Expectile Regression on Distributed Large-Scale Data

AIJUN HU1,2, CHUJIN LI3, AND JING WU3

1School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, China
2School of Mathematics and Economics, Hubei University of Education, Wuhan 430205, China
3Electronic Information School, Wuhan University, Wuhan 430072, China

Corresponding author: Chujin Li (lichujin@hust.edu.cn)

This work was supported in part by the National Key Research and Development Project under Grant 2017YFC0503801.

ABSTRACT Large-scale data presents great challenges to data analysis due to the limited computer storage capacity and the heterogeneous data structure. In this article, we propose a distributed expectile regression model to resolve the challenges of large-scale data by designing a surrogate loss function and using the Iterative Local Alternating Direction Method of the Multipliers (IL-ADMM) algorithm, which is developed for the calculation of the proposed estimator. To obtain nice performance only after fewer rounds of communications, the proposed method only needs to solve an M-estimation problem on the master machine while the other working machines only to compute the gradients based on local data. Moreover, we show the consistency and the asymptotic normality of the proposed estimator, and illustrate the efficient proof by numerical simulations and positive analysis on the superconductor data.

INDEX TERMS Expectile regression, surrogate loss, distributed statistical learning, IL-ADMM algorithm.

I. INTRODUCTION

In recent years, machine learning techniques based on large-scale data have been widely used in inspection systems, correct classification and Internet of Things, as well as in social networks and smart cities [1]–[3]. As for the large-scale data, the problem is not only the size of the data, but in fact the data is usually stored in multiple locations. This means that the statistical analysis involving the entire data must deal data communication between different storages, which may slow down the calculation. Hence, the impact of statistical procedures related to calculations and data storage has been exciting exuberant research interest.

Several illuminating methods have been proposed to study the statistical inference of such large-scale data, such as Mcdonald et al. [4], Boyd et al. [5], Duchi et al. [6], Dekel et al. [7], Zhang et al. [8], Lee et al. [9]. Among these methods, the divided-and-conquer (DC) is widely used. The main idea of DC is to first compute local estimates on each machine and then take average, where only one round of communication between machines is required and it could reduce the communication cost significantly. However, as discussed in Jordan et al. [10], these averaged-based one-shot methods may cause efficiency loss. For example, the number of machines for dividing the data must not be too large for achieving the minimax convergence rate [11].

Usually, the above-mentioned studies on DC are mainly focused on least squares linear regression, and the ordinary least squares linear regressions give the same weight to negative and positive residuals and estimate the expected values of a response variable as a liner function of covariates. Although, quantile regressions [12]–[14] based on asymmetric $l_1$-norm present some improvements by giving different weights to the positive and negative parts of the residual. As an alternative to quantile regression, Newey and Powell [15] criticized the use of regression quantiles in liner models, and proposed the following asymmetric quadratic loss function based on $l_2$-norm

$$\phi_\tau(\lambda) = \begin{cases} \tau \lambda^2, & \lambda \geq 0, \\ (1-\tau)\lambda^2, & \lambda < 0, \end{cases}$$

with the $\tau$-expectile of random variable $\xi$ as

$$\mu_\tau = \arg \min_{\mu \in \mathbb{R}} E [\phi_\tau(\xi - \mu)].$$

Since the classical article of Newey and Powell [15], there are numerous and extensive studies on expectile and related statistical inference [16]–[22]. Moreover, there has been developed several popular applied expectile regression packages, such as expectreg [23] and ER-Boost [21]. But the two packages can only be applied to the calculation of medium-sized sample data.
The first advantage of the asymmetric least squares method over quantiles lies in the convenience of calculation of sample expectiles, which only uses grading or iteratively re-weighted least squares method. The second advantage is that the statistical inferences of expectiles are more efficient than the cases of quantiles by considering the distance and location information on the data sufficiently, meanwhile the empirical quantiles utilize only the information on whether the observations are below or above the predictor [24]. In addition, unlike sample quantiles, sample expectiles provide a class of smooth curves as functions of the level \( \tau \), and demonstrate much more robustness for heavy-tailed distribution [25]–[26].

As for the distributed large scale data, we confront many interesting challenges. Jordan et al. [10] proposed the communication-efficient surrogate likelihood (CSL) procedure to solve distributed statistical learning problems. Especially, CSL only needs to exchange the gradients of local data, so can effectively reduce the communication cost of large data. Moreover, CSL can make statistical inferences in the same way as the global likelihood function by using all samples, and the estimation based on CSL can achieve the same convergence rate as oracle method based on the whole data. Note the theoretical results in Jordan et al. [10] need a very important assumption on the Hessian matrix of the population risk function, which should be invertible. In fact, this condition may be not satisfied usually, which could bring many difficulties to the theoretical proof and limit the application of the CSL. Here, we improve the assumption on Hessian matrix by Jordan et al. [10], then establish the asymptotic properties of the proposed estimator by M-estimation and the asymptotics for minimisers of convex processes, which bring much more remarkable theoretical and practical worth.

Another challenge arises from the numerical implementation of the proposed estimator. To this end, we adopt the Alternating Direction Method of Multipliers (ADMM) algorithm [5], [27]–[28], which is competent to distributed convex optimization problems and large-scale statistical inference problems. Combining the idea of CSL with the ADMM algorithm, we develop an Iterative Local Alternating Direction Method of Multipliers (IL-ADMM) algorithm for the distributed estimator in the linear expectile regression model. Compared to these existing approaches, our proposed method can not only solve the expectile regression with distributed data immediately, but also reduce the storage and transmission cost in distributed computing effectively. Most importantly, its efficiency priority can be compared with the oracle method.

The layout of this article is as follows. In section 2, we develop to fit the linear expectile model for distributed large-scale data, and present the IL-ADMM algorithm for the realization of the proposed estimator in Section 3. We further study the consistency and the asymptotic normality of the estimator in Section 4. Then, we carry out numerical simulations to evaluate the finite-sample performance of the proposed estimator in Section 5, and illustrate a real data example in Section 6. Finally, we remind of some concluding remarks, and offer some proofs in the Appendix.

