The Annals of Statistics
2019, Vol. 47, No. 6, 3244–3273
https://doi.org/10.1214/18-AOS1777

QUANTILE REGRESSION UNDER MEMORY CONSTRAINT

BY XI CHEN[*,1] WEIDONG LIU[†,2] AND YICHEN ZHANG[*]

New York University[*] and Shanghai Jiao Tong University[†]

This paper studies the inference problem in quantile regression (QR) for a large sample size n but under a limited memory constraint, where the memory can only store a small batch of data of size m. A natural method is the naive divide-and-conquer approach, which splits data into batches of size m, computes the local QR estimator for each batch and then aggregates the estimators via averaging. However, this method only works when $n = o(m^2)$ and is computationally expensive. This paper proposes a computationally efficient method, which only requires an initial QR estimator on a small batch of data and then successively refines the estimator via multiple rounds of aggregations. Theoretically, as long as n grows polynomially in m, we establish the asymptotic normality for the obtained estimator and show that our estimator with only a few rounds of aggregations achieves the same efficiency as the QR estimator computed on all the data. Moreover, our result allows the case that the dimensionality p goes to infinity. The proposed method can also be applied to address the QR problem under distributed computing environment (e.g., in a large-scale sensor network) or for real-time streaming data.

1. Introduction. The development of modern technology has enabled data collection of unprecedented size, which leads to large-scale datasets that cannot be fit into memory or are distributed in many machines over limited memory. For example, the memory of a personal computer only has a storage size in GBs while the data set on the hard disk could have a much larger size. In addition, in a sensor network, each sensor is designed to collect and store a limited amount of data, and computations are performed via communications and aggregations among sensors (see, e.g., Wang and Li (2018)). Other examples include high-speed data streams that are transient and arrive at the processor at a high speed. In online streaming computation, the memory is usually limited as compared to the length of the data stream (Gama, Sebastião and Rodrigues (2013), Zhang and Wang (2007)). Under memory constraints in all these scenarios, classical statistical methods, which are

Received October 2017; revised October 2018.

[1]Supported by Adobe research award, Alibaba innovation research award and Bloomberg data science research award.

[2]Supported by NSFC Grants 11825104, 11431006 and 11690013, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, Youth Talent Support Program, 973 Program (2015CB856004) and a grant from the Australian Research Council.

MSC2010 subject classifications. Primary 62F12; secondary 62J02.

Key words and phrases. Quantile regression, sample quantile, divide-and-conquer, distributed inference, streaming data.

developed under the assumption that the memory can fit all the data, are no longer applicable; thus, many estimation and inference methods need to be reinvestigated. For example, suppose that there are n samples for some very large n, a fundamental question in data analysis is as follows:

> How to calculate the sample quantiles of n samples when the memory can only store m samples with $n \gg m$?

As one of the most popular interview questions from high-tech companies, this problem has attracted much attention from computer scientists over the last decade; see Greenwald and Khanna (2004), Guha and McGregor (2008/09), Manku, Rajagopalan and Lindsay (1998), Zhang and Wang (2007) and the references therein. However, this is mainly a computation problem with a fixed dataset, which does not involve any statistical modeling.

Motivated by this sample quantile calculation problem, we study a more general problem of quantile regression (QR) under memory constraints. Quantile regression, which models the conditional quantile of the response variable given covariates, finds a wide range of applications to survival analysis (e.g., Wang and Wang (2014), Xu et al. (2017)), health care (e.g., Luo, Huang and Wang (2013), Sherwood, Wang and Zhou (2013)), and economics (e.g., Belloni et al. (2011)). In the classical QR model, assume that there are n *i.i.d.* samples $\{(\boldsymbol{X}_i, Y_i)\}$ from the following model:

$$Y_i = \boldsymbol{X}_i'\boldsymbol{\beta}(\tau) + \varepsilon_i \qquad \text{for } i = 1, \ldots, n, \tag{1}$$

where $\boldsymbol{X}_i' = (1, X_{i1}, \ldots, X_{ip})$ is the random covariate vector with the dimension $p+1$ drawn from a common population $\boldsymbol{X}$. The error ε_i is an unobserved random variable satisfying $\mathbb{P}(\varepsilon_i \le 0 | \boldsymbol{X}_i) = \tau$ for some specified $0 < \tau < 1$ (known as the quantile level). In other words, $\boldsymbol{X}_i'\boldsymbol{\beta}(\tau)$ is the τth quantile of Y_i given $\boldsymbol{X}_i$. When all the n samples can be fit into memory, one can estimate $\boldsymbol{\beta}(\tau)$ via the classical QR estimator (Koenker (2005)),

$$\widehat{\boldsymbol{\beta}}_{\mathrm{QR}} = \operatorname*{arg\,min}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^{n} \rho_\tau(Y_i - \boldsymbol{X}_i'\boldsymbol{\beta}), \tag{2}$$

where $\rho_\tau(x) = x(\tau - I\{x \le 0\})$ is the asymmetric absolute deviation function (a.k.a. check function) and $I(\cdot)$ is the indicator function. However, when samples are distributed across many machines or the sample size n is extremely large, and thus the samples cannot be fit into memory, it is natural to ask the following question:

> How to estimate and conduct inference about $\boldsymbol{\beta}(\tau)$ when the memory can only store m samples with $n \gg m$?

The divide-and-conquer (DC), as one of the most important algorithms in computer science, has been commonly adopted to deal with this kind of big data chal-

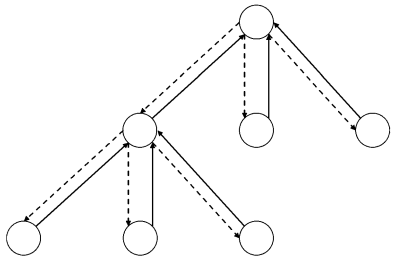

FIG. 1. *An illustration of tree-structured sensor network where the root is the base station. Each sensor collects its a small batch of data. The dashed lines indicate the prior information sent by the base station to all sensors* (*e.g.*, *initializations*) *and the solid lines indicate the information flow* (*i.e.*, *the paths for transferring local statistics* $T_{(\cdot)}$).

lenge. We below describe a general DC algorithm for statistical estimation. Specifically, we split the data indices $\{1, 2, \ldots, n\}$ into N subsets $\mathcal{H}_1, \ldots, \mathcal{H}_N$ with equal size m and $N = n/m$. Correspondingly, the entire data set $\{Y_i, \boldsymbol{X}_i, 1 \le i \le n\}$ is divided into N batches $\mathcal{D}_1, \ldots, \mathcal{D}_N$, where $\mathcal{D}_k = \{Y_i, \boldsymbol{X}_i, i \in \mathcal{H}_k\}$ for $1 \le k \le N$. By swapping each batch of data $\mathcal{D}_k$ into the memory, one constructs a low dimension statistic $T_k = g_k(\mathcal{D}_k)$ for $\mathcal{D}_k$ with some function $g_k(\cdot)$. Then the estimator $\widehat{\boldsymbol{\beta}}$ is obtained by the aggregation of $\{T_k\}_{k=1}^N$ (i.e., $\widehat{\boldsymbol{\beta}} = G(T_1, \ldots, T_N)$ for some aggregation function $G(\cdot)$). In recent years, this DC framework has been widely adopted in distributed statistical inference (see, e.g., Banerjee, Durot and Sen (2018), Battey et al. (2018), Chen and Xie (2014), Li, Lin and Li (2013), Shi, Lu and Song (2017), Volgushev, Chao and Cheng (2018), Zhao, Cheng and Liu (2016) and Section 2 for detailed descriptions).

In addition to memory-constrained estimation on a single machine (where the size of the dataset is much larger than memory size), another natural situation for using the DC framework comes from the application of large-scale wireless sensor networks (see, e.g., Greenwald and Khanna (2004), Huang et al. (2011), Rajagopal, Wainwright and Varaiya (2006), Shrivastava et al. (2004), Wang and Li (2018)). In a sensor network with N sensors, the data are collected and stored in different sensors. Moreover, due to limited energy carried by sensors, communication cost is one of the main concerns in data aggregation. The samples are not transferred to the base station or neighboring sensors directly. Instead, each sensor first summarizes the samples into a low dimensional statistic T_k, which can be transferred with a low communication cost. Figure 1 visualizes a typical sensor network with data flows as a routing tree with the base station as the root. An internal sensor node in the ith layer receives statistics $T_{(\cdot)}$ from its children nodes in $(i+1)$th layer, and then combines received statistics with its own $T_{(\cdot)}$ and sends the resulting statistic to its parent in the $(i-1)$th layer. The final estimator in the base station (or central node) can be computed by $\widehat{\boldsymbol{\beta}} = G(T_1, \ldots, T_N)$.

A critical problem in statistical DC framework is how to construct local statistics $g_k(\cdot)$ and aggregation function $G(\cdot)$. In many existing studies, a typical choice of

$g_k(\cdot)$ is to use the same estimator as the one designed for the estimation from the entire data. For example, in QR, one may choose

$$\widehat{\boldsymbol{\beta}}_{\mathrm{QR},k} := g_k(\mathcal{D}_k) = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i\in\mathcal{H}_k} \rho_\tau(Y_i - \boldsymbol{X}_i'\boldsymbol{\beta}), \tag{3}$$

and a simple averaging function $G(T_1,\ldots,T_N) = \frac{1}{N}\sum_{k=1}^N T_k$, where $T_k = g_k(\mathcal{D}_k)$ is the local statistic. We call this kind of DC methods the naive-DC algorithm where the estimator is denoted by $\widehat{\boldsymbol{\beta}}_{\mathrm{ndc}} = \sum_{k=1}^N \widehat{\boldsymbol{\beta}}_{\mathrm{QR},k}/N$. Despite its popularity, the naive-DC algorithm might fail when n/m is large. For example, in a special case of quantile estimation (i.e., $p = 0$), it is straightforward to show that $\sqrt{n}|\widehat{\beta}_{\mathrm{ndc}} - \beta(\tau)| \to \infty$ in probability when $n/m^2 \to \infty$ (see Theorem B.1 in the supplementary material). Similar phenomenon occurs for general p; see Volgushev, Chao and Cheng (2018). In fact, the local QR estimators $\widehat{\boldsymbol{\beta}}_{\mathrm{QR},k}$ are biased estimators with the bias $O(1/m)$. Although the averaging aggregation is useful for variance reduction, it is unable to reduce the bias, which makes the naive-DC fail when n is large as compared to m. In the DC framework, bias reduction in T_k is more critical than the variance reduction. This is a fundamental difference from many classical inference problems that require to balance the variance and bias (cf. nonparametric estimation). Furthermore, in the naive-DC algorithm, we need to solve $N = n/m$ optimization problems, which could be computationally expensive.

The deficiency of the naive-DC approach calls for a new DC scheme to achieve the following two important goals in distributed inference:

1. The obtained estimator $\widehat{\boldsymbol{\beta}}$ should achieve the same statistical efficiency as merging all the data together under a weak condition on the sample size n as a function of m. More precisely, it is desirable to remove the constraint $n = o(m^2)$ in naive-DC so that the procedure can be applied to situations such as large-scale sensor networks, where the number of sensors $N = n/m$ is excessively large.
2. The second goal is on the *computational efficiency*. For example, the naive-DC requires solving a nonsmooth optimization for computing the local statistic $T_k = g_k(\mathcal{D}_k)$. Since there is no explicit formula for $g_k(\cdot)$, the computation is quite heavy (especially considering each sensor has a limited computational power).

This paper develops new constructions of $g_k(\cdot)$ and $G(\cdot)$ with a multi-round aggregation scheme in Algorithm 1, which simultaneously achieves the two goals referred above. Our method is applicable to both scenarios of small memory on a single machine and large-scale sensor networks. Instead of using the local QR estimation as in (3), we adopt a smoothing technique in the literature (see, e.g., Horowitz (1998), Pang, Lu and Wang (2012), Wang, Stefanski and Zhu (2012), Whang (2006), Wu, Ma and Yin (2015)), and propose a new estimator for QR called *linear estimator of QR (LEQR)*, which serves as the cornerstone for our DC approach. Our LEQR has an explicit formula in the form of direct sums of the

transformation of $\{Y_i, \boldsymbol{X}_i\}$, which is quite different from the optimization-based QR estimator in (3). It is also worthwhile noting that the linearity is the most desirable property in the DC framework for both theoretical development and computation efficiency.

