Annals of the Institute of Statistical Mathematics (2022) 74:869–893
https://doi.org/10.1007/s10463-021-00816-0

Multi-round smoothed composite quantile regression for distributed data

Fengrui Di[1] · **Lei Wang**[1]

Received: 12 May 2021 / Revised: 25 September 2021 / Accepted: 26 November 2021 /
Published online: 10 January 2022

Abstract
Statistical analysis of large-scale dataset is challenging due to the limited memory constraint and computation source and calls for the efficient distributed methods. In this paper, we mainly study the distributed estimation and inference for composite quantile regression (CQR). For computational and statistical efficiency, we propose to apply a smoothing idea to the CQR loss function for the distributed data and then successively refine the estimator via multiple rounds of aggregations. Based on the Bahadur representation, we derive the asymptotic normality of the proposed multi-round smoothed CQR estimator and show that it also achieves the same efficiency of the ideal CQR estimator by analyzing the entire dataset simultaneously. Moreover, to improve the efficiency of the CQR, we propose a multi-round smoothed weighted CQR estimator. Extensive numerical experiments on both simulated and real data validate the superior performance of the proposed estimators.

Keywords Bahadur representation · Composite quantile regression · Divide-and-conquer · Multiple rounds · Kernel smoothing · Weighted composite quantile regression

1 Introduction

With the rapid development of science and technologies, massive data are increasingly being collected and stored in the distributed environment with many machines. Naturally, the traditional method, which processes all of data simultaneously in one central machine, is not practical due to the storage space, limited computational source and privacy problem. As a common and effective way to reduce the computational burden, the parallel and distributed estimation has attracted increasing

✉ Lei Wang
lwangstat@nankai.edu.cn

[1] School of Statistics and Data Science & LPMC, Nankai University, No. 94 Weijin Road, Tianjin 300071, China

attention in the statistical and machine learning literature. See Boyd et al. (2011), Dekel et al. (2012), Zhang et al. (2013) and the references therein. Among these distributed estimation methods, the divide-and-conquer (DC) approach has become the simplest and most popular method to deal with these challenges. The general DC framework firstly divides the entire dataset of sample size N on m machines with size $n = N/m$, then computes the local statistical estimator on each machine with smaller sample size n and outputs the calculation results, and finally combines the local estimators from each machine to obtain the global estimator. In this way, the information of entire dataset can be utilized. However, constructing the local calculation algorithm and combining the local results from each machine to make estimation statistically and computationally efficient are the main obstacles of implementing the DC method. In many existing DC studies, such as Chen and Xie (2014), Lee et al. (2017) and Battey et al. (2018), the global estimator is obtained by a simple average of the local estimators computed on each machine, which is called as the naive method in DC framework.

Various statistical models have been investigated based on the DC framework, examples include density parameter estimation (Li et al. 2013), M-estimator (Shi et al. 2018), least squares estimator (Fan et al. 2007) and so on. Since the large-scale data are collected from different locations and times, the homoscedasticity assumption may not hold such that the ordinary M-estimator and least squares estimator do not perform well. Volgushev et al. (2019) studied distributed inference for quantile regression (QR; Koenker and Bassett 1978) to provide more robust estimation and a complete picture of effects of the covariates on the response variable. However, the distributed QR estimator is hampered by the following three issues. Firstly, the loss function of the QR model is not differentiable at some points, which may cause some problems in the subsequent asymptotic analysis and computation. Secondly, the QR estimator is less efficient for certain light-tailed distributions (Zou and Yuan 2008; Gu and Zou 2020). Thirdly, the QR estimator considers only one quantile at a time and may not fully grasp the distributional information to produce more efficient estimation.

To address the first problem, Chen et al. (2019) circumvented the non-differentiability of the loss function by smoothing the indicator part of check function via a kernel function (Horowitz 1998; Whang 2006; Heller 2007; Kaplan and Sun 2017) and then developed a multi-round distributed approach for the QR estimation. The idea of smoothing the non-smooth QR loss function was firstly introduced by Horowitz (1998) and then has been widely applied to different areas of QR problem. For example, Whang (2006) applied the smoothed empirical likelihood method for the QR problem, Kaplan and Sun (2017) considered the smoothed estimating equations for instrumental QR and so on. However, different from adopting the traditional smoothing approach for calculating a one-stage estimator in the existing literature, we use this smoothing technique to construct multi-round smoothed estimators, which heavily rely on the first-order optimality condition of the loss function (Chen et al. 2019). To solve the last two problems, in this paper we consider the composite quantile regression (CQR; Zou and Yuan 2008), which is a mixture of the objective functions from different quantile regression models and can provide gains in estimation efficiency over the single

QR. Furthermore, Zou and Yuan (2008), Kai et al. (2010, 2011) and many others demonstrated that the CQR estimator is potentially much more efficient than the M-estimator and least squares estimator. In addition, different from the CQR estimation based on a sum of different quantile regressions with equal weights, Jiang et al. (2012) considered weighted composite quantile regression (WCQR) as a more efficient alternative to the regular CQR estimator. Given some appropriate weights, the WCQR estimator outperforms the CQR estimator when comparing asymptotic relative efficiency theoretically and numerically (Zhao and Lian 2016).

Distributed data with heteroscedasticity and demands for great computation and estimation efficiency encourage us to develop robust and efficient CQR and WCQR estimation methods. To the best of our knowledge, the multi-round distributed approach in conjunction with the smoothing idea has not previously been investigated for the CQR and WCQR models for the distributed data. Thus, we are motivated to adopt these approaches and establish theoretical properties for the smoothed CQR and WCQR estimators, which will significantly expand the applicability of Chen et al. (2019). Our contributions of this paper are in three aspects.

(1) We propose a multi-round smoothed CQR estimator for the distributed data. To illustrate our idea, we first apply the smoothing technique to the loss function of CQR based on the entire data, then set the derivatives of the smoothed loss function to zero and ultimately derive the explicit expressions of the CQR estimator, which only rely on initial value and individual data points. Motivated by the above concise formulation, we propose to design the local calculation form on each machine and construct the final estimator by adding up the local results as the components of derived CQR estimator expressions. The proposed estimation approach can use the last iteration result as the consistent initial value and successively refine the estimator via multiple rounds to improve the efficiency.

(2) We show the proposed multi-round smoothed CQR estimator has the following outstanding merits. Firstly, it can achieve the same efficiency as the ideal CQR estimator computed based on the entire data. Secondly, our proposed estimator improves the naive DC CQR framework. In statistical theory, since the local estimators are biased with the bias $O(1/n)$, the naive DC CQR estimator works on a small number of machines $m = o(\sqrt{N})$ and requires the large sample size on each machine to achieve the same asymptotic distribution as pooling the entire data together. However, these conditions are easily to be violated in practice, while our proposed method removes this strict constraint and achieves the same asymptotic efficiency through multiple rounds. Finally, our method only needs to solve one optimization problem to obtain the initial estimator during the whole process and the iterative process converges rapidly due to the consistent initial value and simple calculation formulas.

(3) To further improve the estimation efficiency based on the CQR model, a multi-round smoothed WCQR estimator for the distributed data is also proposed. Our simulation results based on $m = (5, 10, 25, 50)$ and $n = (50, 100, 2000)$ show that

our proposed multi-round smoothed CQR and WCQR estimators with only a few rounds of aggregations can achieve the same efficiency as the corresponding ideal CQR and WCQR estimators computed on the entire data and perform better than the naive DC estimators.

The rest of this paper is organized as follows. In Sect. 2, we propose the multi-round smoothed CQR estimator for the distributed data and give the asymptotic properties. In Sect. 3, we study the multi-round smoothed WCQR estimator. Sects. 4 and 5 show the simulation studies and an application of a real dataset. We summarize this paper in Sect. 6 and display our future work. All the proofs of theoretical results are given in the Supplementary Material.

2 Multi-round smoothed CQR estimation

2.1 Smoothed CQR estimator for the entire data

Given independent and identically distributed samples $(\boldsymbol{x}_i, y_i)$, $i = 1, \dots, N$, we consider the linear model as follows:

$$y_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \epsilon_i, \quad i = 1, \dots, N,$$

where y_i is a univariate response, $\boldsymbol{x}_i^T = (x_{i1}, \dots, x_{ip})$ is a vector of p-dimensional covariates, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^T$ is a true but unknown parameter vector, ϵ_i is an unobserved random variable. For multiple quantile levels $0 < \tau_1 < \cdots < \tau_K < 1$, when all of data fit into one machine, the ideal CQR estimator $(b_{\tau_1}, \dots, b_{\tau_K}, \boldsymbol{\beta})$ can be estimated by solving

$$(\check{b}_{\tau_1}, \dots, \check{b}_{\tau_K}, \check{\boldsymbol{\beta}}) = \underset{b_{\tau_1}, \dots, b_{\tau_K}, \boldsymbol{\beta}}{\arg\min} \sum_{k=1}^{K} \left\{ \sum_{i=1}^{N} \rho_{\tau_k}(y_i - b_{\tau_k} - \boldsymbol{x}_i^T \boldsymbol{\beta}) \right\}, \tag{1}$$

where $\rho_{\tau_k}(u) = u(I\{u > 0\} + \tau_k - 1)$ is the check function, $I\{\cdot\}$ is the indicator function, and b_{τ_k} is the τ_k-th quantile of error term ϵ. We usually use $\tau_k = k/(K + 1)$ for $k = 1, \dots, K$. The main challenge in the above CQR estimation is that the check function is piecewise linear and not differentiable such that the first-order optimization method can not be performed directly on (1).

