\boldsymbol{\alpha}}^{(0)} - \boldsymbol{\alpha}^{*}|_2 = O_p(K^{1/2}a_n)$, where $a_n = (s \log N/n)^{1/2}$.

Condition (C1) is standard on the smoothness of the noise density (Gu and Zou (2020); Chen et al. (2020)). The irrepresentable condition (C2) is widely used in the high-dimensional statistics literature to establish support recovery; see, for example, Zhao and Yu (2006),Wainwright (2009), Hastie, Tibshirani and Wainwright (2015), and Wainwright (2019). Condition (C3) imposes regular conditions on the kernel function, and is mild and satisfied by many common kernel functions. Condition (C4) is commonly assumed in the distributed estimation literature; see, for example, Chen et al. (2020), Wang et al. (2017), and Jordan, Lee and Yang (2019). In Algorithm 1, the initial estimator $\widehat{\boldsymbol{\beta}}^{(0)}$ is obtained using the data scattered at the central node. Such an initial estimate satisfies (C5) under Conditions (C1), (C2), and (C4) (see Theorem 1 of Gu and Zou (2020)). We use $\log N$ throughout for simplicity, because we have $\log\{\max(N, p)\} = C_1 \log N$, for some constant $C_1 > 0$, by Condition (C4).

We first present the convergence rates of the distributed estimates $\widehat{\boldsymbol{\alpha}}^{(1)}$ and $\widehat{\boldsymbol{\beta}}^{(1)}$ from the first communication.

Theorem 1. *Set* $\lambda_N = C_0\{(\log N/N)^{1/2} + (s \log N/n)^{1/2}a_n\}$ *and the bandwidth* $h \asymp a_n$*, where* $C_0 > 0$ *is a sufficiently large constant. Under Conditions* (C1)–(C5)*, we have*

$$|\widehat{\boldsymbol{\alpha}}^{(1)} - \boldsymbol{\alpha}^{*}|_2 = O_p\left\{\left(\frac{Ks \log N}{N}\right)^{1/2} + \left(\frac{Ks^2 \log N}{n}\right)^{1/2} a_n\right\}$$

and

$$|\widehat{\boldsymbol{\beta}}^{(1)} - \boldsymbol{\beta}^{*}|_2 = O_p\left\{\left(\frac{s \log N}{N}\right)^{1/2} + \left(\frac{s^2 \log N}{n}\right)^{1/2} a_n\right\}.$$

Let $a_N^{(g)} = (s \log N/N)^{1/2} + s^{(2g+1)/2}(\log[N]/n)^{(g+1)/2}$, for $g = 1, \ldots, t$. Applying Theorem 1 leads to the convergence rates of the distributed estimates $\widehat{\boldsymbol{\alpha}}^{(t)}$ and $\widehat{\boldsymbol{\beta}}^{(t)}$ from the tth communication.

Theorem 2. *Set* $\lambda_N^{(g)} = C_0\{(\log N/N)^{1/2} + (s\log N/n)^{1/2}a_N^{(g-1)}\}$ *and the bandwidth* $h^{(g)} \asymp a_N^{(g-1)}$, *for* $g = 1, \ldots, t$, *where* $C_0 > 0$ *is a sufficiently large constant. Under Conditions* (C1)–(C5), *we have*

$$|\widehat{\boldsymbol{\alpha}}^{(t)} - \boldsymbol{\alpha}^*|_2 = O_p\left\{\left(\frac{Ks\log N}{N}\right)^{1/2} + (K)^{1/2}s^{(2t+1)/2}\left(\frac{\log N}{n}\right)^{(t+1)/2}\right\}$$

and

$$|\widehat{\boldsymbol{\beta}}^{(t)} - \boldsymbol{\beta}^*|_2 = O_p\left\{\left(\frac{s\log N}{N}\right)^{1/2} + s^{(2t+1)/2}\left(\frac{\log N}{n}\right)^{(t+1)/2}\right\}.$$

When the number of communications t is large enough, that is,

$$t \geq \log\left(\frac{N}{n}\right)\Big/\log\left\{\frac{c_0 n}{s^2\log N}\right\}, \text{ for some } c_0 > 0, \tag{3.1}$$

we have $s^{(2t+1)/2}(\log N/n)^{(t+1)/2} = O\{(s\log N/N)^{1/2}\}$. Therefore, $|\widehat{\boldsymbol{\beta}}^{(t)} - \boldsymbol{\beta}^*|_2 = O_p\{(s\log N/N)^{1/2}\}$, and the distributed estimate $\widehat{\boldsymbol{\beta}}^{(t)}$ attains the minimax optimal rate $O\{(s\log N/N)^{1/2}\}$. This is also the optimal rate when the data are pooled at a single central node (Gu and Zou (2020)). In view of Condition (C4), the right-hand side of (3.1) is bounded by a constant. To achieve the oracle rate, our distributed algorithm requires the number of communications, t, to increase logarithmically with the number of nodes, m. In contrast, existing distributed first-order algorithms require the number of communications to increase polynomially with m; see Table 1 of Zhang and Xiao (2018) for details.