II. EXPECTILE REGRESSION WITH DISTRIBUTED DATA

Let \( y \) be a response variable, let \( x \) be an observable \( p \times 1 \) vector of covariates. Here, the sample size \( N \) of the observable data \( \{x_i, y_i\}_{i=1}^N \) is so large that the data cannot fit in the memory of a single machine. We assume that the data are stored in one distributed way across multiple machines, such that each machine stores a sub-sample of \( n \) observations. Let \( \{x_i, y_i\}_{i=1}^N = \{(x_{ki}, y_{ki}) : i \in [n], k \in [K]\} \), where the notation \( [Z] \) denotes the set \( \{1, \ldots, Z\} \) for any non-negative integer \( Z \). The data set \( \{(x_{ki}, y_{ki}) : i \in [n]\} \) denotes the sub-sample stored in the \( k \)th machine \( M_k \) for \( k \in [K]\).

Our main objective is to estimate the \( p \)-dimensional vector expectile coefficient \( \beta_0(\tau) \) for certain \( \tau \in (0, 1) \) in the following linear expectile regression model

\[
y_i = x_i^T \beta_0(\tau) + \epsilon_i(\tau), \quad i \in [N],
\]

where \( \epsilon_i(\tau) \) is the random error with the \( \tau \)th expectile conditional on \( x_i \) equaling 0. We would drop \( \tau \) in the parameter and error term for notation simplicity in the following without ambiguity.

Define the local and global loss functions as follows:

\[
L_k(\beta) = \frac{1}{n} \sum_{i=1}^{n} \phi_\tau(y_{ki} - x_{ki}^T \beta), \quad k \in [K],
\]

\[
L_N(\beta) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n} \phi_\tau(y_{ki} - x_{ki}^T \beta),
\]

where \( \phi_\tau(\cdot) \) is defined by (1). Note that \( L_k(\beta) \) is the loss function evaluated at \( \beta \) by using the local data stored in the \( k \)th machine \( M_k \), and \( L_N(\beta) = K^{-1}\sum_{k=1}^{K} L_k(\beta) \). Without loss of generality, we call the first machine that stores local data set \( \{x_{1i}, y_{1i}\}_{i \in [n]} \) as the master machine, and call the remaining machines as worker machines.

Expectile regression is to solve the following optimization problem:

\[
\tilde{\beta} = \arg \min_{\beta \in \mathcal{B}} L_N(\beta),
\]

where \( \mathcal{B} \) denotes a compact subset of \( \mathbb{R}^p \) that contains \( \beta_0 \). Here the global empirical loss minimizer \( \tilde{\beta} \) is the best estimator of \( \beta \), and Newey and Powell [15] had established the asymptotic behavior of this estimator.

The above global model could achieve the optimal statistical error, however the expensive communications make it impractical. Inspired by the ideas on CSL methodology, we now consider a surrogate loss function that approximates the global loss function as follows:

\[
L(\beta) = L_1(\beta) + \left( \nabla L_N(\beta^0) - \nabla L_1(\beta^0), \beta \right),
\]

where \( \beta^0 \) is any initial estimator of \( \beta \), \( \langle \cdot, \cdot \rangle \) denotes the inner product and \( \nabla \) denotes the partial derivative with respect to \( \beta \). For example, \( \beta^0 = \arg \min_{\beta} L_1(\beta) \). Note that the
construction of the surrogate loss function only uses the local data on the main machine, and the gradient vectors calculated from the data on other working machines, so that the method will greatly reduce the cost of data transmission and communication. The master machine calculates an initial estimator \( \hat{\beta} \) firstly, and then \( \hat{\beta} \) is broadcasted to all other machines, with each worker machine computing the gradient \( \nabla L_i(\beta) \) and passing it to the master machine. Hence, it is called one round of communication.

In the distributed learning setting, the proposed estimator with the surrogate loss as the objective optimization function is
\[
\hat{\beta} = \arg \min_{\beta \in \mathcal{B}} L(\beta). \quad (6)
\]

### III. IL-ADMM ALGORITHM

We now present an IL-ADMM algorithm to solve the proposed estimator (6). Here, the key principle of the IL-ADMM algorithm is to construct a surrogate loss function as in (5), and then transfer the optimization problems from the original objective function (3) to the surrogate function.

To construct the surrogate loss function, by replacing \( \beta^0 \) with the current \( m \)-th iteration value \( \beta^{(m)} \), instead of (5), we rewrite the surrogate loss function as
\[
L_i^n(\beta) = L_i(\beta) + \left\langle \nabla L_i(\beta^{(m)}), \beta - \beta^{(m)} \right\rangle, \quad (7)
\]
and design the optimization problem as
\[
\beta^{(m+1)} = \arg \min_{\beta \in \mathcal{B}} L_i^n(\beta). \quad (8)
\]
It is noted that the Hessian matrix \( \nabla^2 L_i(\beta) \) is not necessarily invertible at \( \beta^{(m)} \), therefore the solution of (8) cannot be obtained by Newton-Ralphson algorithm as in [10].

To crack this problem, we consider the following ADMM algorithm as in [2]. Denote \( F_i(z) = n^{-1} \sum_{i=1}^{n} \phi_i(z_i), \quad z = (z_1, z_2, \ldots, z_n)^T \equiv y - \mu x, \quad \text{where} \quad y = (y_1, \ldots, y_n)^T, \quad x = (x_1, \ldots, x_m)^T. \) Without loss of generality, we let \( \{x, y\} \) to be the sub-sample stored in the first machine.

By convexity, we can transfer optimization problem (8) to
\[
\begin{cases}
\min_{x, \beta} F_i(z) + \left\langle \nabla L_i^n(\beta^{(m)}), \beta - \beta^{(m)} \right\rangle, \\
\text{s.t.} \quad x + z = y.
\end{cases} \quad (9)
\]

Define the augmented Lagrangian function of (9) as follows:
\[
H(\beta, z, y) = F_i(z) + \left\langle \nabla L_i^n(\beta^{(m)}), \beta - \beta^{(m)} \right\rangle - \left\langle y, x + z - y \right\rangle + \frac{\sigma}{2} \| x + z - y \|_2^2, \quad (10)
\]
where \( \sigma > 0 \) is a fixed constant and \( y \in \mathbb{R}^n \) is the Lagrangian multiplier.