The high-level description of the proposed multiround DC approach is provided as follows. Our method only needs to compute an initial QR estimator $\widehat{\boldsymbol{\beta}}_0$ based on a small part of samples (e.g., $\mathcal{D}_1$). Based on $\widehat{\boldsymbol{\beta}}_0$, for each batch of data $\mathcal{D}_k$, we compute local statistics $\{T_k\}$ using the proposed LEQR. The local statistics are in a simple form of weighted sums of $\boldsymbol{X}_i$ and $\boldsymbol{X}_i\boldsymbol{X}_i'$. The aggregation function is constructed by adding up the local statistics and then solving a linear system, which gives the first-round estimator $\widehat{\boldsymbol{\beta}}^{(1)}$. Now, we can repeat this DC algorithm using $\widehat{\boldsymbol{\beta}}^{(1)}$ as the initial estimator. After q iterations, we denote our final estimator by $\widehat{\boldsymbol{\beta}}^{(q)}$. Theoretically, under some conditions on the growth rate of $p \to \infty$ as a function m and n, we first establish the Bahadur representation of $\widehat{\boldsymbol{\beta}}^{(q)}$ and show that the Bahadur remainder term achieves a nearly optimal rate (up to a log-factor) when q satisfies some mild conditions (see (17) and Theorem 4.3). Furthermore, as long as $n = o(m^A)$ for some constant A, the final estimator $\widehat{\boldsymbol{\beta}}^{(q)}$ achieves the same asymptotic efficiency as the QR estimator (2) computed on the entire data (see Theorem 4.4).

The new DC approach is particularly suitable for QR in sensor networks in which communication cost is one of the major concerns. The proposed procedure only requires $O(p^2)$ bits communication between any two sensors. We also highlight two other important applications of our method:

1. Our method can be adapted to make inference for online streaming data, which arrives at the processor at a high speed. Our method provides a sequence of successively refined estimators of $\boldsymbol{\beta}(\tau)$ for streaming data and can deal with an arbitrary length of data stream. The online quantile estimation problem (which is a special case of QR with $p = 0$) for streaming data has been extensively studied in computer science literature (see, e.g., Guha and McGregor (2008/09), Munro and Paterson (1980), Wang et al. (2013), Zhang and Wang (2007)). However, these works mainly focus on developing approximations to the sample quantile, which are insufficient to obtain limiting distribution results for the purpose of inference. We extend the quantile estimation to the more general QR problem and provide the asymptotical normality result for the proposed online estimator (see Theorem 4.5).

2. Our method also serves as an efficient optimization solver for classical QR on a single machine. As compared to the standard interior-point method for solving the QR estimator in (2) that requires the computational complexity of $O(n^{1.25}p^3 \log n)$ (Portnoy and Koenker (1997)), our approach requires $O(m^{1.25}p^3 \log m + np^2 + p^3)$ since it only solves an optimization on a small batch of data for construction initial estimator. Therefore, our method is computationally more efficient.

We will illustrate them in Section 3.2 after we provide the detailed description of the method.

1.1. *Organization and notations*. The rest of the paper is organized as follows. In Section 2, we review the related literature on recent works on distributed estimation and inference. Section 3 describes the proposed inference procedure for QR under memory constraints. Section 4 presents the theoretical results. In Section 5, we demonstrate the performance of the proposed inference procedure by simulated experiments, followed by conclusions in Section 6. The proofs and additional experimental results are provided in the supplementary material (Chen, Liu and Zhang (2019)).

In the QR model in (1), let $F(\cdot|\boldsymbol{x})$ and $f(\cdot|\boldsymbol{x})$ denote the CDF and PDF of ε conditioning on $\boldsymbol{X} = \boldsymbol{x}$, respectively, throughout the paper. Then, for any $\boldsymbol{x}$, we have $F(0|\boldsymbol{x}) = \tau$. For two sequences of real numbers $f(n)$ and $g(n)$, let $f(n) = \Omega(g(n))$ denote that f is bounded below by g (up to constant factor) asymptotically. For a set of random variables X_n and a corresponding set of constants a_n, $X_n = O_p(a_n)$ means that X_n/a_n is stochastically bounded and $X_n = o_p(a_n)$ means that X_n/a_n converges to zero in probability as n goes to infinity. For a real number c, we will use $\lfloor c \rfloor$ to denote largest integer less than or equal to c. Finally, denote the Euclidean norm for a vector $\boldsymbol{x} \in \mathbb{R}^p$ by $\|\boldsymbol{x}\|_2$, and denote the spectral norm for a matrix $\boldsymbol{X}$ by $\|\boldsymbol{X}\|$.

2. Related works. The explosive growth of data presents new challenges for many classical statistical problems. In recent years, a large body of literature has emerged on studying estimation and inference problems under memory constraints or in distributed environments (please see the references described below as well as other works such as Kleiner et al. (2014), Wang and Dunson (2014), Wang et al. (2015)). Examples include but are not limited to density parameter estimation (Li, Lin and Li (2013)), generalized linear regression with nonconvex penalties (Chen and Xie (2014)), kernel ridge regression (Zhang, Duchi and Wainwright (2015)), high-dimensional sparse linear regression (Lee et al. (2017)), high-dimensional generalized linear models (Battey et al. (2018)), semiparametric partial linear models (Zhao, Cheng and Liu (2016)), QR processes (Volgushev, Chao and Cheng (2018)), M-estimators with cubic rate (Shi, Lu and Song (2017)) and some nonstandard problems where rates of convergence are slower than $n^{1/2}$ and limit distributions are non-Gaussian (Banerjee, Durot and Sen (2018)). All of these results rely on averaging, where the global estimator is the average of the local estimators computed on each batch of data. For the averaging estimators to achieve the same asymptotic distribution for inference as pooling the data together, it usually requires the number of batches (i.e., the number of machines) to be $o(m)$ (i.e., $n = o(m^2)$). However, in some applications such as sensor networks and streaming data, the number of batches can be large.

To address the challenge, instead of using one-shot aggregation via averaging, recent works by Jordan, Lee and Yang (2018) and Wang et al. (2017) proposed iterative methods with multiple rounds of aggregations, which relaxes the condition $n = o(m^2)$. These methods have been applied to the M-estimator and Bayesian inference. Their framework is based on an approximate Newton algorithm (Shamir, Srebro and Zhang (2014)) and thus requires the twice-differentiability of the loss function. However, the QR loss is nondifferentiable, and thus their approach cannot be utilized. More detailed discussions on these two works will be provided in Remark 4.3. Rajagopal, Wainwright and Varaiya (2006) proposed a multiround decentralized quantile estimation algorithm under a restrictive communication-constrained setup. However, their method cannot be applied to solve QR problems.

There is a large body of literature on estimation and inference for QR and its variant (e.g., censored QR). We will not be able to provide a detailed survey here and we refer the readers to Koenker (2005), Koenker et al. (2018) for more background knowledge and recent development of QR. However, it is worth noting that the smoothing idea in QR literature has been adopted for developing our linear estimator for QR (LEQR in Section 3.1), which serves as the cornerstone of our method. The idea of smoothing the nonsmooth QR objective goes back to Horowitz (1998), where he studied the bootstrap refinement in inference in quantile models. Since the smoothing idea overcomes the difficulty in higher-order expansion of the scores associated with the QR objective, it plays an important role in solving various QR problems. For example, Wang, Stefanski and Zhu (2012) and Wu, Ma and Yin (2015) proposed different smoothed objectives to determine the corrected scores under the presence of covariate measurement errors for QR and censored QR. Galvao and Kato (2016) proposed a fix-effects estimator for the smoothed QR in linear panel data models and derived the corresponding limiting distribution. Pang, Lu and Wang (2012) proposed an induced-smoothing idea for estimating the variance of inverse-censoring-probability weighted estimator in Bang and Tsiatis (2002). Whang (2006) considered the problem of inference using the empirical likelihood mehod for QR and demonstrated that the smoothed empirical likelihood can help achieve higher-order refinements (i.e., $O(n^{-1})$ of the coverage error). The smoothing idea can also facilitate the computation, especially for the first-order optimization methods (see, e.g., Zheng (2011)). We also adopt the smoothing idea for constructing our LEQR estimator (see Section 3.1), which heavily relies on the first-order optimality condition of the objective. Instead of using the smoothing technique for computing a one-stage estimator as in existing literature, our use of the smoothing technique enables successive refinement of the LEQR estimator (see Propositions 4.1 and 4.2).

3. Methodology. In this section, we introduce the proposed method. We start with a new linear type estimator for quantile regression, which serves as an important building block for our inference approach.

3.1. *A linear type estimator of quantile regression.* We first propose a linear type estimator for quantile regression, which is named as LEQR Linear Estimator for Quantile Regression (LEQR). Recall the classical quantile regression estimator from (2). Note that for the quantile regression, the loss function $\rho_\tau(x) = x(\tau - I\{x \le 0\}) = x(I\{x > 0\} + \tau - 1)$ is nondifferentiable. Using the smoothing idea (see the literature surveyed in Section 2), we approximate the indicator factor $I\{x > 0\}$ with a smooth function $H(x/h)$, where $h \to 0$ is the bandwidth. With this approximation, we replace $\rho_\tau(x)$ in quantile regression by $K_h(x) = x(H(x/h) + \tau - 1)$ and define

$$\widehat{\boldsymbol{\beta}}_h = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} K_h(Y_i - \boldsymbol{X}_i'\boldsymbol{\beta}). \tag{4}$$

Now the right-hand side in (4) is differentiable and we note that $\frac{\mathrm{d}K_h(x)}{\mathrm{d}x} = H(x/h) + \tau - 1 + (x/h)H'(x/h)$. Here, the function $H(u)$ is a smooth approximation of the indicator factor $I\{x > 0\}$ satisfying $H(u) = 1$ when $u \ge 1$ and $H(u) = 0$ when $u \le -1$ (see more details on the condition of $H(\cdot)$ in Condition (C3) on page 3256). For example, one may choose $H(u)$ for $u \in [-1, 1]$ to be the integral of a smooth kernel function with support on $[-1, 1]$, for example, a biweight (or quartic) kernel (see (25)). We further note that (4) is a nonconvex optimization, and thus it is difficult to compute the minimizer $\widehat{\boldsymbol{\beta}}_h$. However, it is not a concern since (4) is only introduced for the motivation purpose. The proposed LEQR estimator, which is explicitly defined in (7) below, does not require to solve (4).

Since $H(\cdot)$ is a smooth function, by the first-order optimality condition, the solution $\widehat{\boldsymbol{\beta}}_h$ in (4) satisfies (see Theorem 2.6 in Beck (2014))

$$\sum_{i=1}^{n} \boldsymbol{X}_i \left\{ H\left(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_h}{h}\right) + \tau - 1 + \frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_h}{h} H'\left(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_h}{h}\right)\right\} = 0. \tag{5}$$

From (5), we can express $\widehat{\boldsymbol{\beta}}_h$ by

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_h = {} & \left(\sum_{i=1}^{n} \boldsymbol{X}_i\boldsymbol{X}_i' \frac{1}{h} H'\left(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_h}{h}\right)\right)^{-1} \\ & \times \left[\sum_{i=1}^{n} \boldsymbol{X}_i \left\{ H\left(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_h}{h}\right) + \tau - 1 + \frac{Y_i}{h} H'\left(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_h}{h}\right)\right\}\right].\end{aligned} \tag{6}$$

However, there is no closed-form expression of $\widehat{\boldsymbol{\beta}}_h$ from this fixed-point equation. Instead of using $\widehat{\boldsymbol{\beta}}_h$ on the right-hand side of (6), we replace $\widehat{\boldsymbol{\beta}}_h$ by a consistent initial estimator $\widehat{\boldsymbol{\beta}}_0$, which leads to the proposed LEQR:

$$\widehat{\boldsymbol{\beta}} = \left(\sum_{i=1}^{n} \boldsymbol{X}_i\boldsymbol{X}_i' \frac{1}{h} H'\left(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{h}\right)\right)^{-1}$$

$$\times \left[\sum_{i=1}^{n} \boldsymbol{X}_i \left\{ H\left(\frac{Y_i - \boldsymbol{X}_i' \widehat{\boldsymbol{\beta}}_0}{h} \right) + \tau - 1 + \frac{Y_i}{h} H'\left(\frac{Y_i - \boldsymbol{X}_i' \widehat{\boldsymbol{\beta}}_0}{h} \right) \right\} \right]. \tag{7}$$

Allowing the choice of initial estimator $\widehat{\boldsymbol{\beta}}_0$ is crucial for our inference procedure under the memory constraint. While it is difficult to solve the fixed-point equation in (6) when all data cannot be loaded into memory, we are still able to compute an initial estimator using only a batch of samples, for example, $\mathcal{D}_1$. Given the initial estimator $\widehat{\boldsymbol{\beta}}_0$, the LEQR in (6) only depends on sums of $\boldsymbol{X}_i$ and $\boldsymbol{X}_i \boldsymbol{X}_i'$, which can be easily implemented via a divide-and-conquer scheme (see Section 3.2 for details). Comparing to the naive-DC method that needs to solve N optimization problems on each batch of data, LEQR only needs to solve one optimization, and thus is computationally more efficient. Moreover, our estimator in (7) is essentially solving a linear equation system, and we do not need to explicitly compute a matrix inversion. There are a number of efficient methods for solving a linear system numerically, such as conjugate gradient method (Hestenes and Stiefel (1952)) and stochastic variance reduced gradient method (Johnson and Zhang (2013)). In our simulation studies, we use the conjugate gradient method, and due to space limitations, more detailed explanations of this method are relegated to Section B in the supplementary material.