We present the support recovery of our distributed method in the following two theorems.

Theorem 3. *Under the conditions of Theorem* 1, *we have* $\mathrm{supp}(\widehat{\boldsymbol{\beta}}^{(1)}) \subseteq \mathcal{S}$, *with probability approaching one. Suppose, in addition, for a sufficiently large positive constant* C,

$$\boldsymbol{\beta}^{*\min} \geq C\|\Sigma_{\mathcal{S}\times\mathcal{S}}^{-1}\|_\infty\left\{\left(\frac{\log N}{N}\right)^{1/2} + |\widehat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}^*|_2\left(\frac{s\log N}{n}\right)^{1/2}\right\}.$$

Then, we have $\mathrm{supp}(\widehat{\boldsymbol{\beta}}^{(1)}) = \mathcal{S}$, *with probability approaching one.*

Theorem 4. *Under the conditions of Theorem* 2, *we have* $\mathrm{supp}(\widehat{\boldsymbol{\beta}}^{(t)}) \subseteq \mathcal{S}$, *with probability approaching one. Suppose, for a sufficiently large positive constant* C,

$$\boldsymbol{\beta}^{*\min} \geq C\|\Sigma_{\mathcal{S}\times\mathcal{S}}^{-1}\|_\infty\left\{\left(\frac{\log N}{N}\right)^{1/2} + s^t\left(\frac{\log N}{n}\right)^{(t+1)/2}\right\}.$$

Then, we have $\text{supp}(\widehat{\boldsymbol{\beta}}^{(t)}) = \mathcal{S}$*, with probability approaching one.*

The "beta-min" condition, which is commonly assumed in the literature on high-dimensional statistics, weakens as t increases and matches the oracle rate for the "beta-min" condition, that is, $\boldsymbol{\beta}^{*\min} \geq C\|\Sigma_{\mathcal{S}\times\mathcal{S}}^{-1}\|_\infty(\log N/N)^{1/2}$, after a constant number of communications (Wainwright (2009)).

We have assumed evenly scattered data across the nodes, for ease of demonstration. In fact, the number of data points, n, is just the "working" sample size at the first node, or the central node, as it is known as in distributed computing. Once it is specified, our approach does not depend on the partition of the data.

Several works in the distributed computing literature examine heavy-tailed noise. Chen et al. (2020) consider the distributed QR estimation and similarly cast the nonsmooth QR problem as an LS problem. Note that our study of the CQR is motivated by the potential loss of efficiency of the QR under certain noise distributions, and our work is technically more challenging than the distributed QR, because our loss function consists of quantile check losses at multiple levels. Furthermore, in contrast to the validation method of Chen et al. (2020), we suggest a new tuning procedure in Section 2.2 that does not incur extra communication and computation costs. Luo, Sun and Zhou (2022) consider the distributed adaptive Huber regression, which can also handle certain cases of heavy-tailed noise. Their theoretical analysis does not assume independence between the noise and the covariates, but does require that the covariates to be bounded. Moreover, the Huber regression does not work under very heavy-tailed noise such as the Cauchy, whereas the CQR does. Battey, Tan and Zhou (2021) suggest a convoluted smoothing of the check loss to handle the nonsmoothness of the QR. This alternative smoothing procedure can be applied to our CQR loss, especially when the construction of the pseudo response is not stable in small samples. However, it may not be as computationally efficient as our method, owing to our simple LS formulation. In addition, the aforementioned works focus mainly on the estimation error bounds of their respective estimates, whereas we establish the support recovery theory in addition to the estimation error. Support recovery is an important topic in high-dimensional distributed settings but is usually very challenging (Neykov, Liu and Cai (2016)).

4. Simulation Studies

4.1. Design of the simulations

We simulate the data from model (1.1) with $\beta_0^* = 0$ and $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, 2, \mathbf{0}_{p-5})^{\mathrm{T}}$, and the covariates $\mathbf{x}_i$ are drawn from $\mathcal{N}(0, \Sigma)$, with $\Sigma = (0.5^{|k-l|})_{p\times p}$.