The iterations for the local ADMM algorithm are given as
\[
\begin{align*}
\beta^{(m+1)} &= \arg \min_{\beta} H(\beta, z^{(m)}, y^{(m)}), \\
z^{(m+1)} &= \arg \min_{z} H(\beta^{(m+1)}, z, y^{(m)}), \\
y^{(m+1)} &= y^{(m)} - \sigma \left( x^{(m+1)} + z^{(m+1)} - y \right),
\end{align*} \quad (11)
\]
where \( (\beta^{(m)}, z^{(m)}, y^{(m)}) \) denotes the \( m \)-th iteration of the algorithm for \( m \geq 0 \).

Specifically,
\[
\begin{align*}
\beta \text{ Step: } \beta^{(m+1)} &= \arg \min_{\beta} \left\{ \nabla L_i^n(\beta^{(m)}) \right\} - \left\langle y^{(m)}, x + z - y \right\rangle + \frac{\sigma}{2} \| x + z - y \|_2^2, \\
z \text{ Step: } z^{(m+1)} &= \arg \min_{z} F_i(z) - \left\{ \nabla L_i^n(\beta^{(m)}) \right\} + \frac{\sigma}{2} \| x^{(m+1)} + z - y \|_2^2, \\
y \text{ Step: } y^{(m+1)} &= y^{(m)} - \sigma \left( x^{(m+1)} + z^{(m+1)} - y \right).
\end{align*}
\]

Remark 1: Noting that \( \phi_i(\lambda) \) is defined by (1), it follows that
\[
\phi_i(\lambda) + \frac{\alpha}{2}(\lambda - \xi)^2 = \begin{cases}
(\tau + \frac{\alpha}{2})\lambda^2 - \alpha \xi \lambda + \frac{\alpha}{2} \xi^2, & \lambda \geq 0, \\
(1 - \tau + \frac{\alpha}{2})\lambda^2 - \alpha \xi \lambda + \frac{\alpha}{2} \xi^2, & \lambda < 0.
\end{cases}
\]

Therefore, for fixed \( \alpha > 0 \), we have
\[
\begin{align*}
\phi_i^*(\lambda) + \frac{\alpha}{2}(\lambda - \xi)^2 &= \frac{\alpha \xi}{2(1 - \tau + \alpha)}, \quad \xi \geq 0, \\
\phi_i(\lambda) + \frac{\alpha}{2}(\lambda - \xi)^2 &= \frac{\alpha \xi}{2(1 - \tau + \alpha)}, \quad \xi < 0.
\end{align*}
\]

Now, we summarize the above iteration procedure on the master machine by Algorithm 1 in the following.
Algorithm 1 IL-ADMM: The Iterative Local ADMM Algorithm for Solving Distributed Linear Expectile Regression

Initialize (β(0), z(0), y(0)) = (0, 0, 0);

for t = 0, 1, 2, . . . , T − 1 do

1. Transmit the current iterate β(t) to machines {M_k}^K_{k=1};
2. Compute the local gradient ∇L_k(β(t)) at machine M_k, k = 1, 2, . . . , K;
3. Transmit the local gradient ∇L_k(β(t)) to the master machine M_1;
4. Calculate the global gradient ∇L_N(β(t)) = K^−1 ∑_k=1^K ∇L_k(β(t)) in the master machine M_1;
5. Calculate the surrogate loss function (7);
6. Do the following iterate in the master machine M_1:
   for m = 0, 1, . . . , M − 1 do
     Update β(m+1) via (12);
     Update z(m+1) via (14);
     Update y(m+1) via (11);
   end for
   Update β(t+1) = β(M), z(t+1) = z(M), y(t+1) = y(M).
end for

Return β(T).

IV. ASYMPTOTIC ANALYSIS

To present the asymptotic results, we introduce some notation first.

For i ∈ [n], define
ε_i = y_i − x_i^T β, ε_i = y_i − x_i^T β_0;
φε(λ) = 2λ I(λ ≥ 0) + 2(1 − t) λ I(λ < 0);
g(τ) = E[τ(ε_i | ε_i ≥ 0)] + (1 − τ) E[ε_i^2 | ε_i < 0];
c(τ) = τ^2 E[ε_i^2 | ε_i ≥ 0] + (1 − τ)^2 E[ε_i^2 | ε_i < 0].

Note that ∥·∥_2 refers to the l_2-norm in the Euclidean space, → represents convergence in distribution, and P → represents convergence in probability.

Moreover, we present some important conditions:
(C1) The parameter space B is compact.
(C2) The zero of a criterion function β → E[∇ϕ(y_i − x_i^T β)] is unique.
(C3) E[∥x_k∥_2^2] < ∞, for k ∈ [K].
(C4) The regression errors {ε_i_k}^n_{i,k=1} are independent and identically distributed, and satisfy μ_t(ε_i_k | x_k) = 0 with E[ε_i_k | x_k] < ∞.
(C5) There exists a positive definite matrix Σ such that lim_n→∞ n^−1 ∑_{i,k=1}^n x_i k^T = Σ, for k ∈ [K].

Condition (C1) is not very strong, but it is often used in the extreme value estimates; Condition (C2) guarantees the identification of the problem; Conditions (C3)–(C5) are common conditions for the expectile regression as in Newey and Powell [15]. And, the condition (C5) is one technical requirement being used in Xu et al. [29].

The asymptotic properties of the proposed estimator β̂ are summarized in the following theorems, with the proofs given in the Appendix.

Theorem 1 (Consistency of β̂): Under conditions (C1)–(C5), we have β̂ P → β_0.

Theorem 2 (Asymptotic Normality of β̂): Under conditions (C4) and (C5), we have

N^{1/2}(β̂ − β_0) → N(0, α(K, τ) σ^2_{N} τ^{-1}),

where α(K, τ) = 4Kc(τ) + (K − 1)A + (2 − 2K)B, A = Var[ϕε(ε_i)] and B = Cov[ϕε(ε_i), ϕε(ε_i)].