3.2. *Divide-and-conquer LEQR.* Based on LEQR, we now introduce a divide-and-conquer LEQR for estimating $\boldsymbol{\beta}(\tau)$. For each batch of data $\mathcal{D}_k$ for $1 \le k \le N$, let us define the following quantities:

$$\begin{aligned} \boldsymbol{U}_k &= \sum_{i \in \mathcal{H}_k} \boldsymbol{X}_i \left\{ H\left(\frac{Y_i - \boldsymbol{X}_i' \widehat{\boldsymbol{\beta}}_0}{h} \right) + \tau - 1 + \frac{Y_i}{h} H'\left(\frac{Y_i - \boldsymbol{X}_i' \widehat{\boldsymbol{\beta}}_0}{h} \right) \right\}, \\ \boldsymbol{V}_k &= \sum_{i \in \mathcal{H}_k} \boldsymbol{X}_i \boldsymbol{X}_i' \frac{1}{h} H'\left(\frac{Y_i - \boldsymbol{X}_i' \widehat{\boldsymbol{\beta}}_0}{h} \right). \end{aligned} \tag{8}$$

The inference procedure is presented in Algorithm 1. Using our theory in Theorem 4.3 and 4.4, for the gth iteration, we choose the bandwidth $h = h_g = \max(\sqrt{p/n}, (p/m)^{2^{g-2}})$ for $1 \le g \le q$.

Algorithm 1 cannot only be used under the memory constraint, but also be applied to distributed setting, to reduce computational cost for classical quantile regression, and to deal with streaming data. We illustrate these important applications as follows:

1. *Quantile regression in large-scale sensor networks.* Algorithm 1 is directly applicable to distributed sensor network with N sensors, where each sensor collects a batch of data $\mathcal{D}_k$ (for $k = 1, \ldots, N$). The base station first broadcasts the initial estimator $\widehat{\boldsymbol{\beta}}_0$ computed on $\mathcal{D}_1$ in (9) to all sensors (see Figure 1 for an illustration). Then each sensor computes $(\boldsymbol{U}_k, \boldsymbol{V}_k)$ locally, which will be transferred

Algorithm 1 Divide-and-conquer (DC) LEQR

Input: Data batches $\mathcal{D}_k$ for $k = 1, \ldots, N$, the number of iterations/aggregations q, quantile level τ, smooth function H, a sequence of bandwidths h_g for $g = 1, \ldots, q$.

1: **for** $g = 1, 2 \ldots, q$ **do**
2: **if** $g = 1$ **then**
3: Calculate the initial estimator (for the first iteration) based on $\mathcal{D}_1$:

$$\widehat{\boldsymbol{\beta}}_0 = \operatorname*{arg\,min}_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i \in \mathcal{H}_1} \rho_\tau(Y_i - \boldsymbol{X}_i' \boldsymbol{\beta}). \tag{9}$$

4: **else**
5: Set the initial estimator to be the estimator from the previous iteration:

$$\widehat{\boldsymbol{\beta}}_0 = \widehat{\boldsymbol{\beta}}^{(g-1)}$$

6: **end if**
7: **for** $k = 1, \ldots, N$ **do**
8: Swap data $\mathcal{D}_k$ into the memory and compute $(\boldsymbol{U}_k, \boldsymbol{V}_k)$ according to (8) using the bandwidth $h := h_g$.
9: Compute and maintain the sums $(\sum_{j=1}^k \boldsymbol{U}_j, \sum_{j=1}^k \boldsymbol{V}_j)$ in the memory (and delete $(\boldsymbol{U}_k, \boldsymbol{V}_k)$).
10: **end for**
11: Compute the estimator $\widehat{\boldsymbol{\beta}}^{(g)}$:

$$\widehat{\boldsymbol{\beta}}^{(g)} = \left(\sum_{k=1}^N \boldsymbol{V}_k \right)^{-1} \left(\sum_{k=1}^N \boldsymbol{U}_k \right). \tag{10}$$

12: **end for**

Output: The final estimator $\widehat{\boldsymbol{\beta}}^{(q)}$.

from bottom to the base station. In particular, each sensor k only keeps the *summation* of $(\boldsymbol{U}_\cdot, \boldsymbol{V}_\cdot)$ from all its children nodes and its own $(\boldsymbol{U}_k, \boldsymbol{V}_k)$ and then transfers the summed statistics to its parent. After receiving $(\sum_{k=1}^N \boldsymbol{U}_k, \sum_{k=1}^N \boldsymbol{V}_k)$, the base station will compute $\widehat{\boldsymbol{\beta}}^{(g)}$ for $g = 1$ in (10). Then this distributed procedure can be repeated for $g = 2, \ldots, q$.

2. *Computational reduction of quantile regression.* Algorithm 1 can also be utilized as an efficient solver for classical quantile regression on a single machine. For the ease of illustration, let us assume the quantile regression estimator in (2) (or (3)) is solved by the standard interior-point method. Using the standard interior-point method, the initial estimator requires the computational complexity $O(m^{1.25} p^3 \log m)$ (Portnoy and Koenker (1997)). The computation of $\sum_{k=1}^N \boldsymbol{U}_k$ and $\sum_{k=1}^N \boldsymbol{V}_k$ require $O(np)$ and $O(np^2)$, respectively. Therefore, the computa-

tional complexity for $\widehat{\boldsymbol{\beta}}^{(q)}$ is at most $O(m^{1.25}p^3 \log m + np^2 + p^3)$, where $O(p^3)$ comes from the inversion of $\sum_{k=1}^{N} \boldsymbol{V}_k$. This greatly saves the computational cost as compared to the interior-point method for computing quantile regression estimator on the entire data, which requires a complexity of $O(n^{1.25}p^3 \log n)$.

3. *Online quantile regression for streaming data.* To deal with online streaming data, it is critical to design a one-pass algorithm since streaming data are transient. To this end, based on Algorithm 1, we develop a new one-pass algorithm (see Algorithm 2) that provides a sequence of successively refined estimators.

For streaming data, we divide the data into intervals $\{(s_l, r_l)\}_{l=1}^{\infty}$. The starting and ending positions of the lth interval are chosen as

$$s_l = \lfloor m^{a_{l-1}} \rfloor + 1 \quad \text{and} \quad r_l = \lfloor m^{a_l} \rfloor$$

for $l \geq 1$ where $a_{2k-1} = 2^{k-1} + 1/2$ and $a_{2k} = 2^{k-1} + 3/4$ for $k \geq 1$ and $a_0 = -\infty$. The intervals are chosen to ensure that the sample size of the lth interval n_l is approximately n_{l-2}^2. As we will show in the proof of Theorem 4.5, if an initial estimator is computed from m samples, there will be no improvement of the online LEQR estimator after m^2 fresh samples and this is the ending point of an interval where we compute a new initial estimator.

In Algorithm 2, for each interval l and each j such that $s_l \leq j \leq r_l$, the memory only maintains $\widehat{\boldsymbol{\beta}}[r_{l-1}]$, $(\boldsymbol{U}(r_{l-1}), \boldsymbol{V}(r_{l-1}))$, $(\boldsymbol{U}(j), \boldsymbol{V}(j))$ and $\widehat{\boldsymbol{\beta}}[j]$. We note that $(\boldsymbol{U}(r_{l-1}), \boldsymbol{V}(r_{l-1}))$ are the weighted sums of $\boldsymbol{X}_i$ and $\boldsymbol{X}_i\boldsymbol{X}_i'$ for $s_{l-1} \leq i \leq r_{l-1}$ and $(\boldsymbol{U}(j), \boldsymbol{V}(j))$ can be easily updated from $(\boldsymbol{U}(j-1), \boldsymbol{V}(j-1))$ in an online fashion. Therefore, except for an $O(m)$ space for deriving the initial estimator $\widehat{\boldsymbol{\beta}}[0]$, the online LEQR only requires $O(p^2)$ memory, which is independent on n.

In Theorem 4.5, we will show that the online LEQR algorithm achieves the same statistical efficiency for any l and j as the standard quantile regression estimator when merging all the streaming data together. Also, the asymptotic normality of $\sqrt{m+j}(\widehat{\boldsymbol{\beta}}[j] - \boldsymbol{\beta}(\tau))$ holds uniformly in $1 \leq j \leq m^A$ for any constant $A > 0$ (note that $m + j$ is the sample size until time j).

4. Theoretical results. In this section, we provide a Bahadur representation of $\widehat{\boldsymbol{\beta}}^{(q)}$, based on which we derive the asymptotic normality result for DC LEQR $\widehat{\boldsymbol{\beta}}^{(q)}$ and the online LEQR $\widehat{\boldsymbol{\beta}}[j]$. We also discuss adaptive choices of bandwidth and extensions to heterogeneous settings.

4.1. *Asymptotics for DC LEQR.* We note that (7) can be equivalently written as

$$\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}(\tau) = \boldsymbol{D}_{n,h}^{-1} \boldsymbol{A}_{n,h}, \tag{12}$$

where

$$\boldsymbol{A}_{n,h} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{X}_i \left\{ H\left(\frac{Y_i - \boldsymbol{X}_i' \widehat{\boldsymbol{\beta}}_0}{h} \right) + \tau - 1 + \frac{\varepsilon_i}{h} H'\left(\frac{Y_i - \boldsymbol{X}_i' \widehat{\boldsymbol{\beta}}_0}{h} \right) \right\},$$

Algorithm 2 Online LEQR

Initialization: For the first m samples $\{Y_i, \boldsymbol{X}_i, -m+1 \le i \le 0\}$, compute the initial standard quantile regression estimator $\widehat{\boldsymbol{\beta}}[0]$. Then constructs the initial $(\boldsymbol{U}(0), \boldsymbol{V}(0))$ based on $\widehat{\boldsymbol{\beta}}[0]$ according to (8) with $h = (p/m)^{1/2}$.

Parameter Setup: Define $a_{2k-1} = 2^{k-1} + 1/2$ and $a_{2k} = 2^{k-1} + 3/4$ for $k \ge 1$ and $a_0 = -\infty$. Let $s_l = \lfloor m^{a_{l-1}} \rfloor + 1$ and $r_l = \lfloor m^{a_l} \rfloor$ for any $l \ge 1$ and let $r_0 = 0$. Define a sequence of bandwidths $h_1 = (p/m)^{1/2}$ and $h_l = (p/m^{a_{l-1}})^{1/2}$ for $l \ge 2$.

1: **for** each interval $l = 1, 2, \ldots,$ **do**

2: **for** indices in the interval l, $j = s_l, s_l + 1, \ldots, r_l$ **do**

3: Receive an online sample $(Y_j, \boldsymbol{X}_j)$.

