*mathematics*

MDPI

*Article*

# Distributed Estimation for $\ell_0$-Constrained Quantile Regression Using Iterative Hard Thresholding

**Zhihe Zhao and Heng Lian ***

Department of Mathematics, City University of Hong Kong, Hong Kong, China; zhzhao5-c@my.cityu.edu.hk
* Correspondence: henglian@cityu.edu.hk

**Abstract:** Distributed frameworks for statistical estimation and inference have become a critical toolkit for analyzing massive data efficiently. In this paper, we present distributed estimation for high-dimensional quantile regression with $\ell_0$ constraint using iterative hard thresholding (IHT). We propose a communication-efficient distributed estimator which is linearly convergent to the true parameter up to the statistical precision of the model, despite the fact that the check loss minimization problem with an $\ell_0$ constraint is neither strongly smooth nor convex. The distributed estimator we develop can achieve the same convergence rate as the estimator based on the whole data set under suitable assumptions. In our simulations, we illustrate the convergence of the estimators under different settings and also demonstrate the accuracy of nonzero parameter identification.

**Keywords:** distributed estimation; iterative hard thresholding; $\ell_0$ constraint; linear convergence; quantile regression

**MSC:** 62F10

## 1. Introduction

In statistical modeling, it is often the case that the response is only associated with a subset of predictors. Researchers tend to exclude irrelevant variables, referred to as variable selection, for better prediction accuracy and model interpretability [1]. A common approach is to constrain or regularize the coefficient estimates, or equivalently shrink the coefficient estimates towards zero. The shrinkage also has the effect of reducing variance. Depending on the type of shrinkage performed, some of the coefficients may be estimated to be exactly zero. The best-known techniques include ridge regression and the lasso [2]. In addition to these methods, we can also add an $\ell_0$ constraint to yield sparse coefficient estimates [3].

In high-dimensional statistics, $\ell_0$ regularization is widely used in feature selection. It removes noninformative features by penalizing nonzero coefficients and can ultimately result in a sparse model with a subset of the most significant features, serving as an efficient tool for model interpretation and prediction. Compared to other popular approaches such as lasso, $\ell_0$ constraint has the advantage of directly specifying the sparsity level, which may be useful in some scientific investigations (for example, biological researchers may be interested in extracting the top 50 genes for further investigation), but at the same time, the nonconvex nature of the constraint causes some algorithmic and theoretical challenges. Among the known methods for optimization problems with $\ell_0$ constraint, iterative hard thresholding (IHT), which combines gradient descent with a projection operation, offers a fast and scalable solution. Ref. [4] applied this method to models in

check for updates

Academic Editor: Jinting zhang

Received: 17 January 2025
Revised: 14 February 2025
Accepted: 16 February 2025
Published: 18 February 2025

**Citation:** Zhao, Z.; Lian, H. Distributed Estimation for $\ell_0$-Constrained Quantile Regression Using Iterative Hard Thresholding. *Mathematics* **2025**, *13*, 669. https://doi.org/10.3390/math13040669

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

high-dimensional statistical settings and provided convergence guarantees that hold for differentiable functions, which do not directly apply to non-smooth losses. Ref. [5] extended his work and established the linear convergence of IHT under certain statistical precision for the quantile regression model where the check loss function is non-smooth.

Quantile regression proposed by [6], a natural extension to linear regression, estimates the conditional median or other quantiles rather than the conditional mean of the response variables given the predictors. It can thus provide more information about the response distribution. Furthermore, the quantile regression estimates are more robust against outliers in the response measurements [7]. It also has a number of excellent statistical properties, such as invariance to monotone transformation [8]. The empirical applications of quantile regression have appeared in many fields, such as economics, survival analysis, and ecology [9–11].

*Literature Review on Distributed Estimation and Our Contribution*

With the rapid development of information technology, there has been a staggering increase in the scale and scope of data collection. Meanwhile, the network bandwidth and privacy or security concerns set limitations on processing a large amount of data on one single machine. Inspired by the idea of divide-and-conquer, distributed frameworks for statistical estimation and inference have become a critical toolkit for researchers to understand complicated large data. Various distributed methods have been proposed [12–14], allowing us to store data separately and utilize the computing power of all machines by analyzing data simultaneously. Carefully designed algorithms could improve the performance in a distributed system. Ref. [15] proposed a refinement of the simple averaging estimator involving Hessian matrices to be computed and transferred, which leads to a heavy communication cost of the order $O(d^2)$ when the parameter dimension $d$ is high. To tackle this problem, ref. [16] conducted Newton-type iteration distributedly instead, without transferring the Hessian matrices. With a similar strategy, ref. [17] suggested a well-designed distributed framework and introduced an approximate likelihood approach. It can dramatically reduce the communication cost by replacing the transmission of higher-order derivatives with local first-order derivatives. Following the existing key ideas, in this paper, our research question is as follows:

> How could distributed estimation methods with $\ell_0$-constraints be designed to achieve convergence rates comparable to centralized estimators in sparse quantile regression models?

The main contribution of the current work is to present distributed estimation for the above-stated problem with nonconvex constraint and non-smooth loss, distinguishing our work from the existing ones. We adapt the framework of [17] to develop a communication-efficient distributed method and provide the convergence rate theoretically. The conclusions of [17] on the convergence of the distributed system that are valid for loss functions with continuous second derivatives cannot be directly applied to the non-smooth check loss function for quantile regression. We show that under suitable assumptions, the distributed estimator has the same convergence rate as the estimator utilizing the whole data set. Existing distributed learning with strong theoretical guarantees is mostly concerned with smooth convex models, and thus, our work fills in an important gap in the literature.

In the next section, we present the distributed estimator for quantile regression with $\ell_0$ constraint. In Section 3, we provide the convergence rate of the distributed estimator. Some numerical experiments are presented in Section 4 to investigate the finite-sample performance of the estimator. We conclude the paper with some discussions in Section 5. The proofs are contained in the Appendix A.

## 2. Background and Methodology

We begin by giving a description of quantile regression with an $\ell_0$ constraint. We then turn to developing a communication-efficient distributed estimator.

### *2.1. Quantile Regression with $\ell_0$ Constraint*

Let the sample $\left(y_i, \mathbf{x}_i = (x_{i1}, \ldots, x_{id})^{\mathrm{T}}\right)$, $i = 1, \ldots, n$ be independent and identically distributed (i.i.d.) and satisfy

$$y_i = \mu_0 + \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\theta}_0 + \epsilon_i, \tag{1}$$

with $P(\epsilon_i \leqslant 0|\mathbf{x}_i) = \tau$, where $\mathbf{x} \in \mathbb{R}^d$ and $(\mu_0, \boldsymbol{\theta}_0)$ denotes the true parameter values. The sparsity of $\boldsymbol{\theta}_0$ (that is, number of nonzero components) is denoted as $s_0$. We assume $s$ is a known upper bound to $s_0$. The estimator for quantile regression with $\ell_0$ constraint is defined as

$$\widehat{\boldsymbol{\theta}} = \underset{\mu, \boldsymbol{\theta} \in \mathbb{R}^d : \|\boldsymbol{\theta}\|_0 \leqslant s}{\arg\min} \frac{1}{n} \sum_i \rho_\tau(y_i - \mu - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\theta}),$$

where $\rho_\tau(x) = x(\tau - I\{x \leqslant 0\})$ is the piecewise linear check loss and $\tau \in (0, 1)$ represents the quantile level. In Figure 1, we show the loss function $\rho_\tau(.)$ for different values of $\tau$. When $\tau = 0.5$, the loss function becomes $\rho_{0.5}(x) = \frac{1}{2}|x|$. To see why this makes sense, we note that it is well-known that given a sample $\{y_1, \ldots, y_n\}$, the minimizer of $\min_a \sum_{i=1}^n \frac{1}{2}|y_i - a|$ is the sample median (the minimizer of the least squares loss is the sample mean). For other values of $\tau$, the loss function has a similar shape to absolute value function but is skewed so that it can recover other quantiles. For details, see [6]. The intercept $\mu$ would be omitted in the following for the simplicity of notation.

