Neurocomputing 448 (2021) 249–262

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Distributed quantile regression for massive heterogeneous data

Aijun Hu [a,b], Yuling Jiao [c], Yanyan Liu [c], Yueyong Shi [d,*], Yuanshan Wu [e]

[a] School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China
[b] School of Mathematics and Economics, Hubei University of Education, Wuhan, Hubei 430205, China
[c] School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei 430072, China
[d] School of Economics and Management, China University of Geosciences, Wuhan, Hubei 430074, China
[e] School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, Hubei 430073, China

ARTICLE INFO

Article history:
Received 22 February 2020
Revised 28 January 2021
Accepted 11 March 2021
Available online 18 March 2021
Communicated by Zidong Wang

2010 MSC:
62F12
62J05
62G35

Keywords:
ADMM
Communication-efficient
Distributed computing
Massive data
Quantile regression

ABSTRACT

Massive data sets pose great challenges to data analysis because of their heterogeneous data structure and limited computer memory. Jordan et al. (2019, *Journal of American Statistical Association*) has proposed a communication-efficient surrogate likelihood (CSL) method to solve distributed learning problems. However, their method cannot be directly applied to quantile regression because the loss function in quantile regression does not meet the smoothness requirement in CSL method. In this paper, we extend CSL method so that it is applicable to quantile regression problems. The key idea is to construct a surrogate loss function which relates to the local data only through subgradients of the loss function. The alternating direction method of multipliers (ADMM) algorithm is used to address computational issues caused by the non-smooth loss function. Our theoretical analysis establishes the consistency and asymptotic normality for the proposed method. Simulation studies and applications to real data show that our method works well.

1. Introduction

Massive data sets not only increase demand for data storage and computer performance, but also pose challenges to the existing statistical theory and computational methodology. In massive data analysis, the size of the data is so large that they often have to be stored in multiple machines, where each contains a subset of the data which can be of big size on its own right. In such situations, both the computational efficiency of statistical processing and the cost of communication between machines are of critical importance. In attempts to solve the problem, recent methodological developments in statistics have mainly focused on subsampling based methods [1,2] and divide-and-conquer (DC) or the one-shot methods [3–11]. DC has become popular for analyzing massive data because it is fast to compute and easy to implement. The key idea of DC is to first conduct statistical inference on each local machine separately and then aggregate the results from each subset to produce a final solution. One advantage of DC methods is that only one round of communication between machines is needed and the communication cost is therefore significantly reduced. However, as discussed in [12], DC methods are essentially average-based and have several drawbacks. For example, to achieve the minimax convergence rate, the number of subsets or machines where the data are stored can not be too large [10]. Particularly in cases where the goal is to estimate some unknown coefficients in a non-linear model, most of these DC methods can only improve estimate efficiency slightly in comparison with local estimates, and their estimation accuracy deteriorates as the number of machines/subsets gets small. More recently, Wang et al. [13] and Jordan et al. [12] have proposed a communication-efficient surrogate likelihood (CSL) procedure to solve this distributed statistical learning problem. The approach in [13] is designed for high-dimensional linear models, while the approach in [12] is more flexible and is applicable in different settings such as M-estimation, high-dimensional regularized estimation and Bayesian inference [14–17]. CSL approach is appealing

* Corresponding author.
E-mail addresses: haj2016@hust.edu.cn (A. Hu), yulingjiaomath@whu.edu.cn (Y. Jiao), liuyy@whu.edu.cn (Y. Liu), yueyongshi@cug.edu.cn (Y. Shi), wu@zuel.edu.cn (Y. Wu).

https://doi.org/10.1016/j.neucom.2021.03.041
0925-2312/

because it is communication cost effective as it only needs to exchange the gradients of local data. Furthermore, it has been shown that CSL approach is as efficient as the traditional likelihood method on entire dataset. Although CSL has many attractive properties, it can not be directly applied to quantile regression (QR) problems for the following reasons. Theoretically speaking, CSL assumes that the loss function has strong convexity and has Lipschitz-continuous second order derivatives. The loss function in QR problems, however, is non-smooth. Therefore, it is unclear whether those theoretical properties of CSL still hold in the context of QR. Numerically speaking, the commonly used Newton algorithm cannot be used to calculate the final solution in QR.

In this paper, we extend CSL to quantile regression by constructing a communication efficient surrogate loss function. As in [12], we first set one machine as the master machine and calculate the subgradients of quantile loss functions on each machine using the current values of parameters. Then we transfer these updated subgradients to the master machine to form a global subgradient and update the surrogate loss function as well as the parameter estimates via solving the surrogate loss function on the master machine. Our extension of [12] to QR problems is nontrivial, because replacing the smooth loss function in [12] by a non-smooth one fundamentally alters the theoretical analysis and raises significant computational issues. We develop the theory by adapting the results from M-estimation and convex processes. We prove the consistency and asymptotic normality for our method. To address the computational challenges, we apply the alternating direction method of multipliers (ADMM) algorithm [18] to obtain the final solution on the master machine. As pointed out in [19], ADMM is well suited for distributed convex optimization in large-scale problems.

There are several existing approaches for large-scale QR problems. For instance, Yang et al. [20,21] have proposed well-conditioned bases and subspace preserving sampling algorithms (SPC). There are two drawbacks with SPC methods. First, large storage space is needed for performing subsampling. It increases the required amount of primary memory. Second, only a part of the entire dataset are utilized. This results in loss of efficiency of their estimates. Xu et al. [22] have developed a block average quantile regression (BAQR) by taking average of the estimators obtained from each local machine using the standard quantile regression method. They have demonstrated through simulation studies that BAQR performs better than SPC methods in terms of predictive accuracy. BAQR still suffers from the aforementioned loss of efficiency, as with DC methods. In comparison, our method reduces such loss of efficiency by borrowing the strengths of CSL on efficient communication between local machines. Our numerical comparison with BAQR shows that our proposed method outperforms BAQR in terms of prediction accuracy. Furthermore, the proposed method performs almost as well as the oracle method (i.e., the standard quantile regression based on the entire dataset). Chen et al. [23,24] have proposed using smoothing approximation to the QR loss and then applying communication efficient Newton type methods to solve the corresponding minimization. In both [23,24] restrictions on the number of machines derived in [12] have been reduced. Here, in our paper we treat the nonsmooth QR loss directly without smoothing approximations. Wang and Lian [25] considered Lasso penalized QR models in high dimensions with $p > n$, while Chen et al. [23] and our paper focus on $p < n$ cases without penalty.

The rest of this paper is organized as follows. Communication-efficient distributed quantile regression is introduced in Section 2. In Section 3, ADMM algorithm is extended to solve the corresponding non-smooth optimization problem. We discuss the asymptotic properties of our proposed method in Section 4. Section 5 evaluates the finite-sample performance of the proposed method by simulation studies. Applications to real-world data are given in Section 6. Section 7 concludes the article. The theoretical proofs are outlined in the Appendix.

2. Distributed quantile regression for massive data

Compared with mean-based regression, quantile regression (QR) [26] provides a more accurate portrayal of the complex association between a response variable Y and covariates $\mathbf{x} = (x_1, \ldots, x_p)^\top$. Instead of modeling the conditional mean $\mathrm{E}(Y|\mathbf{x})$, QR models the whole conditional distribution of Y. Consider the following linear quantile regression model,

$$Q_\tau(Y|\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta}_0(\tau), \tag{1}$$

where $Q_\tau(Y|\mathbf{x})$is the conditional τ-th quantile function of response Y, and $\boldsymbol{\beta}_0(\tau)$ is a vector of regression coefficients depending on a specified quantile level τ with $\tau \in (0,1)$. Suppose that there are N i.i.d. samples $\{(y_i, \mathbf{x}_i)\}_{i=1,\ldots,N}$ from the model (1). For ease of expression, we suppress the argument τ in $\boldsymbol{\beta}_0(\tau)$ and denote them by $\boldsymbol{\beta}_0$ in the following. The estimates of $\boldsymbol{\beta}_0$ can be obtained by solving the following optimization problem,

$$\widehat{\boldsymbol{\beta}}_N = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \sum_{i=1}^{N} \rho_\tau(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}), \tag{2}$$

where $\rho_\tau(u) = u(\tau - I_{\{u\leqslant 0\}})$ is a check function and $I(\cdot)$ is the indicator function.

Consider the scenario that N is very large, where the solving of (2) is infeasible on a single machine. Assume that the data are distributed among K machines. For simplicity, we suppose that every machine holds the same sample size n, i.e. $N = nK$. For $k = 1, \ldots, K$, the sample on the k-th machine $\mathscr{M}_k$ is $\left\{y_{ki}, \mathbf{x}_{ki} = (x_{ki1}, \ldots, x_{kip})^\top,\ i = 1, \ldots, n\right\}$. For $k = 1, \ldots, K$, define

$$L_k(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_{ki} - \mathbf{x}_{ki}^\top \boldsymbol{\beta})$$

and

$$T_N(\boldsymbol{\beta}) = \frac{1}{N}\sum_{i=1}^{N} \rho_\tau(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}) = \frac{1}{K}\sum_{k=1}^{K} L_k(\boldsymbol{\beta})$$

to be the local and global QR loss functions, respectively. Without of loss of generality, we set the first machine as the master machine. Since the loss functions are not smooth as required by the CSL method, we define the surrogate loss function by replacing the gradient in [12] with a subgradient as follows,

$$\widetilde{L}(\boldsymbol{\beta}) := L_1(\boldsymbol{\beta}) + \langle \nabla T_N(\boldsymbol{\beta}^0) - \nabla L_1(\boldsymbol{\beta}^0), \boldsymbol{\beta} \rangle, \tag{3}$$

where $\boldsymbol{\beta}^0$ is any reasonable initial estimator of $\boldsymbol{\beta}_0$, $\langle \cdot, \cdot \rangle$ denotes the inner product, and $\nabla$is the subgradient. After some algebra, we have

$$\begin{cases} \nabla L_1(\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}}\left[\frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_{1i} - \mathbf{x}_{1i}^\top \boldsymbol{\beta})\right] = -\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_{1i}\psi_\tau(y_{1i} - \mathbf{x}_{1i}^\top \boldsymbol{\beta}), \\ \nabla T_N(\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}}\left[\frac{1}{N}\sum_{i=1}^{N} \rho_\tau(y_i - \mathbf{x}_i^\top \boldsymbol{\beta})\right] = -\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\psi_\tau(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}), \end{cases}$$

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

The estimator of $\boldsymbol{\beta}_0$ can be obtained by minimizing the surrogate loss function on the master machine, which is defined as

$$\widetilde{\boldsymbol{\beta}}_N = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \widetilde{L}(\boldsymbol{\beta}). \tag{4}$$

3. Algorithm

For simplicity, let $\mathbf{X}^{(1)} = (\mathbf{x}_{11}, \ldots, \mathbf{x}_{1n})^\top$ and $\mathbf{y}^{(1)} = (y_{11}, \ldots, y_{1n})^\top$ be the design matrix and the response vector on the first machine $\mathscr{M}_1$, respectively. Define $\mathbf{r} = \mathbf{y}^{(1)} - \mathbf{X}^{(1)}\boldsymbol{\beta}$ and $Q_\tau(\mathbf{r}) = n^{-1}\sum_{i=1}^n \rho_\tau(r_i)$, where $r_i\,(i = 1, \ldots, n)$ is the i-th element of $\mathbf{r}$. According to [12], solving (4) is equivalent to solving the following problem with linear constraint,

$$\min_{\boldsymbol{\beta},\mathbf{r}} Q_\tau(\mathbf{r}) + \mathbf{g}^\top\boldsymbol{\beta}, \quad \text{subject to} \quad \mathbf{X}^{(1)}\boldsymbol{\beta} + \mathbf{r} = \mathbf{y}^{(1)}, \tag{5}$$

where $\mathbf{g} = \nabla T_N(\boldsymbol{\beta}^0) - \nabla L_1(\boldsymbol{\beta}^0)$.