4: Compute

$$\widetilde{\boldsymbol{U}}(j) = \boldsymbol{X}_j \Big\{ H\Big(\frac{Y_j - \boldsymbol{X}_j' \widehat{\boldsymbol{\beta}}[r_{l-1}])}{h_l}\Big) + \tau - 1 + \frac{Y_j}{h_l} H'\Big(\frac{Y_j - \boldsymbol{X}_j' \widehat{\boldsymbol{\beta}}[r_{l-1}]}{h_l}\Big)\Big\}$$

$$\widetilde{\boldsymbol{V}}(j) = \boldsymbol{X}_j \boldsymbol{X}_j' \frac{1}{h_l} H'\Big(\frac{Y_j - \boldsymbol{X}_j' \widehat{\boldsymbol{\beta}}[r_{l-1}]}{h_l}\Big),$$

where $\widehat{\boldsymbol{\beta}}[r_{l-1}]$ is the estimator computed up to the end of $(l-1)$th interval.

5: Update $(\boldsymbol{U}(j), \boldsymbol{V}(j))$ by

$$\boldsymbol{U}(j) \triangleq \sum_{i=s_l}^{j} \widetilde{\boldsymbol{U}}(j) = \begin{cases} \widetilde{\boldsymbol{U}}(j) & \text{if } j = s_l, \\ \boldsymbol{U}(j-1) + \widetilde{\boldsymbol{U}}(j) & \text{if } j > s_l, \end{cases}$$

$$\boldsymbol{V}(j) \triangleq \sum_{i=s_l}^{j} \widetilde{\boldsymbol{V}}(j) = \begin{cases} \widetilde{\boldsymbol{V}}(j) & \text{if } j = s_l, \\ \boldsymbol{V}(j-1) + \widetilde{\boldsymbol{V}}(j) & \text{if } j > s_l. \end{cases}$$

6: Calculate $(\boldsymbol{U}(r_{l-1}) + \boldsymbol{U}(j), \boldsymbol{V}(r_{l-1}) + \boldsymbol{V}(j))$ and compute the online quantile regression estimator at time j:

$$\widehat{\boldsymbol{\beta}}[j] = \big(\boldsymbol{V}(r_{l-1}) + \boldsymbol{V}(j)\big)^{-1} \big(\boldsymbol{U}(r_{l-1}) + \boldsymbol{U}(j)\big). \tag{11}$$

7: Remove $(\widetilde{\boldsymbol{U}}(j), \widetilde{\boldsymbol{V}}(j))$, $(\boldsymbol{U}(j-1), \boldsymbol{V}(j-1))$ from the memory.

8: **end for**

9: Remove $(\boldsymbol{U}(r_{l-1}), \boldsymbol{V}(r_{l-1}))$ from the memory and only keep $\widehat{\boldsymbol{\beta}}[r_l]$, $(\boldsymbol{U}(r_l), \boldsymbol{V}(r_l))$ in the memory.

10: **end for**

$$\boldsymbol{D}_{n,h} = \frac{1}{nh}\sum_{i=1}^{n} \boldsymbol{X}_i \boldsymbol{X}_i' H'\Big(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{h}\Big).$$

We first state some regularity conditions for our theoretical development and then give Propositions 4.1–4.2 on the expansions of $\boldsymbol{A}_{n,h}$ and $\boldsymbol{D}_{n,h}$. We assume the model (1) with n *i.i.d.* samples $\{(\boldsymbol{X}_i, Y_i)\}$ (the non-i.i.d. case will be discussed in the next subsection). Let $f(\cdot|\boldsymbol{X})$ be the conditional density function of the noise ε given $\boldsymbol{X}$. Further, we define $\boldsymbol{D} = \mathbb{E}(\boldsymbol{X}\boldsymbol{X}' f(0|\boldsymbol{X}))$.

(C1) The conditional density function $f(\cdot|\boldsymbol{X})$ is Lipschitz continuous (i.e., $|f(x|\boldsymbol{X}) - f(y|\boldsymbol{X})| \le L|x-y|$ for any $x, y \in \mathbb{R}$ and some constant $L > 0$). Also, assume that $0 < c_1 \le \lambda_{\min}(\boldsymbol{D}) \le \lambda_{\max}(\boldsymbol{D}) \le c_2 < \infty$ for some constants c_1, c_2.

(C2) Let the smoothing function $H(\cdot)$ satisfy $H(u) = 1$ if $u \ge 1$ and $H(u) = 0$ if $u \le -1$. Further, suppose $H(\cdot)$ is twice differentiable and its second derivative $H^{(2)}(\cdot)$ is bounded. Moreover, we assume the bandwidth $h = o(1)$.

(C3) Assume that $p = o(nh/(\log n))$ and $\sup_{\|\boldsymbol{\theta}\|_2=1} \mathbb{E}e^{\eta(\boldsymbol{\theta}'\boldsymbol{X})^2} \le C$ for some $\eta, C > 0$.

(C3*) Assume for some $\kappa > 0$, $p = o((n^{1-4\kappa}h/\log n)^{1/3})$. Suppose that $\sup_j \mathbb{E}|X_{1,j}|^a \le C_1$ for some $a \ge 2/\kappa$ and $\sup_{\|\boldsymbol{\theta}\|_2=1} \mathbb{E}(\boldsymbol{\theta}'\boldsymbol{X})^4 \le C_2$ for some $C_1, C_2 > 0$.

Condition (C1) contains a standard eigenvalue condition related to covariates $\boldsymbol{X}$ and the smoothness of the conditional density function f. Condition (C2) is a mild condition on H for smooth approximation.

Condition (C3) and (C3*) illustrate the relationship between the dimension p and sample size n and the moment condition on covariates $\boldsymbol{X}$. Either one of them leads to our theoretical results in Propositions 4.1–4.2. As compared to Condition (C3*), Condition (C3) is weaker in terms of the relationship of p and n, but requires a stronger moment condition on $\boldsymbol{X}$.

Under these conditions, we have the following Propositions 4.1 and 4.2 for the asymptotic behavior of $\boldsymbol{A}_{n,h}$ and $\boldsymbol{D}_{n,h}$.

PROPOSITION 4.1. *Suppose we have an initial estimator* $\widehat{\boldsymbol{\beta}}_0$ *with* $\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}(\tau)\|_2 = O_{\mathbb{P}}(a_n)$ *with* $a_n = O(h)$. *Assume that* (C1), (C2) *and* (C3) (*or* (C3*)) *hold. We have*

$$\Big\|\boldsymbol{A}_{n,h} - \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{X}_i\big(I\{\varepsilon_i \ge 0\} + \tau - 1\big)\Big\|_2 = O_{\mathbb{P}}\Big(\sqrt{\frac{ph\log n}{n}} + a_n^2 + h^2\Big).$$

PROPOSITION 4.2. *Suppose the conditions in Proposition* 4.1 *hold. We have*

$$\|\boldsymbol{D}_{n,h} - \boldsymbol{D}\| = O_{\mathbb{P}}\Big(\sqrt{\frac{p\log n}{nh}} + a_n + h\Big).$$

Combining Propositions 4.1 and 4.2 with (12) and with some algebraic manipulations, we have

$$\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}(\tau) = \frac{\boldsymbol{D}^{-1}}{n} \sum_{i=1}^{n} \boldsymbol{X}_i \big(I\{\varepsilon_i \geq 0\} + \tau - 1\big) + \boldsymbol{r}_n \tag{13}$$

with

$$\|\boldsymbol{r}_n\|_2 = O_{\mathbb{P}}\bigg(\sqrt{\frac{p^2 \log n}{n^2 h}} + \sqrt{\frac{ph \log n}{n}} + a_n^2 + h^2\bigg). \tag{14}$$

By choosing the bandwidth h shrinking at an appropriate rate, the dominating term of (14) is a_n^2, which means that one round of aggregation enables a refinement of the estimator with its bias reducing from a_n to a_n^2 (note that $\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}(\tau)\|_2 = O_{\mathbb{P}}(a_n)$). Therefore, an iterative refinement of the initial estimator will successively improve the estimation accuracy. The effect of bias reduction is mainly due to the term $\frac{Y_i}{h} H'(\frac{Y_i - \boldsymbol{X}_i' \widehat{\boldsymbol{\beta}}_0}{h})$ in (7), which is induced by the smoothing technique (please see more details in the proof of Proposition 4.1).

The previous discussions only involve one round of aggregation. Now we are ready to present the theoretical results for our DC LEQR $\widehat{\boldsymbol{\beta}}^{(q)}$ in Algorithm 1 with multiple rounds of aggregations. By a recursive argument based on (13), we establish the following Bahadur representation.

THEOREM 4.3. *Assume the initial estimator* $\widehat{\boldsymbol{\beta}}_0$ *in* (9) *satisfies* $\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}(\tau)\|_2 = O_{\mathbb{P}}(\sqrt{p/m})$. *Let* $h_g = \max(\sqrt{p/n}, (p/m)^{2^{g-2}})$ *for* $1 \leq g \leq q$. *Assuming that* (C1), (C2) *and* (C3) (*or* (C3*)) *hold with* $h = h_q$ *and* p *also satisfies* $p = O(m/(\log n)^2)$, *then we have*

$$\widehat{\boldsymbol{\beta}}^{(q)} - \boldsymbol{\beta}(\tau) = \frac{\boldsymbol{D}^{-1}}{n} \sum_{i=1}^{n} \boldsymbol{X}_i \big(I\{\varepsilon_i \geq 0\} + \tau - 1\big) + \boldsymbol{r}_n \tag{15}$$

with

$$\|\boldsymbol{r}_n\|_2 = O_{\mathbb{P}}\bigg(\sqrt{\frac{ph_q \log n}{n}} + \bigg(\frac{p}{m}\bigg)^{2^{q-1}}\bigg). \tag{16}$$

The classical initial estimator based on QR in (9) will satisfy $\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}(\tau)\|_2 = O_{\mathbb{P}}(\sqrt{p/m})$ under some regularity conditions; see He and Shao (2000). According to our choice of bandwidth, we have $h_1 = \sqrt{p/m}$ and thus the convergence rate of the initial estimator is $O(h_1)$, which satisfies the condition in Proposition 4.1. Furthermore, we note that any initial estimator $\widehat{\boldsymbol{\beta}}_0$ with $\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}(\tau)\|_2 = O_{\mathbb{P}}(\sqrt{p/m})$ can be used in the first iteration and the same Bahadur representation in Theorem 4.3 holds.

The condition $p = O(m/(\log n)^2)$ in Theorem 4.3 ensures that $\sqrt{p/m} = o(1)$, which implies the consistency of the initial estimator (and the $1/(\log n)^2$ factor

is used for balancing the terms in $\|\boldsymbol{r}_n\|_2$ in (14)). This condition on p cannot be implied by either (C3) or (C3*); and we choose to present (C3) (or (C3*)) at the beginning since they are the minimum requirements to obtain Propositions 4.1–4.2. On the other hand, the condition $p = o(nh_q/\log n)$ in (C3) will be satisfied if $p = o(n/(\log n)^2)$ since $h_q \geq \sqrt{p/n}$. Therefore, we can simply impose a stronger condition $p = o(m/(\log n)^2)$ that unifies the condition $p = O(m/(\log n)^2)$ and that in (C3).

REMARK 4.1 (Nearly optimal rate of the Bahadur remainder term). From Theorem 4.3, as long as

$$q \geq 2 + \log\{\log(\sqrt{p/n})/\log(p/m)\}/\log 2, \tag{17}$$

the bandwidth for the qth iteration is $h_q = \sqrt{p/n}$. Then the first term in the right-hand side of (16) is $(p/n)^{3/4}\sqrt{\log n}$ and the second term is bounded by p/n, which is dominated by the first term. Therefore, the Bahadur remainder term $\boldsymbol{r}_n$ of our method achieves a nearly optimal rate

$$\|\boldsymbol{r}_n\|_2 = O_{\mathbb{P}}\big((p/n)^{3/4}(\log n)^{1/2}\big). \tag{18}$$

In fact, for classical QR estimator $\widehat{\boldsymbol{\beta}}_{\mathrm{QR}}$ and fixed p, it is known that the rate $n^{-3/4}$ cannot be improved except for a $\log n$ term (Koenker (2005)). Note that in a common scenario when $n = O(m^A)$ and $p = O(m^\delta)$ for some constants $A \geq 1$ and $0 < \delta < 1$, the right-hand side of (17) is bounded by a constant. Therefore, a constant number of rounds of aggregations is sufficient to obtain a nearly optimal rate in Bahadur representation.