To illustrate our smoothing idea, we first propose the smoothed CQR estimator based on the entire data and then extend this technique to the distributed data in Sect. 2.2. Motivated by Chen et al. (2019), we approximate the indicator function $I\{u > 0\}$ with a smooth kernel function $H(u/h)$, where $h \to 0$ is the bandwidth. The smooth function satisfies $H(+\infty) = 1$ and $H(-\infty) = 0$ such that $H(u/h) = 1$ when $u > 0$ and $H(u/h) = 0$ when $u < 0$. If it is possible to analyze the entire dataset, the smoothed loss function of the CQR model can be written as

$$L = \mathop{\arg\min}_{b_{\tau_1},\ldots,b_{\tau_K},\boldsymbol{\beta}} \sum_{k=1}^{K} \left\{ \sum_{i=1}^{N} (y_i - b_{\tau_k} - \boldsymbol{x_i}^T\boldsymbol{\beta}) \left[H\left(\frac{y_i - b_{\tau_k} - \boldsymbol{x_i}^T\boldsymbol{\beta}}{h} \right) + \tau_k - 1 \right] \right\}. \tag{2}$$

By the first-order optimality conditions on (2), the ideal smoothed CQR estimator $(\hat{b}_{\tau_1}, \ldots, \hat{b}_{\tau_K}, \hat{\boldsymbol{\beta}})$ satisfies

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \sum_{k=1}^{K}\sum_{i=1}^{N} \boldsymbol{x}_i \left\{ H\left(\frac{y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta}}{h} \right) + \tau_k - 1 + \frac{y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta}}{h} H'\left(\frac{y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta}}{h} \right) \right\} = 0,$$

$$\frac{\partial L}{\partial b_{\tau_k}} = \sum_{i=1}^{N} \left\{ H\left(\frac{y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta}}{h} \right) + \tau_k - 1 + \frac{y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta}}{h} H'\left(\frac{y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta}}{h} \right) \right\} = 0,$$

for $k = 1, \ldots, K$. Here, $H'(\cdot)$ denotes the first-order derivative. Given a set of consistent initial estimator $(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}})$, it is straightforward to deduce simple and explicit closed-form expressions of the smoothed CQR estimator $(\hat{b}_{\tau_1}, \ldots, \hat{b}_{\tau_K}, \hat{\boldsymbol{\beta}})$ for the entire data as follows:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{W}^{-1}\boldsymbol{M}, \quad \hat{b}_{\tau_k} = \boldsymbol{V}_k^{-1}\boldsymbol{U}_k, \quad k = 1, \ldots, K, \tag{3}$$

where

$$\boldsymbol{W} = \sum_{k=1}^{K}\sum_{i=1}^{N} \frac{\boldsymbol{x}_i\boldsymbol{x}_i^T}{h} H'\left(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h} \right), \quad \boldsymbol{V}_k = \sum_{i=1}^{N} \frac{1}{h} H'\left(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h} \right),$$

$$\boldsymbol{M} = \sum_{k=1}^{K}\sum_{i=1}^{N} \boldsymbol{x}_i \left\{ H\left(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h} \right) + \tau_k - 1 + \frac{y_i - \tilde{b}_{\tau_k}}{h} H'\left(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h} \right) \right\},$$

$$\boldsymbol{U}_k = \sum_{i=1}^{N} \left\{ H\left(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h} \right) + \tau_k - 1 + \frac{y_i - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h} H'\left(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h} \right) \right\}. \tag{4}$$

After obtaining $(\hat{b}_{\tau_1}, \ldots, \hat{b}_{\tau_K}, \hat{\boldsymbol{\beta}})$, we can employ it as a set of new initial estimator $(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}})$ to recalculate (3) until convergence.

Unfortunately, due to the storage, computation capacity and privacy, it is impractical to compute (3) in one machine based on the whole data. Recently, the DC method has become popular in statistical literature to handle the distributed data. A typical approach is one-shot simple averaging DC CQR, which is also called as the naive DC CQR method. In specific, assume that the total data indices $\{1, \ldots, N\}$ are divided into m subsets $\{\mathcal{H}_1, \ldots, \mathcal{H}_m\}$ with equal size $n = N/m$ and denote the data in the j-th local machine by $\mathcal{D}_j = \{\boldsymbol{x}_i, y_i : i \in \mathcal{H}_j\}$, $j = 1, \ldots, m$. The naive DC CQR method will firstly compute the classical CQR estimator $\check{\boldsymbol{\beta}}_{\boldsymbol{j}}$ on each $\mathcal{D}_j$ and then calculate the final CQR estimator by taking a simple average, i.e., $\hat{\boldsymbol{\beta}}_{Ave} = \sum_{j=1}^{m} \check{\boldsymbol{\beta}}_{\boldsymbol{j}}/m$.

However, as we mentioned in Sect. 1, the naive DC CQR estimator $\hat{\boldsymbol{\beta}}_{Ave}$ is suboptimal and our simulation results in Sect. 4 show that it does not perform well in most of the cases. Next, we will introduce the multi-round smoothed CQR estimator for the distributed data to overcome the existing problems in statistical and computational efficiency.

2.2 The proposed estimator

As we discussed in Sect. 2.1, the smoothed CQR estimator for the entire data in (3) is constructed by the quantities $\boldsymbol{W}$, $\boldsymbol{M}$, $\boldsymbol{V}_k$ and $\boldsymbol{U}_k$ defined in (4). While these formulas only involve the summation of matrices and vectors computed for each individual data point $\boldsymbol{x}_i$ and y_i. This concise formulation makes it simple to adopt the distributed method and greatly facilitates the distributed computing. Therefore, we propose the multi-round smoothed CQR estimator for the distributed data based on (3). We are able to compute the initial estimator using the traditional CQR estimation method based on a small part of samples, e.g., $\mathcal{D}_1$. The specific calculation steps can be described as follows:

S1: In the first iteration, using the traditional CQR estimation method, we obtain the initial value $(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}})$ based on $\mathcal{D}_1$.

S2: For each batch of data $\mathcal{D}_j$, $1 \le j \le m$, define the following quantities .

$$\boldsymbol{W}_j = \sum_{k=1}^{K} \sum_{i\in\mathcal{D}_j} \frac{\boldsymbol{x}_i\boldsymbol{x}_i^T}{h_j} H'\Big(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h_j}\Big), \quad \boldsymbol{V}_{kj} = \sum_{i\in\mathcal{D}_j} \frac{1}{h_j} H'\Big(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h_j}\Big),$$

$$\boldsymbol{M}_j = \sum_{k=1}^{K} \sum_{i\in\mathcal{D}_j} \boldsymbol{x}_i \Big\{ H\Big(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h_j}\Big) + \tau_k - 1 + \frac{y_i - \tilde{b}_{\tau_k}}{h_j} H'\Big(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h_j}\Big) \Big\},$$

$$\boldsymbol{U}_{kj} = \sum_{i\in\mathcal{D}_j} \Big\{ H\Big(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h_j}\Big) + \tau_k - 1 + \frac{y_i - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h_j} H'\Big(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h_j}\Big) \Big\}.$$

Notice that $(\boldsymbol{W}_j, \boldsymbol{M}_j, \boldsymbol{V}_{kj}, \boldsymbol{U}_{kj})$ can be calculated separately on the j-th machine and only the summed statistics $(\boldsymbol{W}_j, \boldsymbol{M}_j, \boldsymbol{V}_{kj}, \boldsymbol{U}_{kj})$ have to be stored and transferred to the central machine, $j = 1, \ldots, m$ and $k = 1, \ldots, K$

S3: After receiving $(\boldsymbol{W}_j, \boldsymbol{M}_j, \boldsymbol{V}_{kj}, \boldsymbol{U}_{kj})$ from all the machines, the central machine can aggregate them and compute

$$\hat{\boldsymbol{\beta}}^{(1)} = (\sum_{j=1}^{m} \boldsymbol{W}_j)^{-1}(\sum_{j=1}^{m} \boldsymbol{M}_j), \quad \hat{b}_{\tau_k}^{(1)} = (\sum_{j=1}^{m} \boldsymbol{V}_{kj})^{-1}(\sum_{j=1}^{m} \boldsymbol{U}_{kj}), \quad k = 1, \ldots, K. \tag{5}$$

S4: After the first round, $(\hat{b}_{\tau_1}^{(1)}, \ldots, \hat{b}_{\tau_K}^{(1)}, \hat{\boldsymbol{\beta}}^{(1)})$ can be treated as the new initial estimator $(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}})$ and then sent to all the machines to repeat the steps S2-S3 described above to construct the second round estimator, denoted as $\hat{\boldsymbol{\beta}}^{(2)}$. The algorithm is repeated q times until the q-th round estimator $\hat{\boldsymbol{\beta}}^{(q)}$ converges with a given threshold δ, and $\hat{\boldsymbol{\beta}}^{(q)}$ is taken to be the final estimator. The details of the entire inference procedure are presented in Algorithm 1. We name the final estimator $\hat{\boldsymbol{\beta}}^{(q)}$ in the distributed environment as the multi-round smoothed CQR (MSCQR) estimator.

For each iteration, we can choose different bandwidths h_j with the same order on different machines for $j = 1, \ldots, m$ in the step S2. This will not change the final asymptotic results of expressions (5). Therefore, in Sect. 2.3, we denote h as a common bandwidth for the proof of theoretical results. Since the initial estimator is consistent and calculation formulas are simple, these make the convergence fast.

Algorithm 1 Multi-round smoothed CQR estimation for the distributed data.

Input: Data batches $\mathcal{D}_1, \ldots, \mathcal{D}_m$, smooth function $H(x/h)$, quantiles $\tau_1, \ldots, \tau_K$, convergence threshold δ.

Output: $\hat{\boldsymbol{\beta}}^{(q)}$.

1: Set $g = 1$.

2: Calculate the initial value based on $\mathcal{D}_1$ using traditional CQR method:

$$(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}}) = \underset{b_{\tau_1}, \ldots, b_{\tau_k}, \beta}{\arg\min} \sum_{k=1}^{K} \Big\{ \sum_{i \in \mathcal{D}_1} \rho_{\tau_k}(y_i - b_{\tau_k} - \boldsymbol{x}_{\boldsymbol{i}}^T \boldsymbol{\beta}) \Big\}.$$

3: **for** $j = 1, \ldots, m$ **do**

4: Compute the bandwidth h_j on dataset $\mathcal{D}_j$ using $(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}})$.

5: Compute $(\boldsymbol{W}_j,\ \boldsymbol{M}_j)$ on dataset $\mathcal{D}_j$ using $(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}})$ with the bandwidth h_j.

6: **for** $k = 1, \ldots, K$ **do**

7: Compute $(\boldsymbol{V}_{kj},\ \boldsymbol{U}_{kj})$ on dataset $\mathcal{D}_j$ using $(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}})$ with the bandwidth h_j.

8: **end for**

9: **end for**

10: Compute

$$\hat{\boldsymbol{\beta}}^{(g)} = (\sum_{j=1}^{m} \boldsymbol{W}_j)^{-1} (\sum_{j=1}^{m} \boldsymbol{M}_j), \quad \hat{\boldsymbol{b}}_{\tau_k}^{(g)} = (\sum_{j=1}^{m} \boldsymbol{V}_{kj})^{-1} (\sum_{j=1}^{m} \boldsymbol{U}_{kj}), \quad k = 1, \ldots, K.$$

11: **while** $(\|\hat{\boldsymbol{\beta}}^{(g)} - \tilde{\boldsymbol{\beta}}\|_2 > \delta)$ **do**

12: $(\tilde{b}_{\tau_1}, \ldots, \tilde{b}_{\tau_K}, \tilde{\boldsymbol{\beta}}) = (\hat{b}_{\tau_1}^{(g)}, \ldots, \hat{b}_{\tau_K}^{(g)}, \hat{\boldsymbol{\beta}}^{(g)})$.