Remark 2: If we choose the initial value β_0 satisfying β̂_0 = β_0 + O_p(n^{−1/2}), i.e. β̂_0 = arg min β̂ E(L_1(β̂)), then it is the empirical loss minimizer in local machine M_1. From the proof of Theorem 2, we have

N^{1/2}(β̂ − β_0) → N(0, c(τ) σ^2_{N} τ^{-1}).

Therefore, the limiting distribution of β̂ is same as that of the global expectile regression estimator β̂ in (4) obtained by analyzing all data sets together [15].

V. NUMERICAL ANALYSIS

In this section, we examine the finite sample performance of our proposed method, and illustrate comparisons with existing methods via numerical studies. The methods to be compared are:

• Proposed: the proposed IL-ADMM algorithm.
• Oracle: the master machine collects all data from different machines together, and implement the ADMM algorithm with 0 as initial estimator to solve the global loss minimization problem.
• OneShot: each machine conduct the ADMM algorithm with 0 as initial value to compute a local estimator with the portion of the data stored locally, and the master machine averages the local estimators to produce an aggregate estimator β̂ OneShot = K^−1 ∑_k=1^K β̂_k.

Recall the linear expectile regression model (2), where x_i is generated from multivariate normal distribution with mean zero vector, and covariance matrix with (i, j)-element Σ_{ij} = 0.5|t−|t|. We assume different distributions for the random error ε_i: (I) homogeneous error ε_i ~ N(0, 1), ε_i ~ x_i^2(2). (II) heterogeneous error ε_i ~ (1 + x_i^2)Z_i, where Z_i ~ N(0, 1), Z_i ~ τ (3) and Z_i ~ x_i^2(2), and x_i^2 is the ith observation value of the second component of x. The scalar response y_i is generated from (2).

And, set the sample size N = 10000 and covariate dimension p = 80, and randomly partition the data as K = 20, 40 and 80 machines respectively. Therefore, the local sample size n on each machine is 500, 250 and 125 respectively.

Choosing τ = 0.3, 0.5 and 0.7, we compare the performance of the three methods mentioned above with different expectile levels of τ.

In each configuration, the results presented below are obtained from 100 independently generated data sets. Table 1 and Table 2 summarize the average (AER) and the standard deviation (SD) of estimation error defined as ∥β̂ − β_0∥_2^2.
### TABLE 1. Simulation results on estimation of $\beta$ under homoskedasticity errors.

| $\tau$ | Method   | $K = 20$ AER | $K = 20$ SD | $K = 40$ AER | $K = 40$ SD | $K = 80$ AER | $K = 80$ SD | $\varepsilon_{ki} \sim N(0, 1)$ | $\varepsilon_{ki} \sim t(3)$ | $\varepsilon_{ki} \sim \chi^2(2)$ |
|-------|----------|--------------|------------|--------------|------------|--------------|------------|---------------------------------|---------------------------------|---------------------------------|
| 0.3   | Oracle   | 0.0147       | 0.0027     | 0.0147       | 0.0027     | 0.0147       | 0.0027     | 0.0495                          | 0.0107                          | 0.0495                          |
|       | OneShot  | 0.0174       | 0.0031     | 0.0212       | 0.0037     | 0.0395       | 0.0072     | 0.0536                          | 0.0116                          | 0.0625                          |
|       | Proposed | 0.0147       | 0.0027     | 0.0148       | 0.0027     | 0.0164       | 0.0032     | 0.0495                          | 0.0106                          | 0.0496                          |
| 0.5   | Oracle   | 0.0138       | 0.0027     | 0.0138       | 0.0027     | 0.0138       | 0.0027     | 0.0393                          | 0.0085                          | 0.0393                          |
|       | OneShot  | 0.0163       | 0.0030     | 0.0202       | 0.0036     | 0.0376       | 0.0068     | 0.0464                          | 0.0106                          | 0.0568                          |
|       | Proposed | 0.0138       | 0.0027     | 0.0138       | 0.0027     | 0.0148       | 0.0029     | 0.0393                          | 0.0086                          | 0.0393                          |
| 0.7   | Oracle   | 0.0147       | 0.0028     | 0.0147       | 0.0028     | 0.0147       | 0.0028     | 0.0500                          | 0.0131                          | 0.0500                          |
|       | OneShot  | 0.0171       | 0.0032     | 0.0209       | 0.0036     | 0.0392       | 0.0068     | 0.0541                          | 0.0135                          | 0.0628                          |
|       | Proposed | 0.0147       | 0.0028     | 0.0148       | 0.0028     | 0.0163       | 0.0032     | 0.0499                          | 0.0131                          | 0.0500                          |

### TABLE 2. Simulation results on estimation of $\beta$ under heteroscedastic errors.

| $\tau$ | Method   | $K = 20$ AER | $K = 20$ SD | $K = 40$ AER | $K = 40$ SD | $K = 80$ AER | $K = 80$ SD | $\varepsilon_{ki} \sim N(0, 1)$ | $\varepsilon_{ki} \sim t(3)$ | $\varepsilon_{ki} \sim \chi^2(2)$ |
|-------|----------|--------------|------------|--------------|------------|--------------|------------|---------------------------------|---------------------------------|---------------------------------|
| 0.3   | Oracle   | 0.0455       | 0.0084     | 0.0455       | 0.0084     | 0.0455       | 0.0084     | 0.1290                          | 0.0258                          | 0.1290                          |
|       | OneShot  | 0.0673       | 0.0110     | 0.0927       | 0.0142     | 0.1679       | 0.0259     | 0.1725                          | 0.0313                          | 0.2287                          |
|       | Proposed | 0.0457       | 0.0085     | 0.0460       | 0.0086     | 0.0503       | 0.0100     | 0.1294                          | 0.0259                          | 0.1306                          |
| 0.5   | Oracle   | 0.0279       | 0.0051     | 0.0279       | 0.0051     | 0.0279       | 0.0051     | 0.0785                          | 0.0183                          | 0.0785                          |
|       | OneShot  | 0.0329       | 0.0057     | 0.0396       | 0.0068     | 0.0762       | 0.0141     | 0.0921                          | 0.0215                          | 0.1125                          |
|       | Proposed | 0.0279       | 0.0051     | 0.0280       | 0.0051     | 0.0297       | 0.0058     | 0.0785                          | 0.0183                          | 0.0785                          |
| 0.7   | Oracle   | 0.0448       | 0.0086     | 0.0448       | 0.0086     | 0.0448       | 0.0086     | 0.1290                          | 0.0258                          | 0.1290                          |
|       | OneShot  | 0.0652       | 0.0107     | 0.0895       | 0.0137     | 0.1599       | 0.0220     | 0.1725                          | 0.0313                          | 0.2287                          |
|       | Proposed | 0.0450       | 0.0087     | 0.0453       | 0.0087     | 0.0494       | 0.0104     | 0.1294                          | 0.0259                          | 0.1297                          |
Figure 1. Boxplots for estimation error $\|\hat{\beta} - \beta_0\|^2$ versus number of machines under three algorithms and homogeneous errors. 1st row: $e_{ki} \sim N(0, 1)$, 2nd row: $e_{ki} \sim t(3)$, 3rd row: $e_{ki} \sim \chi^2(2)$.