We fix $p = 500$ throughout. For the noise distribution, we follow Zou and Yuan (2008) and Gu and Zou (2020), and consider three shapes:

(a) the normal distribution, $\varepsilon \sim \mathcal{N}(0, 1)$,

(b) the Student's t distribution with three degrees of freedom, $\varepsilon \sim t(3)$, and

(c) the Cauchy distribution, $\varepsilon \sim f(\varepsilon) = 1/\{\pi(1 + \varepsilon^2)\}$.

The initial estimator is taken as the ℓ_1-regularized CQR estimator defined in (2.12) using the local data at the first node. The constant C_0 of $\lambda_N^{(g)}$ is chosen using majority voting along the solution paths calculated using the method of Huang et al. (2018). We use the bi-weight kernel function $\mathcal{K}(x) = 105(1-3x^2)(1-x^2)^2 I(|x| \le 1)/64$, and set the bandwidth as $h^{(g)} = ca_N^{(g-1)}$, for some constant $c > 0$ (Theorem 2). We take $c = 1$, for simplicity. A sensitivity analysis of the choice of c is provided in Section 4.5. Throughout, we take $K = 19$ and $\tau_k = k/(K + 1)$, for $k = 1, \ldots, K$.

We compare our distributed estimate with its pooled counterpart and the divide-and-conquer estimate. Specifically, the divide-and-conquer method computes the ℓ_1-regularized CQR at each local node and combines the local estimates using simple averaging.

Two criteria are used to evaluate the performance of the methods: the estimation error, $|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*|_2$, and the F_1-score,

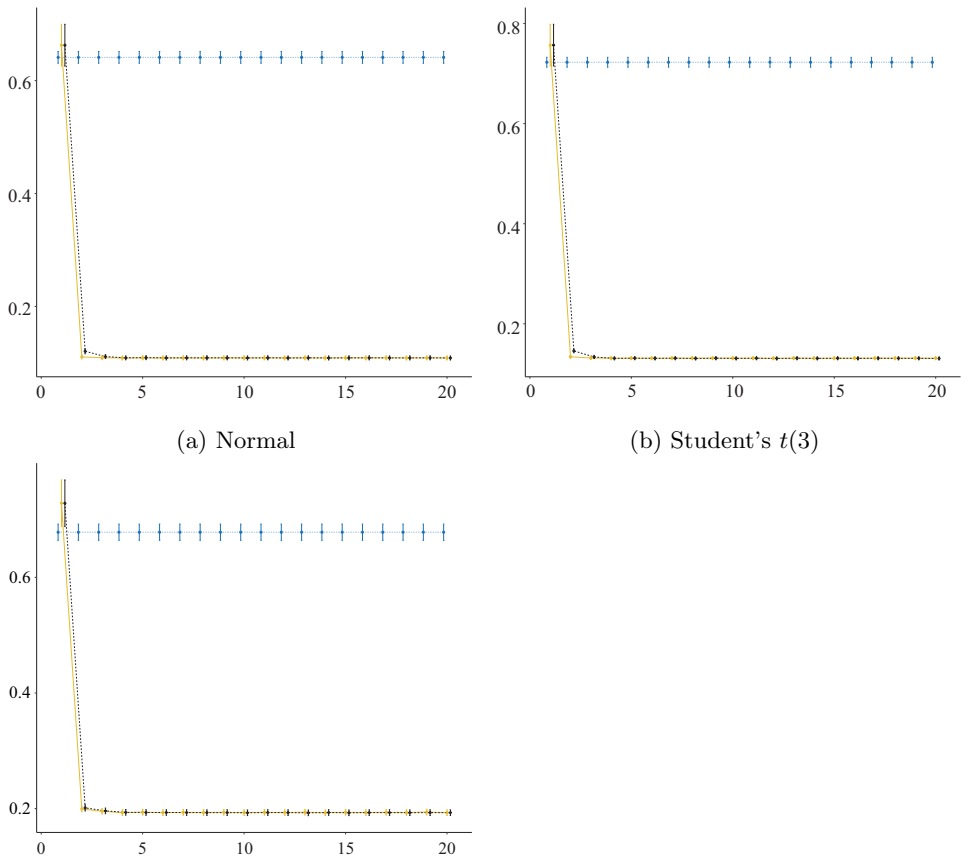
$$F_1 \stackrel{\text{def}}{=} \frac{2(\text{TP})}{2(\text{TP}) + \text{FP} + \text{FN}},$$

where TP, FP, and FN denote the numbers of true positives, false positives, and false negatives, respectively. The F_1-score ranges from zero to one, with larger values indicating better performance (see, e.g., Goutte and Gaussier (2005)), and is widely used in the literature to evaluate support recovery accuracy.