13: $g = g + 1$.

14: Repeat from 3.

15: **end while**

2.3 Asymptotic theories

In this subsection, our main objectives are to provide a Bahadur representation of the MSCQR estimator $\hat{\boldsymbol{\beta}}^{(q)}$ and to establish correspondingly asymptotic normality result. Firstly, from (3), we show that the difference between the smoothed CQR estimator after the first round and its true parameter can be equivalently rearranged as:

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} = \boldsymbol{Q}_N^{-1} \boldsymbol{P}_N, \tag{6}$$

where $\boldsymbol{Q}_N = \frac{1}{Nh} \sum_{k=1}^{K} \sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T H'\left(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}}{h}\right)$ and

$$\boldsymbol{P}_N = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N}\boldsymbol{x}_i\Big\{H\Big(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h}\Big) + \tau_k - 1 + \frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta}}{h}H'\Big(\frac{y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h}\Big)\Big\}.$$

Let $f(\cdot|\boldsymbol{x})$ be the conditional density function of the noise ϵ given $\boldsymbol{x}$. Define $\boldsymbol{P} = \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{N}\boldsymbol{x}_i\{I\{y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta} \geq 0\} + \tau_k - 1\}$ and $\boldsymbol{Q} = \sum_{k=1}^{K}E(\boldsymbol{x}\boldsymbol{x}^T f(b_{\tau_k}|\boldsymbol{x}))$. Then we state some regularity conditions for theoretical development and show that $\boldsymbol{P}_N$ and $\boldsymbol{Q}_N$ can be close to their corresponding population quantities $\boldsymbol{P}$ and $\boldsymbol{Q}$ when N is large.

(**C1**) The function $f(\cdot|\boldsymbol{x})$ is Lipschitz continuous ($|f(x_1|\boldsymbol{x}) - f(x_2|\boldsymbol{x})| \leq C|x_1 - x_2|$ for any $x_1, x_2 \in \mathbb{R}$ and some constant $C > 0$). There also exist constants c_1 and c_2 such that $0 < c_1 \leq \lambda_{\min}(\boldsymbol{Q}) \leq \lambda_{\max}(\boldsymbol{Q}) \leq c_2 < \infty$.

(**C2**) The smooth function $H(\cdot)$ is twice differentiable and its second derivative $H^{(2)}(\cdot)$ is bounded. Moreover, we assume the bandwidth $h = o(1)$.

(**C3**) $p = o(Nh/(\log(KN))$ as well as $\sup_{\|\theta\|_2=1} Ee^{\eta(\theta^T\boldsymbol{x})^2} < \infty$ for some $\eta > 0$.

(**C3**)* $p = o((N^{1-4\kappa}h/\log(KN))^{1/3})$ for some $\kappa > 0$ as well as $\sup_j E|x_{1j}|^a < \infty$ for some $a \geq 2/\kappa$ and $\sup_{\|\theta\|_2=1} E(\theta^T\boldsymbol{x})^4 < \infty$.

Condition (C1) contains the smoothness of the conditional density function $f(\cdot|\boldsymbol{x})$, which can be used to obtain the upper bounds of inequalities in the proof of Propositions 1 and 2, and involves a normal eigenvalue condition related to covariates $\boldsymbol{x}$. Condition (C2) is a mild condition on $H(\cdot)$ for the smooth approximation and can be easily satisfied by a properly chosen $H(\cdot)$. Conditions (C3) and (C3)* illustrate the relationship between the dimension p and sample size N, and the moment conditions on covariates $\boldsymbol{x}$ are also presented. However, Condition (C3) has weaker constraints compared with Condition (C3)*. Both Conditions (C3) and (C3)* can reach the same theoretical conclusions in the Propositions 1 and 2. Under these conditions, we derive the following asymptotic analysis of $\boldsymbol{P}_N$ and $\boldsymbol{Q}_N$, respectively. The proofs are relegated to the Supplementary Material.

Proposition 1 *Under conditions (C1), (C2) and (C3) (or (C3)*), assume that the initial estimator $(\tilde{b}_{\tau_1}, \dots, \tilde{b}_{\tau_k}, \tilde{\boldsymbol{\beta}})$ satisfies $\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 = O_P(a_n)$ and $|\tilde{b}_{\tau_k} - b_{\tau_k}| = O_P(b_n)$ for $k = 1, \dots, K$, in which $a_n = O(h)$ and $b_n = O(h)$. We have*

$$\|\boldsymbol{P}_N - \boldsymbol{P}\|_2 = O_P\Big(\sqrt{\frac{ph\log(kN)}{N}} + a_n^2 + b_n^2 + h^2\Big).$$

Proposition 2 *Under the same conditions in Proposition 1, we have*

$$\|\boldsymbol{Q}_N - \boldsymbol{Q}\|_2 = O_P\Big(\sqrt{\frac{p\log(kN)}{Nh}} + a_n + b_n + h\Big).$$

Combining Propositions 1 and 2, the expression (6) with some algebraic manipulations, we have

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} = \boldsymbol{Q}^{-1}\boldsymbol{P} + r_N, \tag{7}$$

with $\|r_N\|_2 = O_P\Big(\sqrt{\frac{p^2\log(KN)}{N^2h}} + \sqrt{\frac{ph\log(KN)}{N}} + a_n^2 + b_n^2 + h^2\Big)$.

Remark 1 When h shrinks at an appropriate rate, we find that the dominant item of r_N shrinks from a_n to a_n^2, while the initial estimator only can attain $||\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}||_2 = O_P(a_n)$. This result indicates that an iterative refinement of the initial estimator will significantly improve the estimation accuracy of $\hat{\boldsymbol{\beta}}$. Therefore, our proposed method could obtain the MSCQR estimator to achieve the desirable estimate efficiency by successively refining the initial estimator only based on data from the first machine after q iterations.

Remark 2 The similar process as Propositions 1 and 2 can be applied to $\hat{b}_{\tau_k}$ for $k = 1, \dots, K$, we have

$$\hat{b}_{\tau_k} - b_{\tau_k} = \frac{1}{N}\sum_{i=1}^{N}\{I\big\{y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta} \geq 0\big\} + \tau_k - 1\}/E(f(b_{\tau_k}|\boldsymbol{x})) + r_N',$$

with $\|r_N'\|_2 = O_P\Big(\sqrt{\frac{p^2\log N}{N^2h}} + \sqrt{\frac{ph\log N}{N}} + a_n^2 + b_n^2 + h^2\Big)$.

According to our Algorithm 1, the previous discussions only involve the asymptotic behaviors after one round aggregation. Based on the above arguments, the theoretical results for our MSCQR estimator $\hat{\boldsymbol{\beta}}^{(q)}$ in Algorithm 1 can also be concluded. By a recursive argument based on (7) and setting the obtained estimator as the new initial value $(\tilde{b}_{\tau_1}, \dots, \tilde{b}_{\tau_k}, \tilde{\boldsymbol{\beta}})$, we establish the following Bahadur representation of $\hat{\boldsymbol{\beta}}^{(q)}$, where the main term is still $\boldsymbol{Q}^{-1}\boldsymbol{P}$.

Theorem 1 *Under the same conditions in Proposition 1, the initial estimator* $(\tilde{b}_{\tau_1}, \dots, \tilde{b}_{\tau_k}, \tilde{\boldsymbol{\beta}})$ *in the first iteration satisfies* $\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 = O_P(\sqrt{p/n})$ *and* $|\tilde{b}_{\tau_k} - b_{\tau_k}| = O_P(\sqrt{1/n})$ *for* $k = 1, \dots, K$ *and* $p = O(n/(\log(KN))^2)$. *We have*

$$\hat{\boldsymbol{\beta}}^{(q)} - \boldsymbol{\beta} = \boldsymbol{Q}^{-1}\boldsymbol{P} + r_N, \tag{8}$$

with $\|r_N\|_2 = O_P\Big(\sqrt{\frac{ph^{(q)}\log(KN)}{N}}\Big)$.

Remark 3 The classical initial CQR estimator based on a single machine will satisfy $\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2 = O_P(\sqrt{p/n})$ and $|\tilde{b}_{\tau_k} - b_{\tau_k}| = O_P(\sqrt{1/n})$. Furthermore, we should point out that any initial estimator satisfying the above conditions in Theorem 1 can be used in the first iteration and the same Bahadur representation in Theorem1 holds. The condition $p = O(n/(\log(KN))^2)$ is used for balancing the terms in r_N in (7).

By applying the central limit theorem to (8), we derive the following result on the asymptotic distribution of the MSCQR estimator $\hat{\boldsymbol{\beta}}^{(q)}$.

Theorem 2 *Under the conditions in Theorem1, for the MSCQR estimator* $\hat{\boldsymbol{\beta}}^{(q)}$, *we have*

$$\sqrt{N}(\hat{\boldsymbol{\beta}}^{(q)} - \boldsymbol{\beta}) \Rightarrow N\Big(0, \sum_{k,k'=1}^{K} \min(\tau_k, \tau_{k'})(1 - \max(\tau_k, \tau_{k'}))\boldsymbol{Q}^{-1}E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{Q}^{-1}\Big),$$

with $\boldsymbol{Q} = \sum_{k=1}^{K} E(\boldsymbol{x}\boldsymbol{x}^T f(b_{\tau_k}|\boldsymbol{x}))$ *as* n *and* $N \to \infty$.

Remark 4 In order to construct confidence intervals for $\hat{\boldsymbol{\beta}}^{(q)}$, the consistent estimators of $\boldsymbol{Q}$ and $E[\boldsymbol{x}\boldsymbol{x}^T]$ are needed. Motivated by Proposition 2 and Chen et al. (2019), we propose to use $\boldsymbol{Q}_N$ in the q-th iteration and $\sum_{i=1}^{N} \boldsymbol{x}_i\boldsymbol{x}_i^T/N$ to estimate $\boldsymbol{Q}$ and $E[\boldsymbol{x}\boldsymbol{x}^T]$, respectively. It is convenient to obtain $\boldsymbol{Q}_N$ and $\sum_{i=1}^{N} \boldsymbol{x}_i\boldsymbol{x}_i^T/N$, since they can be separately calculated on each machine when computing the MSCQR for the distributed data and then taken for a simple summation.

Remark 5 Theorem2 shows that $\hat{\boldsymbol{\beta}}^{(q)}$ achieves the same asymptotic efficiency as $\check{\boldsymbol{\beta}}$ in (1) computed directly on the entire samples. When p is fixed, although the naive DC CQR method also can achieve the same efficiency, it requires a small number of machines, i.e., $m = o(\sqrt{N})$, to achieve better performance. However, in some applications such as sensor networks, the number of batches can be large. While our proposed approach removes the restriction by applying multiple rounds of aggregations. Since these three methods achieve the same asymptotic distribution, we could use the same asymptotic variance as the MSCQR when constructing the confidence intervals of above mentioned methods.