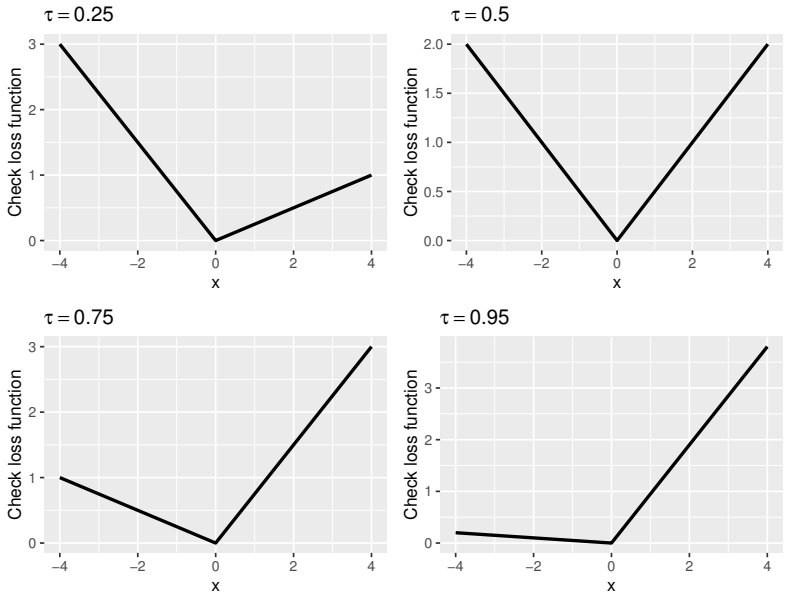



**Figure 1.** The curves of the piecewise linear check loss function at $\tau \in \{0.25, 0.5, 0.75, 0.95\}$.

A subgradient of the check loss function is $(1/n)\sum_i \mathbf{x}_i(I\{y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\theta} \leqslant 0\} - \tau)$. With an initial value $\boldsymbol{\theta}^0$, we can apply the iterative hard thresholding algorithm, which is a projected gradient descent method for the nonconvex case, to compute the estimator $\widehat{\boldsymbol{\theta}}$ iteratively as

$$\boldsymbol{\theta}^{t+1} = P_s\Big(\boldsymbol{\theta}^t - \eta \frac{1}{n} \sum_i \mathbf{x}_i(I\{y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\theta}^t \leqslant 0\} - \tau)\Big), t = 0, 1, \ldots,$$

where $\eta > 0$ denotes the step size. After a sufficiently large number of iterations $T$, we set $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}^T$. In the above, the projection operator $P_s(\boldsymbol{\theta})$ enforces the $\ell_0$ constraint by retaining $s$

elements with the largest absolute values (in magnitude) and setting the other entries of $\boldsymbol{\theta}$ to be zero.

### *2.2. Distributed Estimation*

In this subsection, we apply the communication-efficient surrogate likelihood (CSL) framework, proposed by [17], to solve $\ell_0$-constraint quantile regression.

We first review the definitions and notations required in the distributed framework. In a distributed setting, $N$ observations are randomly and evenly distributed to $m$ machines with $n$ observations stored on each one. Let $\mathbf{Z} := \{Z_{ij}, i = 1, \ldots, n, j = 1, \ldots, m\}$ be a set of $N = mn$ observations independently and identically distributed with the distribution $\mathbb{P}_{\boldsymbol{\theta}_0}$, where $\{\mathbb{P}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ is a family of statistical models parameterized by $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^d$. Here, $\boldsymbol{\Theta}$ is the parameter space and $\boldsymbol{\theta}_0$ is the true parameter generating the data set. In a regression problem, we have $Z_{ij} = (\mathbf{x}_{ij}, y_{ij})$. We use $\mathbf{Z}_j := \{Z_{ij}, i = 1, \ldots, n\}$ to denote the $n$ observations stored on the $j$th machine. In particular, $\mathbf{Z}_1$ stands for the data on the first machine, which is regarded as the central machine (server) that can directly communicate with all others, while other machines are worker machines that can only communicate with the central machine.

Ref. [17] presented a surrogate loss function to approximate the global loss function, from which we can derive the surrogate loss function $\widetilde{\mathcal{L}}(\boldsymbol{\theta})$ for the quantile regression model as

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}) = \mathcal{L}_1(\boldsymbol{\theta}) - \left\langle \boldsymbol{\theta}, \nabla \mathcal{L}_1(\widetilde{\boldsymbol{\theta}}) - \nabla \mathcal{L}_N(\widetilde{\boldsymbol{\theta}}) \right\rangle, \tag{2}$$

where $\widetilde{\boldsymbol{\theta}}$ is a suitable initial estimator. For example, $\widetilde{\boldsymbol{\theta}}$ can be the minimizer of the empirical loss function on the first machine. $\mathcal{L}_1(\boldsymbol{\theta})$ is the local loss function on the first machine and $\nabla \mathcal{L}_1(\boldsymbol{\theta})$ is the subgradient of $\mathcal{L}_1(\boldsymbol{\theta})$, that is, $\mathcal{L}_1(\boldsymbol{\theta}) = (1/n) \sum_{i=1}^{n} \rho_\tau(y_{i1} - \mathbf{x}_{i1}^{\mathrm{T}} \boldsymbol{\theta})$, $\nabla \mathcal{L}_1(\boldsymbol{\theta}) = (1/n)) \sum_{i=1}^{n} \mathbf{x}_{i1}(I\{y_{i1} - \mathbf{x}_{i1}^{\mathrm{T}} \boldsymbol{\theta} \leqslant 0\} - \tau)$. $\mathcal{L}_N(\boldsymbol{\theta})$ is the global loss function based on all observations, and $\nabla \mathcal{L}_N(\boldsymbol{\theta})$ is the subgradient of $\mathcal{L}_N(\boldsymbol{\theta})$. To understand the meaning of $\widetilde{\mathcal{L}}$ in (2), we note that the first term $\mathcal{L}_1(\boldsymbol{\theta})$ is the loss based on data on machine 1. The desired global loss is $\mathcal{L}_N(\boldsymbol{\theta})$. Thus, to correct for the difference between $\mathcal{L}_1(\boldsymbol{\theta})$ and $\mathcal{L}_N(\boldsymbol{\theta})$, the second term $\left\langle \boldsymbol{\theta}, \nabla \mathcal{L}_1(\widetilde{\boldsymbol{\theta}}) - \nabla \mathcal{L}_N(\widetilde{\boldsymbol{\theta}}) \right\rangle$ in (2) adjusts the gradient of the loss at the current estimate $\widetilde{\boldsymbol{\theta}}$, to make $\widetilde{\mathcal{L}}(\boldsymbol{\theta})$ approximate $\mathcal{L}_N(\boldsymbol{\theta})$ better.

Using the surrogate loss function, the communication-efficient distributed estimator is

$$\breve{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta}: \|\boldsymbol{\theta}\|_0 \leqslant s}{\arg\min}\, \widetilde{\mathcal{L}}(\boldsymbol{\theta}).$$

In practice, we apply IHT to compute the distributed estimator $\breve{\boldsymbol{\theta}}$, noting that the subgradient of the surrogate loss function is $\nabla \widetilde{\mathcal{L}}(\boldsymbol{\theta}) = \nabla \mathcal{L}_1(\boldsymbol{\theta}) - \left(\nabla \mathcal{L}_1(\widetilde{\boldsymbol{\theta}}) - \nabla \mathcal{L}_N(\widetilde{\boldsymbol{\theta}})\right)$. Note that to obtain $\nabla \mathcal{L}_N(\widetilde{\boldsymbol{\theta}})$ in the distributed setting, $\widetilde{\boldsymbol{\theta}}$ needs to be broadcast to all local machines and the local subgradients are computed on all machines and sent to the central machine to form the global gradient $\nabla \mathcal{L}_N(\widetilde{\boldsymbol{\theta}})$.

We can repeat the above procedure for multiple stages. In stage $t$, we set the starting value $\boldsymbol{\theta}_0^{(t)}$ to be the estimator obtained in the previous stage, and then form the surrogate loss function

$$\widetilde{\mathcal{L}}^{(t)}(\boldsymbol{\theta}) = \mathcal{L}_1(\boldsymbol{\theta}) - \left\langle \boldsymbol{\theta}, \nabla \mathcal{L}_1(\boldsymbol{\theta}_0^{(t)}) - \nabla \mathcal{L}_N(\boldsymbol{\theta}_0^{(t)}) \right\rangle,$$

and then obtain $\boldsymbol{\theta}_0^{(t+1)}$ by minimizing the surrogate loss function above (using IHT)

$$\boldsymbol{\theta}_0^{(t+1)} \in \underset{\boldsymbol{\theta}: \|\boldsymbol{\theta}\|_0 \leqslant s}{\arg\min}\, \widetilde{\mathcal{L}}^{(t)}(\boldsymbol{\theta}),$$

which serves as the initial value for the next stage (see Algorithm 1 for pseudocode).