We repeat each setting with one hundred independent runs.

4.2. Effect of the number of communications

We investigate how the performance of our distributed estimate varies according to the number of iterations (or communications). We fix the sample size at $N = 5000$, the local sample size at $n = 500$, and the number of nodes at $m = 10$. Shown in Figure 1 is the plot of the mean estimation error (over one hundred independent runs) versus the number of iterations, where the error bar corresponds to one standard error. The distributed and pooled estimates exhibit similar performance under all three noise scenarios, and both become stable in



(a) Normal

(b) Student's $t(3)$

(c) Cauchy

Figure 1. The horizontal axis shows the number of iterations (communications), and the vertical axis represents the estimation errors of the divide-and-conquer (···✳···), distributed (-◇-), and pooled (—⊟—) estimates when the noise comes from the (a) normal, (b) Student's $t(3)$, and (c) Cauchy distributions. The error bar corresponds to one standard error. The horizontal values are jittered to avoid overlapping. The overall sample size, local sample size, and dimension are fixed at $N = 5000$, $n = 500$, and $p = 500$, respectively.

just a few iterations. In addition, they both outperform the divide-and-conquer estimate by a large margin.

4.3. Effect of the noise distribution

Here, we demonstrate the robustness of the CQR to heavy-tailed noise and its preservation of efficiency under light-tailed noise by considering the three aforementioned noise distributions. We additionally include the LS lasso estimate in a single-node setting (pooled data) in the comparison. We vary the sample size

Table 1. The estimation errors and F_1-scores of the distributed, pooled, and divide-and-conquer estimates, and the least squares lasso estimate fitted using the pooled data under varying sample sizes N when the noise comes from the normal, Student's $t(3)$, and Cauchy distributions. The local sample size n is fixed at 500.

N	distr		pooled		dc		lasso	
	est. error	F_1-score	est. error	F_1-score	est. error	F_1-score	est. error	F_1-score
				Normal noise				
2,500	0.0998	1.0000	0.0997	1.0000	0.1770	0.0826	0.0817	0.4636
5,000	0.0718	1.0000	0.0717	1.0000	0.1716	0.0409	0.0576	0.4793
10,000	0.0515	1.0000	0.0515	1.0000	0.1673	0.0241	0.0398	0.4521
15,000	0.0378	1.0000	0.0379	1.0000	0.1639	0.0186	0.0307	0.4748
20,000	0.0349	1.0000	0.0348	1.0000	0.1662	0.0158	0.0295	0.5512
25,000	0.0308	1.0000	0.0307	1.0000	0.1655	0.0146	0.0288	0.8413
				Student's $t(3)$ noise				
2,500	0.1242	1.0000	0.1244	1.0000	0.1974	0.0756	0.1433	0.4199
5,000	0.0884	1.0000	0.0885	1.0000	0.1894	0.0394	0.0990	0.4067
10,000	0.0596	1.0000	0.0596	1.0000	0.1841	0.0238	0.0695	0.4858
15,000	0.0465	1.0000	0.0465	1.0000	0.1826	0.0183	0.0538	0.4761
20,000	0.0432	1.0000	0.0432	1.0000	0.1824	0.0159	0.0481	0.4482
25,000	0.0383	1.0000	0.0383	1.0000	0.1816	0.0143	0.0441	0.4417
				Cauchy noise				
2,500	0.1713	1.0000	0.1713	1.0000	0.1908	0.1042	3.1826	0.2245
5,000	0.1254	1.0000	0.1255	1.0000	0.1897	0.0470	3.1247	0.2505
10,000	0.0846	1.0000	0.0846	1.0000	0.1878	0.0276	3.0907	0.2454
15,000	0.0732	1.0000	0.0732	1.0000	0.1845	0.0211	3.1835	0.2361
20,000	0.0619	1.0000	0.0618	1.0000	0.1819	0.0175	3.1540	0.2490
25,000	0.0541	1.0000	0.0540	1.0000	0.1837	0.0158	2.9773	0.2418

N, and summarize the estimation errors and F_1-scores of the four estimates in Table 1. In all settings, the performance of our distributed estimate matches that of the pooled estimate, and both outperform the divide-and-conquer and lasso estimates. In particular, the distributed and pooled estimates perform similarly to the lasso estimate under normal noise, but exhibit much better performance under the Cauchy noise. We omit the lasso estimate in subsequent comparisons, owing to its instability under heavy-tailed noises.