Noting that $Q_\tau(\mathbf{r})$ is not smooth at zero, the solution of (5) cannot be obtained by the Newton-Ralphson algorithm. To address this problem, we resort to the ADMM algorithm [19]. We first define the augmented Lagrangian function of (5) as follows,

$$\phi_\lambda(\boldsymbol{\beta}, \mathbf{r}, \boldsymbol{\theta}) = Q_\tau(\mathbf{r}) + \mathbf{g}^\top\boldsymbol{\beta} + \left\langle \boldsymbol{\theta}, \mathbf{X}^{(1)}\boldsymbol{\beta} + \mathbf{r} - \mathbf{y}^{(1)} \right\rangle + \frac{\lambda}{2}\left\| \mathbf{X}^{(1)}\boldsymbol{\beta} + \mathbf{r} - \mathbf{y}^{(1)} \right\|_2^2, \tag{6}$$

where $\lambda > 0$ is a fixed constant and $\boldsymbol{\theta} \in \mathbb{R}^n$ is the Lagrange multiplier. As a result, we arrive at the iteration for the standard ADMM algorithm as follows,

$$\begin{aligned} \boldsymbol{\beta}^{m+1} &= \arg\min_{\boldsymbol{\beta}} \phi_\lambda(\boldsymbol{\beta}, \mathbf{r}^m, \boldsymbol{\theta}^m), \\ \mathbf{r}^{m+1} &= \arg\min_{\mathbf{r}} \phi_\lambda(\boldsymbol{\beta}^{m+1}, \mathbf{r}, \boldsymbol{\theta}^m), \\ \boldsymbol{\theta}^{m+1} &= \boldsymbol{\theta}^m + \lambda\left(\mathbf{X}^{(1)}\boldsymbol{\beta}^{m+1} + \mathbf{r}^{m+1} - \mathbf{y}^{(1)}\right), \end{aligned} \tag{7}$$

where $(\boldsymbol{\beta}^m, \mathbf{r}^m, \boldsymbol{\theta}^m)$ denotes the m-th iteration of the algorithm.

Specifically, $\boldsymbol{\beta}^{m+1}$ has the following closed solution as

$$\boldsymbol{\beta}^{m+1} = \left(\mathbf{X}^{(1)\top}\mathbf{X}^{(1)}\right)^{-1}\left[\mathbf{X}^{(1)\top}\left(\mathbf{y}^{(1)} - \mathbf{r}^m - \boldsymbol{\theta}^m/\lambda\right) - \mathbf{g}/\lambda\right].$$

$\mathbf{r}^{m+1}$ also has a closed-form solution and can be implemented component-wise. Namely, for $i = 1, \ldots, n$, we have

$$\begin{aligned} r_i^{m+1} &= \arg\min_{r_i}\left\{\frac{1}{n}\rho_\tau(r_i) + \theta_i^m r_i + \frac{\lambda}{2}\left(\mathbf{x}_{1i}^\top\boldsymbol{\beta}^{m+1} + r_i - y_{1i}\right)^2\right\} \\ &= \arg\min_{r_i}\left\{\rho_\tau(r_i) + \frac{n\lambda}{2}\left[r_i - \left(y_{1i} - \mathbf{x}_{1i}^\top\boldsymbol{\beta}^{m+1} - \frac{1}{\lambda}\theta_i^m\right)^2\right]\right\} \mathrm{Prox}_{\rho_\tau}\left(y_{1i} - \mathbf{x}_{1i}^\top\boldsymbol{\beta}^{m+1} - \frac{1}{\lambda}\theta_i^m, n\lambda\right), \end{aligned} \tag{9}$$

where the proximal mapping operator $\mathrm{Prox}_{\rho_\tau}(\cdot,\cdot)$ is given in [27] as follows,

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

We summarize the above iterative procedure performed on the master machine in Algorithm 1.

Algorithm 1. QR-ADMM (QA).

Input: Data $\{y_{ki}, \mathbf{x}_{ki}, k = 1, \cdots, K, i = 1, \cdots, n\}$; constants $\lambda > 0$ and $\tau > 0$.

Initialize the algorithm with $\left(\boldsymbol{\beta}^0, \mathbf{r}^0, \boldsymbol{\theta}^0\right) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$ and the maximum number of iterations M ($M = 1000$ by default); set $m = 0$.

for $m = 0, 1, 2, \ldots, M-1$ **do**

until the convergence criterion is met.

Update $\boldsymbol{\beta}^{m+1}$ via (8).

Update $\mathbf{r}^{m+1}$ via (9).

Update $\boldsymbol{\theta}^{m+1}$ via (7).

end for

Output: $(\boldsymbol{\beta}^M, \mathbf{r}^M, \boldsymbol{\theta}^M)$.

In what follows, the communication-efficient distributed estimation is outlined in Algorithm 2.

Algorithm 2. Distributed algorithm for QR.

Input: Constants $\lambda > 0$ and $\tau > 0$. The number of rounds of communication $B = B(K)$, which is allowed to increase logarithmically with the total number of machines.

Initialize $\left(\boldsymbol{\beta}^0, \mathbf{r}^0, \boldsymbol{\theta}^0\right) = (\boldsymbol{\beta}^M, \mathbf{r}^M, \boldsymbol{\theta}^M)$ via Algorithm 1 by setting $\mathbf{g} = \mathbf{0}$.

for $b = 0, 1, 2, \ldots, B-1$ **do**

Broadcast the current values $\boldsymbol{\beta}^b$ to local machines $\mathscr{M}_2, \ldots, \mathscr{M}_K$;

Calculate the sub-gradient $\nabla L_k\left(\boldsymbol{\beta}^b\right)$ at local machines $\mathscr{M}_k$, $k = 1, \ldots, K$;

Reduce $\nabla L_k\left(\boldsymbol{\beta}^b\right)(k = 1, \ldots, K)$ to the master machine $\mathscr{M}_1$ to calculate the current value of $\mathbf{g}$ as $\mathbf{g}^b = \nabla T_N\left(\boldsymbol{\beta}^b\right) - \nabla L_1\left(\boldsymbol{\beta}^b\right)$ and update the augmented Lagrangian function $\phi_\lambda^b(\boldsymbol{\beta}, \mathbf{z}, \boldsymbol{\theta})$ by replacing $\mathbf{g}$ with $\mathbf{g}^b$ in (6);

Update $(\boldsymbol{\beta}^{b+1}, \mathbf{r}^{b+1}, \boldsymbol{\theta}^{b+1})$ via Algorithm 1 at the master machine $\mathscr{M}_1$.

end for

Output: $\boldsymbol{\beta}^B$ is regarded as the estimate of $\widetilde{\boldsymbol{\beta}}_N$ in (4).

4. Asymptotic properties

In this section, we present the theoretical properties of the proposed estimator $\widetilde{\boldsymbol{\beta}}_N$. We begin by introducing the regular conditions. Here $\boldsymbol{\beta}_0$ is the true parameter.

C1 The parameter space $\mathscr{B}$ is a compact subset of $\mathbb{R}^p$, and $\boldsymbol{\beta}_0$ is an inner point of $\mathscr{B}$.

C2 $\boldsymbol{\beta}_0$ is the unique minimizer of the objective function $\mathrm{E}\{\rho_\tau(Y - \mathbf{x}^\top\boldsymbol{\beta})\}$.

C3 The conditional distribution function $F_i(y) = P(Y \leqslant y|\mathbf{x}_i)$ is absolutely continuous in y. The corresponding conditional density $f_i(\cdot)$ is uniformly bounded away from 0 and ∞at each conditional τ-th quantile $\xi_i = Q_\tau(Y|\mathbf{x}_i) = \mathbf{x}_i^\top\boldsymbol{\beta}_0$.

C4 For $k = 1, \ldots, K$, there exists positive definite matrices Σ_0 and Σ_1 such that

(i) $\lim_{n\to\infty} n^{-1}\sum_{i=1}^n \mathbf{x}_{ki}\mathbf{x}_{ki}^\top = \Sigma_0$,

(ii) $\lim_{n\to\infty} n^{-1}\sum_{i=1}^n f_i(\xi_i)\mathbf{x}_{ki}\mathbf{x}_{ki}^\top = \Sigma_1$, where $\xi_i = \mathbf{x}_i^\top\boldsymbol{\beta}_0$,

(iii) $\max_{1\leqslant i\leqslant n} \|\mathbf{x}_{ki}\|_2/\sqrt{n} \to 0$,

where $\|\cdot\|_2$ is the L_2 norm, and the above convergence is almost everywhere. Note that the conditions C1–C4 are common in standard quantile regression [28], and the condition C4 is the technical condition, with (i) and (ii) being used in [22]. The following theorems show the consistency and asymptotic normality of the proposed estimators. Unless otherwise stated, all the limits are taken as $N \to \infty$.