Applying the central limit theorem to Theorem 4.3, we obtain the asymptotic distribution of $\widehat{\boldsymbol{\beta}}^{(q)} - \boldsymbol{\beta}(\tau)$ as follows.

THEOREM 4.4. *Suppose that all the conditions in Theorem 4.3 hold. Further, assume that $n = O(m^A)$ for some constant $A \geq 1$, $p = o(\min\{n^{1/3}/(\log n)^{2/3}, m^\delta\})$ for some $0 < \delta < 1$ and the number of iterations q satisfies (17). By choosing the bandwidth sequence $h_g = \max(\sqrt{p/n}, (p/m)^{2^{g-2}})$ for $1 \leq g \leq q$, for any $\boldsymbol{v} \in \mathbb{R}^p$ with $\boldsymbol{v} \neq 0$,*

$$\frac{n^{1/2}\boldsymbol{v}'(\widehat{\boldsymbol{\beta}}^{(q)} - \boldsymbol{\beta}(\tau))}{\sqrt{\boldsymbol{v}'\boldsymbol{D}^{-1}\mathbb{E}[\boldsymbol{X}\boldsymbol{X}']\boldsymbol{D}^{-1}\boldsymbol{v}}} \Rightarrow N\big(0, \tau(1-\tau)\big) \tag{19}$$

as $m, n \to \infty$.

Note that to establish central limit theorem, we need $p = o(n^{1/3}/(\log n)^{2/3})$ in Theorem 4.4, which ensures that $\|\boldsymbol{r}_n\|_2 = o(1/\sqrt{n})$ (see (16)). Therefore, as the first term in (15) is an average of n *i.i.d.* zero-mean random vectors, the reminder term $\boldsymbol{r}_n$ is dominated by each coordinate of the first term in (15). For the classical

QR estimator $\widehat{\boldsymbol{\beta}}_{\mathrm{QR}}$ in (2) (assuming all data is pooled together), the corresponding condition should be $p = o(n^{1/3}/(\log n)^{2/3})$; see He and Shao (2000). This is the same with our condition except for the term m^{δ} which is required for the consistency of the initial estimator in our method. We also note that the conditions $n = O(m^A)$ and $p = o(m^{\delta})$ ensure that the number of required iterations q from (17) is a constant (see Remark 4.1). Therefore, we only need to perform a constant number of aggregations as $m, n \to \infty$.

Theorem 4.4 shows that $\widehat{\boldsymbol{\beta}}^{(q)}$ achieves the same asymptotic efficiency as $\widehat{\boldsymbol{\beta}}_{\mathrm{QR}}$ in (2) computed directly on all the samples. When p is fixed, as compared to the naive-DC that also achieves (19) but under the condition $n = o(m^2)$, our approach removes the restriction on the relationship of m and n by applying multiple rounds of aggregations. It is also important to note that the required number of rounds q in (17) is usually quite small even with a large dimension p.

Given (19), we only need consistent estimators of $\boldsymbol{D}$ and $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}']$ to construct confidence interval of $\boldsymbol{v}'\boldsymbol{\beta}(\tau)$ for any given $\boldsymbol{v}$. It is natural to use $\boldsymbol{D}_{n,h}$ and $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i\boldsymbol{X}_i'$ to estimate $\boldsymbol{D}$ and $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}']$, respectively. These estimators can be easily implemented under memory constraint by averaging the local sample estimators on each batch of data. The proofs of Propositions 4.1, 4.2 and Theorems 4.3, 4.4 are provided in the supplementary material of Chen, Liu and Zhang (2019).

REMARK 4.2 (Data-adaptive choices of bandwidth). In practice, one could use the bandwidth $h_g = c_g \max(\sqrt{p/n}, (p/m)^{2^{g-2}})$ with a scaling constant c_g to further improve the empirical performance. An intuitive data-adaptive way of choosing c_g is provided as follows. Given a set of candidate choices for c_g (e.g., $\{c_1, \ldots, c_L\}$), we choose the best c_g by minimizing

$$S(c) := \left\| \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{X}_i \big(I\{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_c^{(g)} \geq 0\} + \tau - 1\big) \right\|_2, \tag{20}$$

where $\widehat{\boldsymbol{\beta}}_c^{(g)}$ denotes the estimator in the gth round of aggregation with the constant c in the bandwidth. That is, $c_g = \arg\min_{c\in\{c_1,\ldots,c_L\}} S(c)$. In a distributed setting, the method only requires a small amount of communication. More specifically, given $\widehat{\boldsymbol{\beta}}_c^{(g)}$, each machine k returns $\sum_{i\in\mathcal{H}_k} \boldsymbol{X}_i(I\{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_c^{(g)} \geq 0\} + \tau - 1)$ (i.e., an $O(p)$ vector) to the center for computing $S(c)$. We also evaluate the performance of our algorithm with the use of data-adaptive bandwidth in Section 5.3.

It is worthwhile noting that the bandwidth tuning is not a critical issue for our algorithm (in contrast to many other smoothed QR estimators) since our estimator is constructed via multiple rounds of aggregations. Even using an inaccurate constant in bandwidth (as long as the bandwidths for different rounds shrink at the right rate of $(p/m)^{2^{g-2}}$), our method can achieve good performance by simply performing more rounds of aggregations. In Section 5.3, we will provide simulation studies to show that our algorithm is insensitive to the scaling constant in the bandwidth.

REMARK 4.3 (Discussions with related literature). Note that our estimator $\widehat{\boldsymbol{\beta}}$ can be written as

$$\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_0 + \boldsymbol{D}_{n,h}^{-1}\Bigg[\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i\Bigg\{H\bigg(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{h}\bigg) + \tau - 1 + \frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{h}H'\bigg(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{h}\bigg)\Bigg\}\Bigg].$$

This formula is closely related to the estimator for quantile regression considered in Pang, Lu and Wang (2012), where they introduced the estimator (noncensored version)

$$\widehat{\boldsymbol{\beta}}_{\mathrm{PLW}} = \widehat{\boldsymbol{\beta}}_0 + \boldsymbol{A}_n^{-1}\Bigg[\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i\Bigg\{H\bigg(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{\sqrt{\boldsymbol{X}_i'W\boldsymbol{X}_i}}\bigg) + \tau - 1\Bigg\}\Bigg],$$

where $\boldsymbol{A}_n = \frac{1}{n}\sum_{i=1}^{n}\frac{\boldsymbol{X}_i\boldsymbol{X}_i'}{\sqrt{\boldsymbol{X}_i'W\boldsymbol{X}_i}}H'(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{\sqrt{\boldsymbol{X}_i'W\boldsymbol{X}_i}})$ and W is a weight matrix with order $O(1/n)$. Pang, Lu and Wang (2012) applied the smoothing to the original score function $\sum_{i=1}^{n}\boldsymbol{X}_i(I\{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0 \geq 0\} + \tau - 1)$, while our estimator comes from the smoothing to the loss function $\rho_\tau(\cdot)$, and hence we have an additional term $\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{h}H'(\frac{Y_i - \boldsymbol{X}_i'\widehat{\boldsymbol{\beta}}_0}{h})$. Note that this term plays a key role in reducing the bias induced by the initial estimator $\widehat{\boldsymbol{\beta}}_0$ to $\|\widehat{\boldsymbol{\beta}}_0 - \boldsymbol{\beta}(\tau)\|_2^2$. As pointed above, this allows our estimator to have a successive improvement on the estimation accuracy by iteratively updating the initial estimator.

Moreover, the recent work by Jordan, Lee and Yang (2018) and Wang et al. (2017) also proposed iterative approaches in distributed setting for successive refinement of an estimator. However, there are a few key differences between our DC LEQR and their approaches. First, their results require the loss function to have Lipschitz continuous second-order derivative, which is not satisfied by the original quantile loss function. Even if we replace the indicator function $I\{x \geq 0\}$ in the quantile loss by the smoothed version $H(x/h)$, the second derivative of the loss function will not satisfy their conditions (e.g., Assumption PD in Jordan, Lee and Yang (2018) requires that the "Lipschitz constant" of the second derivative has a uniform upper bound). Furthermore, our results allow $p \to \infty$ in the inference problem without ℓ_1-regularization.

REMARK 4.4 (General heterogenous case). We further consider a more general heterogenous case where $\{\boldsymbol{X}_i, \varepsilon_i\}$'s are independent, but not identically distributed from the model (1). Due to space limitations, this heterogenous case is relegated to Section C in the supplementary material.

4.2. *Asymptotics for online LEQR.* The next theorem gives the limiting behavior of the online LEQR in Algorithm 2.

THEOREM 4.5. *Suppose that* (C1)–(C3) *hold and* $p = o(m/(\log m)^2)$. *We have for any* $A > 0$ *and uniformly in* $1 \le j \le m^A$,

$$\widehat{\boldsymbol{\beta}}[j] - \boldsymbol{\beta}(\tau) = \frac{\boldsymbol{D}^{-1}}{m+j} \sum_{i=-m+1}^{j} \boldsymbol{X}_i \big(I\{\varepsilon_i \ge 0\} + \tau - 1\big) + \boldsymbol{r}_{m,j}, \tag{21}$$

where

$$\|\boldsymbol{r}_{m,j}\|_2 = O_{\mathbb{P}}\left(\sqrt{\frac{p}{m+j}}\left\{\left(\frac{p}{m}\right)^{1/4}\sqrt{\log m} + \frac{\sqrt{p}}{m^{1/4}}\right\}\right). \tag{22}$$

Furthermore, when $p = o(m^{1/4})$, *we have for any* $\boldsymbol{v} \in \mathbb{R}^p$ *with* $\boldsymbol{v} \ne 0$,

$$\frac{(m+j)^{1/2}\boldsymbol{v}'(\widehat{\boldsymbol{\beta}}[j] - \boldsymbol{\beta}(\tau))}{\sqrt{\boldsymbol{v}'\boldsymbol{D}^{-1}\mathbb{E}[\boldsymbol{X}\boldsymbol{X}']\boldsymbol{D}^{-1}\boldsymbol{v}}} \Rightarrow N\big(0, \tau(1-\tau)\big) \tag{23}$$

as $m \to \infty$.

We note that $m + j$ is the total number of used samples (including the samples for initialization) up to time j. From (21), we have $\|\widehat{\boldsymbol{\beta}}[j] - \boldsymbol{\beta}(\tau)\|_2 = O_{\mathbb{P}}(\sqrt{p/(m+j)})$ for any $1 \le j \le m^A$ when $\boldsymbol{r}_{m,j}$ is dominated by the first term. By Theorem 1 in Siegmund (1969), we can further obtain $\|\widehat{\boldsymbol{\beta}}[j] - \boldsymbol{\beta}(\tau)\|_2 = O_{\mathbb{P}}(\sqrt{\frac{p \log\log m}{m+j}})$ uniformly for $1 \le j \le m^A$. To establish the asymptotic distribution in (23), we need $p = o(m^{1/4})$, which ensures that $\|\boldsymbol{r}_{m,j}\|_2 = o(1/\sqrt{m+j})$ (see (22)).

5. Simulations. In this section, we provide simulation studies to illustrate the performance of DC LEQR for constructing confidence intervals for QR problems. We generate data from a linear regression model

$$Y_i = \boldsymbol{X}_i'\boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, 2, \ldots, n, \tag{24}$$

where $\boldsymbol{X}_i = (1, X_{i1}, \ldots, X_{ip})' \in \mathbb{R}^{p+1}$ is a random covariate vector. Here, $(X_{i1}, \ldots, X_{ip})'$ follows a multivariate uniform distribution $\mathrm{Unif}([0,1]^p)$ with $\mathrm{Corr}(X_{ij}, X_{ik}) = 0.5^{|j-k|}$ for $1 \le j \ne k \le p$. Please refer to Falk (1999) for the construction of such a multivariate uniform distribution. The regression coefficient vector $\boldsymbol{\beta} = \mathbf{1}_{p+1}$. The errors ε_i's are generated independently from the following three distributions:

(1) homoscedastic normal, $\varepsilon_i \sim N(0,1)$;
(2) heteroscedastic normal, $\varepsilon_i \sim N(0, (1 + 0.3X_{i1})^2)$;
(3) exponential, $\varepsilon_i \sim \mathrm{Exp}(1)$.