3 Multi-round smoothed WCQR estimation

The above CQR estimator is investigated based on a sum of different quantile regressions with equal weights and may not be optimal. Let $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_K)^T$ be a vector of weights and the components in the weight vector $\boldsymbol{\omega}$ are allowed to be negative. Jiang et al. (2012) proposed the WCQR estimator $(b_{\tau_1}, \ldots, b_{\tau_K}, \boldsymbol{\beta})$ by solving

$$(\check{b}^{\omega}_{\tau_1}, \ldots, \check{b}^{\omega}_{\tau_K}, \check{\boldsymbol{\beta}}^{\omega}) = \underset{b_{\tau_1},\ldots,b_{\tau_K},\boldsymbol{\beta}}{\arg\min} \sum_{k=1}^{K} \Big\{ \sum_{i=1}^{N} \omega_k \rho_{\tau_k}(y_i - b_{\tau_k} - \boldsymbol{x}_i^T\boldsymbol{\beta}) \Big\}.$$

Similarly, the multi-round smoothed WCQR (MSWCQR) estimator for the distributed data can also be obtained by using our proposed four-step method in Sect. 2. Given the consistent and suitable initial estimator $(\tilde{b}^{\omega}_{\tau_1}, \ldots, \tilde{b}^{\omega}_{\tau_K}, \tilde{\boldsymbol{\beta}}^{\omega})$, in the step S2, we define

$$\boldsymbol{W}_j^{\omega} = \sum_{k=1}^{K}\sum_{i\in\mathcal{D}_j}\frac{\omega_k \boldsymbol{x}_i\boldsymbol{x}_i^T}{h_j}H'\Big(\frac{y_i-\tilde{b}_{\tau_k}^{\omega}-\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}^{\omega}}{h_j}\Big),\ \ \boldsymbol{V}_{kj}^{\omega} = \sum_{i\in\mathcal{D}_j}\frac{1}{h_j}H'\Big(\frac{y_i-\tilde{b}_{\tau_k}^{\omega}-\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}^{\omega}}{h_j}\Big),$$

$$\boldsymbol{M}_j^{\omega} = \sum_{k=1}^{K}\sum_{i\in\mathcal{D}_j}\omega_k\boldsymbol{x}_i\Big\{H\Big(\frac{y_i-\tilde{b}_{\tau_k}^{\omega}-\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}^{\omega}}{h_j}\Big)+\tau_k-1+\frac{y_i-\tilde{b}_{\tau_k}^{\omega}}{h_j}H'\Big(\frac{y_i-\tilde{b}_{\tau_k}^{\omega}-\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}^{\omega}}{h_j}\Big)\Big\},$$

$$\boldsymbol{U}_{kj}^{\omega} = \sum_{i\in\mathcal{D}_j}\Big\{H\Big(\frac{y_i-\tilde{b}_{\tau_k}^{\omega}-\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}^{\omega}}{h_j}\Big)+\tau_k-1+\frac{y_i-\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}^{\omega}}{h_j}H'\Big(\frac{y_i-\tilde{b}_{\tau_k}^{\omega}-\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}^{\omega}}{h_j}\Big)\Big\},$$

and then in the step S3 we obtain

$$\hat{\boldsymbol{\beta}}^{\omega(1)} = (\sum_{j=1}^{m}\boldsymbol{W}_j^{\omega})^{-1}(\sum_{j=1}^{m}\boldsymbol{M}_j^{\omega}),\ \hat{b}_{\tau_k}^{\omega(1)} = (\sum_{j=1}^{m}\boldsymbol{V}_{kj}^{\omega})^{-1}(\sum_{j=1}^{m}\boldsymbol{U}_{kj}^{\omega}),\quad k=1,\dots,K.$$

The MSWCQR for the distributed data algorithm is shown in Algorithm 2.

Theorem 3 *Under the conditions in Theorem1 and* $\sum_{k=1}^{K}\omega_k = 1$, *for the MSWCQR estimator* $\hat{\boldsymbol{\beta}}^{\omega(q)}$, *we have*

$$\sqrt{N}(\hat{\boldsymbol{\beta}}^{\omega(q)}-\boldsymbol{\beta}) \Rightarrow N\Big(0,\sum_{k,k'=1}^{K}\omega_k\omega_{k'}\min(\tau_k,\tau_{k'})(1-\max(\tau_k,\tau_{k'}))(\boldsymbol{Q}^{\omega})^{-1}E[\boldsymbol{x}\boldsymbol{x}^T](\boldsymbol{Q}^{\omega})^{-1}\Big),$$

with $\boldsymbol{Q}^{\omega} = \sum_{k=1}^{K}\omega_k E(\boldsymbol{x}\boldsymbol{x}^T f(b_{\tau_k}|\boldsymbol{x}))$ *as n and* $N\to\infty$.

Remark 6 $\boldsymbol{Q}^{\omega}$ can be estimated by $\boldsymbol{Q}_N^{\omega} = \frac{1}{Nh}\sum_{k=1}^{K}\sum_{i=1}^{N}\omega_k\boldsymbol{x}_i\boldsymbol{x}_i^T H'\Big(\frac{y_i-\tilde{b}_{\tau_k}-\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}}{h}\Big)$.

As pointed in Jiang et al. (2016), the optimal weight $\boldsymbol{\omega}_{opt}$ is

$$\boldsymbol{\omega}_{opt} = \Omega^{-1}f,$$

where $f=(f(b_{\tau_1}),\dots,f(b_{\tau_K}))^T$ and Ω is a $K\times K$ matrix with $\Omega_{kk'}=\min(\tau_k,\tau_{k'})(1-\max(\tau_k,\tau_{k'}))$ for $k=1,\dots,K$. In practice, $\boldsymbol{\omega}_{opt}=\Omega^{-1}f$ can be estimated by $\hat{\boldsymbol{\omega}}_{opt}=\Omega^{-1}\hat{f}=(\hat{\omega}_1,\dots,\hat{\omega}_K)^T$. Furthermore, the usual nonparametric density estimation methods, such as kernel smoothing based on the estimated residuals, can provide a consistent estimator $\hat{f}$.

Algorithm 2 Multi-round smoothed WCQR estimation for the distributed data.

Input: Data batches $\mathcal{D}_1, \ldots, \mathcal{D}_m$, smooth function $H(x/h)$, quantiles $\tau_1, \ldots, \tau_K$, weights $\omega_1, \ldots, \omega_K$, convergence threshold δ.

Output: $\hat{\boldsymbol{\beta}}^{\omega(q)}$.

1: Set $g = 1$.
2: Given consistent initial estimator $(\tilde{b}^{\omega}_{\tau_1}, \ldots, \tilde{b}^{\omega}_{\tau_K}, \tilde{\boldsymbol{\beta}}^{\omega})$.
3: **for** $j = 1, \ldots, m$ **do**
4: Compute the bandwidth h^{ω}_j on dataset $\mathcal{D}_j$ using $(\tilde{b}^{\omega}_{\tau_1}, \ldots, \tilde{b}^{\omega}_{\tau_K}, \tilde{\boldsymbol{\beta}}^{\omega})$.
5: Compute $(\boldsymbol{W}^{\omega}_j,\ \boldsymbol{M}^{\omega}_j)$ on dataset $\mathcal{D}_j$ using $(\tilde{b}^{\omega}_{\tau_1}, \ldots, \tilde{b}^{\omega}_{\tau_K}, \tilde{\boldsymbol{\beta}}^{\omega})$ with the bandwidth h^{ω}_j.
6: **for** $k = 1, \ldots, K$ **do**
7: Compute $(\boldsymbol{V}^{\omega}_{kj},\ \boldsymbol{U}^{\omega}_{kj})$ on dataset $\mathcal{D}_j$ using $(\tilde{b}^{\omega}_{\tau_1}, \ldots, \tilde{b}^{\omega}_{\tau_K}, \tilde{\boldsymbol{\beta}}^{\omega})$ with the bandwidth h^{ω}_j.
8: **end for**
9: **end for**
10: Compute

$$\hat{\boldsymbol{\beta}}^{\omega(g)} = (\sum_{j=1}^{m} \boldsymbol{W}^{\omega}_j)^{-1}(\sum_{j=1}^{m} \boldsymbol{M}^{\omega}_j),\ \hat{b}^{\omega(g)}_{\tau_k} = (\sum_{j=1}^{m} \boldsymbol{V}^{\omega}_{kj})^{-1}(\sum_{j=1}^{m} \boldsymbol{U}^{\omega}_{kj}), \quad k = 1, \ldots, K.$$

11: **while** $(\|\hat{\boldsymbol{\beta}}^{\omega(g)} - \tilde{\boldsymbol{\beta}}^{\omega}\|_2 > \delta)$ **do**
12: $(\tilde{b}^{\omega}_{\tau_1}, \ldots, \tilde{b}^{\omega}_{\tau_K}, \tilde{\boldsymbol{\beta}}^{\omega}) = (\hat{b}^{\omega(g)}_{\tau_1}, \ldots, \hat{b}^{\omega(g)}_{\tau_K}, \hat{\boldsymbol{\beta}}^{\omega(g)})$.
13: $g = g + 1$.
14: Repeat from 3.
15: **end while**

4 Simulations

In this section, we provide simulation experiments to illustrate the performance of our proposed estimators. The data are generated from a linear regression model

$$y_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \epsilon_i, \quad \text{for} \quad i = 1, \ldots, N,$$

where $x_i = (x_{i1}, \ldots, x_{ip})^T$ is generated from a p-dimensional normal distribution with mean being $(0, \ldots, 0)^T$ and covariance matrix Σ being a $p \times p$ symmetric matrix with $\Sigma_{jj'} = 4 \times 0.5^{|j-j'|}$ for $1 \leq j \leq j' \leq p$. The true value of $\boldsymbol{\beta} = \mathbf{1}_p$. The errors ϵ_i are generated independently from the following homogeneous and quadratic heteroscedastic distributions: (1) $\epsilon_i \sim N(0, 4^2)$; (2) $\epsilon_i \sim t(2)$; (3) $\epsilon_i \sim exp(1)$; (4) $\epsilon_i \sim \chi^2(2)$; (5) $\epsilon_i = (0.5 + 0.5(x_{i1})^2)e_i$ and $e_i \sim N(0, 1)$; (6) $\epsilon_i = (0.5 + 0.5(x_{i1})^2)e_i$ and $e_i \sim t(2)$; (7) $\epsilon_i = (0.5 + 0.5(x_{i1})^2)e_i$ and $e_i \sim exp(1)$; (8) $\epsilon_i = (0.5 + 0.5(x_{i1})^2)e_i$ and $e_i \sim \chi^2(2)$.