4.4. Effect of the overall and local sample sizes

We investigate the performance of the estimates under different combinations of the overall sample size N and the local sample size n. The estimation errors and F_1-scores are reported in Table 2. In terms of the estimation error, both the distributed and the pooled estimates outperform the divide-and-conquer estimate in almost all the settings, except when $N = 5000$ and $n = 1000$ under the Cauchy noise. In this exceptional case, however, they outperform the divide-and-

Table 2. The estimation errors and F_1-scores of the distributed, pooled, and divide-and-conquer estimates under different combinations of the overall sample size N and the local sample size n when the noise comes from the normal, Student's $t(3)$, and Cauchy distributions.

	n	200			500			1,000		
	N	5,000	10,000	20,000	5,000	10,000	20,000	5,000	10,000	20,000
Normal noise										
distr	est. error	0.0711	0.0506	0.0377	0.0711	0.0491	0.0342	0.0719	0.0478	0.0358
	F_1-score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
pooled	est. error	0.0710	0.0505	0.0376	0.0710	0.0490	0.0342	0.0718	0.0478	0.0357
	F_1-score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
dc	est. error	0.2565	0.2544	0.2536	0.1698	0.1661	0.1636	0.1251	0.1185	0.1174
	F_1-score	0.0187	0.0136	0.0121	0.0407	0.0239	0.0156	0.0812	0.0415	0.0254
Student's $t(3)$ noise										
distr	est. error	0.0870	0.0606	0.0442	0.0860	0.0623	0.0427	0.0873	0.0577	0.0448
	F_1-score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
pooled	est. error	0.0869	0.0612	0.0436	0.0861	0.0622	0.0425	0.0873	0.0576	0.0447
	F_1-score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
dc	est. error	0.2953	0.2929	0.2906	0.1868	0.1876	0.1825	0.1394	0.1302	0.1292
	F_1-score	0.0192	0.0139	0.0121	0.0403	0.0239	0.0158	0.0794	0.0424	0.0238
Cauchy noise										
distr	est. error	0.1236	0.0878	0.0606	0.1326	0.0897	0.0614	0.1215	0.0865	0.0608
	F_1-score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
pooled	est. error	0.1241	0.0878	0.0611	0.1327	0.0897	0.0615	0.1215	0.0866	0.0610
	F_1-score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
dc	est. error	0.3674	0.3657	0.3630	0.1900	0.1876	0.1828	0.1083	0.1019	0.0941
	F_1-score	0.0199	0.0144	0.0123	0.0492	0.0278	0.0174	0.1384	0.0797	0.0407

conquer estimate again when the sample size N keeps growing, for example, from $N = 5000$ to $N = 10000$. This demonstrates the sub-optimality of the divide-and-conquer estimate compared with our distributed estimate when the number of nodes m grows. In terms of the support recovery, the F_1-scores of the distributed and pooled estimates are equal to one in all settings, and are much better than that of the divide-and-conquer estimate. This is not surprising, because the divide-and-conquer method usually results in a dense estimate.

4.5. Sensitivity analysis for the bandwidth

We investigate the sensitivity of the bandwidth selection by varying the sample size N and the constant c in the bandwidth $h^{(g)} = ca_N^{(g-1)}$ from 1 to 20. We summarize the results for the Cauchy noise in Table 3. The results for the other two noise distributions are relegated to the Supplementary Material. We can see from Table 3 that the distributed and pooled estimates are quite robust to the choice of the bandwidth constant c, and exhibit similar performance under all

Table 3. The estimation errors and F_1-scores of the distributed, pooled, and divide-and-conquer estimates under different combinations of the sample size N and the bandwidth constant c when the noise comes from the Cauchy distribution. The local sample size n is fixed at 500.

N	c	distr		pooled		dc	
		est. error	F_1-score	est. error	F_1-score	est. error	F_1-score
5,000	1	0.1216	1.0000	0.1216	1.0000	0.1953	0.0483
10,000	1	0.0868	1.0000	0.0865	1.0000	0.1870	0.0268
20,000	1	0.0610	1.0000	0.0610	1.0000	0.1835	0.0177
5,000	5	0.1213	1.0000	0.1211	1.0000	0.1904	0.0503
10,000	5	0.0912	1.0000	0.0912	1.0000	0.1889	0.0278
20,000	5	0.0606	1.0000	0.0606	1.0000	0.1852	0.0179
5,000	10	0.1251	1.0000	0.1252	1.0000	0.1917	0.0494
10,000	10	0.0853	1.0000	0.0853	1.0000	0.1859	0.0289
20,000	10	0.0586	1.0000	0.0585	1.0000	0.1862	0.0176
5,000	20	0.1192	1.0000	0.1193	1.0000	0.1844	0.0486
10,000	20	0.0880	1.0000	0.0885	1.0000	0.1886	0.0277
20,000	20	0.0615	1.0000	0.0616	1.0000	0.1851	0.0178

choices of the constant c.