For each quantile level τ, we compute the corresponding true QR coefficient vector $\boldsymbol{\beta}(\tau)$ in the QR model (1) by shifting ε_i such that $\mathbb{P}(\varepsilon_i \le 0 | \boldsymbol{X}_i) = \tau$:

(1) homoscedastic normal, $\boldsymbol{\beta}(\tau) = \boldsymbol{\beta} + \Phi^{-1}(\tau)\boldsymbol{e}_1$;
(2) heteroscedastic normal, $\boldsymbol{\beta}(\tau) = \boldsymbol{\beta} + \Phi^{-1}(\tau)\boldsymbol{e}_1 + 0.3\Phi^{-1}(\tau)\boldsymbol{e}_2$;
(3) exponential, $\boldsymbol{\beta}(\tau) = \boldsymbol{\beta} + F_{\exp}^{-1}(\tau)\boldsymbol{e}_1$.

Here, Φ and $F_{\exp}$ are the cumulative distribution function of standard normal distribution and exponential distribution with parameter 1. The vector $\boldsymbol{e}_i$ (for $i = 1, \ldots, p+1$) is the $(p+1)$-dimensional canonical vector with the ith element being one and all the other elements being zero.

We use the integral of a biweight (or quartic) kernel as the smoothing function H:

$$H(v) = \begin{cases} 0, & \text{if } v \le -1, \\ \dfrac{1}{2} + \dfrac{15}{16}\left(v - \dfrac{2}{3}v^3 + \dfrac{1}{5}v^5\right), & \text{if } |v| < 1, \\ 1, & \text{if } v \ge 1. \end{cases} \tag{25}$$

It is easy to see that it satisfies Condition (C2).

5.1. *Coverage rates.* We compute the DC LEQR $\widehat{\boldsymbol{\beta}}^{(q)}$ in (10) and measure the performance of $\widehat{\boldsymbol{\beta}}^{(q)}$ in terms of the statistical inference. In particular, we report average coverage rates of the confidence interval of $\boldsymbol{v}_0'\boldsymbol{\beta}(\tau)$, where $\boldsymbol{v}_0 = (p+1)^{-1/2}\mathbf{1}_{p+1}$. We set the nominal coverage probability $1 - \alpha_0$ to 95%. From Theorem 4.4, an oracle $(1-\alpha_0)$th confidence interval for $\boldsymbol{v}_0'\boldsymbol{\beta}(\tau)$ is given by

$$\boldsymbol{v}_0'\widehat{\boldsymbol{\beta}}^{(q)} \pm n^{-1/2}\sqrt{\tau(1-\tau)\boldsymbol{v}_0'\boldsymbol{D}^{-1}\mathbb{E}[\boldsymbol{X}\boldsymbol{X}']\boldsymbol{D}^{-1}\boldsymbol{v}_0}\, z_{\alpha_0/2}, \tag{26}$$

where $z_{\alpha_0/2}$ is the $(1-\alpha_0/2)$-quantile of the standard normal distribution. To construct the confidence interval, we estimate $\boldsymbol{D}$ and $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}']$ by $\boldsymbol{D}_{n,h}$ and $\frac{1}{n}\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i'$, respectively. There are two major advantages of this approach. First, since $\boldsymbol{D}_{n,h}$ has already been obtained in computing DC LEQR, we estimate $\boldsymbol{D}$ without any extra computation. Second, both $\boldsymbol{D}_{n,h}$ and $\frac{1}{n}\sum_{i=1}^n \boldsymbol{X}_i\boldsymbol{X}_i'$ are in the form of summation over n terms, which can be easily computed in a distributed setting with little communication cost. As we will show in Table 5, the proposed estimator is very close to the truth. The scaling constant in bandwidth is simply set to one (as in our theorems) and more detailed experiments on the sensitivity analysis of the scaling constant is provided in Section 5.3. We report empirical coverage rates as an average of 1000 independent runs of the simulations.

In Tables 1–3, we present the empirical coverage rates of our DC LEQR estimator, the naive-DC estimator, and the oracle QR estimator in (2) computed on all data points (denoted as QR All) for three different noise models. More precisely, we generate the error from one of three distributions (i.e., homoscedastic normal for Table 1, heteroscedastic normal for Table 2, and exponential for Table 3) and

TABLE 1

Coverage rates, bias and variance of DC LEQR vs. QR All and naive-DC when n varies from $m^{1.6}$ to m^3. Noises ε_i's are generated from homoscedastic normal distribution. Dimension $p = 15$. Batch size $m = 100$. Quantile level $\tau \in \{0.1, 0.5, 0.9\}$

$\log_m(n)$	Coverage Rate	Bias ($\times 10^{-2}$)	Var ($\times 10^{-4}$)	Coverage Rate	Bias ($\times 10^{-2}$)	Var ($\times 10^{-4}$)
		DC LEQR $q = 4$			DC LEQR $q = 5$	
$\tau = 0.1$						
1.6	0.954	0.38	20.81	0.953	0.26	21.14
2.0	0.956	0.13	3.04	0.953	0.09	3.05
2.4	0.946	−0.02	0.52	0.949	−0.02	0.52
3.0	0.942	0.00	0.04	0.943	0.00	0.03
$\tau = 0.5$						
1.6	0.943	0.00	12.07	0.938	0.03	12.01
2.0	0.947	0.02	1.81	0.947	0.02	1.81
2.4	0.944	0.00	0.29	0.945	0.00	0.29
3.0	0.951	0.00	0.02	0.952	0.00	0.02
$\tau = 0.9$						
1.6	0.938	−0.45	21.37	0.940	−0.36	21.77
2.0	0.942	−0.05	3.65	0.932	−0.02	3.65
2.4	0.960	−0.02	0.51	0.959	−0.02	0.51
3.0	0.952	0.00	0.04	0.955	0.00	0.03
		QR All			Naïve-DC	
$\tau = 0.1$						
1.6	0.948	0.15	23.04	0.638	7.86	13.82
2.0	0.949	0.04	3.21	0.000	7.96	1.93
2.4	0.952	0.03	0.50	0.000	7.97	0.31
3.0	0.953	0.01	0.03	0.000	7.95	0.02
$\tau = 0.5$						
1.6	0.954	−0.20	11.16	0.978	0.46	8.71
2.0	0.951	−0.01	1.68	0.968	0.40	1.32
2.4	0.950	0.02	0.28	0.916	0.36	0.21
3.0	0.930	0.00	0.02	0.222	0.35	0.01
$\tau = 0.9$						
1.6	0.942	−0.16	23.12	0.945	3.35	14.36
2.0	0.946	−0.04	3.30	0.531	3.46	2.20
2.4	0.947	0.02	0.52	0.000	3.45	0.35
3.0	0.944	0.00	0.03	0.000	3.41	0.02

consider three different quantile levels $\tau = 0.1, 0.5, 0.9$. In our experiment, we set $m = 100$, $p = 15$ and vary n from $m^{1.6}$ to m^3 (i.e., $\log_m(n)$ from 1.6 to 3). From (17), it is easy to see that we need number of aggregations $q \geq 4$. Thus, we report the performance of DC LEQR $\widehat{\boldsymbol{\beta}}^{(q)}$ for $q = 4$ and $q = 5$. We also report the case of $p = 3$ in the supplementary material (see Section E.1).

TABLE 2

Coverage rates, bias and variance of DC LEQR vs. QR All and naive-DC when n varies from $m^{1.6}$ to m^3. Noises ε_i's are generated from heteroscedastic normal distribution. Dimension $p = 15$. Batch size $m = 100$. Quantile level $\tau \in \{0.1, 0.5, 0.9\}$

$\log_m(n)$	Coverage Rate	Bias ($\times 10^{-2}$)	Var ($\times 10^{-4}$)	Coverage Rate	Bias ($\times 10^{-2}$)	Var ($\times 10^{-4}$)
		DC LEQR $q = 4$			DC LEQR $q = 5$	
$\tau = 0.1$						
1.6	0.941	0.62	30.11	0.940	0.56	30.24
2.0	0.942	0.03	4.67	0.938	−0.04	4.79
2.4	0.947	0.01	0.72	0.947	0.00	0.70
3.0	0.952	0.00	0.06	0.950	0.00	0.05
$\tau = 0.5$						
1.6	0.941	−0.19	16.14	0.939	−0.18	16.22
2.0	0.946	0.02	2.48	0.944	0.02	2.48
2.4	0.940	0.01	0.40	0.940	0.01	0.40
3.0	0.947	0.00	0.03	0.947	0.00	0.02
$\tau = 0.9$						
1.6	0.925	−0.61	31.55	0.926	−0.48	31.46
2.0	0.943	−0.05	4.81	0.947	−0.01	4.73
2.4	0.955	−0.03	0.67	0.956	−0.02	0.67
3.0	0.953	−0.01	0.09	0.957	0.00	0.04
		QR All			Naïve-DC	
$\tau = 0.1$						
1.6	0.951	0.25	31.53	0.362	11.95	17.03
2.0	0.959	0.10	4.42	0.000	11.92	2.71
2.4	0.937	−0.01	0.79	0.000	11.88	0.44
3.0	0.954	0.01	0.04	0.000	11.91	0.02
$\tau = 0.5$						
1.6	0.950	−0.09	16.70	0.976	0.16	11.99
2.0	0.948	0.03	2.38	0.974	0.29	1.81
2.4	0.943	−0.03	0.40	0.944	0.33	0.28
3.0	0.960	0.00	0.02	0.320	0.36	0.02
$\tau = 0.9$						
1.6	0.942	−0.32	31.16	0.972	2.10	19.16
2.0	0.946	−0.13	4.68	0.882	2.00	2.94
2.4	0.954	0.00	0.72	0.303	1.99	0.45
3.0	0.946	0.00	0.05	0.000	2.01	0.03

As one can see from Tables 1–3, for most of the settings, the coverage rates of our DC LEQR are close to the nominal level of 95% after 4 rounds of aggregations ($q = 4$). The coverage performance becomes quite stable for $q = 5$ iterations. On the other hand, for the naive-DC estimator, the coverage rates are quite low in most settings, especially when n is larger than m^2.

TABLE 3

Coverage rates, bias and variance of DC LEQR vs. QR All and naive-DC when n varies from $m^{1.6}$ to m^3. Noises ε_i's are generated from exponential distribution. Dimension $p = 15$. Batch size $m = 100$. Quantile level $\tau \in \{0.1, 0.5, 0.9\}$

$\log_m(n)$	Coverage Rate	Bias ($\times 10^{-2}$)	Var ($\times 10^{-4}$)	Coverage Rate	Bias ($\times 10^{-2}$)	Var ($\times 10^{-4}$)
		DC LEQR $q = 4$			DC LEQR $q = 5$	
$\tau = 0.1$						
1.6	0.880	0.71	1.19	0.916	0.45	0.90
2.0	0.920	0.05	0.42	0.942	0.02	0.22
2.4	0.915	0.00	0.13	0.931	0.02	0.09
3.0	0.907	0.04	0.33	0.931	−0.04	0.58
$\tau = 0.5$						
1.6	0.933	−0.12	7.76	0.931	−0.09	7.75
2.0	0.937	−0.05	1.59	0.939	−0.04	1.12
2.4	0.928	0.01	0.51	0.933	0.00	0.19
3.0	0.934	0.00	0.39	0.941	−0.01	0.04
$\tau = 0.9$						
1.6	0.931	−0.43	61.03	0.933	−0.21	58.31
2.0	0.908	0.00	10.75	0.916	0.12	10.37
2.4	0.906	−0.05	1.95	0.915	0.03	1.52
3.0	0.885	−0.02	0.15	0.917	0.01	0.10
		QR All			Naïve-DC	
$\tau = 0.1$						
1.6	0.945	0.14	0.87	0.422	1.98	1.11
2.0	0.957	0.02	0.12	0.001	1.99	0.16
2.4	0.958	0.00	0.02	0.000	1.99	0.03
3.0	0.944	0.00	0.00	0.000	2.00	0.00
$\tau = 0.5$						
1.6	0.959	0.11	7.16	0.799	3.49	5.32
2.0	0.944	0.02	1.15	0.070	3.46	0.93
2.4	0.948	0.01	0.18	0.000	3.46	0.15
3.0	0.953	0.00	0.01	0.000	3.45	0.01
$\tau = 0.9$						
1.6	0.952	0.07	65.66	0.798	11.20	36.00
2.0	0.944	0.16	10.14	0.010	11.66	5.76
2.4	0.953	0.00	1.59	0.000	11.68	0.96
3.0	0.948	0.02	0.10	0.000	11.69	0.06

Note that for naive-DC and QR All, we use the same estimator of the limiting variance in (26) as in our DC LEQR when $q = 4$. More precisely, we use $\boldsymbol{D}_{n,h}$ computed in the 4th iteration to estimate $\boldsymbol{D}$ in (26) when constructing the confidence intervals of naive-DC and QR All estimators. We will show in Table 5 below that the proposed estimator of the limiting variance performs well. In fact, we also

use the true limiting variance to construct the confidence interval and the coverage rates for all the methods are almost the same.