In the simulations, let $\tau_k = k/(1+K)$ for $k = 1, \dots, K$ and we consider $K = 5$, $m = (5, 10, 25, 50)$, $(n, p) = (50, 10)$, (100, 20) and (2000, 50). The Gaussian kernel is employed as the smooth function $H(\cdot)$. All of the simulations are based on 500 replications. Furthermore, we include the following four competitors in the simulations.

(1) $\hat{\boldsymbol{\beta}}^{(q)}$: the proposed MSCQR estimator on the distributed data;
(2) $\hat{\boldsymbol{\beta}}_{Cen}$: the central CQR estimator, which is computed on the entire data using traditional CQR model;
(3) $\hat{\boldsymbol{\beta}}_{Ave}$: the naive DC CQR estimator on the distributed data, which is computed by taking a simple average of traditional local CQR estimators on each machine;
(4) $\hat{\boldsymbol{\beta}}_{Sub}$: the subsample CQR estimator, which is only computed on the dataset in one single machine using traditional CQR model.

The corresponding four WCQR estimators $\hat{\boldsymbol{\beta}}^{\omega(q)}$, $\hat{\boldsymbol{\beta}}^{\omega}_{Cen}$, $\hat{\boldsymbol{\beta}}^{\omega}_{Ave}$, $\hat{\boldsymbol{\beta}}^{\omega}_{Sub}$ are also computed. Here, the estimators $\hat{\boldsymbol{\beta}}^{(q)}$, $\hat{\boldsymbol{\beta}}_{Cen}$, $\hat{\boldsymbol{\beta}}_{Ave}$, $\hat{\boldsymbol{\beta}}_{Sub}$ are used to compute the optimal weights, respectively. Thus, eight different estimators are included. For our proposed estimators, we choose the bandwidth $h_j = 1.5\sigma_{\hat{\epsilon}_j}(Kn)^{-1/3}$ on j-th machine, where $\sigma_{\hat{\epsilon}_j}$ means the sample standard deviation of $\hat{\epsilon}_j = y_i - \tilde{b}_{\tau_k} - \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}, i \in \mathcal{D}_j, k = 1, \dots, K$.

4.1 RMSE and MAD

To access the accuracy of our proposed estimators in terms of estimation errors, we compute root of mean square error (RMSE) and mean absolute deviation (MAD) of $\hat{\boldsymbol{\beta}}$:

$$\text{RMSE} = \sqrt{\frac{1}{p}\sum_{j=1}^{p}(\hat{\beta}_j - \beta_j)^2}, \quad \text{MAD} = \frac{1}{p}\sum_{j=1}^{p}|\hat{\beta}_j - \beta_j|.$$

When $(n, p) = (50, 10)$ and (100, 20), the simulated RMSEs and MADs with eight different errors for the CQR and WCQR estimators are given in Tables 1, 2, 3 and 4. When $(n, p) = (2000, 50)$, due to the computation time issue, we only present the simulated RMSEs and MADs for the CQR estimators under four errors in Table 5. A few conclusions can be drawn from the simulation results.

(1) As shown in Tables 1 and 2, for any given number of machines m, as expected, the RMSEs and MADs of the subsample CQR estimator $\hat{\boldsymbol{\beta}}_{Sub}$ are the largest because it only uses the local data on one machine. The naive DC estimator $\hat{\boldsymbol{\beta}}_{Ave}$ can reduce the RMSEs and MADs of $\hat{\boldsymbol{\beta}}_{Sub}$ by averaging; however, these values of $\hat{\boldsymbol{\beta}}_{Ave}$ are still larger than those of $\hat{\boldsymbol{\beta}}^{(q)}$ and $\hat{\boldsymbol{\beta}}_{Cen}$. Under the normal error, the RMSEs and MADs of our proposed estimator $\hat{\boldsymbol{\beta}}^{(q)}$ are comparable with these of the central estimator $\hat{\boldsymbol{\beta}}_{Cen}$, while our proposed CQR estimator has the smallest RMSEs and MADs in the other errors, which is in accord with our Remark 5. The similar phenomenon was also reported by Chen et al. (2019) for the multi-round QR model. As the number of machines m increases, the RMSEs and

Table 1 Simulated RMSEs $\times 10$ under eight errors with $(n, p) = (50, 10)$

m	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{\omega(q)}$	$\hat{\beta}^{\omega}_{Cen}$	$\hat{\beta}^{\omega}_{Ave}$	$\hat{\beta}^{\omega}_{Sub}$
$\epsilon_i \sim N(0, 4^2)$								
5	1.623	1.593	1.808	4.036	1.625	1.596	1.856	4.010
10	1.171	1.149	1.296	4.009	1.169	1.152	1.317	3.977
25	0.717	0.703	0.812	4.077	0.716	0.704	0.829	4.028
50	0.510	0.501	0.582	4.006	0.508	0.500	0.635	3.975
$\epsilon_i \sim \boldsymbol{t(2)}$								
5	0.551	0.596	0.723	1.614	0.557	0.579	0.753	1.697
10	0.381	0.412	0.512	1.573	0.385	0.400	0.535	1.678
25	0.238	0.259	0.320	1.542	0.241	0.248	0.334	1.636
50	0.165	0.178	0.228	1.576	0.167	0.171	0.238	1.678
$\epsilon_i \sim \boldsymbol{exp(1)}$								
5	0.276	0.320	0.380	0.834	0.194	0.202	0.293	0.682
10	0.182	0.215	0.269	0.856	0.127	0.128	0.206	0.709
25	0.117	0.138	0.171	0.835	0.079	0.079	0.129	0.690
50	0.083	0.098	0.121	0.837	0.056	0.055	0.091	0.695
$\epsilon_i \sim \boldsymbol{\chi^2(2)}$								
5	0.547	0.608	0.732	1.638	0.387	0.375	0.555	1.333
10	0.372	0.415	0.513	1.626	0.259	0.245	0.383	1.297
25	0.231	0.261	0.332	1.649	0.158	0.149	0.246	1.320
50	0.163	0.183	0.233	1.601	0.112	0.104	0.170	1.257
$\epsilon_i \sim \boldsymbol{(0.5 + 0.5(x_{i1})^2)N(0, 1)}$								
5	0.800	0.860	1.078	2.404	0.776	0.747	1.062	2.503
10	0.523	0.603	0.785	2.366	0.501	0.506	0.753	2.494
25	0.374	0.375	0.501	2.356	0.357	0.307	0.480	2.480
50	0.224	0.259	0.345	2.445	0.212	0.212	0.351	2.598
$\epsilon_i \sim \boldsymbol{(0.5 + 0.5(x_{i1})^2)t(2)}$								
5	1.211	1.147	1.561	3.408	1.163	0.908	1.467	3.453
10	0.717	0.794	1.102	3.424	0.633	0.605	1.033	3.458
25	0.547	0.507	0.711	3.360	0.460	0.383	0.691	3.409
50	0.400	0.348	0.499	3.353	0.267	0.255	0.509	3.445
$\epsilon_i \sim \boldsymbol{(0.5 + 0.5(x_{i1})^2)exp(1)}$								
5	0.612	0.705	0.997	2.130	0.440	0.309	0.508	1.329
10	0.470	0.477	0.713	2.170	0.322	0.198	0.365	1.367
25	0.306	0.298	0.451	2.175	0.190	0.119	0.227	1.359
50	0.198	0.208	0.321	2.190	0.103	0.082	0.167	1.412
$\epsilon_i \sim \boldsymbol{(0.5 + 0.5(x_{i1})^2)\chi^2(2)}$								
5	1.298	1.317	1.916	4.124	0.882	0.564	1.162	2.765
10	0.812	0.850	1.308	3.950	0.520	0.372	0.767	2.680
25	0.517	0.526	0.837	3.992	0.299	0.228	0.515	2.590
50	0.400	0.366	0.587	3.989	0.209	0.158	0.359	2.600

Table 2 Simulated MADs (×10) under homogeneous errors with $(n, p) = (50, 10)$

m	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{\omega(q)}$	$\hat{\beta}^{\omega}_{Cen}$	$\hat{\beta}^{\omega}_{Ave}$	$\hat{\beta}^{\omega}_{Sub}$
$\epsilon_i \sim N(0, 4^2)$								
5	1.341	1.319	1.488	3.321	1.346	1.324	1.535	3.295
10	0.964	0.946	1.068	3.295	0.965	0.949	1.085	3.284
25	0.589	0.578	0.672	3.340	0.586	0.578	0.687	3.305
50	0.417	0.409	0.477	3.301	0.415	0.410	0.520	3.275
$\epsilon_i \sim t(2)$								
5	0.454	0.491	0.593	1.334	0.459	0.477	0.619	1.402
10	0.314	0.340	0.419	1.293	0.317	0.329	0.438	1.380
25	0.196	0.213	0.264	1.271	0.197	0.204	0.276	1.349
50	0.137	0.147	0.189	1.297	0.138	0.141	0.197	1.385
$\epsilon_i \sim exp(1)$								
5	0.229	0.264	0.314	0.689	0.160	0.167	0.241	0.565
10	0.149	0.176	0.222	0.706	0.104	0.105	0.169	0.584
25	0.096	0.113	0.141	0.688	0.065	0.065	0.106	0.567
50	0.068	0.080	0.099	0.690	0.046	0.045	0.074	0.574
$\epsilon_i \sim \chi^2(2)$								
5	0.447	0.497	0.601	1.352	0.318	0.307	0.455	1.098
10	0.308	0.342	0.423	1.336	0.213	0.202	0.315	1.059
25	0.190	0.215	0.273	1.356	0.131	0.123	0.203	1.083
50	0.134	0.151	0.191	1.317	0.092	0.086	0.140	1.033
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)N(0, 1)$								
5	0.638	0.690	0.874	1.954	0.614	0.589	0.856	2.036
10	0.523	0.603	0.785	2.366	0.501	0.506	0.753	2.494
25	0.295	0.300	0.407	1.900	0.279	0.242	0.390	2.006
50	0.176	0.206	0.279	1.987	0.166	0.165	0.284	2.114
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)t(2)$								
5	0.976	0.926	1.277	2.758	0.938	0.723	1.198	2.806
10	0.573	0.638	0.899	2.795	0.502	0.479	0.842	2.824
25	0.441	0.405	0.580	2.735	0.368	0.301	0.563	2.781
50	0.321	0.280	0.406	2.733	0.212	0.201	0.415	2.809
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)exp(1)$								
5	0.574	0.575	0.813	1.757	0.471	0.252	0.430	1.105
10	0.387	0.386	0.579	1.765	0.266	0.161	0.298	1.107
25	0.248	0.240	0.367	1.770	0.155	0.097	0.186	1.102
50	0.162	0.168	0.261	1.787	0.084	0.066	0.136	1.148
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)\chi^2(2)$								
5	1.056	1.061	1.566	3.353	0.719	0.458	0.950	2.247
10	0.657	0.684	1.068	3.236	0.421	0.302	0.625	2.186
25	0.419	0.421	0.684	3.250	0.243	0.185	0.420	2.123
50	0.322	0.294	0.478	3.240	0.169	0.129	0.291	2.108