**Remark 1.** *For convenience of presentation only, we assume data are evenly distributed on all machines. However, when the total sample size $N$ is fixed, the only important quantity is the sample size $n$ on machine 1, which determines the accuracy of the initial estimator, as well as the surrogate function, while sample sizes for other machines play no role since other machines only contribute the gradients and all gradients are aggregated by machine 1.*

*To see this point more clearly, in (2), the only term that requires the participation of other machines is the term $\mathcal{L}_N(\widetilde{\boldsymbol{\theta}})$. Whatever the local sample sizes for machines $2, \ldots, m$, denoted by $n_2, \ldots, n_m$, each machine simply sends $\nabla \mathcal{L}_j(\widetilde{\boldsymbol{\theta}}) = \frac{1}{n_j} \sum_{i=1}^{n_j} \mathbf{x}_{ij}(I\{y_{ij} - \mathbf{x}_{ij}^{\mathrm{T}}\widetilde{\boldsymbol{\theta}} \leqslant 0\} - \tau)$, and machine 1 can aggregate them by an weighted average to obtain exactly $\nabla \mathcal{L}_N(\widetilde{\boldsymbol{\theta}}) = \frac{n_j}{N} \sum_j \nabla \mathcal{L}_j(\widetilde{\boldsymbol{\theta}})$. Alternatively, the machines $2, \ldots, m$ can simply send the sum of gradients on $n_j$ observations (instead of the mean of gradients on $n_j$ observations) and then machine 1 can sum up these quantities and divide it by $N$, to obtain exactly $\nabla \mathcal{L}_N(\widetilde{\boldsymbol{\theta}})$.*

**Algorithm 1** DISTRIBUTED ESTIMATION FOR QUANTILE REGRESSION USING IHT

**Input**: sparsity level $s$, number of stages $T$, number of iterations of IHT $Q$, step size $\eta$
**Initialize** $\boldsymbol{\theta}_0^{(0)}$
**for** $t = 0, \ldots, T-1$ **do**
  **for** $j = 1, \ldots, m$ **do**
    (in parallel)
    Receive $\boldsymbol{\theta}_0^{(t)}$ from machine 1
    Compute the local subgradient $\nabla \mathcal{L}_j(\boldsymbol{\theta}_0^{(t)})$ on machine $j$
    Transmit the local subgradient $\nabla \mathcal{L}_j(\boldsymbol{\theta}_0^{(t)})$ to machine 1 (central machine)
  **end for**
  Calculate $\nabla \mathcal{L}_N(\boldsymbol{\theta}_0^{(t)}) = \frac{1}{m} \sum_{j=1}^{m} \nabla \mathcal{L}_j(\boldsymbol{\theta}_0^{(t)})$ on machine 1
  **for** $q = 0, \ldots, Q-1$ **do**
    $\boldsymbol{\theta}_{q+1}^{(t)} = P_s\big(\boldsymbol{\theta}_q^{(t)} - \eta \nabla \widetilde{\mathcal{L}}^{(t)}(\boldsymbol{\theta}_q^{(t)})\big)$ where $\nabla \widetilde{\mathcal{L}}^{(t)}(\boldsymbol{\theta}) = \nabla \mathcal{L}_1(\boldsymbol{\theta}) - \big(\nabla \mathcal{L}_1(\boldsymbol{\theta}_0^{(t)}) - \nabla \mathcal{L}_N(\boldsymbol{\theta}_0^{(t)})\big)$
  **end for**
  Set $\boldsymbol{\theta}_0^{(t+1)} = \boldsymbol{\theta}_Q^{(t)}$;
**end for**
**Output** $\boldsymbol{\theta}_0^{(T)}$

## 3. Main Results

In this section, we provide the theoretical result showing the convergence rate of the distributed estimator. We will see that the distributed estimator can achieve the same convergence rate as the global estimator based on the whole data set under suitable assumptions. We note that the check loss function for quantile regression is neither strongly convex nor second-order differentiable, and $\ell_0$ constraint is nonconvex, which renders the theory in [17] not directly applicable to our current problem.

Let $f(.|\mathbf{x})$ be the conditional density of $\epsilon_i$ (as defined in (1)). The following assumptions are imposed.

(A1) Define $\boldsymbol{\Theta} = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\| \leqslant \|\boldsymbol{\theta}_0\| \vee 1\}$ and $H(\boldsymbol{\theta}) = E[f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}|\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ (it can be shown that $H(\boldsymbol{\theta})$ actually contains the second-order partially derivatives of $E[\rho_\tau(y - \mathbf{x}^{\mathrm{T}}\boldsymbol{\theta})]$ and is thus referred to as the population Hessian matrix). We assume that $\sup_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbf{a}^{\mathrm{T}} H(\boldsymbol{\theta}) \mathbf{a}$ is bounded from above by a constant $C_2 > 0$ and $\inf_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbf{a}^{\mathrm{T}} H(\boldsymbol{\theta}) \mathbf{a}$ is bounded from below by a constant $C_1 > 0$, for all $(2s + s_0)$-sparse unit vectors $\mathbf{a}$ (that is, $\|\mathbf{a}\| = 1$).

(A2) Components of $\mathbf{x} = (x_1, \ldots, x_d)^{\mathrm{T}}$ are sub-Gaussian random variables in the sense that $E(e^{ax_j}) \leqslant e^{C_3 a^2}$ for any $a \in \mathbb{R}$ and some positive constants $C_3$.

Assumption (A1) imposes a constraint on the population Hessian matrix. To see that $E[f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}|\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ is the Hessian matrix (in other words, it contains the second partial derivatives of $E[\rho_\tau(y-\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta})]$), we note that it is easy to see the first-order partial derivatives (gradient) are given by $E[\mathbf{x}(I\{y-\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}\leqslant 0\}-\tau)]=E[\mathbf{x}(F(\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}|\mathbf{x})-\tau)]$, where $F(.|\mathbf{x})$ denotes the conditional distribution function of $y$. Thus, taking the derivative with respect to $\mathbf{x}$ gives us $E[f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}|\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]$. This is the quantile regression counterpart of the standard assumption in least squares regression that usually requires that eigenvalues of $E[\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ are bounded from above and below by some positive constants. This assumption is also adapted to deal with the sparse regression case where we only need to deal with a sparse unit vector $\mathbf{a}$. Assumption (A2) requires the predictors to be sub-Gaussian, which is necessary for our technical analysis using empirical processes theory.

In the following, $C$ will denote a generic positive constant whose value may vary in different places. In the statement of the theorem, $C_1$ and $C_2$ are constants bounding the eigenvalues of the population Hessian matrix as in (A1). $\boldsymbol{\theta}_0^{(0)}$ and $\boldsymbol{\theta}_0^{(T)}$ are the initial estimator and the final estimator output by our algorithm, respectively.

**Theorem 1.** *Under the assumptions (A1) and (A2), let* $\eta=\frac{2}{C_1+C_2}$, $s\geqslant cs_0$ *with* $c\geqslant 1$ *large enough such that* $(1+\sqrt{s_0/s})\frac{C_2-C_1}{C_2+C_1}=:1-\delta<1$; *when the initial estimator* $\boldsymbol{\theta}_0^{(0)}$ *satisfies* $\|\boldsymbol{\theta}_0^{(0)}-\boldsymbol{\theta}_0\|\leqslant C\left(s^{\frac{5}{2}}\frac{log^{\frac{3}{2}}(d\vee n)}{n}+\sqrt{\frac{slog(d\vee n)}{n}}\right)$, *after* $T\geqslant Clog\left(\frac{N\|\boldsymbol{\theta}_0^{(0)}-\boldsymbol{\theta}_0\|}{slog(d\vee n)}\right)$ *stages with* $Q\geqslant Clog\left(\frac{N\|\boldsymbol{\theta}_0^{(0)}-\boldsymbol{\theta}_0\|}{slog(d\vee n)}\right)$ *iterations for each stage using IHT, with a probability of at least* $1-(d\vee n)^{-C}$, *it holds that*

$$\|\boldsymbol{\theta}_0^{(T)}-\boldsymbol{\theta}_0\|\leqslant C\left(\frac{s^{\frac{5}{2}}log^{\frac{3}{2}}(d\vee n)}{n}+\sqrt{\frac{slog(d\vee n)}{N}}\right),$$

*where* $d\vee n$ *denotes* $\max\{d,n\}$.