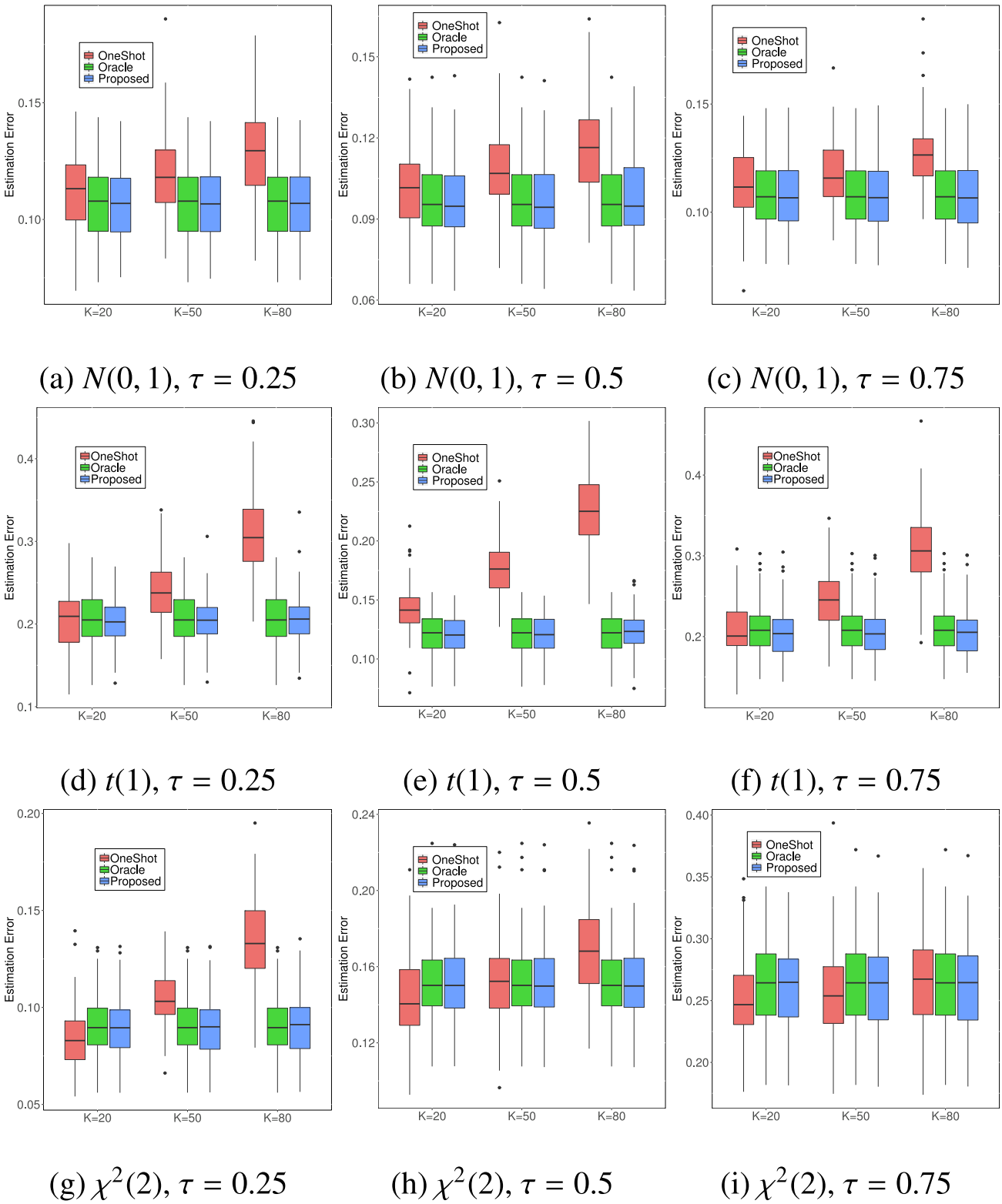

Fig. 1. Boxplots of various methods on simulated data for $K \in \{20, 50, 80\}$ and $\sigma = 0$ (the homogeneous case).

Theorem 1. *Under conditions C1–C4, we have* $\widetilde{\boldsymbol{\beta}}_N$ *converges weakly to* $\boldsymbol{\beta}_0$.

Theorem 2. *Under conditions C3 and C4, we have*

$$\sqrt{N}\left(\widetilde{\boldsymbol{\beta}}_N - \boldsymbol{\beta}_0\right) \xrightarrow{d} \mathrm{N}\left(\mathbf{0}, D\Sigma_1^{-1}\Sigma_0\Sigma_1^{-1}\right), \tag{10}$$

where $D = K\tau(1-\tau) + (K-1)\{\mathrm{Var}(\psi_\tau(\bar{\varepsilon}_i)) - 2\mathrm{Cov}(\psi_\tau(\bar{\varepsilon}_i), \psi_\tau(\varepsilon_i))\}$, $\bar{\varepsilon}_i = y_i - x_i^\top \boldsymbol{\beta}^0$, $\varepsilon_i = y_i - x_i^\top \boldsymbol{\beta}_0$, *and* $\xrightarrow{d}$ *represents convergence in distribution.* $\bar{\varepsilon}_i$ *and* ε_i *represent the residuals corresponding to the i-th observation value calculated with the initial value of any parameter and the real value of the parameter, respectively.*

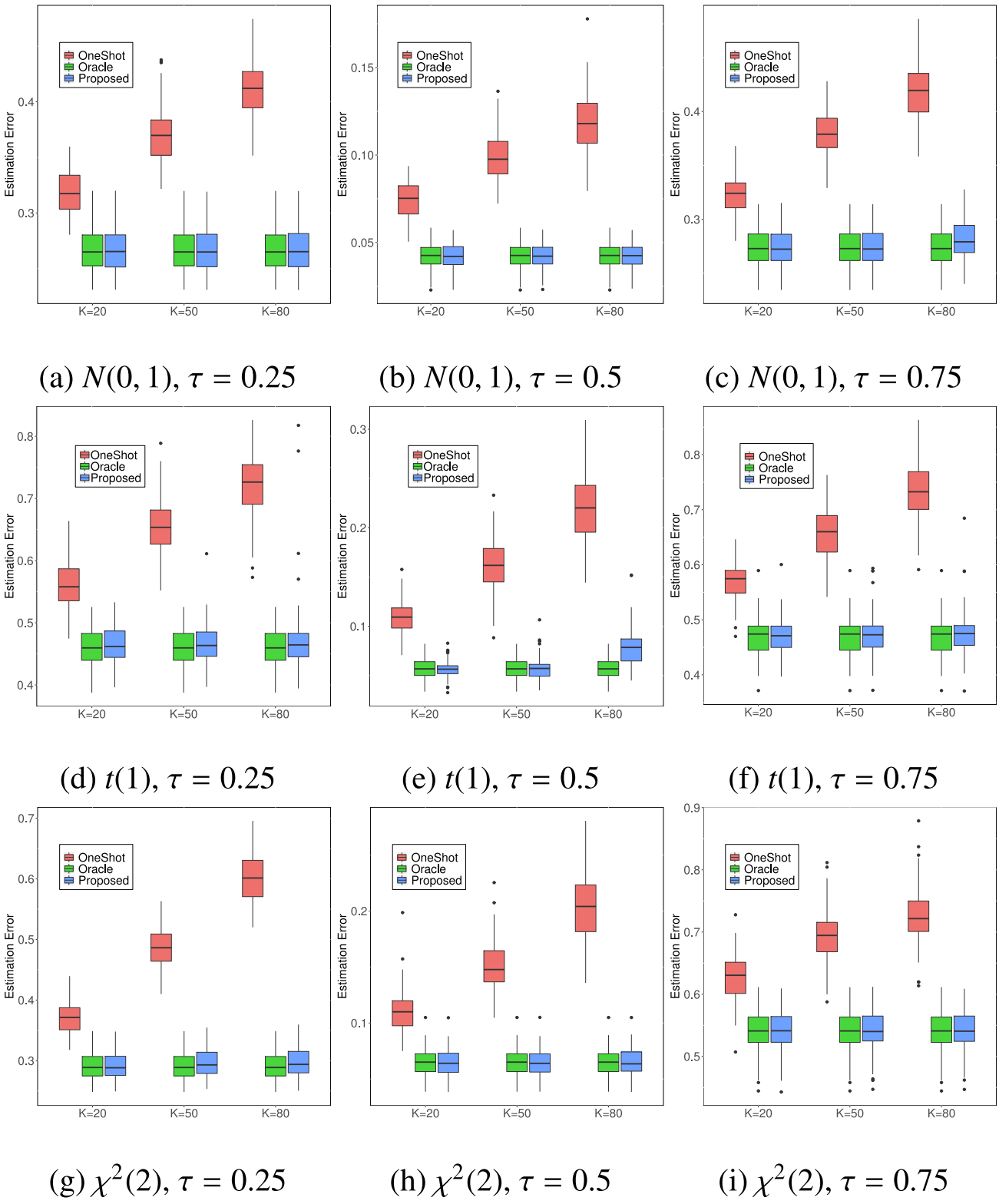

Fig. 2. Boxplots of various methods on simulated data for $K \in \{20, 50, 80\}$ and $\sigma = x_{ki,1}$ (the heterogeneous case).

Remark 1. If the initial value $\boldsymbol{\beta}^0$ satisfies $\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}_0\|_2 = O_p(n^{-1/2})$, then we can prove that $D = \tau(1-\tau)$. Therefore, the limiting distribution of $\widetilde{\boldsymbol{\beta}}_N$ in (10) is the same as that of the global QR estimator $\widehat{\boldsymbol{\beta}}_N$ based on the whole dataset. Generally, the initial value $\boldsymbol{\beta}^0$ can be obtained from the first machine and satisfies $\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}_0\|_2 = O_p(n^{-1/2})$. This means that our proposed estimator is asymptotically unbiased and applicable.

5. Numerical examples

We present numerical examples to illustrate the finite sample performance of the proposed method. All the experiments are performed in MATLAB R2010b on a laptop with a quad-core Intel Core i5 (2.60 GHz) CPU and 8 GB of RAM running 64-bit Windows 8.1. We compare the performance of the proposed method (Proposed) with other two approaches. The first one is referred to as the oracle approach (Oracle), which implements the standard quantile regression on the entire dataset. The second one is the averaging-based one-shot communication approach (OneShot), which is also known as the BAQR method in [22].

The data are generated from

$$y_{ki} = \mathbf{x}_{ki}^{\top}\boldsymbol{\beta}_0 + (1+\sigma)\epsilon_{ki}, \quad k = 1,\ldots,K, \quad i = 1,\ldots,n,$$

where the elements of $\boldsymbol{\beta}_0$ are generated from $U[-3,3]$. We consider two kinds of $\sigma : \sigma = 0$ for homogeneous data and $\sigma = x_{ki,1}$ for heterogeneous data, where $x_{ki,1}$ denotes the first element of $\mathbf{x}_{ki}$. The p-dimensional covariates $\mathbf{x}_{ki}$ are generated from a multivariate normal distribution $N_p(\mathbf{0}, \boldsymbol{\Sigma})$, with $\boldsymbol{\Sigma} = (\Sigma_{jl})_{p\times p}$, where $\Sigma_{jl} = 0.5^{|j-l|}$. We generate the error ϵ from three different distributions: the standard normal distribution $N(0,1)$, the t distribution with 1 degree of freedom $t(1)$, and the chi-squared distribution with 2 degrees of freedom $\chi^2(2)$. These three distributions represent a symmetric thin tail distribution, a symmetric thick tail distribution and an asymmetric distribution, respectively. We perform 100 simulations for each setting. We consider two cases of the sample size: the medium sample size and the large scale one.