Table 3 Simulated RMSEs ($\times$ 10) under homogeneous errors with $(n, p) = (100, 20)$

m	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{\omega(q)}$	$\hat{\beta}^{\omega}_{Cen}$	$\hat{\beta}^{\omega}_{Ave}$	$\hat{\beta}^{\omega}_{Sub}$
$\epsilon_i \sim N(0, 4^2)$								
5	1.198	1.177	1.308	2.900	1.199	1.181	1.336	2.945
10	0.845	0.830	0.931	2.880	0.843	0.831	0.953	2.951
25	0.538	0.530	0.593	2.932	0.535	0.527	0.604	2.961
50	0.377	0.370	0.415	2.914	0.374	0.369	0.427	2.972
$\epsilon_i \sim t(2)$								
5	0.398	0.430	0.509	1.125	0.400	0.414	0.516	1.179
10	0.270	0.293	0.354	1.125	0.270	0.278	0.357	1.184
25	0.170	0.184	0.225	1.121	0.170	0.175	0.227	1.182
50	0.119	0.129	0.159	1.127	0.119	0.122	0.160	1.191
$\epsilon_i \sim exp(1)$								
5	0.198	0.234	0.274	0.609	0.134	0.142	0.202	0.478
10	0.133	0.158	0.191	0.607	0.089	0.091	0.139	0.480
25	0.084	0.101	0.124	0.602	0.055	0.055	0.089	0.477
50	0.059	0.070	0.086	0.608	0.039	0.038	0.062	0.484
$\epsilon_i \sim \chi^2(2)$								
5	0.396	0.444	0.527	1.146	0.272	0.267	0.379	0.877
10	0.273	0.308	0.379	1.189	0.182	0.175	0.270	0.914
25	0.167	0.190	0.235	1.173	0.111	0.106	0.166	0.905
50	0.118	0.135	0.168	1.165	0.078	0.074	0.117	0.891
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)N(0, 1)$								
5	0.507	0.594	0.744	1.651	0.472	0.486	0.677	1.692
10	0.352	0.412	0.527	1.657	0.324	0.331	0.474	1.722
25	0.216	0.253	0.330	1.636	0.197	0.200	0.296	1.705
50	0.152	0.179	0.235	1.626	0.138	0.140	0.213	1.718
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)t(2)$								
5	0.696	0.793	1.040	2.289	0.643	0.592	0.911	2.306
10	0.485	0.552	0.750	2.347	0.416	0.402	0.663	2.386
25	0.298	0.338	0.472	2.287	0.248	0.242	0.442	2.350
50	0.206	0.234	0.331	2.285	0.171	0.167	0.326	2.338
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)exp(1)$								
5	0.378	0.482	0.666	1.450	0.252	0.203	0.318	0.840
10	0.262	0.332	0.479	1.453	0.158	0.132	0.219	0.838
25	0.159	0.201	0.294	1.478	0.098	0.079	0.134	0.866
50	0.137	0.143	0.211	1.460	0.092	0.055	0.095	0.863
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)\chi^2(2)$								
5	0.771	0.873	1.242	2.732	0.511	0.383	0.634	1.571
10	0.516	0.590	0.871	2.709	0.312	0.248	0.413	1.595
25	0.321	0.363	0.555	2.720	0.194	0.150	0.259	1.591
50	0.222	0.249	0.390	2.725	0.137	0.107	0.182	1.584

Table 4 Simulated MADs (×10) under homogeneous errors with $(n, p) = (100, 20)$

m	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{\omega(q)}$	$\hat{\beta}^{\omega}_{Cen}$	$\hat{\beta}^{\omega}_{Ave}$	$\hat{\beta}^{\omega}_{Sub}$
$\epsilon_i \sim N(0, 4^2)$								
5	0.971	0.953	1.059	2.354	0.971	0.955	1.080	2.388
10	0.685	0.673	0.755	2.346	0.683	0.673	0.773	2.395
25	0.436	0.428	0.481	2.390	0.433	0.426	0.491	2.402
50	0.307	0.301	0.336	2.365	0.304	0.300	0.347	2.406
$\epsilon_i \sim t(2)$								
5	0.324	0.350	0.415	0.916	0.326	0.336	0.421	0.959
10	0.218	0.237	0.288	0.911	0.218	0.225	0.289	0.958
25	0.138	0.149	0.182	0.905	0.138	0.142	0.184	0.952
50	0.096	0.104	0.129	0.912	0.096	0.099	0.130	0.965
$\epsilon_i \sim exp(1)$								
5	0.161	0.190	0.222	0.494	0.108	0.115	0.164	0.387
10	0.107	0.128	0.155	0.491	0.072	0.073	0.112	0.389
25	0.068	0.082	0.101	0.486	0.045	0.045	0.072	0.386
50	0.048	0.057	0.070	0.494	0.032	0.031	0.050	0.393
$\epsilon_i \sim \chi^2(2)$								
5	0.320	0.360	0.425	0.931	0.220	0.216	0.307	0.712
10	0.221	0.250	0.305	0.962	0.147	0.142	0.218	0.739
25	0.135	0.154	0.190	0.953	0.090	0.086	0.134	0.734
50	0.096	0.109	0.136	0.943	0.063	0.060	0.095	0.724
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)N(0, 1)$								
5	0.403	0.475	0.597	1.325	0.372	0.384	0.541	1.359
10	0.278	0.328	0.424	1.335	0.254	0.260	0.380	1.388
25	0.171	0.201	0.264	1.310	0.155	0.157	2.370	1.369
50	0.119	0.141	0.188	1.301	0.107	0.109	0.170	1.374
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)t(2)$								
5	0.555	0.634	0.837	1.848	0.510	0.468	0.731	1.862
10	0.385	0.439	0.602	1.883	0.328	0.317	0.532	1.915
25	0.238	0.270	0.380	1.840	0.198	0.193	0.356	1.892
50	0.163	0.186	0.267	1.841	0.135	0.132	0.263	1.883
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)exp(1)$								
5	0.378	0.482	0.666	1.450	0.252	0.203	0.318	0.840
10	0.211	0.267	0.386	1.172	0.128	0.106	0.177	0.674
25	0.127	0.161	0.237	1.191	0.079	0.063	0.109	0.697
50	0.110	0.114	0.171	1.177	0.075	0.044	0.077	0.696
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)\chi^2(2)$								
5	0.621	0.702	0.100	2.201	0.411	0.309	0.511	1.267
10	0.416	0.473	0.701	2.192	0.252	0.200	0.334	1.289
25	0.258	0.292	0.450	2.208	0.156	0.121	0.208	1.284
50	0.178	0.200	0.316	2.197	0.110	0.086	0.147	1.276

Table 5 Simulated RMSEs (× 100) and MADs (×100) with $(n, p) = (2000, 50)$

m	RMSE				MAD			
	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$
$\epsilon_i \sim t(2)$								
5	0.878	0.940	0.957	2.128	0.701	0.750	0.764	1.705
10	0.613	0.658	0.672	2.106	0.493	0.528	0.539	1.688
25	0.391	0.419	0.429	2.147	0.314	0.337	0.345	1.718
50	0.274	0.294	0.300	2.134	0.219	0.236	0.241	1.707
$\epsilon_i \sim exp(1)$								
5	0.427	0.506	0.516	1.145	0.342	0.405	0.414	0.918
10	0.306	0.361	0.369	1.164	0.245	0.291	0.296	0.937
25	0.192	0.225	0.230	1.148	0.154	0.180	0.184	0.921
50	0.135	0.159	0.162	1.151	0.108	0.128	0.130	0.925
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)N(0, 1)$								
5	1.060	1.229	1.261	2.828	0.841	0.978	1.004	2.247
10	0.744	0.866	0.894	2.282	0.590	0.687	0.711	2.252
25	0.466	0.542	0.561	2.813	0.368	0.430	0.446	2.238
50	0.331	0.385	0.398	2.815	0.261	0.305	0.315	2.241
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)t(2)$								
5	1.371	1.647	1.710	3.787	1.086	1.310	1.361	3.021
10	0.956	1.149	1.198	3.763	0.763	0.918	0.958	2.999
25	0.601	0.722	0.754	3.784	0.477	0.574	0.602	3.012
50	0.426	0.512	0.536	3.773	0.338	0.409	0.427	3.007

MADs of $\hat{\boldsymbol{\beta}}_{Sub}$ vary slightly since its local sample size n is still fixed, while these values of the other estimators decrease due to the increased total sample size. At the same time, the RMSE and MAD difference ratios between $\hat{\boldsymbol{\beta}}_{Ave}$ become obviously smaller than those between $\hat{\boldsymbol{\beta}}_{Cen}$ due to the fact that the naive method requires the constraint on the number of machines for bias reduction.

(2) From Tables 1 and 2, the RMSEs and MADs of our proposed WCQR estimator $\hat{\boldsymbol{\beta}}^{\omega(q)}$ are comparable with these of the central estimator $\hat{\boldsymbol{\beta}}^{\omega}_{Cen}$. Apart from that, the other WCQR estimators have the similar findings as the corresponding CQR estimators. In addition, it can be seen that the optimal WCQR estimators have smaller RMSEs and MADs than those of CQR estimators in most of cases, which agree with the theoretical result that equal weights for CQR might not be optimal. However, this improvement is not significant when $\epsilon_i \sim N(0, 4^2)$ and $t(2)$.

(3) From Tables 3 and 4, when both the local sample size n and dimension p increase (i.e., $(n, p) = (100, 20)$), we have the same conclusions as $(n, p) = (50, 10)$, which shows our proposed estimators are robust.

(4) When $(n, p) = (2000, 50)$, we have the similar conclusions. In these cases, the values of iterations for proposed CQR estimator are around 5 even with the larger n and p on each machine.

In summary, our proposed estimators can achieve the desirable performance with few iterations even when (n, p) are large.