Figure 2. Boxplots for estimation error $\|\hat{\beta} - \beta_0\|^2$ versus number of machines under three algorithms and heterogeneous errors. 1st row: $e_{ki} \sim (1 + x_i^2)N(0, 1)$, 2nd row: $e_{ki} \sim (1 + x_i^2)t(3)$, 3rd row: $e_{ki} \sim (1 + x_i^2)\chi^2(2)$.

As for the boxplots comparing the estimation error over the number of machines, Figure 1 and Figure 2 show boxplots of the estimated errors of the above three methods for different machine numbers under homogeneous and heterogeneous errors, respectively. Figure 3 and Figure 4 illustrate the estimation for the proposed IL-ADMM algorithm with different...
rounds of communications and the estimation error based on the Oracle and OneShot method.

Summing up all the cases considered here, we have the following observations:

- The Oracle approach gives us the nice estimation and statistical performance. Our proposed approach obtains an estimation that could compete with the Oracle approach. However, the OneShot approach is relatively inferior. Moreover, we can see from Figures 1 and 2 that the estimation error of the OneShot method is significantly increased as dealing with heterogeneous data versus the homogenous data. However, the estimation error of the proposed method varies small and is close to that of the Oracle method.

- Since the total sample size \( N \) is fixed, the local sample size \( n \) decreases when the number of machines \( K \) varies from 20, 40 and 80. But, the AER and the SD with the Oracle method remain fixed. The OneShot and the proposed estimators present an ascent in the AER and SD as the number of machines increasing. Meticulously, our proposed method can still compete with the Oracle method.

- The rounds of communications are fixed for the Oracle and OneShot method, with the estimation error seeming as an horizontal line. Even though the number of machines grows, by fewer rounds of communications the estimation error of our proposed method decreases to be truly competitive with respect to the Oracle method.

- Note that in the OneShot method, we need to solve \( K = N/n \) optimization problems, which could be computationally expensive. However, our method only needs to calculate the gradient of the local data and an optimization problem on the master in each round of communication. As in Jordan et al. [10] and Zhang et al. [8], the communication complexity of our method is \( O(pK) \), and the computation complexity of the OneShot method is \( O(n\log(nK)) \).

### VI. REAL DATA ANALYSIS

To illustrate the effectiveness of our proposed approach, we also practice a large-scale sample of real dataset to compare the performance of the three methods mentioned above. The dataset comes from the Superconducting Material Database maintained by Japan’s National Institute for Materials Science (NIMS) and is also available from the UCI Machine Learning Repository. The goal here is to predict the critical temperature based on the features extracted from the superconductor’s chemical formula. Readers may refer to Hamidieh [30] for a detailed description of this dataset. Li et al. [31] proposed a distributed screening framework for this dataset.

The dataset contains 81 covariates extracted from 21263 superconductors along with the associated critical temperature. Noting that the covariates are linked to the critical temperature with a non-linear relationship, we take log-critical temperature, denoted by \( Y \) (the response). Since the non-Gaussian empirical distribution of \( Y \), the expectile regression approach may be more attractive to analyze this data.

To evaluate the performance of the Proposed relative to the Oracle and the OneShot, we use cross-validation for three methods with three levels as 0.3, 0.5 and 0.7. By randomly partitioning the sample 100 times, randomly selecting 16000 samples each time as the training set \( D_{\text{train}} \) and keeping 5263 samples as the test set \( D_{\text{test}} \). We store training set data with machine number \( K \) of 20, 40 and 80 respectively, and then set the local sample size \( n \) to be 800, 400 and 200. Inspired by Wang et al. [14], the prediction error is defined as \((1/5263)\sum_{i\in D_{\text{test}}} \Phi_\tau(y_i - \hat{y}_i)\). We use the above three methods to calculate the average (APE) and the standard deviation (SD) of the prediction errors.

| \( \tau \) | Method | \( K = 20 \) | \( K = 40 \) | \( K = 50 \) |
|---|---|---|---|---|
| 0.3 | Oracle | 0.3657 ± 0.0192 | 0.3657 ± 0.0192 | 0.3657 ± 0.0192 |
| | OneShot | 0.3672 ± 0.0197 | 0.3705 ± 0.0202 | 0.3841 ± 0.0211 |
| | Proposed | 0.3657 ± 0.0192 | 0.3657 ± 0.0192 | 0.3661 ± 0.0193 |
| 0.5 | Oracle | 0.3540 ± 0.0159 | 0.3540 ± 0.0159 | 0.3540 ± 0.0159 |
| | OneShot | 0.3555 ± 0.0160 | 0.3582 ± 0.0162 | 0.3665 ± 0.0163 |
| | Proposed | 0.3540 ± 0.0159 | 0.3540 ± 0.0159 | 0.3545 ± 0.0159 |
| 0.7 | Oracle | 0.2758 ± 0.0109 | 0.2758 ± 0.0109 | 0.2758 ± 0.0109 |
| | OneShot | 0.2780 ± 0.0109 | 0.2821 ± 0.0112 | 0.2931 ± 0.0116 |
| | Proposed | 0.2758 ± 0.0109 | 0.2758 ± 0.0109 | 0.2767 ± 0.0125 |

![FIGURE 5. Boxplots for prediction error versus numbers of machines on the superconductor data under three methods.](https://archive.ics.uci.edu/ml/datasets/Superconductivity+Data)
of communications. All these show that our method is more efficient and robust when dealing with large-scale real data.