**Remark 2.** *We require the initial estimator to be sufficiently accurate, with* $\|\boldsymbol{\theta}_0^{(0)}-\boldsymbol{\theta}_0\|\leqslant C\left(s^{\frac{5}{2}}\frac{log^{\frac{3}{2}}(d\vee n)}{n}+\sqrt{\frac{slog(d\vee n)}{n}}\right)$. *This can be guaranteed when* $\boldsymbol{\theta}_0^{(0)}$ *is computed on the first machine using its local data of size n if n is not too small, as shown in [5]. Intuitively, the initial estimator should be sufficiently accurate for the surrogate loss to be a good approximation of the global loss. If the initial estimator is not accurate enough (for example in our simulation later, this is the case when the local sample size n is small), the method might have bad performance.*

**Remark 3.** *The iterative bound for the estimation error is shown in* (A11) *in the proof, which indicates that the distributed estimator can be linearly convergent to the true parameter* $\boldsymbol{\theta}_0$. *Compared to the iterative bound for the non-distributed global estimator provided by [5], which is* $e_{t+1}=(1-\delta/2)e_t+O\left(\frac{s^{5/2}log^{3/2}(d\vee N)}{N}+\sqrt{\frac{slog(d\vee N)}{N}}\right)$, *we can conclude that the distributed estimator can achieve the same convergence rate as the estimator with all data stored on a single machine, when the term* $\sqrt{\frac{slog(d\vee n)}{N}}$ *is the dominating term. Here,* $\sqrt{\frac{slog(d\vee n)}{N}}$ *is the statistical precision since it is the error bound when using the entire sample with size N. In other words, if all data with sample size N are stored on a single machine, which performs IHT to obtain an estimator* $\widehat{\boldsymbol{\theta}}$, *then we have* $\|\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}_0\|\leqslant\sqrt{\frac{slog(d\vee n)}{N}}$, *according to the result in [5]. Thus, in this paper, we call this term the statistical precision, which is the benchmark error for the global estimator. In particular, this holds when* $\frac{s^{\frac{5}{2}}log^{\frac{3}{2}}(d\vee n)}{n}\leqslant\sqrt{\frac{slog(d\vee n)}{N}}$, *or equivalently,* $m\leqslant\frac{\sqrt{N}}{s^2log(d\vee n)}$. *In other words, when the number of machines is not too large, the distributed estimation has the same convergence rate as the global estimator.*

## 4. Simulations

In this section, we numerically illustrate the convergence of the distributed estimator to verify the theoretical conclusions we obtain in Section 3, comparing it to the local estimator $\widetilde{\boldsymbol{\theta}}$ (only using the data stored on the first machine) and the global estimator $\boldsymbol{\theta}^G$ (assuming all data are stored on a single machine). We then turn to demonstrating the robustness to different parameter setups and the accuracy of nonzero parameter identification.

### *4.1. Convergence Illustration*

We first generate $N$ i.i.d observations by $y_i = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\theta}_0 + \epsilon_i, i = 1, \ldots, N$, where $\mathbf{x}_i$ is from the multivariate normal distribution with mean $\mathbf{0}$ and the covariance matrix $0.2^{|j-k|}, j, k = 1, \ldots, d$. The noise $\epsilon$ is generated from the normal distribution with mean 0 and variance 0.16. We set $\boldsymbol{\theta}_0 = (1, 2, -1, 0.5, -0.3, 0.1, 0.2, -0.1, 0.05, -0.03, 0, \ldots, 0)^{\mathrm{T}}$, with the dimension $d = 100$, $s = 15$. Strictly speaking, as a tuning parameter, we could use cross-validation to fix the proper value of $s$. Here, we solely fix $s$ to discuss the convergence of the estimators.

We will report the estimation error (EE) $\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2$ of the estimators using 100 repetitions in each setting, and the prediction error (PE) based on additional independently generated 5000 observations.

For the first simulation, we fix the whole sample size $N = 10{,}000$ and vary the number of machines $m$, with the results shown in Figure 2. We can observe that the errors decrease with the number of stages used, and with a five-stage estimator, the errors are usually close to those of the global estimator. We see that the one-stage estimation is typically insufficient, with much larger errors. For $m = 5$, the estimator quickly converges with a very small number of stages, while for a larger number of machines, more stages are probably required. We also note that the estimation errors increase with the number of machines $m$.

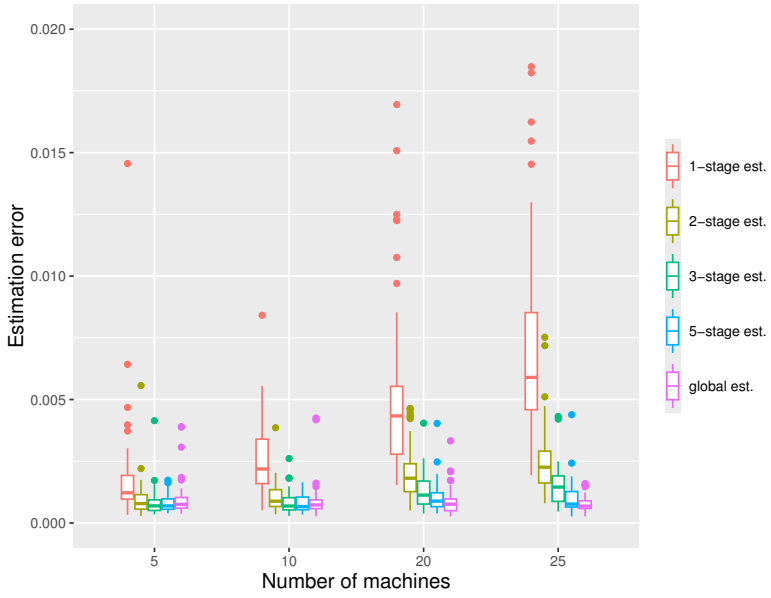



**Figure 2.** The boxplots displaying the estimation errors of the distributed estimators at the $\{1, 2, 3, 5\}$-stage and the global estimator, respectively, under the settings $N = 10{,}000, s = 15$, $d = 100$, $m \in \{5, 10, 20, 25\}$ at quantile level $\tau = 0.25$.

In our second simulation, we keep the local sample size $n = 400$ fixed and increase the number of machines $m$. According to Figure 3, the estimation errors of all estimators decrease with $m$ as expected, since the total sample size $N$ is proportional to $m$. Again, we see one-stage estimators perform much worse than multiple-stage estimators.

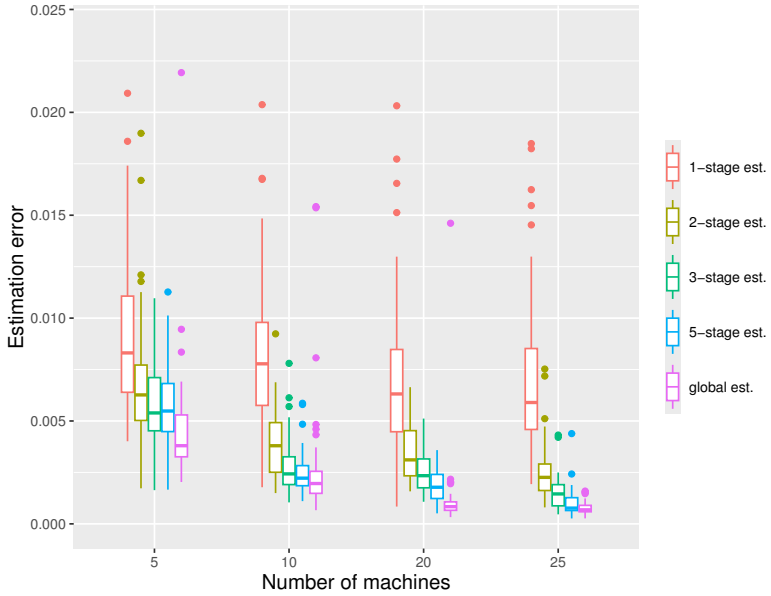



**Figure 3.** The boxplot displaying the estimation errors of the distributed estimators and the global estimator respectively under the settings $n = 400$, $m \in \{5, 10, 20, 25\}$ at quantile level $\tau = 0.25$.

In the third simulation, we fix $m = 5$ and increase the local sample size $n$. Again, we see in Figure 4 that the errors decrease with $n$. When the local sample size $n$ is sufficiently large, two or three stages suffice for the distributed estimators to gain good performances comparable to the global estimators. We also note that when $n$ is small ($n = 400$), there is still a significant gap between the distributed estimator and the global estimator after five stages.

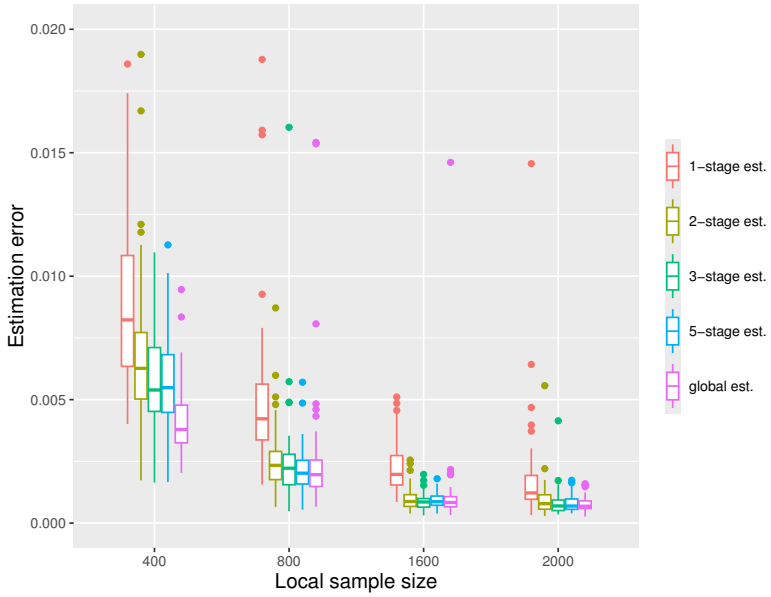



**Figure 4.** The boxplot displaying the estimation errors of the distributed estimators and the global estimator, respectively, under the settings $m = 5, n \in \{400, 800, 1600, 2000\}$ at quantile level $\tau = 0.25$.