4.2 Sensitivity analysis

The sensitivity analysis of the proposed estimators with respect to bandwidth and K is also investigated. In specific, we consider $h_j = 1.5\sigma_{\hat{\epsilon}_j}(Kn)^{\nu}$ with $\nu = (-1/5, -1/3, -2/5)$ and $h_j = c\sigma_{\hat{\epsilon}_j}(Kn)^{-1/3}$ with $c = (1, 1.5, 2)$, respectively, based on the error $N(0, 4^2)$ and $(n, p) = (100, 20)$. The simulation results are reported in Table 6. It can be seen that the RMSEs and MADs vary little over a wide range of ν and c values. The simulations show that our proposed estimators are insensitive to the choice of the bandwidth. Finally, $\nu = -1/3$ and $c = 1.5$ are recommended in practice. Furthermore, we consider $\tau_k = k/(K+1)$ for $k = 1, \ldots, K$ with $K = 5, 7, 9$ based on the error $N(0, 4^2)$ and $(n, p) = (100, 20)$. It can be seen from Table 7 that the RMSEs and MADs with different K vary little.

4.3 Coverage probability

To further measure the performance of our proposed method in terms of the statistical inference, we study the coverage probability (CP) of 95% confidence interval for $v_0^T\boldsymbol{\beta}$, where $v_0 = \mathbf{1}_p$. From Theorem2, an oracle 95% confidence interval for $v_0^T\boldsymbol{\beta}$ is given by

$$v_0^T\hat{\boldsymbol{\beta}}^{(q)} \pm N^{-1/2}\sqrt{\sum_{k,k'=1}^{K}\min(\tau_k, \tau_{k'})(1 - \max(\tau_k, \tau_{k'}))v_0^T Q^{-1}E[xx^T]Q^{-1}v_0 z_{0.975}},$$

where $z_{0.975}$ is the 97.5%-quantile of the standard normal distribution. Table 8 show that CPs of the different methods. The CPs of our proposed CQR method are around nominal level 0.95, which are similar to those of $\hat{\boldsymbol{\beta}}_{Cen}$. On the other hand, the CPs of $\hat{\boldsymbol{\beta}}_{Ave}$ and $\hat{\boldsymbol{\beta}}_{Sub}$ are much lower, which also demonstrate our proposed CQR method has good performance than the naive DC method.

5 Real data

In this section, we carry out our proposed methods on the Gas Turbine CO and NOx Emission Data in Year 2013 from the UCI machine learning repository (https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set). The dataset contains $N = 6000$ samples, which were aggregated over one hour (by means of average) from a gas turbine located in Turkey's north western region. Our interest is to study the flue gas CO (Y) emissions with covariates turbine inlet temperature (X_1), turbine after temperature (X_2), turbine energy yield (X_3) and compressor discharge pressure (X_4) at different quantiles $\tau_k = k/6$, $k = 1, \ldots, 5$.

Table 6 Simulated RMSEs (×10) and MADs (×10) for different bandwidths under normal error with $(n, p) = (100, 20)$

m	bandwidth	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{\omega(q)}$	$\hat{\beta}^{\omega}_{Cen}$	$\hat{\beta}^{\omega}_{Ave}$	$\hat{\beta}^{\omega}_{Sub}$
RMSEs									
5	(1.5, − 1/5)	1.223	1.188	1.318	2.953	1.202	1.191	1.346	2.957
	(1.5, − 1/3)	1.198	1.177	1.308	2.900	1.199	1.181	1.336	2.945
	(1.5, − 2/5)	1.196	1.173	1.304	2.923	1.196	1.175	1.325	2.905
	(1, − 1/3)	1.202	1.181	1.300	2.896	1.203	1.182	1.325	2.889
	(2, − 1/3)	1.212	1.190	1.320	2.910	1.212	1.193	1.350	2.894
10	(1.5, − 1/5)	0.843	0.823	0.928	2.979	0.828	0.822	0.946	2.958
	(1.5, − 1/3)	0.845	0.830	0.931	2.880	0.843	0.831	0.953	2.951
	(1.5, − 2/5)	0.837	0.822	0.928	2.915	0.838	0.823	0.947	2.888
	(1, − 1/3)	0.839	0.825	0.931	2.963	0.838	0.825	0.950	2.950
	(2, − 1/3)	0.839	0.824	0.926	2.925	0.833	0.824	0.948	2.922
25	(1.5, − 1/5)	0.531	0.518	0.579	2.901	0.522	0.518	0.589	2.876
	(1.5, − 1/3)	0.538	0.530	0.593	2.932	0.535	0.527	0.604	2.961
	(1.5, − 2/5)	0.523	0.513	0.583	2.943	0.522	0.513	0.594	2.924
	(1, − 1/3)	0.531	0.522	0.585	2.912	0.531	0.522	0.594	2.917
	(2, − 1/3)	0.522	0.512	0.582	2.935	0.518	0.512	0.596	2.917
50	(1.5, − 1/5)	0.378	0.369	0.413	2.938	0.370	0.368	0.587	2.918
	(1.5, − 1/3)	0.377	0.370	0.415	2.914	0.374	0.369	0.427	2.972
	(1.5, − 2/5)	0.376	0.370	0.413	2.879	0.374	0.369	0.560	2.847
	(1, − 1/3)	0.373	0.366	0.417	2.940	0.371	0.365	0.579	2.932
	(2, − 1/3)	0.371	0.364	0.414	2.851	0.366	0.362	0.566	2.846
MADs									
5	(1.5, − 1/5)	0.990	0.961	1.068	2.392	0.973	0.964	1.091	2.395
	(1.5, − 1/3)	0.971	0.953	1.059	2.354	0.971	0.955	1.080	2.388
	(1.5, − 2/5)	0.968	0.949	1.054	2.374	0.969	0.951	1.069	2.362
	(1, − 1/3)	0.974	0.957	1.056	2.340	0.976	0.959	1.076	2.335
	(2, − 1/3)	0.984	0.965	1.074	2.368	0.987	0.970	1.097	2.353
10	(1.5, − 1/5)	0.687	0.670	0.756	2.415	0.675	0.670	0.771	2.396
	(1.5, − 1/3)	0.685	0.673	0.755	2.346	0.683	0.673	0.773	2.395
	(1.5, − 2/5)	0.678	0.667	0.750	2.369	0.680	0.667	0.766	2.345
	(1, − 1/3)	0.678	0.667	0.752	2.396	0.678	0.668	0.770	2.380
	(2, − 1/3)	0.681	0.668	0.749	2.364	0.677	0.670	0.767	2.360
25	(1.5, − 1/5)	0.431	0.419	0.469	2.344	0.422	0.418	0.477	2.327
	(1.5, − 1/3)	0.436	0.428	0.481	2.390	0.433	0.426	0.491	2.402
	(1.5, − 2/5)	0.423	0.415	0.472	2.387	0.422	0.415	0.480	2.374
	(1, − 1/3)	0.428	0.421	0.472	2.353	0.428	0.421	0.479	2.363
	(2, − 1/3)	0.424	0.416	0.472	2.370	0.421	0.416	0.484	2.354

Table 6 (continued)

m	bandwidth	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{\omega(q)}$	$\hat{\beta}^{\omega}_{Cen}$	$\hat{\beta}^{\omega}_{Ave}$	$\hat{\beta}^{\omega}_{Sub}$
50	(1.5, − 1/5)	0.307	0.299	0.335	2.378	0.299	0.298	0.474	2.363
	(1.5, − 1/3)	0.307	0.301	0.336	2.365	0.304	0.300	0.347	2.406
	(1.5, − 2/5)	0.305	0.300	0.335	2.337	0.304	0.299	0.455	2.308
	(1, − 1/3)	0.302	0.296	0.336	2.379	0.300	0.295	0.467	2.370
	(2, − 1/3)	0.301	0.296	0.336	2.299	0.298	0.294	0.458	2.296

Table 7 Simulated RMSEs (×10) and MADs (×10) for different quantile levels under normal error with $(n, p) = (100, 20)$

m	K	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{\omega(q)}$	$\hat{\beta}^{\omega}_{Cen}$	$\hat{\beta}^{\omega}_{Ave}$	$\hat{\beta}^{\omega}_{Sub}$
RMSEs									
5	5	1.198	1.177	1.308	2.900	1.199	1.181	1.336	2.945
	7	1.188	1.172	1.303	2.886	1.189	1.170	1.354	2.937
	9	1.183	1.170	1.300	2.880	1.185	1.168	1.387	2.945
10	5	0.845	0.830	0.931	2.880	0.843	0.831	0.953	2.951
	7	0.838	0.827	0.928	2.869	0.836	0.825	0.971	2.933
	9	0.835	0.825	0.926	2.863	0.833	0.823	0.996	2.938
25	5	0.538	0.530	0.593	2.932	0.535	0.527	0.604	2.961
	7	0.535	0.528	0.590	2.918	0.529	0.524	0.614	2.956
	9	0.533	0.527	0.589	2.911	0.528	0.523	0.634	2.975
50	5	0.377	0.370	0.415	2.914	0.374	0.369	0.427	2.972
	7	0.374	0.369	0.413	2.905	0.369	0.366	0.436	2.956
	9	0.373	0.368	0.412	2.899	0.368	0.364	0.451	2.960
MADs									
5	5	0.971	0.953	1.059	2.354	0.971	0.955	1.080	2.388
	7	0.963	0.949	1.054	2.343	0.963	0.948	1.095	2.384
	9	0.959	0.947	1.052	2.338	0.960	0.946	1.122	2.385
10	5	0.685	0.673	0.755	2.346	0.683	0.673	0.773	2.395
	7	0.679	0.670	0.752	2.338	0.677	0.668	0.786	2.383
	9	0.677	0.669	0.751	2.333	0.674	0.667	0.808	2.387
25	5	0.436	0.428	0.481	2.390	0.433	0.426	0.491	2.402
	7	0.433	0.427	0.479	2.377	0.428	0.424	0.499	2.401
	9	0.431	0.426	0.478	2.371	0.427	0.422	0.514	2.418
50	5	0.307	0.301	0.336	2.365	0.304	0.300	0.347	2.406
	7	0.304	0.300	0.335	2.357	0.300	0.297	0.354	2.393
	9	0.303	0.300	0.334	2.354	0.299	0.296	0.366	2.403

Since the true value of $\boldsymbol{\beta}$ is unknown for a real data set, for the purpose of comparison, we randomly divide this dataset into 5000 training data and 1000 testing data, apply the eight methods to train the model, and then compare the performance of these estimates in terms of prediction errors based on the testing data. In particular, we