In addition, we also report the simulation study with large dimension ($p = 1000$). The results and analysis are relegated to Section E.3 in the supplementary material. From the results, we can infer that the coverage rates get better as the iterative refinement proceeds. In particular, the coverage rates are close to the nominal level 95% after 4 iterations when the dimension $p = 1000$. In summary, when the dimension p is large, the proposed DC LEQR algorithm still achieves desirable performance with a small number of iterations.

5.2. *Bias and variance analysis.* To see the improvement of DC LEQR over naive-DC when n is excessively larger than the subset size m, we also report the mean bias and variance of our proposed DC LEQR, naive-DC and QR All in Tables 1–3. The mean bias and variance of $\boldsymbol{v}_0'\widehat{\boldsymbol{\beta}}$ are based on 1000 independent runs of simulations.

From Tables 1–3, the bias of our method is quite small while the naive-DC approach has a much larger bias regardless of the sample size n. For the variance, it decays with the rate $1/n$ as n goes large for all methods. For most cases of using naive-DC, as $\log_m(n)$ exceeding 2, the squared bias becomes comparable or larger than the variance, which explains the reason of the failure of naive-DC when n is large as compared to m. On the other hand, the bias of our proposed DC LEQR is similar to that of QR All and much smaller than that of niave-DC.

5.3. *Sensitivity analysis and data-adaptive choice of the bandwidth.* In this section, we show the empirical performance of the data-adaptive choice of bandwidth in Remark 4.2 and the sensitivity of the scaling constant in bandwidth. Due to space limitations, we report $\tau = 0.1$, homoscedastic normal noise case as an example. More noise cases (e.g., heteroscedastic normal and exponential cases) are relegated to Section E.2 in the supplementary material, and observations are similar to the homoscedastic normal case.

Table 4 shows coverage rates of the DC LEQR with $q = 1, 2, 3, 4, 5$ iterations. Similar to the setting in Tables 1–3, we choose $m = 100$, $p = 15$ and n varies from $m^{1.6}$ to m^3. We report the performance of DC LEQR using different fixed constants $c = 1, 3, 5, 10$ in bandwidth h_g for $1 \le g \le q$ (using the same constant in all iterations) as well as our data-adaptive choice of bandwidth. For the data-adaptive bandwidth, we choose the best scaling constant from a list of 1000 equally spaced constants from a very small number (0.1) to a large one (100) according to Remark 4.2. Note that different scaling constants will be chosen for different iterations. As one can see, for $q = 1$ and $q = 2$, the adaptive method indeed achieves better coverage than other scaling constants. On the other hand, when $q \ge 3$, all different choices of scaling constants lead to coverage rates close to the nominal level of 95%. This experiment suggests that for our proposed iterative aggregation

TABLE 4

Coverage rates for DC LEQR with iterations $q = 1, 2, 3, 4, 5$ for different choices of the scaling constant c. Noises ε_i's are generated from homoscedastic normal distribution. Dimension $p = 15$. Batch size $m = 100$. Sample size n varies from $m^{1.6}$ to m^3. Quantile level $\tau = 0.1$

	$\log_m(n)$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$
$c=1$						
	1.6	0.642	0.909	0.949	0.954	0.953
	2.0	0.311	0.911	0.956	0.956	0.953
	2.4	0.077	0.427	0.877	0.946	0.949
	3.0	0.000	0.231	0.611	0.942	0.943
$c=3$						
	1.6	0.711	0.907	0.951	0.949	0.944
	2.0	0.303	0.891	0.947	0.946	0.946
	2.4	0.111	0.521	0.889	0.952	0.949
	3.0	0.000	0.306	0.687	0.953	0.950
$c=5$						
	1.6	0.644	0.900	0.959	0.950	0.950
	2.0	0.196	0.849	0.942	0.952	0.951
	2.4	0.079	0.469	0.815	0.944	0.944
	3.0	0.000	0.120	0.588	0.947	0.949
$c=10$						
	1.6	0.597	0.814	0.955	0.949	0.947
	2.0	0.129	0.729	0.944	0.951	0.951
	2.4	0.000	0.402	0.777	0.952	0.949
	3.0	0.000	0.011	0.513	0.950	0.946
data-adaptive						
	1.6	0.724	0.929	0.946	0.949	0.948
	2.0	0.336	0.914	0.947	0.954	0.949
	2.4	0.187	0.564	0.912	0.941	0.944
	3.0	0.000	0.385	0.710	0.954	0.951

approach, even when a suboptimal scaling constant is used, one can still achieve good performance by performing more iterations.

Moreover, to investigate the sensitivity of the scaling constant in terms of variance estimation, we also present the square root of the ratio of the estimated variance of our approach versus the true limiting variance, that is,

$$\frac{\sqrt{\boldsymbol{v}_0' \boldsymbol{D}_{n,h}^{-1} \frac{1}{n} \sum_{i=1}^{n} [\boldsymbol{X}_i \boldsymbol{X}_i'] \boldsymbol{D}_{n,h}^{-1} \boldsymbol{v}_0}}{\sqrt{\boldsymbol{v}_0' \boldsymbol{D}^{-1} \mathbb{E}[\boldsymbol{X}\boldsymbol{X}'] \boldsymbol{D}^{-1} \boldsymbol{v}_0}}, \tag{27}$$

where $\boldsymbol{D}_{n,h}$ is computed for iterations $q = 1, 2, 3, 4, 5$ and for each fixed q, the bandwidth $h_g = c \max(\sqrt{p/n}, (p/m)^{2^{g-2}})$ for $1 \le g \le q$. In Table 5, we report the performance of the variance estimation with different choices of the scaling constant c of the bandwidth h in constructing $\boldsymbol{D}_{n,h}$.

TABLE 5

Square root of the ratio of the estimated variance and the true limiting variance using different choices of the scaling constant c in bandwidths. Noises ε_i's are generated from homoscedastic normal distribution. Dimension $p = 15$. Batch size $m = 100$. Sample size n varies from $m^{1.6}$ to m^3. Quantile level $\tau = 0.1$

	$\log_m(n)$	$q = 1$	$q = 2$	$q = 3$	$q = 4$	$q = 5$
$c = 1$						
	1.6	1.05	1.10	1.11	1.07	1.05
	2.0	1.14	1.09	0.99	1.04	1.03
	2.4	1.27	1.24	1.12	0.99	1.00
	3.0	1.12	1.07	1.03	1.00	1.01
$c = 3$						
	1.6	1.14	1.01	0.99	0.99	1.00
	2.0	1.06	1.04	1.02	1.00	1.01
	2.4	1.08	1.03	1.03	1.01	1.01
	3.0	1.04	1.00	0.99	0.99	0.99
$c = 5$						
	1.6	1.09	1.04	0.99	0.98	0.99
	2.0	1.11	1.12	1.07	1.03	1.04
	2.4	1.04	1.07	1.00	1.00	1.01
	3.0	1.07	1.02	1.01	1.01	1.01
$c = 10$						
	1.6	0.74	0.84	0.89	0.94	0.96
	2.0	0.86	0.86	0.92	0.96	0.99
	2.4	0.90	0.91	0.96	0.99	1.01
	3.0	0.88	0.90	0.98	1.00	1.00
data-adaptive						
	1.6	1.01	1.06	1.03	1.01	1.00
	2.0	1.03	1.03	1.01	0.98	0.97
	2.4	1.01	1.06	1.01	1.00	1.01
	3.0	1.01	1.04	1.02	1.01	1.01

From Table 5, the ratio is very close to 1 when $q = 4$ or 5. Therefore, when q is large, the proposed variance estimator is a reliable one in the distributed setting. Moreover, we notice that the ratio is very stable for different choices of the scaling constant c, which illustrates the robustness of the estimator.

5.4. *Computation efficiency.* We further conduct experiments to illustrate the computation efficiency of our algorithm for different m and n with $\tau = 0.1$, $p = 15$ and $\varepsilon_i \sim N(0, 1)$. We compare the computation time of DC LEQR versus that of naive-DC as well as QR All in Table 6. We report the bias and variance and the coverage rates for reference of the performance of the estimators.

First of all, the computation time of DC LEQR is about as twice faster than that of the QR All, especially when n is large. It is also faster than the naive-DC and

TABLE 6

Bias ($\times 10^{-2}$), *variance* ($\times 10^{-4}$), *coverage rates* (*nominal level* 95%) *and computation time* ($\times 100$ *seconds*) *of DC LEQR for different* q *versus the standard QR estimator on the entire data* (*QR All*), *and naive-DC. Noises* ε_i*'s are generated from homoscedastic normal distribution. Dimension* $p = 15$. *Quantile level* $\tau = 0.1$

	DC LEQR				QR	Naive
	$q=1$	$q=2$	$q=3$	$q=4$	All	DC
$m = 100, n = 10^6$						
Bias	6.112	0.865	0.038	−0.020	−0.020	7.947
Variance	35.464	1.637	0.037	0.035	0.031	0.24
Coverage	0.001	0.314	0.940	0.942	0.942	0.000
Time	0.409	0.821	1.233	1.643	8.015	2.421
$m = 500, n = 10^6$						
Bias	1.334	0.029	−0.008	−0.010	−0.009	0.214
Variance	0.642	0.042	0.029	0.029	0.029	0.029
Coverage	0.087	0.914	0.947	0.951	0.951	0.132
Time	0.499	0.993	1.488	1.982	7.909	2.549
$m = 500, n = 10^7$						
Bias	1.171	0.046	−0.005	−0.005	−0.005	0.237
Variance	0.885	0.016	0.002	0.002	0.002	0.002
Coverage	0.024	0.642	0.943	0.948	0.947	0.000
Time	4.555	9.106	13.648	18.188	143.521	25.289
$m = 1000, n = 10^7$						
Bias	0.786	0.016	−0.007	−0.007	−0.007	0.140
Variance	0.167	0.002	0.003	0.003	0.003	0.003
Coverage	0.014	0.897	0.952	0.947	0.947	0.013
Time	4.547	9.087	13.625	18.164	137.425	25.353

with a much better coverage when n is much larger than m. Moreover, the time of our algorithm grows almost linearly in both n and q, which is consistent with the computation time analysis in Section 3.2. In contrast, we observe that the time of QR grows faster than a linear function in the sample size n. We also observe that, for each fixed n, the value m has little effect on the computation time of DC LEQR. For naive-DC, the squared bias in these cases dominate the variance, so the coverage rates are far below the nominal level. In the meantime, DC LEQR has around 95% coverage in all four cases after 2 iterations, and shows a similar behavior of bias and variance as in Table 1.

Recently, researchers have developed new optimization techniques based on alternating direction method of multiplier (ADMM) for solving QR problems (see, e.g., Gu et al. (2018), Yu, Lin and Wang (2017)). We further conduct comparisons to the ADMM approach and the details are provided in the supplementary material due to space limitations.

Finally, we also conduct simulation studies for online LEQR and the results are presented in Section E.5 in the supplementary material.

6. Conclusions and future works. In this paper, we propose a novel inference approach for quantile regression under the memory constraint. The proposed method achieves the same asymptotic efficiency as the quantile regression estimator using all the data. Furthermore, it allows a weak condition on the sample size n as a function of memory size m and is computationally attractive. One key insight from this work is that naively splitting data and averaging local estimators could be suboptimal. Instead, the iterative refinement idea can lead to much improved performance for some inference problems in distributed environments.