We also ran experiments to verify the performance of the distributed estimator we developed in some more extreme cases. We explored the scenarios in which the number of machines $m$ and the dimension $d$ became large with the entire sample size $N = 120{,}000$. We use $m = 10$ and a much larger $m = 100$. With $m = 100$, the local data size is $n = 120$, which is much smaller than $d = 1500$. As shown in Table 1, the errors grow with the dimension $d$, as expected. The distributed estimator largely reduces the estimation errors of the local

estimator and can decrease the errors stage by stage in high-dimensional settings. The estimation errors exhibit a surge when the number of machines is 100, showing the method would fail if $m$ is too big.

**Table 1.** The estimation and prediction errors (%) of the local estimator $\widetilde{\boldsymbol{\theta}}$, the two-stage distributed estimator $\boldsymbol{\theta}_0^{(2)}$, the three-stage distributed estimator $\boldsymbol{\theta}_0^{(3)}$, and the global estimator $\boldsymbol{\theta}^G$ with $N = 12{,}000$, $s = 15, d \in \{100, 1500\}, m \in \{10, 100\}$ at quantile level $\tau = 0.25$.

| $N$ | $d$ | $m$ | $\widetilde{\boldsymbol{\theta}}$ | | $\boldsymbol{\theta}_0^{(2)}$ | | $\boldsymbol{\theta}_0^{(3)}$ | | $\boldsymbol{\theta}^G$ * | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | **PE** (%) | **EE** (%) | **PE** (%) | **EE** (%) | **PE** (%) | **EE** (%) | **PE** (%) | **EE** (%) |
| 12,000 | 100 | 10 | 23.131 (0.484) | 0.811 (0.043) | 23.125 (0.486) | 0.328 (0.025) | 23.122 (0.486) | 0.238 (0.023) | 23.219 (0.313) | 0.091 (0.025) |
| | | 100 | 26.264 (0.553) | 21.373 (6.840) | 24.229 (0.508) | 9.491 (5.547) | 23.278 (0.491) | 6.931 (5.265) | | |
| | 1500 | 10 | 23.701 (0.495) | 1.188 (0.714) | 23.519 (0.490) | 0.351 (0.024) | 23.420 (0.489) | 0.289 (0.047) | 23.332 (0.353) | 0.166 (0.052) |
| | | 100 | 26.771 (0.560) | 26.874 (2.656) | 24.853 (0.497) | 16.396 (2.307) | 23.763 (0.495) | 13.187 (2.083) | | |

* The results for the global estimator $\boldsymbol{\theta}^G$ are the same when $m = 10$ and $m = 100$; thus, we do not present the latter.

We now illustrate what happens if $n$ is very small. We set $N = 5000$, and $n = \{10, 100, 500\}$, with the other settings unchanged as in the first simulation. We can observe from Figure 5 that when the first machine has significantly less data, the initial estimator would not be accurate enough. Correspondingly, the distributed estimators perform much worse, and, even after five stages, the errors are still large; increasing the number of stages further does not help.

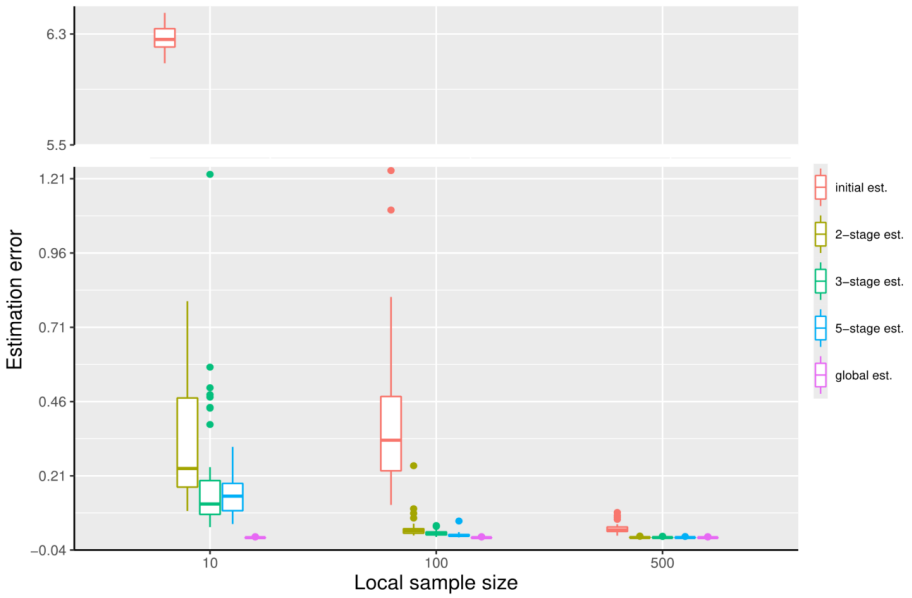



**Figure 5.** The boxplots displaying the estimation errors of the initial estimator, the distributed estimators at the $\{2, 3, 5\}$-stage, and the global estimator, respectively, under the settings $N = 5000$, $s = 15, d = 100, n \in \{10, 100, 500\}$ at quantile level $\tau = 0.25$.

Finally, we illustrate the convergence of the proposed distributed estimator in the case where the noise distribution is chi-squared with 5 degrees of freedom. The other settings remain unchanged as in the first simulation. The boxplots displaying the estimation errors are shown in Figure 6. We see that qualitatively, the results are similar to those of Figure 2. When $m$ is small, the two-stage estimator is almost as good as the global estimator. When

$m$ becomes larger, more stages are required and there is a perceivable performance gap between the distributed estimator and the global estimator.

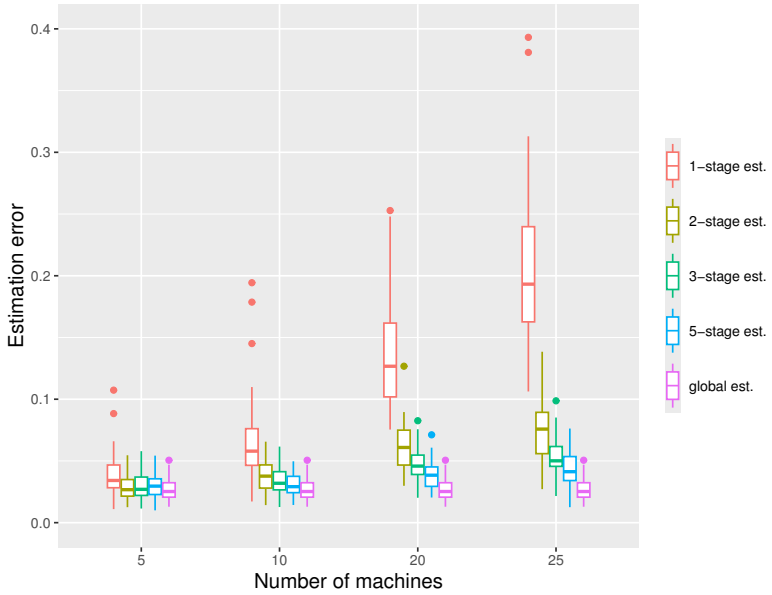



**Figure 6.** The boxplots displaying the estimation errors of the distributed estimators at the $\{1, 2, 3, 5\}$-stage and the global estimator, respectively, under the settings $N = 10,000, s = 15, d = 100, m \in \{5, 10, 20, 25\}$ at quantile level $\tau = 0.25$. The noise distribution used is chi-squared with 5 degrees of freedom.

### *4.2. Variable Identification Performance*

In this simulation, to examine whether the magnitude of the true parameter and the sparsity affect the estimation and variable selection, we generate $N = 5000$ i.i.d observations with $\boldsymbol{\theta}_0 = (1, 0.98, 0.96, \ldots, 0.02, 0, \ldots, 0)^{\mathrm{T}}$, where there are 50 nonzero parameters following a decreasing pattern and the dimension $d = 100$. Using 100 repetitions, we record the number of times each of the 100 parameters is identified as nonzero. With this, we can obtain the corresponding identification frequency. We also report the overall misclassification rate (MCR), which is the proportion of the sum of false positives and false negatives in identifying nonzero parameters.

In our simulations, we set $s \in \{10, 30\}, m \in \{10, 20\}$. As shown in Figure 7 and Table 2, with the special setting of $\boldsymbol{\theta}_0$, the identification rates decrease as the parameter index grows. At $s = 30$, the three-stage estimator slightly outperforms the local estimator, with higher identification rate for the parameters with large magnitudes and a lower misclassification rate. At $s = 10$, the local estimator and the three-stage estimator have almost identical performance. We can also observe that the misclassification rate of the three-stage distributed estimator increases slightly with the number of machines. In addition, we report the computational time of the four cases, each using 100 repetitions, as shown in the footnotes of Table 2. Our algorithm is implemented in R (R-4.4.2) on our desktop computer with 12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz and 16 GB RAM.