Table 8 CPs under homogeneous errors for the CQR estimators with $(n, p) = (50, 10)$ and $(n, p) = (100, 20)$

m	$(n,p) = (50, 10)$				$(n,p) = (100, 20)$			
	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$
$\epsilon_i \sim N(0, 4^2)$								
5	0.940	0.944	0.918	0.914	0.956	0.954	0.924	0.936
10	0.930	0.934	0.890	0.916	0.968	0.964	0.920	0.916
25	0.954	0.954	0.922	0.932	0.962	0.962	0.932	0.946
50	0.946	0.954	0.900	0.930	0.954	0.956	0.924	0.926
$\epsilon_i \sim t(2)$								
5	0.962	0.942	0.878	0.868	0.930	0.918	0.872	0.876
10	0.968	0.948	0.862	0.898	0.968	0.954	0.894	0.886
25	0.946	0.918	0.854	0.896	0.982	0.972	0.902	0.870
50	0.958	0.946	0.882	0.882	0.962	0.942	0.870	0.892
$\epsilon_i \sim exp(1)$								
5	0.966	0.938	0.892	0.860	0.950	0.904	0.846	0.842
10	0.968	0.942	0.868	0.898	0.962	0.918	0.854	0.852
25	0.968	0.946	0.870	0.862	0.966	0.928	0.838	0.854
50	0.980	0.942	0.864	0.884	0.950	0.906	0.854	0.886
$\epsilon_i \sim \chi^2(2)$								
5	0.964	0.944	0.866	0.874	0.966	0.940	0.880	0.886
10	0.968	0.928	0.864	0.888	0.954	0.944	0.876	0.886
25	0.958	0.934	0.866	0.874	0.956	0.930	0.866	0.868
50	0.964	0.952	0.902	0.886	0.966	0.944	0.880	0.888
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)N(0, 1)$								
5	0.918	0.904	0.844	0.842	0.936	0.912	0.810	0.824
10	0.900	0.884	0.814	0.836	0.910	0.858	0.792	0.802
25	0.932	0.898	0.808	0.826	0.950	0.908	0.796	0.820
50	0.936	0.896	0.798	0.808	0.940	0.896	0.840	0.832
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)t(2)$								
5	0.960	0.940	0.866	0.868	0.950	0.924	0.852	0.844
10	0.958	0.940	0.848	0.840	0.950	0.926	0.850	0.844
25	0.948	0.936	0.864	0.858	0.966	0.956	0.872	0.870
50	0.962	0.946	0.856	0.880	0.964	0.930	0.864	0.844
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)exp(1)$								
5	0.966	0.932	0.808	0.840	0.986	0.932	0.864	0.828
10	0.980	0.932	0.852	0.830	0.980	0.932	0.830	0.852
25	0.986	0.960	0.830	0.814	0.992	0.938	0.808	0.816
50	0.988	0.942	0.808	0.828	0.980	0.942	0.828	0.832
$\epsilon_i \sim (0.5 + 0.5(x_{i1})^2)\chi^2(2)$								
5	0.962	0.946	0.812	0.862	0.992	0.970	0.866	0.874
10	0.984	0.968	0.872	0.894	0.982	0.970	0.870	0.856
25	0.978	0.962	0.860	0.844	0.986	0.970	0.834	0.840
50	0.974	0.964	0.854	0.826	0.980	0.966	0.846	0.872

Table 9 The RMSEs and MADs for the gas emission data

m		$\hat{\beta}^{(q)}$	$\hat{\beta}_{Cen}$	$\hat{\beta}_{Ave}$	$\hat{\beta}_{Sub}$	$\hat{\beta}^{\omega(q)}$	$\hat{\beta}^{\omega}_{Cen}$	$\hat{\beta}^{\omega}_{Ave}$	$\hat{\beta}^{\omega}_{Sub}$
5	$X_1(\beta_1)$	− 1.494	− 1.611	− 1.226	− 0.893	− 1.502	− 1.645	− 1.341	− 0.770
	RMSE	0.347	0.381	0.387	0.530	0.363	0.366	0.373	0.520
	MAD	0.261	0.299	0.310	0.436	0.278	0.283	0.295	0.421
10	$X_1(\beta_1)$	− 1.486	− 1.611	− 1.504	− 0.442	− 1.507	− 1.648	− 1.447	− 0.414
	RMSE	0.348	0.381	0.403	0.549	0.365	0.367	0.373	0.541
	MAD	0.261	0.299	0.330	0.438	0.281	0.283	0.291	0.426
25	$X_1(\beta_1)$	− 1.485	− 1.611	− 1.697	− 0.965	− 1.518	− 1.655	− 1.576	− 0.684
	RMSE	0.341	0.381	0.414	0.672	0.367	0.368	0.379	0.646
	MAD	0.260	0.299	0.325	0.567	0.283	0.284	0.299	0.530
50	$X_1(\beta_1)$	− 1.476	− 1.611	− 1.667	1.904	− 1.536	− 1.662	− 1.555	1.034
	RMSE	0.347	0.381	0.449	0.873	0.369	0.369	0.386	0.780
	MAD	0.259	0.299	0.347	0.663	0.285	0.285	0.297	0.578

consider the number of machines $m = (5, 10, 25, 50)$ and then obtain the estimates $\hat{\boldsymbol{\beta}}^{(q)}$, $\hat{\boldsymbol{\beta}}_{Cen}$, $\hat{\boldsymbol{\beta}}_{Ave}$, $\hat{\boldsymbol{\beta}}_{Sub}$, $\hat{\boldsymbol{\beta}}^{\omega(q)}$, $\hat{\boldsymbol{\beta}}^{\omega}_{Cen}$, $\hat{\boldsymbol{\beta}}^{\omega}_{Ave}$ and $\hat{\boldsymbol{\beta}}^{\omega}_{Sub}$, respectively, based on the training data. Thereafter, we use the estimated coefficients to construct forecasts of the other 1000 testing data. We compute both the RMSE $= \sqrt{\sum_i (Y_i - \hat{Y}_i)^2/1000}$ and MAD $= \sum_i |Y_i - \hat{Y}_i|/1000$ based on the testing data, where $\hat{Y}_i$ is the fitted value of the testing data Y_i, $i = 1, \ldots, 1000$. The results are given in Table 9.

(1) For $m = (5, 10, 25, 50)$, we find that only the estimated coefficients of X_1 are remarkable and negatively related with Y among these eight different methods, while the other estimated coefficients are close to zero. This coincides with the fact that the CO emissions will increase when incomplete combustion with lower turbine inlet temperature occurs. Thus, we only present the estimated coefficients of X_1 in Table 9. Among these results, it can be seen that the subsample estimates are the worst since their absolute values are the smallest when m=5, 10, 25 and they even become positive when $m = 50$.
(2) In terms of RMSEs and MADs, our proposed estimates $\hat{\boldsymbol{\beta}}^{(q)}$ and $\hat{\boldsymbol{\beta}}^{\omega(q)}$ have similar results with these of $\hat{\boldsymbol{\beta}}_{Cen}$ and $\hat{\boldsymbol{\beta}}^{\omega}_{Cen}$, and perform better than the naive DC method in both CQR and WCQR models. This indicates that we can obtain desirable coefficient estimates via our proposed distributed methods.

6 Discussion

In this paper, we propose the multi-round smoothed CQR and WCQR estimators for the distributed data. The proposed methods only require consistent initial value, and the rest of operations are convenient matrix manipulations. Some interesting issues still merit further research. Firstly, the proposed methods are designed for small to moderate covariate dimensionality. The multi-round penalized and smoothed estimators for sparse high-dimensional CQR and WCQR model are interesting consideration. Secondly, it is also of interest to investigate the proposed estimation methods for longitudinal data.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10463-021-00816-0.

Acknowledgements We are grateful to the Editor, an associate editor and one anonymous referee for their insightful comments and suggestions, which have led to significant improvements. This paper was supported by the National Natural Science Foundation of China [grant numbers 11871287, 11771144, 11801359], the Natural Science Foundation of Tianjin [18JCYBJC41100], Fundamental Research Funds for the Central Universities and the Key Laboratory for Medical Data Analysis and Statistical Research of Tianjin.

References

Battey, H., Fan, J., Liu, H., Lu, J., Zhu, Z. (2018). Distributed estimation and inference with statistical guarantees. *The Annals of Statistics, 46*, 1352–1382.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning, 3*, 1–122.

Chen, X., Xie, M. G. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica, 24*, 1655–1684.

Chen, X., Liu, W., Zhang, Y. (2019). Quantile regression under memory constraint. *The Annals of Statistics, 47*, 3244–3273.

Dekel, O., Gilad-Bachrach, R., Shamir, O., Xiao, L. (2012). Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research, 13*, 165–202.

Fan, T. H., Lin, D., Cheng, K. F. (2007). Regression analysis for massive datasets. *Data and Knowledge Engineering, 61*, 554–562.

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

Heller, G. (2007). Smoothed rank regression with censored data. *Journal of the American Statistical Association, 102*, 552–559.

Horowitz, J. L. (1998). Bootstrap methods for median regression models. *Econometrica, 66*, 1327–1351.

Jiang, R., Qian, W. M., Zhou, Z. G. (2016). Weighted composite quantile regression for single-index models. *Journal of Multivariate Analysis, 148*, 34–48.

Jiang, X. J., Jiang, J. C., Song, X. Y. (2012). Oracle model selection for nonlinear models based on weighted composite quantile regression. *Statistica Sinica, 22*, 1479–1506.

Kai, B., Li, R., 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.

Kai, B., Li, R., Zou, H. (2011). New efficient estimation and variable selection methods for semiparametric varying-coefficient partially linear models. *The Annals of Statistics, 39*, 305–332.

Kaplan, D. M., Sun, Y. (2017). Smoothed estimating equations for instrumental variables quantile regression. *Econometric Theory, 33*, 105–157.
Koenker, R., Bassett, J. G. (1978). Regression quantiles. *Econometrica, 46*, 33–50.
Lee, J. D., Liu, Q., Sun, Y., Taylor, J. E. (2017). Communication-efficient sparse regression. *The Journal of Machine Learning Research, 18*, 115–144.
Li, R., Lin, D. K., Li, B. (2013). Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry, 29*, 399–409.
Shi, C., Lu, W., Song, R. (2018). A massive data framework for M-estimators with cubic-rate. *Journal of the American Statistical Association, 113*, 1698–1709.
Volgushev, S., Chao, S. K., Cheng, G. (2019). Distributed inference for quantile regression processes. *The Annals of Statistics, 47*, 1634–1662.
Whang, Y. J. (2006). Smoothed empirical likelihood methods for quantile regression models. *Econometric Theory, 22*, 173–205.
Zhang, Y., Duchi, J. C., Wainwright, M. J. (2013). Communication-efficient algorithms for statistical optimization. *The Journal of Machine Learning Research, 14*, 3321–3363.
Zhao, K., Lian, H. (2016). A note on the efficiency of composite quantile regression. *Journal of Statistical Computation and Simulation, 86*, 1334–1341.
Zou, H., Yuan, M. (2008). Composite quantile regression and the oracle model selection theory. *The Annals of Statistics, 36*, 1108–1126.