5.1. Medium sample size case

Set the total sample size $N = 8000$ and the covariate dimension $p = 30$. The number of machines is set to be $K \in \{20, 50, 80\}$. Then, $n = \lfloor N/K \rfloor$ is the sample size on each machine, where $\lfloor a \rfloor$ denotes the integer part of a positive number a. For quantile regression, we consider three different values of $\tau : 0.25, 0.5$ and 0.75. We compare the performance of the aforementioned different methods in terms of the estimation error defined as $\|\widehat{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}_0\|_2$. Figs. 1 and 2 show the boxplots of the estimation error for homogeneous ($\sigma = 0$) and heterogeneous ($\sigma = x_{ki,1}$) data, respectively. Figs. 3 and 4 show the change of the estimation error with the rounds of communication for homogeneous and heterogeneous data, respectively. From Figs. 1 and 2, we make the following observations. The performance of our proposed method is very close to that of the Oracle method. In the OneShot method, the estimation error increases with the number of machines, but the estimation error of our proposed method is very robust to the number of machines. When $\sigma = 0$, the OneShot method performs the worst with the largest estimation error in most settings, especially when the distribution of ϵ is symmetric (e.g. $N(0,1)$ and $t(1)$). For heterogeneous ($\sigma = x_{ki,1}$) data, both our method and the Oracle method perform much better than the OneShot method.

From Figs. 3 and 4, we observe that the proposed method converges quickly and it clearly performs much better than the OneShot method after a few rounds of communication. On average, our proposed method only needs about 80 rounds of communication to reach the level close to the Oracle method. To reach the level close to the Oracle method, our method needs more rounds of communication for heterogeneous data than for the homogeneous data.

5.2. Large scale case

Set the total sample size $N = 50,000$, the dimension $p \in \{100, 150\}$, and the number of machines $K \in \{5, 10, 20, 50, 80, 100, 200\}$. We fix the quantile level $\tau = 0.25$ to compare the Proposed method with the OneShot method. All simulation results are based on 100 independent replications.

To compare the performance of two solvers, we use the average estimation error (AEE) over 100 independent repetitions defined as

$$100^{-1}\sum_{s=1}^{100}\|\widehat{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}\|_2.$$

The change of AEE of our method and the Oneshot method to the machine number K is shown in Fig. 5.

From Fig. 5, we observe that in the case of homogeneous data, if the number of machines is less than 50, our proposed method is comparable to the OneShot method. With increasingly larger K, our approach significantly outperforms the OneShot method. For heterogeneous data, even if the number of machines is small, our method outperforms the OneShot method with smaller AEEs. In addition, the AEEs of the OneShot method increase with increasing K, whereas those of our method are more stable with respect to K.

6. Application to real-world data

We illustrate our proposed method via two real data examples. We compare the proposed method with the Oracle and OneShot methods via cross-validation. To this end, we partition N samples 100 times. In each partition, we randomly select $2/3$ of the N samples as the training dataset and use the remaining as the test dataset. For all three methods, we compute the estimation coefficients using the training dataset D_{train} and then calculate the prediction error (PE) based on the test dataset D_{test}, where PE is defined as $\sum_{i\in D_{\text{test}}}\rho_\tau(y_i - \widehat{y}_i)$ [29]. We set the quantile level $\tau \in \{0.25, 0.5, 0.75\}$.

6.1. Analysis of wine quality data

We apply the proposed method to the wine quality data, which consists of two datasets, namely, red and white vinho verde wine samples from the north of Portugal. The datasets are available from the *UCI Machine Learning Repository* [1]. The goal is to model the wine quality based on physicochemical tests (cf., e.g., [30]). We focus on the white wine dataset, since it contains a larger number of samples. This dataset contains 4898 samples and 12 attribute variables, where the latter includes 11 physicochemical (explanatory) variables and one sensory (response) variable. Thus, the total sample size is $N = 4898$, and the dimensionality is $p = 11$. In the wine quality dataset, the score of response variable quality ranges from 3 to 9; the greater the score is, the better the wine quality. We transform the response variable into the natural logarithm form but use the original explanatory variables rather than their logarithms. The names of the explanatory variables and the corresponding explanations are listed in Table 1.

Set $K = 31$ and 62. Then the local sample size are $n = \lfloor N/K \rfloor = 158$ and 79, respectively. We use the 11 explanatory variables to predict the wine quality. The results are summarized

[1] The repository's website is located at https://archive.ics.uci.edu/ml/datasets/Wine+Quality

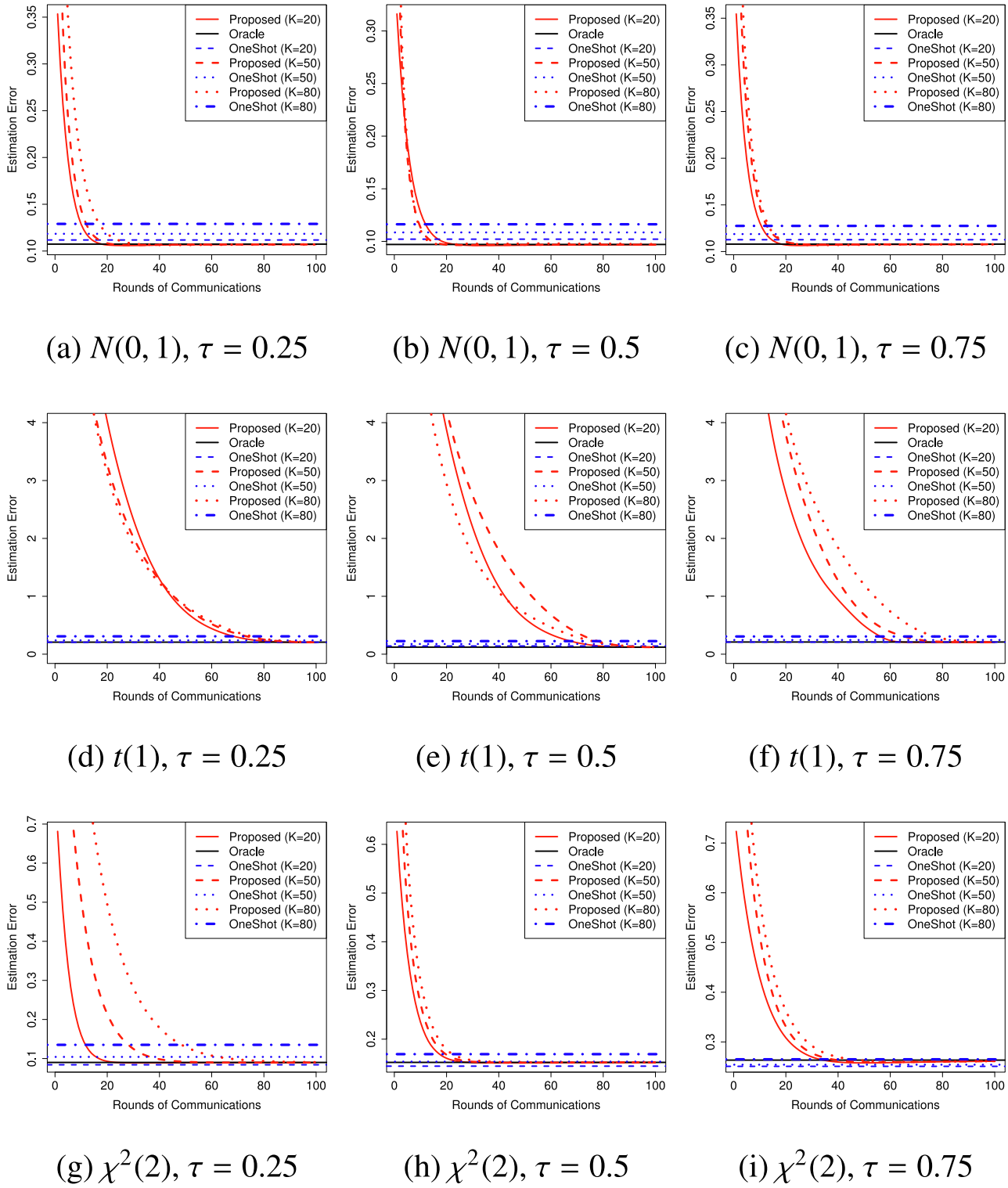

Fig. 3. Comparison of three methods in terms of estimation errors for $K \in \{20, 50, 80\}$ and $\sigma = 0$ (the homogeneous case) with 3 error distributions.

in Fig. 6. From Fig. 6, we observe clearly that at quantile $\tau = 0.5$, all the three methods are comparable.

6.2. Analysis of remote sensing data

The second example is a collection of geographic characteristics of Xuchang city in China, which contains 18,301 samples (observations) and 30 attribute variables. All the attribute variables are continuous. The response variable is the sandy land type named as "LengthWidth". The remaining 29 variables are predictors. Explanatory analysis shows that the distribution of "LengthWidth" is non-normal, therefore, the quantile regression approach may be more attractive than other methods in analyzing this data.