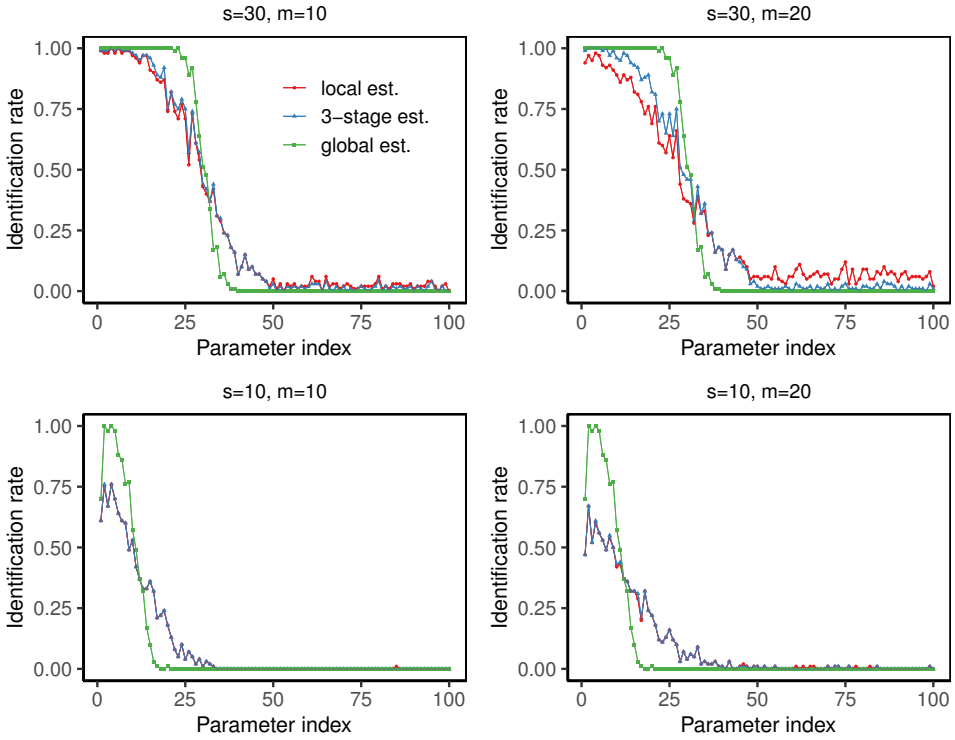



**Figure 7.** The line charts displaying the identification rate of the local estimator $\widetilde{\boldsymbol{\theta}}$, the three-stage distributed estimator $\boldsymbol{\theta}_0^{(3)}$, and the global estimator $\boldsymbol{\theta}^G$, respectively, under the settings $N = 5000$, $d = 100, s \in \{10, 30\}, m \in \{10, 20\}$ at quantile level $\tau = 0.25$. In all cases, each point corresponds to the identification rate over 100 trials for each of the 100 parameters.

**Table 2.** The misclassification rate (%), estimation and prediction errors of the local estimator $\widetilde{\boldsymbol{\theta}}$, the three-stage distributed estimator $\boldsymbol{\theta}_0^{(3)}$, and the global estimator $\boldsymbol{\theta}^G$ with $N = 5000, d = 100$, $s \in \{10, 30\}, m \in \{10, 20\}$ at quantile level $\tau = 0.25$.

| $N$ | $s$ | $m$ | $\widetilde{\boldsymbol{\theta}}$ | | | $\boldsymbol{\theta}_0^{(3)}$ | | | $\boldsymbol{\theta}^{G}$ * | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **MCR (%)** | **PE** | **EE** | **MCR (%)** | **PE** | **EE** | **MCR (%)** | **PE** | **EE** |
| 5000 | 30 | 10 [1] | 22.320 (0.202) | 2.425 (0.053) | 2.523 (0.052) | 21.320 (0.171) | 2.249 (0.049) | 2.093 (0.047) | 20.000 (0.000) | 1.813 (0.026) | 1.191 (0.009) |
| | | 20 [2] | 26.420 (0.303) | 4.940 (0.114) | 4.102 (0.102) | 21.400 (0.194) | 2.619 (0.060) | 2.304 (0.069) | | | |
| | 10 | 10 [3] | 40.000 (0.020) | 13.313 (0.308) | 10.316 (0.057) | 40.000 (0.000) | 12.956 (0.300) | 10.100 (0.059) | 40.000 (0.000) | 12.196 (0.173) | 8.829 (0.023) |
| | | 20 [4] | 40.280 (0.081) | 13.324 (0.298) | 11.398 (0.103) | 40.160 (0.055) | 12.958 (0.294) | 10.871 (0.082) | | | |

* The results for the global estimator $\boldsymbol{\theta}^G$ are the same when $m = 10$ and $m = 20$; thus, we do not present the latter. [1] The computational time is 20.777 min when $(s, m) = (30, 10)$. [2] The computational time is 25.502 min when $(s, m) = (30, 20)$. [3] The computational time is 22.565 min when $(s, m) = (10, 10)$. [4] The computational time is 21.565 min when $(s, m) = (10, 20)$.

## 5. Conclusions

In this article, we theoretically establish the convergence rate of the distributed estimator for quantile regression with $\ell_0$ constraint. We extend the conclusions of [17] to the model with a non-smooth loss function and a nonconvex constraint. The numerical experiments show that the distributed estimators have satisfactory performance. The distributed estimator can decrease the estimation and prediction errors stage by stage. It also largely reduces the errors of the local estimator and two or three stages often suffice

for the distributed framework to gain almost the same accuracy as the global estimator. The errors increase with the number of machines $m$ when $N$ is fixed, decrease with $m$ with $n$ fixed, and also decrease as $n$ grows with $m$ fixed.

Our work has its limitations; for example, in our simulations, we notice that only with each machine having sufficient observations can we have satisfactory performance. This is probably due to the initial estimator being important for performance, and when the local sample size is too small, the initial estimator is not good enough. This makes the proposed method sometimes unreliable in practice.

The proposed method contributes theoretically and businesswise. Existing distributed frameworks with high-dimensional statistical settings have convergence guarantees held only for smooth convex models. We bridge the gap and extend it to the non-smooth and nonconvex situation. In addition, it is effective for addressing large-scale statistical optimization problems. It divides a substantial task into smaller components that can be processed in parallel across multiple machines. This approach can be utilized by commercial companies to analyze vast amounts of browsing data and identify the most influential factors for personalized recommendations. Additionally, biological researchers may resort to this method to pinpoint the most significant genes to diagnose certain diseases.

Some extensions can be considered. For example, one can consider composite quantile regression (CQR) proposed by [18], which is regarded as a robust and efficient alternative to mean regression. One can also consider the quantile matrix regression problem, where the predictors and parameters are in the form of matrices. Also, it is well-known that standard quantile regression based on the check loss does not work for extreme quantile levels such as $\tau = 0.99$, for which more complicated methods such as [19] should be applied. Furthermore, we did not consider the variance estimation problem, which would be useful if we want to perform statistical inference. Furthermore, we chose one particular approach of distributed estimation for its simplicity and feasibility to demonstrate its extension to the particular non-smooth nonconvex problem that interests us; other approaches such as using DGD (distributed gradient descent) or ADMM (alternating direction method of multipliers) could also be investigated [20,21]. Finally, federated learning emphasizes the case of heterogeneous data distribution and asynchronous update [22] and is more challenging to deal with. These interesting problems are left for future work to solve.

**Author Contributions:** Conceptualization, H.L.; methodology, Z.Z. and H.L.; software, Z.Z.; validation, Z.Z. and H.L.; investigation, Z.Z.; resources, H.L.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, H.L. and Z.Z.; visualization, Z.Z.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A. Proof of Theorem 1

**Proof of Theorem 1.** Let $\mathbf{g}^{(t)}$ denote the derivative of the surrogate loss function $\nabla\widetilde{\mathcal{L}}^{(t)}(\boldsymbol{\theta})$. We first focus on the first stage $t = 0$. Define $I_q^{(0)} = S_q^{(0)} \cup S_0$, where $S_q^{(0)}$ and $S_0$ are the

support (set of indices of nonzero entries) of $\boldsymbol{\theta}_q^{(0)}$ and the true parameter $\boldsymbol{\theta}_0$, respectively. We have

$$\begin{aligned}
&\|\boldsymbol{\theta}_{q+1}^{(0)} - \boldsymbol{\theta}_0\| \\
&= \|(\boldsymbol{\theta}_{q+1}^{(0)})_{I_{q+1}^{(0)}} - (\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}}\| \\
&= \|P_s\Big(\boldsymbol{\theta}_q^{(0)} - \eta \mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)})\Big)_{I_{q+1}^{(0)}} - (\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}}\| \\
&\leqslant \|P_s\Big(\boldsymbol{\theta}_q^{(0)} - \eta \mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)})\Big)_{I_{q+1}^{(0)}} - \Big(\boldsymbol{\theta}_q^{(0)} - \eta \mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)})\Big)_{I_{q+1}^{(0)}}\| \\
&\quad + \|\Big(\boldsymbol{\theta}_q^{(0)} - \eta \mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)})\Big)_{I_{q+1}^{(0)}} - (\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}}\| \\
&\leqslant \left(1 + \sqrt{\frac{|I_{q+1}^{(0)}| - s}{|I_{q+1}^{(0)}| - s_0}}\right) \|\Big(\boldsymbol{\theta}_q^{(0)} - \eta \mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)})\Big)_{I_{q+1}^{(0)}} - (\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}}\|,
\end{aligned} \tag{A1}$$

where the last inequality applies Lemma 1 in [4].