VII. DISCUSSION

For large-scale dataset, certain partition of data across multiple machines is the only practical way to overcome the limitation of storage and computer memory. In this article, we extend the CSL method by Jordan et al. [10] to the distributed expectile regression model. Since the target loss function does not satisfy the smooth assumption as in the common CSL, both the theoretical results and computation method cannot be applied directly. We propose an efficient approach by using a surrogate loss function to fit the expectile regression model for distributed large-scale data, and develop an IL-ADMM algorithm for the calculation of the proposed estimator.

Estimation error can be improved by fewer rounds of communications. Asymptotic properties of our proposed estimator are established. The theoretical results show that if we choose the appropriate initial values of the regression parameters, our proposed method can match the ideal Oracle method. Extensive simulation studies and real data example demonstrate that our proposed approach performs better than the OneShot averaging approach, and also show that our method is still comparable to the Oracle method even with the heterogeneous error.

In real implementation of the ADMM algorithm, much more flexible strategies than z Step can be involved. Future works also include developments of the proposed IL-ADMM algorithm. For example, to improve the communication efficiency, we could consider the other versions of ADMM algorithm as in [32]–[33]. Here we assumed that the dataset are split into K machines, that is to say, the sample size N is large. However, when dataset is of ultra-high dimensionality, it would be an interesting issue to discuss how to split the data along the p direction, and how to implement the corresponding ADMM algorithm as in [34]–[35].

Moreover, the expectile regression is more applicable in engineering, especially in communication network. For example, the base stations of 5G network are smaller, picocells and femtocells are widely used, and micro data centers neighboring access network edge are popular trends. Large-scale data distributed in the micro data centers are always unbalanced, and usually the estimation object based on the data are heavy-tailed distributed, hence the expectile regression in that condition could be more robust than quantile regression. The network performance prediction is another example. Although deep learning and statistical algorithm are used to predict network performance, the error range varies dramatically due to heterogeneity of networks. Here, the expectile regression gives play as the a generalized form of mean regression, and could be considered as one more robust alternative.

APPENDIX: PROOFS OF THE THEOREMS

Proof of Theorem 1: Based on the surrogate loss function (5), by direct computation, we obtain

\[
\Psi_n(\beta) \triangleq \nabla L(\beta) = -\nabla L_1(\beta) - \nabla L_N(\beta^0) + \nabla L_1(\beta^0) = \frac{1}{n} \sum_{i=1}^n \left[ x_i \varphi_\tau(y_{1i} - x_i^T \beta) + \frac{1}{K} \sum_{k=1}^K x_{ki} \varphi_\tau(y_{ki} - x_{ki}^T \beta^0) - x_i \varphi_\tau(y_{1i} - x_i^T \beta^0) \right] = \mathbb{P}_n \psi_\beta, \tag{A.15}
\]

where \( \varphi_\tau(\lambda) = 2 \tau \lambda \mathbb{1}(\lambda \geq 0) + 2 (1 - \tau) \lambda \mathbb{1}(\lambda < 0) \), \( \psi_\beta = x_1 \varphi_\tau(y_{1i} - x_1^T \beta) + \frac{1}{K} \sum_{k=1}^K x_{ki} \varphi_\tau(y_{ki} - x_{ki}^T \beta^0) - x_i \varphi_\tau(y_{1i} - x_i^T \beta^0) \), and \( \mathbb{P}_n \) is the empirical distribution.

Note that \( \varphi_\tau(\lambda) \) is the first order derivative of function \( \phi_\tau(\lambda) \). Define \( \Psi(\beta) = \mathbb{E} \psi_\beta \). According to condition (C4), we have \( \Psi(\beta_0) = \mathbb{E} \left[ x_1 \varphi_\tau(y_{1i} - x_1^T \beta_0) \right] = 0 \), which means that \( \beta_0 \) is a stationary point of \( \Psi(\beta) \). By condition (C1) and the Theorem 5.9 in Van der Vaart [36], we only need to verify the following two conditions:

(a) \( \sup_{\beta \in \mathcal{B}} \| \Psi_n(\beta) - \Psi(\beta) \|_F \xrightarrow{p} 0 \).

(b) \( \inf_{\beta, d(\beta, \beta_0) \leq \epsilon} \| \Psi(\beta) \| > 0 \equiv \| \Psi(\beta_0) \| \), where \( d(\cdot, \cdot) \) is the Euclidean distance function, and \( \epsilon \) is any positive constant.

In fact, the first condition (a) could be guaranteed by the Theorem 2 in Jennrich [37]. Hence we only need to verify three conditions of Theorem 2 in Jennrich [37]. Notice that the first condition on parameter space \( \mathcal{B} \) is satisfied by condition (C1). And, \( \psi_\beta \) is continuous with probability one at each \( \beta \) by condition (C4). Next, we check that the third condition of Theorem 2 in Parikh and Boyd [34]. Under conditions (C1) and (C4), for \( k = 1, \ldots, K \), we have

\[
\begin{align*}
\| x_{ki} \varphi_\tau(y_{ki} - x_{ki}^T \beta^0) \|_2 & \leq 2 \| x_{ki} [ x_1^T (\beta_0 - \beta^0) + \varepsilon_{ki} ] \|_2 \\
& \leq 2 \| x_{ki} x_1^T (\beta_0 - \beta^0) \|_2 + 2 \| x_{ki} \varepsilon_{ki} \|_2 \\
& \leq 2 \| x_{ki} x_1^T \|_F \| \beta_0 - \beta^0 \|_2 + 2 \| x_{ki} \varepsilon_{ki} \|_2 \\
& \leq 2C \| x_{ki} x_1^T \|_F + 2 \| x_{ki} \varepsilon_{ki} \|_2,
\end{align*}
\]