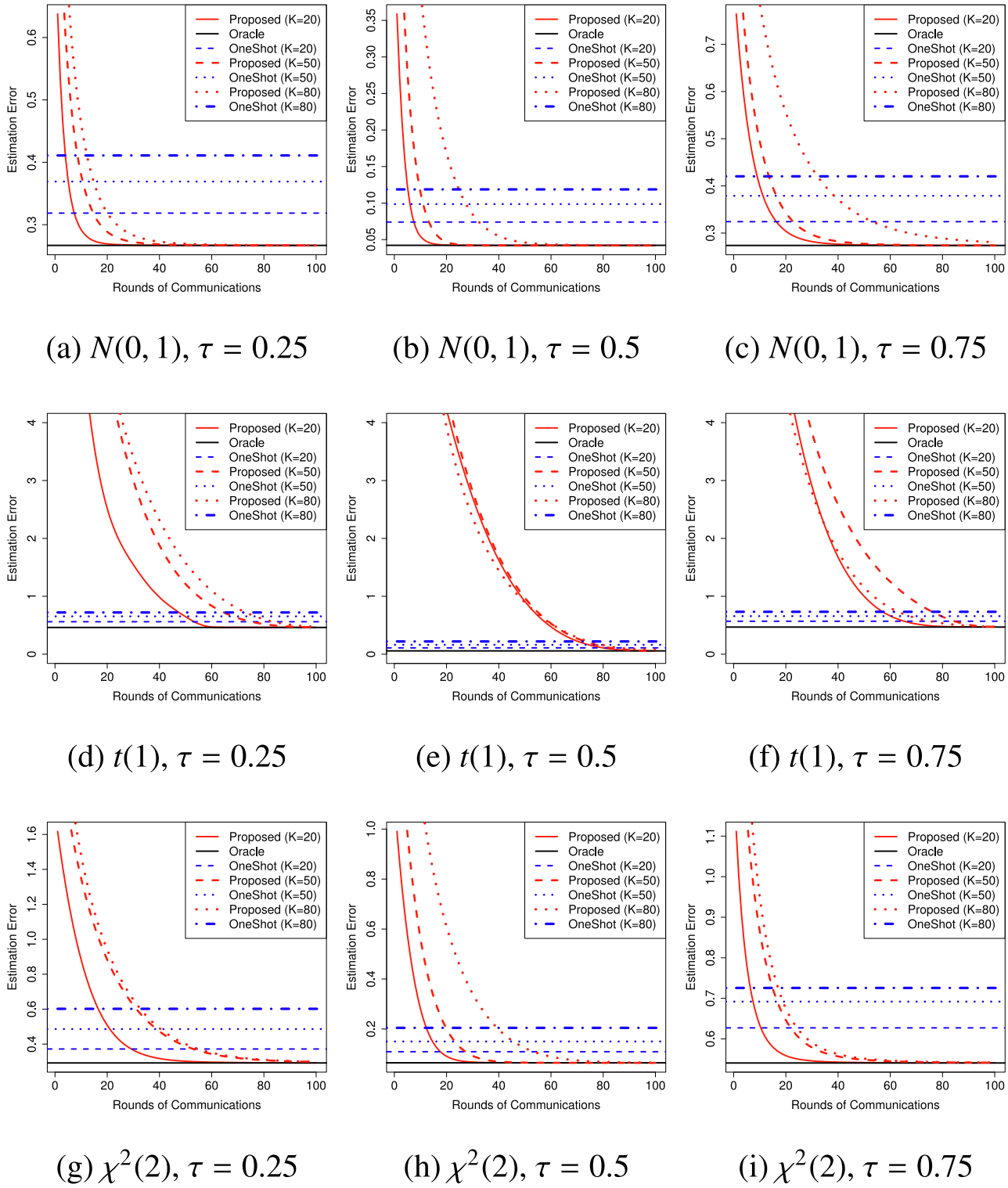

Fig. 4. Comparison of three methods in terms of estimation errors for $K \in \{20, 50, 80\}$ and $\sigma = x_{ki,1}$ (the heterogeneous case) with 3 error distributions.

The sizes of the training data and the test data are 12,000 and 6301, respectively. We set the number of machines $K \in \{40, 80\}$. Simulation results for the quantile levels $\tau = 0.25, 0.5$ and 0.75 are shown in Fig. 7. From Fig. 7, we see that the PEs for each of the three considered methods are smaller when $K = 40$ than those when $K = 80$. This is because the loss incurred by the partition data is larger as K increases. Our proposed method has smaller PEs than the OneShot method and is comparable to the Oracle method. These results indicate that, compared with the OneShot method, our proposed method is more precise and stable.

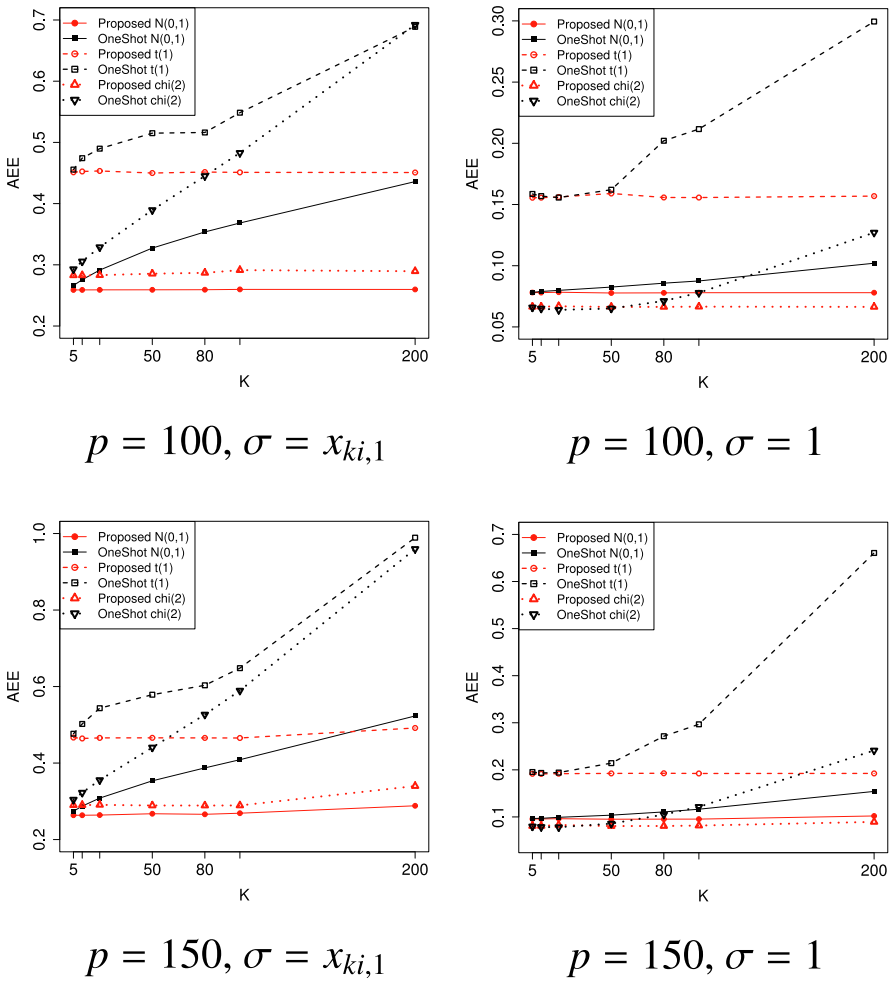

Fig. 5. Comparison of accuracy in terms of the average estimation error (AEE) between the proposed method and the OneShot method for $K \in \{5, 10, 20, 50, 80, 100, 200\}$ under three error distributions.

Table 1
Variable names and corresponding explanations for the wine quality data.

Name	Explanation
fixed.acidity	Most acids involved with wine or fixed or nonvolatile (do not evaporate readily)
volatile.acidity	Refers to the steam distillable acids present in wine
citric.acid	A weak organic acid that has the chemical formula $C_6H_8O_7$
residual.sugar	Refers to any natural grape sugars that are leftover after fermentation ceases
chlorides	A compound of chlorine with another element or group especially: a salt or ester of hydrochloric acid
free.sulfur.dioxide	A measure of the amount of SO_2 that is not bound to other molecules
total.sulfur.dioxide	A measure of both the free and bound forms of SO_2
density	The mass per unit volume of wine or must at 20°C
pH	A scale used to specify how acidic or basic a water-based solution is
sulphates	A chemical formed from sulphur, oxygen, and another element
alcohol	An organic substance formed when a hydroxyl group is substituted for a hydrogen atom in a hydrocarbon
quality	Score between 3 and 9

To further evaluate the influence of K on PE, we set $K \in \{5, 10, 20, 50, 80, 100, 200\}$. Fig. 8 shows the comparison of average PE over 100 repetitions for $\tau = 0.25, 0.5$ and 0.75. We can see in Fig. 8 that the Xuchang dataset is obviously heterogeneous. With increasing K, the average PE of both our method and the OneShot method increases.

However, the PE obtained by our method increases steadily with a lower rate while the PE of the OneShot method increases sharply, which further shows that our method is more accurate and stable than the OneShot method, especially for heterogeneous data.

7. Conclusions

For massive datasets, partition of data across multiple machines is the only practical way to overcome the limitation of storage and

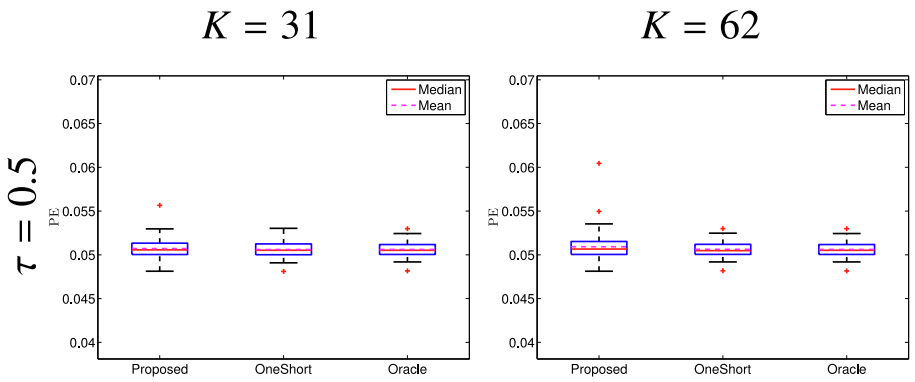

Fig. 6. Boxplots of PE for all methods using cross validation based on 100 random partitions of the wine quality dataset.

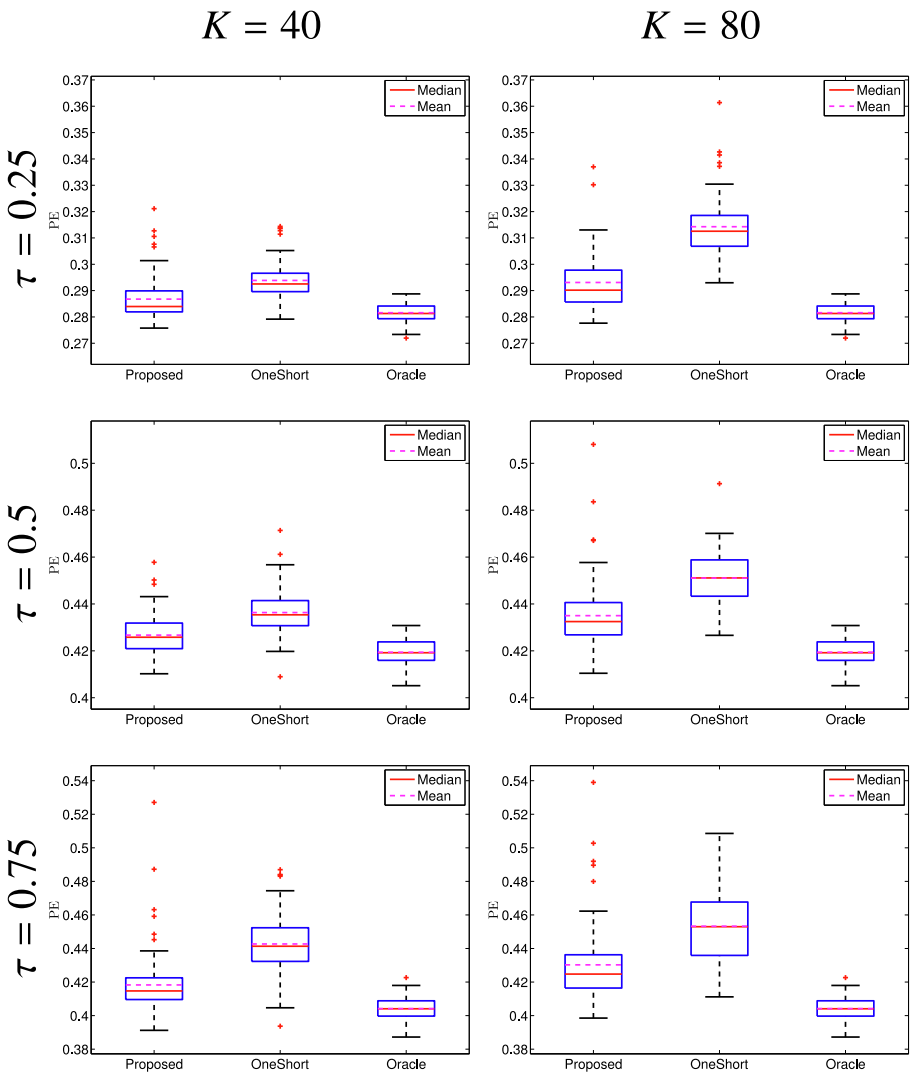

Fig. 7. Boxplots of PE for all methods using cross validation based on 100 random partitions of the remote sensing data.