By adding and subtracting terms, we obtain

$$\begin{aligned}
&\|\Big(\boldsymbol{\theta}_q^{(0)} - \eta \mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)})\Big)_{I_{q+1}^{(0)}} - (\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}}\| \\
&\leqslant \|\Big(\boldsymbol{\theta}_q^{(0)}\Big)_{I_{q+1}^{(0)}} - (\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}} - \eta\Big(E\mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)}) - E\mathbf{g}^{(0)}(\boldsymbol{\theta}_0)\Big)_{I_{q+1}^{(0)}}\| \\
&\quad + \eta\|\Big(\mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)}) - \mathbf{g}^{(0)}(\boldsymbol{\theta}_0) - E\mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)}) + E\mathbf{g}^{(0)}(\boldsymbol{\theta}_0)\Big)_{I_{q+1}^{(0)}}\| \\
&\quad + \eta\|\mathbf{g}^{(0)}(\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}}\|.
\end{aligned} \tag{A2}$$

Let $e_0^{(0)} = \|\boldsymbol{\theta}_0^{(0)} - \boldsymbol{\theta}_0\|$ and define

$$e_q^{(0)} = (1 - \delta/2)e_{q-1}^{(0)} + \frac{1}{2\delta}a_n^2 + b_{N,n}, \tag{A3}$$

where $a_n = Cs^{5/4}\sqrt{c_n}\sqrt{\log(d \vee n)/n}$, $b_{N,n} = Cs^{1/2}\sqrt{\log(d \vee n)/N} + Cs^{3/2}c_n\log(d \vee n)/n + Cs^{5/4}\sqrt{c_n}\sqrt{\log(d \vee n)/n}\sqrt{e_0^{(0)}}$, and $c_n = C\sqrt{\log(d \vee n)}$. We will show by induction that $\Delta_q^{(0)} := \|\boldsymbol{\theta}_q^{(0)} - \boldsymbol{\theta}_0\| \leqslant e_q^{(0)}$ with a probability at least $1 - (d \vee n)^{-C}$ for all $q$.

Trivially, $\Delta_0^{(0)} = e_0^{(0)}$ by definition. Assume $\Delta_q^{(0)} \leqslant e_q^{(0)}$. Through Lemma 1 in [5], we know that for any $k > 1$, with a probability at least $1 - (d \vee n)^{-C} - nP(\|\mathbf{x}\|_\infty > c_n)$

$$\begin{aligned}
&\|\Big(\mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)}) - \mathbf{g}^{(0)}(\boldsymbol{\theta}_0) - E\mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)}) + E\mathbf{g}^{(0)}(\boldsymbol{\theta}_0)\Big)_{I_{q+1}^{(0)}}\| \\
&= \|\Big(\nabla\mathcal{L}_1(\boldsymbol{\theta}_q^{(0)}) - \nabla\mathcal{L}_1(\boldsymbol{\theta}_0) - E\nabla\mathcal{L}_1(\boldsymbol{\theta}_q^{(0)}) + E\nabla\mathcal{L}_1(\boldsymbol{\theta}_0)\Big)_{I_{q+1}^{(0)}}\| \\
&\leqslant C\sqrt{|I_{q+1}^{(0)}|}\left(s^{1/4}\sqrt{c_n e_q^{(0)}}\sqrt{\frac{s\log(d \vee n)}{n}} + c_n\frac{s\log(d \vee n)}{n} + P(\|\mathbf{x}\|_\infty > c_n)^{1/k}\right).
\end{aligned} \tag{A4}$$

By sub-Gaussianity, we can obtain $P(\|\mathbf{x}\|_\infty > c_n) \leqslant (d \vee n)^{-C}$. When the constant $C$ in $c_n$ is set sufficiently large, $P(\|\mathbf{x}\|_\infty > c_n)$ is very small.

By Taylor's expansion,

$$\Big(E\mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)}) - E\mathbf{g}^{(0)}(\boldsymbol{\theta}_0)\Big)_{I_{q+1}^{(0)}} = E[f(\alpha|\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]_{I_{q+1}^{(0)},:}(\boldsymbol{\theta}_q^{(0)} - \boldsymbol{\theta}_0),$$

where $\alpha \in [-|\mathbf{x}^{\mathrm{T}}(\boldsymbol{\theta}_q^{(0)} - \boldsymbol{\theta}_0)|, |\mathbf{x}^{\mathrm{T}}(\boldsymbol{\theta}_q^{(0)} - \boldsymbol{\theta}_0)|]$ and $E[f(\alpha|\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]_{I_{q+1}^{(0)},:}$ denotes the sub-matrix with rows in $I_{q+1}^{(0)}$. Let $J_{q+1}^{(0)} = I_{q+1}^{(0)} \cup S_q^{(0)}$. Then, we obtain

$$\begin{aligned}
&\|\left(\boldsymbol{\theta}_q^{(0)}\right)_{I_{q+1}^{(0)}} - (\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}} - \eta\left(E\mathbf{g}^{(0)}(\boldsymbol{\theta}_q^{(0)}) - E\mathbf{g}^{(0)}(\boldsymbol{\theta}_0)\right)_{I_{q+1}^{(0)}}\| \\
&\leqslant \|\left(I - \eta E[f(\alpha|\mathbf{x})\mathbf{x}\mathbf{x}^{\mathrm{T}}]\right)_{J_{q+1}^{(0)}, I_{q+1}^{(0)}}(\boldsymbol{\theta}_q^{(0)} - \boldsymbol{\theta}_0)\| \\
&\leqslant \max\{|1 - \eta C_1|, |1 - \eta C_2|\}\|\boldsymbol{\theta}_q^{(0)} - \boldsymbol{\theta}_0\|,
\end{aligned} \tag{A5}$$

where the last step is based on assumption (A1).

Finally, we have

$$\begin{aligned}
&\|\mathbf{g}^{(0)}(\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}}\| \\
&= \|\left(\nabla\mathcal{L}_1(\boldsymbol{\theta}_0) - \nabla\mathcal{L}_1(\boldsymbol{\theta}_0^{(0)}) + \nabla\mathcal{L}_N(\boldsymbol{\theta}_0^{(0)})\right)_{I_{q+1}^{(0)}}\| \\
&\leqslant \|\left(\nabla\mathcal{L}_1(\boldsymbol{\theta}_0) - \nabla\mathcal{L}_1(\boldsymbol{\theta}_0^{(0)}) + \nabla\mathcal{L}_N(\boldsymbol{\theta}_0^{(0)}) - \nabla\mathcal{L}_N(\boldsymbol{\theta}_0)\right)_{I_{q+1}^{(0)}}\| + \|\nabla\mathcal{L}_N(\boldsymbol{\theta}_0)_{I_{q+1}^{(0)}}\| \\
&\leqslant \|\nabla\mathcal{L}_1(\boldsymbol{\theta}_0) - \nabla\mathcal{L}_1(\boldsymbol{\theta}_0^{(0)}) - E(\nabla\mathcal{L}_1(\boldsymbol{\theta}_0)) + E(\nabla\mathcal{L}_1(\boldsymbol{\theta}_0^{(0)}))\|_\infty\sqrt{|I_{q+1}^{(0)}|} \\
&\quad + \|\nabla\mathcal{L}_N(\boldsymbol{\theta}_0) - \nabla\mathcal{L}_N(\boldsymbol{\theta}_0^{(0)}) - E(\nabla\mathcal{L}_N(\boldsymbol{\theta}_0)) + E(\nabla\mathcal{L}_N(\boldsymbol{\theta}_0^{(0)}))\|_\infty\sqrt{|I_{q+1}^{(0)}|} \\
&\quad + \|\nabla\mathcal{L}_N(\boldsymbol{\theta}_0)\|_\infty\sqrt{|I_{q+1}^{(0)}|}.
\end{aligned} \tag{A6}$$