4.6. The initial estimation method

The initial estimates play a key role in determining the performance of our distributed estimate. We investigate the sensitivity of our final estimate to the initialization by considering different initial estimates: (1) the LS lasso estimate based on the local sample at the central node; (2) the ℓ_1-regularized CQR estimate (CQR lasso); (3) the ℓ_1-regularized adaptive Huber regression estimate (Huber lasso, Sun, Zhou and Fan (2020); Wang et al. (2021)); and (4) the perturbed true parameters with a normal noise, that is, $\widehat{\alpha}_k^{(0)} \sim \mathcal{N}(\alpha_k^*, \sigma^2)$, for $k = 1, \ldots, K$, and $\widehat{\beta}_j^{(0)} \sim I(\beta_j^* \neq 0) \cdot \mathcal{N}(\beta_j^*, \sigma^2)$, for $j = 1, \ldots, p$. We examine two noise levels, $\sigma = 0.05$ and $\sigma = 0.1$, for the last type of initialization. For the first and third types of initialization, we set $\widehat{\boldsymbol{\alpha}}^{(0)}$ as the empirical quantiles of the response at the central node. Because our algorithm may diverge without a carefully chosen initialization, we compare the estimates after only one iteration. We fix the overall sample size at $N = 5000$, the local sample size at $n = 500$, and the dimension at $p = 500$. The results are reported in Table 4.

Comparing the fourth type of initialization under different noise levels, we see that a more precise initial estimate yields a better distributed estimate. Comparing the first three types of initialization, we see that under a heavy-tailed noise,

Table 4. The estimation errors and F_1-scores of the distributed estimates under different types of initialization when the noise comes from the normal, Student's $t(3)$, and Cauchy distributions. The overall sample size is $N = 5000$, the local sample size is $n = 500$, and the dimension is $p = 500$.

Initialization	Normal noise		Student's $t(3)$ noise		Cauchy noise	
	est. error	F_1-score	est. error	F_1-score	est. error	F_1-score
LS lasso	0.1149	1.0000	0.1534	1.0000	1.0561	0.9986
CQR lasso	0.1220	1.0000	0.1472	1.0000	0.1993	1.0000
Huber lasso	0.1145	1.0000	0.1500	1.0000	0.6364	0.9986
$\sigma = 0.05$	0.1179	1.0000	0.1424	1.0000	0.1973	1.0000
$\sigma = 0.1$	0.1318	1.0000	0.1466	1.0000	0.2108	1.0000

such as the Cauchy, the distributed estimate with the LS lasso initialization has the largest estimation error, which is likely caused by this "bad" initialization. Though the Huber lasso initialization mitigates this issue, its performance is not nearly as good as that of the CQR lasso initialization. Moreover, the latter gives stable estimates under all types of noise. Therefore, we suggest using the CQR lasso initialization to handle arbitrary noise.

4.7. Computational efficiency

We investigate the computational efficiency of our distributed algorithm by comparing the timing with that of competing methods. In addition to the pooled and divide-and-conquer estimates, we include the ℓ_1-regularized CQR estimate based on the pooled data in the comparison. We fix the local sample size at $n = 500$, and vary the overall sample size N. The estimation errors, F_1-scores, and wall times of the four methods are reported in Table 5. It can be seen that our distributed estimate is computationally much more efficient than the single-node ℓ_1-regularized CQR, while exhibiting similar performance.

5. Conclusion

We have developed a distributed algorithm for the penalized CQR by transforming the highly nonsmooth CQR problem into an ordinary LS, which facilitates both computational and theoretical developments. We have proposed a communication-efficient distributed implementation of the transformed problem that communicates gradient information only. Note that our distributed algorithm assumes a centralized system, so the local workers are idle when the central node executes the optimization. Future work should consider a decentralized distributed algorithm that uses all of the system's computing power.

Table 5. The estimation errors, F_1-scores, and wall times of the distributed, pooled, divide-and-conquer, and single-node ℓ_1-regularized CQR estimates under varying sample size N and the Cauchy noise. The local sample size is fixed at $n = 500$.