computer memory. In this paper, we extend the CSL method in [12] to distributed quantile regression. Inspired by the idea of CSL, we proposed a communicate-efficient surrogate loss function to approximate the global loss function that uses all samples and obtain the estimate by minimizing the surrogate loss function at the master machine. At each iteration, only the subgradient of loss function at each local machine needs to be transferred to the master machine. Since the target loss function in quantile regression does not satisfy the smoothness assumption in CSL, the existing theoretical analysis and computing algorithms can not handle the problem properly. In our extension, we make use of convex process theory to establish the consistency of the proposed estima-

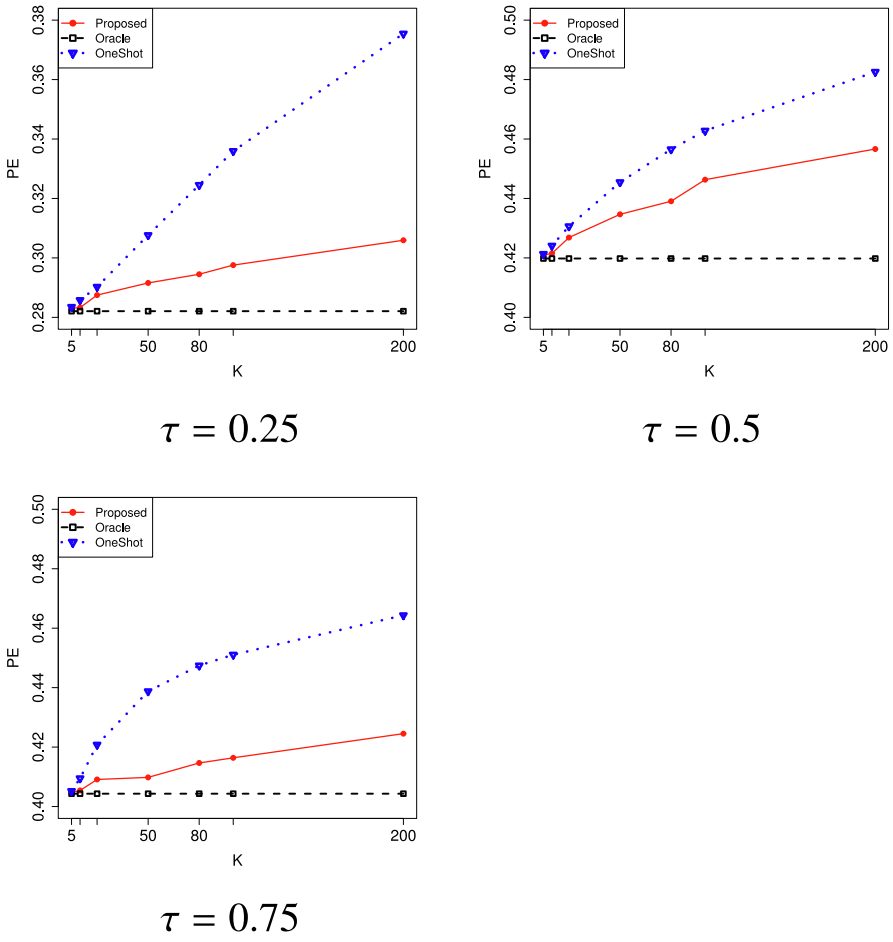

Fig. 8. The average prediction errors (PE) of all methods for $K \in \{5, 10, 20, 50, 80, 100, 200\}$ based on the remote sensing data.

tor. We also show that our distributed estimator has the same asymptotic distribution as the oracle estimate, which is based on the entire dataset. To address computational issues, we utilize ADMM to solve the non-smooth optimization problem. Simulation studies show that our proposed method is more accurate and stable than the OneShot method, especially in cases where the data are heterogeneous.

It is worth mentioning that the implementation of our proposed Algorithm 2 can be further improved in several ways. For example, at each iteration of the ADMM algorithm, we have to update the subgradient on each machine before the iteration can be completed, therefore the total computation speed is limited by the slowest computing machine. The asynchronous version of ADMM algorithm suggested by Zhang and Kwok [31] is helpful in addressing this limitation. In addition, splitting data along sample size and dimension directions (e.g., [32]) can be useful in extending our method to cases where the data is both large in size and high in dimension.

CRediT authorship contribution statement

Aijun Hu: Investigation, Formal analysis, Software, Visualization, Writing - original draft, Writing - review & editing. **Yuling Jiao:** Methodology, Validation, Writing - original draft, Writing - review & editing. **Yanyan Liu:** Resources, Supervision, Validation, Writing - review & editing. **Yueyong Shi:** Software, Visualization, Validation, Writing - original draft, Writing - review & editing. **Yuanshan Wu:** Conceptualization, Methodology, Data curation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant Nos. 11801531, 11971362, 11871474, 11671311). The authors would like to thank the Editor-in-Chief Steven Hoi, an associate editor and anonymous reviewers for their helpful and insightful comments that led to significant improvements in this article. We would also like to thank Dr. Y. Kang of Bentley University for his careful reading of this article and helpful suggestions in the writing.

Appendix A. Proofs

A.1. Proof of Theorem 1

We compute the negative subgradient of the objective function $\widetilde{L}(\boldsymbol{\beta})$ defined in (3) as

$$\begin{aligned}\Psi_n(\boldsymbol{\beta}) &:= -\nabla\widetilde{L}(\boldsymbol{\beta}) = -\nabla L_1(\boldsymbol{\beta}) + \nabla L_1(\boldsymbol{\beta}^0) - \nabla T_N(\boldsymbol{\beta}^0)\\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}) - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}^0)\\ &\quad + \frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{K}\mathbf{x}_{ji}\psi_\tau\left(y_{ji}-\mathbf{x}_{ji}^\top\boldsymbol{\beta}^0\right)\\ &= \frac{1}{n}\sum_{i=1}^{n}\left\{\mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}) - \mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}^0) + \frac{1}{K}\sum_{j=1}^{K}\mathbf{x}_{ji}\psi_\tau\left(y_{ji}-\mathbf{x}_{ji}^\top\boldsymbol{\beta}^0\right)\right\}.\end{aligned}$$

Denote

$$\Psi(\boldsymbol{\beta}) := \mathrm{E}\left[\mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}) - \mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}^0) + \frac{1}{K}\sum_{j=1}^{K}\mathbf{x}_{ji}\psi_\tau\left(y_{ji}-\mathbf{x}_{ji}^\top\boldsymbol{\beta}^0\right)\right].$$

Then we obtain

$$\Psi(\beta_0) = \mathrm{E}\left(\mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\beta_0)\right) = 0.$$

Because $\boldsymbol{\beta}_0$ is the solution of $\Psi(\boldsymbol{\beta})$, it is in the interior of a compact parameter space $\mathscr{B}$. Therefore, the consistency of $\widetilde{\boldsymbol{\beta}}_N$ is the direct conclusion of Theorem 5.9 in [33], which can be shown by verifying the following two conditions.

(a) $\sup_{\boldsymbol{\beta}\in\mathscr{B}}\|\Psi_n(\boldsymbol{\beta})-\Psi(\boldsymbol{\beta})\|_2 \xrightarrow{p} 0$, where $\xrightarrow{p}$ denotes converge in probability.
(b) $\inf_{\boldsymbol{\beta}:d(\boldsymbol{\beta},\boldsymbol{\beta}_0)\geqslant\epsilon}\|\Psi(\boldsymbol{\beta})\|_2 > 0 = \|\Psi(\boldsymbol{\beta}_0)\|_2$, where $d(\cdot,\cdot)$ is the Euclidean distance function and ϵ is any positive constant.

We first verify that condition (a) holds. Define

$$\begin{aligned}g(y_i,\mathbf{x}_i,\boldsymbol{\beta}) &= \mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}) - \mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}^0)\\ &\quad + \frac{1}{K}\sum_{j=1}^{K}\mathbf{x}_{ji}\psi_\tau\left(y_{ji}-\mathbf{x}_{ji}^\top\boldsymbol{\beta}^0\right).\end{aligned}$$

It is continuous with a probability of one at each $\boldsymbol{\beta}$ by condition C3. In addition,

$$\|g_\tau(y_i,\mathbf{x}_i,\boldsymbol{\beta})\|_2 \leqslant h(\mathbf{x}),$$

where $h(\mathbf{x}) = 2\|\mathbf{x}_{1i}\|_2 + \frac{1}{K}\sum_{j=1}^{K}\|\mathbf{x}_{ji}\|_2$. Under condition C4, we have $\mathrm{E}\|h(\mathbf{x})\|_2 < \infty$. Together with condition C1, it leads to the conclusion that condition (a) holds in accordance with Theorem 2 in [34].

Below, we verify that condition (b) holds. Note that

$$\Psi(\boldsymbol{\beta}) = \mathrm{E}\left[\mathbf{x}_{1i}\psi_\tau(y_{1i}-\mathbf{x}_{1i}^\top\beta)\right]$$

is a continuous function with respect to $\boldsymbol{\beta}$ under condition C3. By combining conditions C1 and C2, the above condition (b) is verified (see Problem 5.27 in [33]). This completes the proof of Theorem 1.