where \( \| \cdot \|_F \) is the Frobenius norm of a matrix, and \( C \) is a definite positive number. By combining the properties of the
trace of a matrix, we have
\[
\begin{align*}
\|x_{ki}^T x_{ki}^T\|_F^2 &= \left[\text{tr}(x_{ki}^T x_{ki} x_{ki}^T x_{ki})\right]^2 = \left[\text{tr}(x_{ki}^T x_{ki} x_{ki}^T x_{ki})\right]^2 \\
&= \left[\|x_{ki}\|_2^2\right]^2 = \|x_{ki}\|_2^2.
\end{align*}
\]
Under conditions (C1) and (C3), we get \(\text{E}\|x_{ki}^T x_{ki}\|_F = E\|x_{ki}\|_2^2 < \infty\), then combined with conditions (C3) and (C4), we have
\[
\begin{align*}
\left[\|x_{ki}^T x_{ki}\|_F\right]^2 &\leq E\left[\|x_{ki} x_{ki}\|_F^2\right] = E\left(\|x_{ki}^T x_{ki}\|_2^2\right) \\
&= E\left[\|x_{ki}\|_2^2 E\left(\|x_{ki}\|_2^2\right)\right] < \infty.
\end{align*}
\]
So we can get \(\|x_{ki}^T x_{ki}\|_F^2 \leq h(x_{ki})\), where \(h(x_{ki}) = 2C \|x_{ki}^T x_{ki}\|_F^2 + 2 \|x_{ki}^T x_{ki}\|_F\), and \(Eh(x_{ki}) < \infty\). Similarly, we can get \(h(x_{ki}) < \infty\).

In addition, to define \(H(x) = 2h(x_{ki}) + \frac{1}{K} \sum_{i=1}^K h(x_{ki})\), then the inequality (A.16) means \(\|\psi_\beta\|_2 \leq H(x)\), and \(E|H(x)| < \infty\). This verifies the third condition of Theorem 2 in Parikh and Boyd [34]. Hence, the first condition (a) is satisfied.

Finally, by conditions (C1)-(C2) and the continuity of \(\psi(\beta)\), the second condition (b) is verified (See Problem 5.27 in Van der Vaart [36]). Thus, we have the convergence of \(\hat{\beta}\) to \(\beta_0\) in probability.

**Proof of Theorem 2:** Let \(V\) be a symmetric and positive definite matrix, \(U\) be a random variable and \(A_n(t)\) be a convex objective function with the minimum point \(a_n\). Since Knight [38] had verified that, if
\[
A_n(t) = \frac{1}{2} t^T V t + U^T t + o_p(1),
\]
then
\[
a_n \overset{d}{\to} -V^{-1}U.
\]
In the following, given the above consistency results, we set up to derive the result (A.17), we then derive the asymptotic normality of \(\hat{\beta}\).

Define \(\hat{\delta}_n = \sqrt{n}(\hat{\beta} - \beta_0)\), then \(\hat{\beta} = \beta_0 + \frac{\hat{\delta}_n}{\sqrt{n}}\). Notice
\[
\hat{\delta}_n = \arg \min_{\delta} \left[ L(\beta_0 + \frac{\delta}{\sqrt{n}}) - L(\beta_0) \right].
\]
Based on the (A.17), the proof of Theorem 2 will be completed if for random variable \(Z \sim N(0, \frac{\sigma^2(K, \tau)}{K})\), we can obtain
\[
\begin{align*}
n \left[ L(\beta_0 + \frac{\delta}{\sqrt{n}}) - L(\beta_0) \right] &= \frac{1}{2} \delta^T \left[ g(\tau) \Sigma \right] \delta + \delta^T Z + o_p(1),
\end{align*}
\]
where
\[
\begin{align*}
\alpha(K, \tau) &= 4Kc(\tau) + (K - 1)A + (2 - 2K)B, \\
A &= \text{Var}[\psi(\epsilon_{i1})], \\
B &= \text{Cov}[\psi(\epsilon_{i1}), \psi(\epsilon_{i1})].
\end{align*}
\]
In fact, we can obtain
\[
\begin{align*}
n \left[ L(\beta_0 + \frac{\delta}{\sqrt{n}}) - L(\beta_0) \right] &= n \left[ L_1(\beta_0 + \frac{\delta}{\sqrt{n}}) - L_1(\beta_0) + \left( \nabla L_N(\beta_0) - \nabla L_1(\beta_0), \frac{\delta}{\sqrt{n}} \right) \right] \\
&= \frac{n}{\sqrt{n}} \left[ \sum_{i=1}^n \phi_\tau(\epsilon_{i1}) - \frac{\delta^T x_{i1}}{\sqrt{n}} - \delta^T \left[ \frac{1}{\sqrt{n}} \sum_{i=1}^n x_{i1} \psi(\epsilon_{i1}) \right] \right] \\
&= \delta^T \left[ \frac{1}{\sqrt{n}K} \sum_{i=1}^n \sum_{k=1}^n x_{ki} \psi(\epsilon_{ki}) \right].
\end{align*}
\]
Under conditions (C4) and (C5), by the Appendix of Zhao and Zhang [26], we obtain
\[
G_n(\delta) = g(\tau) \delta^T \sum_{i=1}^n x_{i1} x_{i1}^T \delta - \delta^T \left[ \frac{1}{\sqrt{n}} \sum_{i=1}^n x_{i1} \psi(\epsilon_{i1}) \right] + o_p(1),
\]
Hence, we have
\[
G_n(\delta) + H_n(\delta)
\]
where
\[
\begin{align*}
G_n(\delta) &= \sum_{i=1}^n \phi(\epsilon_{i1}) - \frac{\delta^T x_{i1}}{\sqrt{n}} - \phi(\epsilon_{i1}), \\
H_n(\delta) &= \delta^T \left[ \frac{1}{\sqrt{n}} \sum_{i=1}^n x_{i1} \psi(\epsilon_{i1}) \right] - \delta^T \left[ \frac{1}{\sqrt{n}K} \sum_{i=1}^n \sum_{k=1}^n x_{ki} \psi(\epsilon_{ki}) \right].
\end{align*}
\]
Under conditions (C4) and (C5), the Appendix of Zhao and Zhang [26], by the Appendix of Theorem 2, we obtain
\[
G_n(\delta) = g(\tau) \delta^T \sum_{i=1}^n x_{i1} x_{i1}^T \delta - \delta^T \left[ \frac{1}{\sqrt{n}} \sum_{i=1}^n x_{i1} \psi(\epsilon_{i1}) \right] + o_p(1),
\]
Hence, we have
\[
G_n(\delta) + H_n(\delta)
\]
where
\[
\begin{align*}
Z_i &= -\xi_i + \eta_i - \frac{1}{K} \xi, \\
\xi_i &= x_{i1} \psi(\epsilon_{i1}), \\
\eta_i &= x_{i1} \psi(\epsilon_{i1}), \\
\xi &= \sum_{i=1}^K x_{ki} \psi(\epsilon_{ki}),
\end{align*}
\]
with \(\xi_i = x_{i1} \psi(\epsilon_{i1}), \eta_i = x_{i1} \psi(\epsilon_{i1}), \xi = \sum_{k=1}^K x_{ki} \psi(\epsilon_{ki})\) and \(I_{p \times p}\) is the \(p\)-order identity matrix.
Let \( \text{Var} \left( \frac{s}{n}, \frac{n}{n} \right) = W \), where