N	distr			pooled		
	est. error	F_1-score	Time	est. error	F_1-score	Time
5,000	0.1302	1.0000	7.5889	0.1310	1.0000	7.8213
10,000	0.0959	1.0000	9.1686	0.0962	1.0000	15.1164
15,000	0.0737	1.0000	9.2264	0.0740	1.0000	19.6047
N	dc			ℓ_1-regularized CQR		
	est. error	F_1-score	Time	est. error	F_1-score	Time
5,000	0.1872	0.0519	19.0113	0.0423	0.8109	9.4916
10,000	0.1866	0.0280	22.9009	0.0339	0.8609	17.8304
15,000	0.1832	0.0213	24.1322	0.0357	0.8387	24.7910

Supplementary Material

The online Supplementary Material contains proofs of all our theoretical results, as well as some additional simulations.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (12225113, 12171477, 11731011, 11931014), Renmin University of China (22XNA026) and National Science Foundation (DMS 1915842, 2015120). Liping Zhu is the corresponding author. The authors contributed equally to this work, and their names are listed in alphabetical order.

References

Bach, F., Jenatton, R., Mairal, J. and Obozinski, G. (2012). Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning* **4**, 1–106.

Battey, H., Fan, J., Liu, H., Lu, J. and Zhu, Z. (2018). Distributed testing and estimation under sparse high dimensional models. *The Annals of Statistics* **46**, 1352–1382.

Battey, H., Tan, K. M. and Zhou, W.-X. (2021). Communication-efficient distributed quantile regression with optimal statistical guarantees. *arXiv:2110.13113*.

Braverman, M., Garg, A., Ma, T., Nguyen, H. L. and Woodruff, D. P. (2016). Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, 1011–1020.

Chen, X., Liu, W., Mao, X. and Yang, Z. (2020). Distributed high-dimensional regression under a quantile loss function. *Journal of Machine Learning Research* **21**, 1–43.

Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 185–212. Springer, New York.

Dassios, I., Fountoulakis, K. and Gondzio, J. (2015). A preconditioner for a primal-dual Newton conjugate gradient method for compressed sensing problems. *SIAM Journal on Scientific Computing* **37**, A2783–A2812.

Fan, J., Fan, Y. and Barut, E. (2014). Adaptive robust variable selection. *The Annals of Statistics* **42**, 324–351.

Fan, J., Guo, Y. and Wang, K. (2021). Communication-efficient accurate statistical estimation. *Journal of the American Statistical Association*. DOI: 10.1080/01621459.2021.1969238.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96**, 1348–1360.

Fan, J., Wang, D., Wang, K. and Zhu, Z. (2019). Distributed estimation of principal eigenspaces. *The Annals of Statistics* **47**, 3009–3031.

Fan, Q., Jiao, Y. and Lu, X. (2014). A primal dual active set algorithm with continuation for compressed sensing. *IEEE Transactions on Signal Processing* **62**, 6276–6285.

Fountoulakis, K., Gondzio, J. and Zhlobich, P. (2014). Matrix-free interior point method for compressed sensing problems. *Mathematical Programming Computation* **6**, 1–31.

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

Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *Advances in Information Retrieval* (Edited by D. E. Losada and J. M. Fernández-Luna), 345–359. Springer, Berlin, Heidelberg.

Gu, Y. and Zou, H. (2020). Sparse composite quantile regression in ultrahigh dimensions with tuning parameter calibration. *IEEE Transactions on Information Theory* **66**, 7132–7154.

Hastie, T., Tibshirani, R. and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. 1st Edition. CRC Press, Boca Raton.

He, Y., Zhou, Y. and Feng, Y. (2022). Distributed feature selection for high-dimensional additive models. *arXiv:2205.07932*.

Huang, J., Jiao, Y., Lu, X. and Zhu, L. (2018). Robust decoding from 1-bit compressive sampling with ordinary and regularized least squares. *SIAM Journal on Scientific Computing* **40**, A2062–A2086.

Jiang, R., Hu, X., Yu, K. and Qian, W. (2018). Composite quantile regression for massive datasets. *Statistics* **52**, 980–1004.

Jordan, M. I., Lee, J. D. and Yang, Y. (2019). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association* **114**, 668–681.

Kai, B., Li, R. and Zou, H. (2010). Local composite quantile regression smoothing: An efficient and safe alternative to local polynomial regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**, 49–69.

Koenker, R. (2005). *Quantile Regression*. 1st Edition. Cambridge University Press, Cambridge.