A.2. Proof of Theorem 2

For $\boldsymbol{\delta}\in\mathbb{R}^p$, define

$$\begin{aligned}Z_N(\boldsymbol{\delta}) &= \widetilde{L}\left(\boldsymbol{\beta}_0+\boldsymbol{\delta}/\sqrt{N}\right)\\ &= L_1\left(\boldsymbol{\beta}_0+\boldsymbol{\delta}/\sqrt{N}\right) - \left\langle\boldsymbol{\beta}_0+\boldsymbol{\delta}/\sqrt{N}, \nabla L_1(\boldsymbol{\beta}^0) - \nabla T_N(\boldsymbol{\beta}^0)\right\rangle\\ &= \frac{1}{n}\sum_{i=1}^{n}\rho_\tau\left(\varepsilon_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\delta}/\sqrt{N}\right)\\ &\quad - \left(\boldsymbol{\beta}_0+\boldsymbol{\delta}/\sqrt{N}\right)^\top\left[\nabla L_1(\boldsymbol{\beta}^0) - \nabla T_N(\boldsymbol{\beta}^0)\right],\end{aligned}$$

where $\varepsilon_{1i} = y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}_0$. Function $Z_N(\boldsymbol{\delta})$ is convex with respect to $\boldsymbol{\delta}$ and is minimized at $\widetilde{\boldsymbol{\delta}}_N = \sqrt{N}\left(\widetilde{\boldsymbol{\beta}}_N-\boldsymbol{\beta}_0\right)$. Furthermore, we have $\widetilde{\boldsymbol{\delta}}_N = \arg\min Z_N^{(1)}(\boldsymbol{\delta})$, where

$$Z_N^{(1)}(\boldsymbol{\delta}) = \sqrt{Nn}\left\{\frac{1}{n}\sum_{i=1}^{n}\left[\rho_\tau\left(\varepsilon_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\delta}/\sqrt{N}\right)-\rho_\tau(\varepsilon_{1i})\right] + \frac{1}{\sqrt{N}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{1i}^\top\boldsymbol{\delta}\psi_\tau(\bar{\varepsilon}_{1i}) - \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i^\top\boldsymbol{\delta}\psi_\tau(\bar{\varepsilon}_i)\right]\right\},$$

$\bar{\varepsilon}_{1i} = y_{1i}-\mathbf{x}_{1i}^\top\boldsymbol{\beta}^0$ and $\bar{\varepsilon}_i = y_i-\mathbf{x}_i^\top\boldsymbol{\beta}^0$. Following [35], it can be shown that the limiting distribution of $\widetilde{\boldsymbol{\delta}}_N$ is determined by the limiting behavior of function $Z_N^{(1)}(\boldsymbol{\delta})$. Using Knight's identity

$$\rho_\tau(u-v)-\rho_\tau(u) = -v\psi_\tau(u) + \int_0^v (I(u\leqslant s)-I(u\leqslant 0))ds,$$

and letting $v_i = \frac{1}{\sqrt{N}}\mathbf{x}_{1i}^\top\boldsymbol{\delta}$, we rewrite

$$\begin{aligned}Z_N^{(1)}(\boldsymbol{\delta}) &= Z_{1N}(\boldsymbol{\delta}) + Z_{2N}(\boldsymbol{\delta})\\ &\quad + \left[\frac{1}{\sqrt{n}}\sum_{i=1}^{n}x_{1i}^\top\boldsymbol{\delta}\psi_\tau(\bar{\varepsilon}_{1i}) - \frac{1}{\sqrt{K}}\frac{1}{\sqrt{N}}\sum_{i=1}^{N}\mathbf{x}_i^\top\boldsymbol{\delta}\psi_\tau(\bar{\varepsilon}_i)\right],\end{aligned}$$

where

$$\begin{aligned}Z_{1N}(\boldsymbol{\delta}) &= -\tfrac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}^\top\boldsymbol{\delta}\psi_\tau(\varepsilon_{1i}),\\ Z_{2N}(\boldsymbol{\delta}) &= \tfrac{\sqrt{Nn}}{n}\sum_{i=1}^{n}\textstyle\int_0^{v_i}(I(\varepsilon_{1i}\leqslant s)-I(\varepsilon_{1i}\leqslant 0))ds \equiv \sum_{i=1}^{n}Z_{2Ni}(\boldsymbol{\delta}).\end{aligned}$$

Using conditions C3 and C4, according to the Lindeberg-Feller central limit theorem, we obtain $Z_{1N}(\boldsymbol{\delta}) \xrightarrow{d} -\boldsymbol{\delta}^\top\mathbf{W}$, where $\mathbf{W}\sim\mathrm{N}(0,\tau(1-\tau)\Sigma_0)$. Now, we write the second term as

$$Z_{2N}(\boldsymbol{\delta}) = \sum_{i=1}^{n}\mathrm{E}Z_{2Ni}(\boldsymbol{\delta}) + \sum_{i=1}^{n}\left(Z_{2Ni}(\boldsymbol{\delta}) - \sum_{i=1}^{n}\mathrm{E}Z_{2Ni}(\boldsymbol{\delta})\right).$$

Under condition C4 (ii), we have

$$\begin{aligned}\sum_{i=1}^{n}\mathrm{E}Z_{2Ni}(\boldsymbol{\delta}) &= \sqrt{Nn}\frac{1}{n}\sum_{i=1}^{n}\int_0^{v_i}(F_i(\xi_i+s)-F_i(\xi_i))ds\\ &= \sqrt{Nn}\frac{1}{n}\frac{1}{N}\sum_{i=1}^{n}\int_0^{\mathbf{x}_{1i}^\top\boldsymbol{\delta}}\sqrt{N}\left(F_i\left(\xi_i+t/\sqrt{N}\right)-F_i(\xi_i)\right)dt\\ &= \frac{1}{\sqrt{K}}\frac{1}{n}\sum_{i=1}^{n}\int_0^{\mathbf{x}_{1i}^\top\boldsymbol{\delta}}f_i(\xi_i)tdt + o(1)\\ &= \frac{1}{\sqrt{K}}\frac{1}{2n}\sum_{i=1}^{n}f_i(\xi_i)\boldsymbol{\delta}^\top\mathbf{x}_{1i}\mathbf{x}_{1i}^\top\boldsymbol{\delta} + o(1) \to \frac{1}{\sqrt{K}}\frac{1}{2}\boldsymbol{\delta}^\top\Sigma_1\boldsymbol{\delta},\end{aligned}$$

almost surely. On the other hand, under condition C4 (iii),

$$\begin{aligned}\mathrm{Var}(Z_{2N}(\boldsymbol{\delta})) &= \sum_{i=1}^{n}\mathrm{Var}(Z_{2Ni}(\boldsymbol{\delta})) \leqslant \sum_{i=1}^{n}\mathrm{E}[Z_{2Ni}(\boldsymbol{\delta})]^2\\ &\leqslant \frac{1}{\sqrt{n}}\max_{1\leqslant i\leqslant n}|\mathbf{x}_i^\top\boldsymbol{\delta}|\sum_{i=1}^{n}\mathrm{E}Z_{2Ni}(\boldsymbol{\delta}) \to 0.\end{aligned}$$

It follows that

$$Z_{1N}(\boldsymbol{\delta}) + Z_{2N}(\boldsymbol{\delta}) \stackrel{d}{=} -\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_i^\top\boldsymbol{\delta}\psi_\tau(u_i) + \frac{1}{\sqrt{K}}\frac{1}{2}\boldsymbol{\delta}^\top\Sigma_1\boldsymbol{\delta},$$

where $\stackrel{d}{=}$ indicates that it has the same distribution on both sides. Consequently,

$$\begin{aligned}Z_N^{(1)}(\boldsymbol{\delta}) &\stackrel{d}{=} -\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}^\top\boldsymbol{\delta}\psi_\tau(\varepsilon_{1i}) + \frac{1}{\sqrt{K}}\frac{1}{2}\boldsymbol{\delta}^\top\Sigma_1\boldsymbol{\delta} + \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}^\top\boldsymbol{\delta}\psi_\tau(\bar{\varepsilon}_{1i}) - \frac{1}{\sqrt{K}}\frac{1}{\sqrt{N}}\sum_{i=1}^{N}\mathbf{x}_i^\top\boldsymbol{\delta}\psi_\tau(\bar{\varepsilon}_i)\\ &= \boldsymbol{\delta}^\top\left\{-\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(\varepsilon_{1i}) + \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(\bar{\varepsilon}_{1i}) - \frac{1}{\sqrt{K}}\frac{1}{\sqrt{N}}\sum_{i=1}^{N}\mathbf{x}_i\psi_\tau(\bar{\varepsilon}_i)\right\} + \frac{1}{\sqrt{K}}\frac{1}{2}\boldsymbol{\delta}^\top\Sigma_1\boldsymbol{\delta}\\ &= \boldsymbol{\delta}^\top\left\{-\xi+\eta-\frac{1}{\sqrt{K}}\zeta\right\} + \frac{1}{\sqrt{K}}\frac{1}{2}\boldsymbol{\delta}^\top\Sigma_1\boldsymbol{\delta},\end{aligned}$$

where

$$\xi = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(\varepsilon_{1i}) \xrightarrow{d} \mathrm{N}(0, \tau(1-\tau)\Sigma_0),$$

$$\boldsymbol{\eta} = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(\bar{\varepsilon}_{1i}), \zeta = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\mathbf{x}_i\psi_\tau(\bar{\varepsilon}_i),$$

$$\mathrm{Var}(\boldsymbol{\eta}) = \mathrm{Var}\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(\bar{\varepsilon}_{1i})\right) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{1i}\mathrm{Var}(\psi_\tau(\bar{\varepsilon}_i))\mathbf{x}_{1i}^\top \to \mathrm{Var}(\psi_\tau(\bar{\epsilon}_i))\Sigma_0.$$

Under conditions C3 and C4, by the Lindeberg-Feller central limit theorem, we have

$$\left(\boldsymbol{\xi}^\top, \boldsymbol{\eta}^\top, \boldsymbol{\zeta}^\top\right)^\top \xrightarrow{d} \mathrm{N}(0, \Omega),$$

where the symmetry matrix $\Omega = \{\Omega_{ij}\}_{1\leqslant i,j\leqslant 3}$ satisfies

$$\Omega_{11} = \tau(1-\tau)\Sigma_0, \quad \Omega_{22} = \Omega_{33} = \mathrm{Var}(\psi_\tau(\bar{\epsilon}_i))\Sigma_0$$

In addition,

$$\mathrm{Cov}(\xi, \boldsymbol{\eta}) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{1i}\mathrm{Cov}(\psi_\tau(\bar{\varepsilon}_{1i}), \psi_\tau(\varepsilon_{1i}))\mathbf{x}_{1i}^\top \xrightarrow{d} \mathrm{Cov}(\psi_\tau(\bar{\epsilon}_i), \psi_\tau(\epsilon_i))\Sigma_0 = \Omega_{21},$$

$$\mathrm{Cov}(\zeta, \xi) = \frac{1}{\sqrt{Nn}}\mathrm{Cov}\left(\sum_{i=1}^{N}\mathbf{x}_i\psi_\tau(\bar{\varepsilon}_i), \quad \sum_{i=1}^{n}\mathbf{x}_{1i}\psi_\tau(\varepsilon_{1i})\right)$$

$$= \frac{1}{\sqrt{K}}\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_{1i}\mathrm{Cov}(\psi_\tau(\bar{\varepsilon}_{1i}), \psi_\tau(\varepsilon_{1i}))\mathbf{x}_{1i}^\top \to \frac{1}{\sqrt{K}}\mathrm{Cov}(\psi_\tau(\bar{\epsilon}_i), \psi_\tau(\epsilon_i))\Sigma_0 = \Omega_{31}.$$

Similarly, we have $\mathrm{Cov}(\zeta, \boldsymbol{\eta}) \to \frac{1}{\sqrt{K}}\mathrm{Var}(\psi_\tau(\bar{\varepsilon}_i))\Sigma_0 = \Omega_{32}$.