\[
W = \left( \begin{array}{ccc}
\text{Var}(A) & \text{Var}(B) & \text{Var}(C) \\
\text{Var}(D) & \text{Var}(E) & \text{Var}(F) \\
\text{Var}(G) & \text{Var}(H) & \text{Var}(I)
\end{array} \right)
\]

Denote \( Z \triangleq \frac{1}{n} \sum_{i=1}^{n} Z_i \), by the fact that \( Z \) is a sum of independent and identically distributed zero-mean random vectors and condition (C5), we have

\[
\text{Var}(Z) = \frac{1}{n} \sum_{i=1}^{n} \text{Var}(Z_i)
\]

Due to (A.18), (A.19) and (A.20), by using the result (A.17) and Slutsky’s theorem, as \( n \to \infty \), we have

\[
\frac{n}{\sqrt{K}} \sum_{i=1}^{K} X_i \xrightarrow{d} N(0, \frac{\alpha(K, r) \Sigma}{K}).
\]

By Lindeberg-Feller central limit theorem, we get the following results:

\[
Z \xrightarrow{d} N(0, \frac{\alpha(K, r) \Sigma}{K}).
\] (A.20)

Due to (A.18), (A.19) and (A.20), by using the result (A.17) and Slutsky’s theorem, as \( n \to \infty \), we obtain that

\[
n^{1/2}(\beta - \beta_0) \xrightarrow{d} N(0, \frac{\alpha(K, r) \Sigma}{4\sqrt{g^2(\tau)}})^{-1}.
\] (A.21)

Then, based on the fact that \( N = nK \) and the result (A.21), as \( N \to \infty \), the asymptotic property of \( \beta \) holds as follows,

\[
N^{1/2}(\beta - \beta_0) \xrightarrow{d} N(0, \frac{\alpha(K, r) \Sigma}{4\sqrt{g^2(\tau)}})^{-1},
\]

which completes the proof of Theorem 2.

REFERENCES

[1] N. Rahmato, A. Paul, F. Saeed, W.-H. Hong, H. Seo, and J. Kim, “Machine learning-based automated image processing for quality management in industrial Internet of Things,” Int. J. Distrib. Sensor Netw., vol. 15, no. 10, Oct. 2019, Art. no. 1505177198355.
[2] C. Venkatesan, P. Karthigaikumar, A. Paul, S. Satheeskumaran, and R. Kumar, “ECG signal preprocessing and SVM classifier-based abnormality detection in remote healthcare applications,” IEEE Access, vol. 6, pp. 9767–9773, 2018.
[3] A. Paul, A. Ahmad, M. M. Rathore, and S. Jabbar, “Smartbuilder: Defining human behaviors using big data analytics in social Internet of Things,” IEEE Wireless Commun., vol. 23, no. 5, pp. 68–74, Oct. 2016.
[4] R. McDonald, M. Mohri, N. Silberman, D. Walker, and G. S. Mann, “Efficient large-scale distributed training of conditional maximum entropy models,” in Proc. Adv. Neural Inf. Process. Syst., 2009, pp. 1231–1239.
[5] S. Boyd, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Found. Trends Mach. Learn., vol. 3, no. 1, pp. 1–122, 2010.
[6] J. C. Duchi, A. Agarwal, and M. J. Wainwright, “Dual averaging for distributed optimization: Convergence analysis and network scaling,” IEEE Trans. Autom. Control, vol. 57, no. 3, pp. 592–606, Mar. 2012.
[7] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, “Optimal distributed online prediction using mini-batches,” J. Mach. Learn. Res., vol. 13, pp. 165–202, Jan. 2012.
[36] A. W. Van der Vaart, *Asymptotic Statistics*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[37] R. I. Jennrich, “Asymptotic properties of non-linear least squares estimators,” *Ann. Math. Statist.*, vol. 40, no. 2, pp. 633–643, Apr. 1969.

[38] K. Knight, “Limiting distributions for $L_1$ regression estimators under general conditions,” *Ann. Statist.*, vol. 26, no. 2, pp. 755–770, 1998.

**AIJUN HU** received the M.S. degree in probability theory and mathematical statistics from Wuhan University, in 2006. He is currently pursuing the Ph.D. degree in statistics with the Huazhong University of Science and Technology. He is also a Lecturer with the Hubei University of Education. His research interests include big data analysis and distributed computing.

**CHUJIN LI** received the Ph.D. degree in probability and statistics from the Huazhong University of Science and Technology, China, in 2005. From 2009 to 2010, he undertook his Postdoctoral Research work at the Illinois Institute of Technology, Chicago, USA. He is currently an Associate Professor with the Huazhong University of Science and Technology. His research interests include stochastic analysis, mathematical statistics, and applied statistics.

**JING WU** received the B.Eng. degree in communication engineering and the Ph.D. degree in communication and information systems from Wuhan University, China, in 2002 and 2007, respectively. From 2004 to 2005, she undertook her Postdoctoral Research work at LIMOS, Clermont-Ferrand, France. She is currently an Associate Professor with Wuhan University. Her research interests include wireless communication networks, network simulation, and intelligence data processing.