Lan, G., Lee, S. and Zhou, Y. (2020). Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming* **180**, 237–284.

Lee, J. D., Liu, Q., Sun, Y. and Taylor, J. E. (2017). Communication-efficient sparse regression. *Journal of Machine Learning Research* **18**, 1–30.

Li, R., Lin, D. K. and Li, B. (2012). Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry* **29**, 399–409.

Luo, J., Sun, Q. and Zhou, W.-X. (2022). Distributed adaptive Huber regression. *Computational Statistics & Data Analysis* **169**, 107419.

Monahan, J. F. (2008). *A Primer on Linear Models*. Chapman & Hall/CRC, Boca Raton.

Neykov, M., Liu, J. S. and Cai, T. (2016). l_1-regularized least squares for support recovery of high dimensional single index models with Gaussian designs. *Journal of Machine Learning Research* **17**, 1–37.

Osborne, M. R., Presnell, B. and Turlach, B. A. (2000). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis* **20**, 389–403.

Pietrosanu, M., Gao, J., Kong, L., Jiang, B. and Niu, D. (2021). Advanced algorithms for penalized quantile and composite quantile regression. *Computational Statistics* **36**, 333–346.

Shamir, O., Srebro, N. and Zhang, T. (2014). Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the 31st International Conference on Machine Learning* **32**, 1000–1008.

Shi, C., Lu, W. and Song, R. (2018). A massive data framework for M-estimators with cubic-rate. *Journal of the American Statistical Association* **113**, 1698–1709.

Song, Q. and Liang, F. (2015). A split-and-merge Bayesian variable selection approach for ultra-high dimensional regression. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **77**, 947–972.

Sun, Q., Zhou, W.-X. and Fan, J. (2020). Adaptive huber regression. *Journal of the American Statistical Association* **115**, 254–265.

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

Tropp, J. A. and Wright, S. J. (2010). Computational methods for sparse solution of linear inverse problems. In *Proceedings of the IEEE* **98**, 948–958.

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory* **55**, 2183–2202.

Wainwright, M. J. (2019). *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. 1st Edition. Cambridge University Press, New York.

Wang, J., Kolar, M., Srebro, N. and Zhang, T. (2017). Efficient distributed learning with sparsity. In *Proceedings of the 34th International Conference on Machine Learning*, 3636–3645.

Wang, L., Zheng, C., Zhou, W. and Zhou, W.-X. (2021). A new principle for tuning-free huber regression. *Statistica Sinica* **31**, 2153–2177.

Yu, Y., Chao, S.-K. and Cheng, G. (2021). Distributed bootstrap for simultaneous inference under high dimensionality. *arXiv:2102.10080 [math, stat]*.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A. et al. (2016). Apache Spark: A unified engine for big data processing. *Communications of the ACM* **59**, 56–65.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* **38**, 894–942.

Zhang, Y. and Xiao, L. (2018). Communication-efficient distributed optimization of self-concordant empirical loss. In *Large-Scale and Distributed Optimization* (Edited by P. Giselsson and A. Rantzer), 289–341. Springer, Cham.

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

Zhao, T., Cheng, G. and Liu, H. (2016). A partially linear framework for massive heterogeneous data. *The Annals of Statistics* **44**, 1400–1437.

Zhou, W.-X., Bose, K., Fan, J. and Liu, H. (2018). A new perspective on robust m-estimation: Finite sample theory and applications to dependence-adjusted multiple testing. *The Annals of Statistics* **46**, 1904–1931.

Zhou, Y., Porwal, U., Zhang, C., Ngo, H. Q., Nguyen, X., Ré, C. et al. (2014). Parallel feature selection inspired by group testing. *Advances in Neural Information Processing Systems* **27**, 3554–3562.

Zou, H. and Yuan, M. (2008). Composite quantile regression and the oracle model selection theory. *The Annals of Statistics* **36**, 1108–1126.

Canyi Chen

Institute of Statistics and Big Data, Renmin University of China, Beijing 100872, China.

E-mail: chency1997@ruc.edu.cn

Yuwen Gu

Department of Statistics, University of Connecticut, Storrs, CT 06269, USA.

E-mail: yuwen.gu@uconn.edu

Hui Zou

School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA.

E-mail: zouxx019@umn.edu

Liping Zhu

Institute of Statistics and Big Data and Center for Applied Statistics, Renmin University of China, Beijing 100872, China.

E-mail: zhu.liping@ruc.edu.cn

(Received March 2022; accepted September 2022)