Then, we obtain

$$-\xi + \boldsymbol{\eta} - \frac{1}{\sqrt{K}}\zeta$$

where $\alpha_K = \left(-I_{p\times p}, I_{p\times p}, -\frac{1}{\sqrt{K}}I_{p\times p}\right)$. Hence, we obtain

$$Z_N^{(1)}(\boldsymbol{\delta}) \overset{d}{=} \boldsymbol{\delta}^\top\left\{-\xi + \boldsymbol{\eta} - \frac{1}{\sqrt{K}}\zeta\right\} + \frac{1}{\sqrt{K}}\frac{1}{2}\boldsymbol{\delta}^\top\Sigma_1\boldsymbol{\delta} \xrightarrow{d} Z_0(\boldsymbol{\delta})$$

$$= \boldsymbol{\delta}^\top\mathbf{W}_0 + \frac{1}{\sqrt{K}}\frac{1}{2}\boldsymbol{\delta}^\top\Sigma_1\boldsymbol{\delta},$$

where $\mathbf{W}_0 \sim \mathrm{N}(\mathbf{0}, \mathbf{D}\Sigma_0/K)$. The convexity of the limiting objective function $Z_0(\boldsymbol{\delta})$ assures the uniqueness of the minimizer. Consequently, by computing the derivative of $Z_0(\boldsymbol{\delta})$ with respect to $\boldsymbol{\delta}$, we obtain the minimizer, given by

$$\tilde{\boldsymbol{\delta}}_0 = -\sqrt{K}\Sigma_1^{-1}\mathbf{W}_0.$$

Immediately, we obtain

$$\tilde{\boldsymbol{\delta}}_0 \xrightarrow{d} \mathrm{N}\left(0, D\Sigma_1^{-1}\Sigma_0\Sigma_1^{-1}\right).$$

Therefore,

$$\sqrt{N}\left(\tilde{\boldsymbol{\beta}}_N - \boldsymbol{\beta}_0\right) = \arg\min Z_N(\boldsymbol{\delta}) = \arg\min Z_N^{(1)}(\boldsymbol{\delta}) = \tilde{\boldsymbol{\delta}}_N \xrightarrow{d} \tilde{\boldsymbol{\delta}}_0$$

$$= \arg\min Z_0(\boldsymbol{\delta}),$$

which completes the proof of Theorem 2 by asymptotics for minimizers of the convex processes.

References

[1] A. Kleiner, A. Talwalkar, P. Sarkar, M.I. Jordan, A scalable bootstrap for massive data, J. Roy. Stat. Soc. Ser. B (Stat. Methodol.) 76 (4) (2014) 795–816.

[2] P. Ma, M.W. Mahoney, B. Yu, A statistical perspective on algorithmic leveraging, J. Mach. Learn. Res. 16 (1) (2015) 861–911.

[3] R. Mcdonald, M. Mohri, N. Silberman, D. Walker, G.S. Mann, Efficient large-scale distributed training of conditional maximum entropy models, Advances in Neural Information Processing Systems (2009) 1231–1239.

[4] L.W. Mackey, M.I. Jordan, A. Talwalkar, Divide-and-conquer matrix factorization, Advances in Neural Information Processing Systems (2011) 1134–1142.

[5] J.C. Duchi, A. Agarwal, M.J. Wainwright, Dual averaging for distributed optimization: convergence analysis and network scaling, IEEE Trans. Autom. Control 57 (3) (2012) 592–606.

[6] M.I. Jordan, On statistics, computation and scalability, Bernoulli 19 (4) (2013) 1378–1390.

[7] X. Chen, M.-G. Xie, A split-and-conquer approach for analysis of extraordinarily large data, Stat. Sin. 24 (4) (2014) 1655–1684.

[8] J.C. Duchi, M.I. Jordan, M.J. Wainwright, Y. Zhang, Optimality guarantees for distributed statistical estimation, arXiv preprint arXiv:1405.0782v2 (2014)..

[9] Y. Zhang, M.J. Wainwright, J.C. Duchi, Communication-efficient algorithms for statistical optimization, J. Mach. Learn. Res. 14 (1) (2013) 3321–3363.

[10] Y. Zhang, J. Duchi, M. Wainwright, Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates, J. Mach. Learn. Res. 16 (1) (2015) 3299–3340.

[11] J.D. Lee, Q. Liu, Y. Sun, J.E. Taylor, Communication-efficient sparse regression, J. Mach. Learn. Res. 18 (5) (2017) 1–30.

[12] M.I. Jordan, J.D. Lee, Y. Yang, Communication-efficient distributed statistical inference, J. Am. Stat. Assoc. 114 (526) (2019) 668–681.

[13] J. Wang, M. Kolar, N. Srebro, T. Zhang, Efficient distributed learning with sparsity, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR.org, 2017, pp. 3636–3645..

[14] O. Shamir, N. Srebro, T. Zhang, Communication-efficient distributed optimization using an approximate newton-type method, in: International Conference on Machine Learning, 2014, pp. 1000–1008.

[15] A. Garg, T. Ma, H. Nguyen, On communication cost of distributed statistical estimation and dimensionality, Advances in Neural Information Processing Systems (2014) 2726–2734.

[16] W. Neiswanger, C. Wang, E.P. Xing, Asymptotically exact, embarrassingly parallel mcmc, in: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, 2014, pp. 623–632.

[17] X. Wang, F. Guo, K.A. Heller, D.B. Dunson, Parallelizing MCMC with random partition trees, in: Advances in Neural Information Processing Systems, 2015, pp. 451–459..

[18] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Comput. Math. Appl. 2 (1) (1976) 17–40.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.

[20] J. Yang, X. Meng, M. Mahoney, Quantile regression for large-scale applications, in: International Conference on Machine Learning, 2013, pp. 881–887..

[21] J. Yang, X. Meng, M.W. Mahoney, Quantile regression for large-scale applications, SIAM J. Sci. Comput. 36 (5) (2014) S78–S110.

[22] Q. Xu, C. Cai, C. Jiang, F. Sun, X. Huang, Block average quantile regression for massive dataset, Stat. Pap. (2017) 1–25.

[23] X. Chen, W. Liu, Y. Zhang, Quantile regression under memory constraint, Ann. Stat. 47 (6) (2019) 3244–3273.

[24] X. Chen, W. Liu, X. Mao, Z. Yang, Distributed high-dimensional regression under a quantile loss function, J. Mach. Learn. Res. 21 (182) (2020) 1–43.

[25] L. Wang, H. Lian, Communication-efficient estimation of high-dimensional quantile regression, Anal. Appl. 18 (06) (2020) 1057–1075.

[26] R.W. Koenker, G. Bassett, Regression quantiles, Econometrica 46 (1) (1978) 33–50.

[27] Y. Gu, J. Fan, L. Kong, S. Ma, H. Zou, ADMM for high-dimensional sparse penalized quantile regression, Technometrics 60 (3) (2018) 319–331.

[28] R. Koenker, Quantile Regression, Cambridge University Press, New York, 2005.

[29] L. Wang, Y. Wu, R. Li, Quantile regression for analyzing heterogeneity in ultra-high dimension, J. Am. Stat. Assoc. 107 (497) (2012) 214–222.

[30] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst. 47 (4) (2009) 547–553.

[31] R. Zhang, J. Kwok, Asynchronous distributed ADMM for consensus optimization, in: International Conference on Machine Learning, 2014, pp. 1701–1709..

[32] L. Yu, N. Lin, ADMM for penalized quantile regression in big data, Int. Stat. Rev. 85 (3) (2017) 494–518.

[33] A.W. Van der Vaart, Asymptotic Statistics, Cambridge University Press, 1998.

[34] R.I. Jennrich, Asymptotic properties of non-linear least squares estimators, Ann. Math. Stat. 40 (2) (1969) 633–643.

[35] K. Knight, Limiting distributions for L_1 regression estimators under general conditions, Ann. Stat. 26 (2) (1998) 755–770.

Aijun Hu received his M.Sc. degree in probability and mathematical statistics from Wuhan University, Wuhan, China, in 2006. He is currently pursuing the Ph. D. degree in statistics from Huazhong University of Science and Technology, Wuhan, China. His research interests include statistical computing, big data analysis, and distributed computing.

Yuling Jiao received his Ph.D. degree in applied mathematics from Wuhan University in 2014. He is currently an Associate Professor with the School of Mathematics and Statistics, Wuhan University, Wuhan, China. He has authored or co-authored over 50 research papers, including SIAM Journal on Numerical Analysis, SIAM Journal on Scientific Computing, Journal of Machine Learning Research, Applied and Computational Harmonic Analysis, IEEE Transactions on Signal Processing, Inverse Problems, IEEE Signal Processing Letters, and Statistical Science. His current research interests include compressed sensing, inverse problem, sparse optimization, statistical computing, fast stochastic, parallel, distributed algorithms, and deep learning.

Yanyan Liu received her B. Sc. Degree from Wuhan University, Wuhan, China in 1989, and the Ph.D. degree from Wuhan University in 2001. She is currently a full professor in the School of Mathematics and Statistics, Wuhan University. She has authored or co-authored more than 50 research papers, including Biometrics, Biostatistics, Statistica Sinica, Lifetime Data Analysis, Scandinavian Journal of Statistics, Computational Statistics and Data Analysis, and Journal of Machine Learning Research. Her current research interests include statistical analysis for high dimensional data, machine learning, statistical computing, distributed algorithm and model average for large scale data.

Yueyong Shi received his Ph.D. degree in probability and mathematical statistics from Wuhan University in 2013. He is currently an Associate Professor with the School of Economics and Management, China University of Geosciences, Wuhan, China. He has published more than 10 papers including IEEE Transactions on Neural Networks and Learning Systems and Journal of Statistical Computation and Simulation. His current research interests include semiparametric models, high-dimensional data analysis, machine learning, and statistical computing.

Yuanshan Wu is currently a full professor in the School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, China. He received his Ph. D. degree in statistics from Wuhan University in 2010. He worked as Assistant Professor (2010–2015) and Associate Professor (2015–2018) at Wuhan University. His main research interests include high-dimensional data analysis, statistical machine learning, distributed algorithm for large-scale data. He has been published over 20 peer-viewed papers on the Journal of American Statistical Association, Biometrika, Bernoulli and so on.