In some applications, one would expect a weaker assumption on the distribution of data where the data could be correlated. It would be an interesting future direction to study the problem of inference for correlated data in distributed settings. Moreover, for the online problem, it is also interesting to consider the case where the model (e.g., $\boldsymbol{\beta}(\tau)$) is evolving over time. In this case, some exponential decaying techniques to down weight historical data might be useful.

In the future, we would also like to further explore this idea to other QR problems under memory constraints or in a distributed setup, for example, ℓ_1-penalized high-dimensional quantile regression (see, e.g., Belloni and Chernozhukov (2011), Fan, Xue and Zou (2016), Wang, Wu and Li (2012)) and censored quantile regression (see, e.g., Kong, Linton and Xia (2013), Leng and Tong (2014), Volgushev, Wagener and Dette (2014), Wang and Wang (2009), Zheng, Peng and He (2018)).

Acknowledgments. The authors are very grateful to three anonymous referees and the Associate Editor for their detailed and constructive comments that considerably improved the quality of this paper.

SUPPLEMENTARY MATERIAL

Supplement to "Quantile regression under memory constraint" (DOI: 10.1214/18-AOS1777SUPP; .pdf). We provide the proofs of all the theorectial results as well as additional simulated experimental results.

REFERENCES

BANERJEE, M., DUROT, C. and SEN, B. (2018). Divide and conquer in non-standard problems and the super-efficiency phenomenon. *Ann. Statist.* To appear.

BANG, H. and TSIATIS, A. A. (2002). Median regression with censored cost data. *Biometrics* **58** 643–649. MR1926117

BATTEY, H., FAN, J., LIU, H., LU, J. and ZHU, Z. (2018). Distributed estimation and inference with statistical guarantees. *Ann. Statist.* **46** 1352–1382.

BECK, A. (2014). *Introduction to Nonlinear Optimization. MOS-SIAM Series on Optimization* **19**. SIAM, Philadelphia, PA. MR3288060

BELLONI, A. and CHERNOZHUKOV, V. (2011). ℓ_1-penalized quantile regression in high-dimensional sparse models. *Ann. Statist.* **39** 82–130. MR2797841

BELLONI, A., CHERNOZHUKOV, V., CHETVERIKOV, D. and FERNANDEZ-VAL, I. (2011). Conditional Quantile Processes based on Series or Many Regressors. Technical report. Preprint. Available at arXiv:1105.6154v3.

CHEN, X., LIU, W. and ZHANG, Y. (2019). Supplement to "Quantile regression under memory constraint." DOI:10.1214/18-AOS1777SUPP.

CHEN, X. and XIE, M. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statist. Sinica* **24** 1655–1684. MR3308656

FALK, M. (1999). A simple approach to the generation of uniformly distributed random variables with prescribed correlations. *Comm. Statist. Simulation Comput.* **28** 785–791. MR1713632

FAN, J., XUE, L. and ZOU, H. (2016). Multitask quantile regression under the transnormal model. *J. Amer. Statist. Assoc.* **111** 1726–1735. MR3601731

GALVAO, A. F. and KATO, K. (2016). Smoothed quantile regression for panel data. *J. Econometrics* **193** 92–112. MR3500178

GAMA, J., SEBASTIÃO, R. and RODRIGUES, P. P. (2013). On evaluating stream learning algorithms. *Mach. Learn.* **90** 317–346. MR3021877

GREENWALD, M. B. and KHANNA, S. (2004). Power-conserving computation of order-statistics over sensor networks. In *Proceedings of the ACM Symposium on Principles of Database Systems*.

GU, Y., FAN, J., KONG, L., MA, S. and ZOU, H. (2018). ADMM for high-dimensional sparse penalized quantile regression. *Technometrics* **60** 319–331. MR3847169

GUHA, S. and MCGREGOR, A. (2008/09). Stream order and order statistics: Quantile estimation in random-order streams. *SIAM J. Comput.* **38** 2044–2059. MR2476286

HE, X. and SHAO, Q.-M. (2000). On parameters of increasing dimensions. *J. Multivariate Anal.* **73** 120–135. MR1766124

HESTENES, M. R. and STIEFEL, E. (1952). Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand., B Math. Sci.* **49** 409–436. MR0060307

HOROWITZ, J. L. (1998). Bootstrap methods for median regression models. *Econometrica* **66** 1327–1351. MR1654307

HUANG, Z., WANG, L., YI, K. and LIU, Y. (2011). Sampling based algorithms for quantile computation in sensor networks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

JOHNSON, R. and ZHANG, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the Advances in Neural Information Processing Systems*.

JORDAN, M. I., LEE, J. D. and YANG, Y. (2018). Communication-efficient distributed statistical inference. *J. Amer. Statist. Assoc.* To appear.

KLEINER, A., TALWALKAR, A., SARKAR, P. and JORDAN, M. I. (2014). A scalable bootstrap for massive data. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **76** 795–816. MR3248677

KOENKER, R. (2005). *Quantile Regression. Econometric Society Monographs* **38**. Cambridge Univ. Press, Cambridge. MR2268657

KOENKER, R., CHERNOZHUKOV, V., HE, X. and PENG, L., eds. (2018). *Handbook of Quantile Regression. Chapman & Hall/CRC Handbooks of Modern Statistical Methods*. CRC Press, Boca Raton, FL. MR3728340

KONG, E., LINTON, O. and XIA, Y. (2013). Global Bahadur representation for nonparametric censored regression quantiles and its applications. *Econometric Theory* **29** 941–968. MR3148821

LEE, J. D., LIU, Q., SUN, Y. and TAYLOR, J. E. (2017). Communication-efficient sparse regression. *J. Mach. Learn. Res.* **18** 1–30. MR3625709

LENG, C. and TONG, X. (2014). Censored quantile regression via Box–Cox transformation under conditional independence. *Statist. Sinica* **24** 221–249. MR3183682

LI, R., LIN, D. K. J. and LI, B. (2013). Statistical inference in massive data sets. *Appl. Stoch. Models Bus. Ind.* **29** 399–409. MR3117826

LUO, X., HUANG, C.-Y. and WANG, L. (2013). Quantile regression for recurrent gap time data. *Biometrics* **69** 375–385. MR3071056

MANKU, G. S., RAJAGOPALAN, S. and LINDSAY, B. G. (1998). Approximate medians and other quantiles in one pass and with limited memory. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

MUNRO, J. I. and PATERSON, M. S. (1980). Selection and sorting with limited storage. *Theoret. Comput. Sci.* **12** 315–323. MR0589312

PANG, L., LU, W. and WANG, H. J. (2012). Variance estimation in censored quantile regression via induced smoothing. *Comput. Statist. Data Anal.* **56** 785–796. MR2888725

PORTNOY, S. and KOENKER, R. (1997). The Gaussian hare and the Laplacian tortoise: Computability of squared-error versus absolute-error estimators. *Statist. Sci.* **12** 279–300. MR1619189

RAJAGOPAL, J., WAINWRIGHT, M. and VARAIYA, P. (2006). Universal quantile estimation with feedback in the communication-constrained setting. In *Proceedings of the IEEE International Symposium on Information Theory*.

SHAMIR, O., SREBRO, N. and ZHANG, T. (2014). Communication efficient distributed optimization using an approximate Newton-type method. In *Proceedings of the International Conference on Machine Learning*.

SHERWOOD, B., WANG, L. and ZHOU, X.-H. (2013). Weighted quantile regression for analyzing health care cost data with missing covariates. *Stat. Med.* **32** 4967–4979. MR3127188

SHI, C., LU, W. and SONG, R. (2017). A massive data framework for M-estimators with cubic-rate. *J. Amer. Statist. Assoc.* To appear.

SHRIVASTAVA, N., BURAGOHAIN, C., AGRAWAL, D. and SURI, S. (2004). Medians and beyond: New aggregation techniques for sensor networks. In *Proceedings of the International Conference on Embedded Networked Sensor Systems*.

SIEGMUND, D. (1969). On moments of the maximum of normed partial sums. *Ann. Math. Stat.* **40** 527–531. MR0239695

VOLGUSHEV, S., CHAO, S.-K. and CHENG, G. (2018). Distributed inference for quantile regression processes. *Ann. Statist.* To appear.

VOLGUSHEV, S., WAGENER, J. and DETTE, H. (2014). Censored quantile regression processes under dependence and penalization. *Electron. J. Stat.* **8** 2405–2447. MR3278338

WANG, X. and DUNSON, D. B. (2014). Parallelizing MCMC via Weierstrass sampler. Technical report. Preprint. Available at arXiv:1312.4605.

WANG, H. and LI, C. (2018). Distributed quantile regression over sensor networks. *IEEE Trans. Signal Inform. Process. Netw.* **4** 338–348. MR3808198

WANG, H. J., STEFANSKI, L. A. and ZHU, Z. (2012). Corrected-loss estimation for quantile regression with covariate measurement errors. *Biometrika* **99** 405–421. MR2931262

WANG, H. J. and WANG, L. (2009). Locally weighted censored quantile regression. *J. Amer. Statist. Assoc.* **104** 1117–1128. MR2562007

WANG, H. J. and WANG, L. (2014). Quantile regression analysis of length-biased survival data. *Stat* **3** 31–47.

WANG, L., WU, Y. and LI, R. (2012). Quantile regression for analyzing heterogeneity in ultra-high dimension. *J. Amer. Statist. Assoc.* **107** 214–222. MR2949353

WANG, L., LUO, G., YI, K. and CORMODE, G. (2013). Quantiles over data streams: An experimental study. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

WANG, X., GUO, F., HELLER, K. and DUNSON, D. (2015). Parallelizing MCMC with random partition trees. In *Proceedings of the Advances in Neural Information Processing Systems*.

WANG, J., KOLAR, M., SREBRO, N. and ZHANG, T. (2017). Efficient distributed learning with sparsity. In: *Proceedings of the International Conference on Machine Learning*.

WHANG, Y.-J. (2006). Smoothed empirical likelihood methods for quantile regression models. *Econometric Theory* **22** 173–205. MR2230386

WU, Y., MA, Y. and YIN, G. (2015). Smoothed and corrected score approach to censored quantile regression with measurement errors. *J. Amer. Statist. Assoc.* **110** 1670–1683. MR3449063

XU, G., SIT, T., WANG, L. and HUANG, C.-Y. (2017). Estimation and inference of quantile regression for survival data under biased sampling. *J. Amer. Statist. Assoc.* **112** 1571–1586. MR3750882

YU, L., LIN, N. and WANG, L. (2017). A parallel algorithm for large-scale nonconvex penalized quantile regression. *J. Comput. Graph. Statist.* **26** 935–939. MR3765357

ZHANG, Y., DUCHI, J. and WAINWRIGHT, M. (2015). Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *J. Mach. Learn. Res.* **16** 3299–3340. MR3450540

ZHANG, Q. and WANG, W. (2007). A fast algorithm for approximate quantiles in high speed data streams. In *Proceedings of the International Conference on Scientific and Statistical Database Management*.

ZHAO, T., CHENG, G. and LIU, H. (2016). A partially linear framework for massive heterogeneous data. *Ann. Statist.* **44** 1400–1437. MR3519928

ZHENG, S. (2011). Gradient descent algorithms for quantile regression with smooth approximation. *International Journal of Machine Learning and Cybernetics* **2** 191.

ZHENG, Q., PENG, L. and HE, X. (2018). High dimensional censored quantile regression. *Ann. Statist.* **46** 308–343. MR3766954

X. CHEN
Y. ZHANG
INFORMATION, OPERATIONS
AND MANAGEMENT SCIENCES
STERN SCHOOL OF BUSINESS
NEW YORK UNIVERSITY
NEW YORK, NEW YORK 10012
USA
E-MAIL: xchen3@stern.nyu.edu
yzhang@stern.nyu.edu

W. LIU
DEPARTMENT OF MATHEMATICS
INSTITUTE OF NATURAL SCIENCES
AND MOE-LSC
SHANGHAI JIAO TONG UNIVERSITY
SHANGHAI
CHINA
E-MAIL: weidongl@sjtu.edu.cn