Using Lemma 1 in [5], with probability at least $1 - (d \vee n)^{-C}$,

$$\begin{aligned}
&\|\nabla\mathcal{L}_1(\boldsymbol{\theta}_0) - \nabla\mathcal{L}_1(\boldsymbol{\theta}_0^{(0)}) - E(\nabla\mathcal{L}_1(\boldsymbol{\theta}_0)) + E(\nabla\mathcal{L}_1(\boldsymbol{\theta}_0^{(0)}))\|_\infty \\
&\leqslant C\left(s^{1/4}\sqrt{c_n e_0^{(0)}}\sqrt{\frac{s\log(d\vee n)}{n}} + c_n\frac{s\log(d\vee n)}{n}\right),
\end{aligned} \tag{A7}$$

$$\begin{aligned}
&\|\nabla\mathcal{L}_N(\boldsymbol{\theta}_0) - \nabla\mathcal{L}_N(\boldsymbol{\theta}_0^{(0)}) - E(\nabla\mathcal{L}_N(\boldsymbol{\theta}_0)) + E(\nabla\mathcal{L}_N(\boldsymbol{\theta}_0^{(0)}))\|_\infty \\
&\leqslant C\left(s^{1/4}\sqrt{c_n e_0^{(0)}}\sqrt{\frac{s\log(d\vee n)}{N}} + c_n\frac{s\log(d\vee n)}{N}\right).
\end{aligned} \tag{A8}$$

By sub-Gaussianity,

$$\|\nabla\mathcal{L}_N(\boldsymbol{\theta}_0)\|_\infty \leqslant C\sqrt{\frac{log(d\vee n)}{N}}. \tag{A9}$$

Combining the bounds (A1)–(A9), we conclude that

$$\begin{aligned}
&\|\boldsymbol{\theta}_{q+1}^{(0)} - \boldsymbol{\theta}_0\| \\
&\leqslant \left(1 + \sqrt{\tfrac{s_0}{s}}\right)\max\{|1 - \eta C_1|, |1 - \eta C_2|\}\|\boldsymbol{\theta}_q^{(0)} - \boldsymbol{\theta}_0\| \\
&\quad + C\eta\left(s^{1/4}\sqrt{c_n e_q^{(0)}}\sqrt{\frac{s\log(d\vee n)}{n}} + c_n\frac{s\log(d\vee n)}{n}\right)\sqrt{|I_{q+1}^{(0)}|} \\
&\quad + C\eta\left(s^{1/4}\sqrt{c_n e_0^{(0)}}\sqrt{\frac{s\log(d\vee n)}{n}} + c_n\frac{s\log(d\vee n)}{n}\right)\sqrt{|I_{q+1}^{(0)}|} \\
&\quad + C\eta\sqrt{\tfrac{log(d\vee n)}{N}}\sqrt{|I_{q+1}^{(0)}|}.
\end{aligned} \tag{A10}$$

By our choice of $\eta = \frac{2}{C_1+C_2}$ which makes $\left(1 + \sqrt{s_0/s}\right)\max\{|1 - \eta C_1|, |1 - \eta C_2|\} = (1 + \sqrt{s_0/s})\frac{C_2 - C_1}{C_2 + C_1} = 1 - \delta$, we can reformulate (A10) as

$$\begin{aligned}
\Delta_{q+1}^{(0)} &\leqslant (1 - \delta)e_q^{(0)} + a_n\sqrt{e_q^{(0)}} + b_{N,n} \\
&\leqslant (1 - \delta/2)e_q^{(0)} + \tfrac{1}{2\delta}a_n^2 + b_{N,n} = e_{q+1}^{(0)},
\end{aligned}$$

where we used the Cauchy–Schwarz inequality in the second inequality. This finishes the induction step.

Using the iterative definition of $e_q^{(0)}$, we have

$$e_q^{(0)} = \left(1 - \frac{\delta}{2}\right)^q e_0^{(0)} + \frac{2}{\delta}\left(\frac{1}{2\delta}a_n^2 + b_{N,n}\right).$$

Hence, after at least $Q_1 = C\log\left(\frac{N\|\boldsymbol{\theta}_0^{(0)} - \boldsymbol{\theta}_0\|}{s\log(d \vee n)}\right)$ iterations, we have $e_{Q_1}^{(0)} = O(a_n^2 + b_{N,n})$. We also note that

$$\begin{aligned} & a_n^2 + b_{N,n} \\ &= Cs^{\frac{5}{2}} c_n \frac{\log(d \vee n)}{n} + Cs^{\frac{1}{2}}\sqrt{\frac{\log(d \vee n)}{N}} + Cs^{\frac{3}{2}} c_n \frac{\log(d \vee n)}{n} + Cs^{\frac{5}{4}}\sqrt{c_n}\sqrt{\frac{\log(d \vee n)}{n}}\sqrt{e_0^{(0)}} \\ &\leqslant Cs^{\frac{1}{2}}\sqrt{\frac{\log(d \vee n)}{N}} + Cs^{\frac{3}{2}}\frac{\log(d \vee n)}{n^{\frac{3}{4}}}. \end{aligned}$$

For other stages, we similarly have

$$\begin{aligned} e_0^{(t)} &= \left(1 - \tfrac{\delta}{2}\right)^Q e_0^{(t-1)} + Cs^{\frac{5}{4}}\frac{\log^{\frac{3}{4}}(d \vee n)}{\sqrt{n}}\sqrt{e_0^{(t-1)}} + Cs^{\frac{5}{2}}\frac{\log^{\frac{3}{2}}(d \vee n)}{n} + Cs^{\frac{1}{2}}\sqrt{\frac{\log(d \vee n)}{N}} \\ &\leqslant Cs^{\frac{5}{4}}\frac{\log^{\frac{3}{4}}(d \vee n)}{\sqrt{n}}\sqrt{e_0^{(t-1)}} + Cs^{\frac{5}{2}}\frac{\log^{\frac{3}{2}}(d \vee n)}{n} + Cs^{\frac{1}{2}}\sqrt{\frac{\log(d \vee n)}{N}} \\ &\leqslant \tfrac{1}{2}e_0^{(t-1)} + Cs^{\frac{5}{2}}\frac{\log^{\frac{3}{2}}(d \vee n)}{n} + Cs^{\frac{1}{2}}\sqrt{\frac{\log(d \vee n)}{N}} \\ &\leqslant \left(\frac{1}{2}\right)^t e_0^{(0)} + Cs^{\frac{5}{2}}\frac{\log^{\frac{3}{2}}(d \vee n)}{n} + Cs^{\frac{1}{2}}\sqrt{\frac{\log(d \vee n)}{N}}. \end{aligned} \tag{A11}$$

Thus, with $T \geqslant Clog\left(\frac{N\|\boldsymbol{\theta}_0^{(0)} - \boldsymbol{\theta}_0\|}{s\log(d \vee n)}\right)$ iterations, we have the desired bound. □

## References


1. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning: With Applications in R*; Springer: New York, NY, USA, 2013.
2. Tibshirani, R. The lasso method for variable selection in the Cox model. *Stat. Med.* **1997**, *16*, 385–395. [CrossRef]
3. Shen, X.; Pan, W.; Zhu, Y. Likelihood-based selection and sharp parameter estimation. *J. Am. Stat. Assoc.* **2012**, *107*, 223–232. [CrossRef] [PubMed]
4. Jain, P.; Tewari, A.; Kar, P. On iterative hard thresholding methods for high-dimensional M-estimation. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
5. Wang, Y.; Lu, W.; Lian, H. Best subset selection for high-dimensional non-smooth models using iterative hard thresholding. *Inf. Sci.* **2023**, *625*, 36–48. [CrossRef]
6. Koenker, R.; Bassett, G., Jr. Regression quantiles. *Econom. J. Econom. Soc.* **1978**, *1*, 33–50. [CrossRef]
7. Koenker, R. *Quantile Regression*; Econometric Society Monographs, Cambridge University Press: Cambridge, UK, 2005; p. xv, 349p.
8. Portnoy, S.; Koenker, R. The Gaussian hare and the Laplacian tortoise: Computability of squared-error versus absolute-error estimators. *Stat. Sci.* **1997**, *12*, 279–300. [CrossRef]
9. Koenker, R.; Hallock, K.F. Quantile regression. *J. Econ. Perspect.* **2001**, *15*, 143–156. [CrossRef]
10. Koenker, R.; Geling, O. Reappraising medfly longevity: A quantile regression survival analysis. *J. Am. Stat. Assoc.* **2001**, *96*, 458–468. [CrossRef]
11. Cade, B.S.; Noon, B.R. A gentle introduction to quantile regression for ecologists. *Front. Ecol. Environ.* **2003**, *1*, 412–420. [CrossRef]
12. Rosenblatt, J.D.; Nadler, B. On the optimality of averaging in distributed statistical learning. *Inf. Inference* **2016**, *5*, 379–404. [CrossRef]
13. Lin, S.; Guo, X.; Zhou, D. Distributed learning with regularized least squares. *J. Mach. Learn. Res.* **2017**, *18*, 1–31.
14. Lian, H.; Fan, Z. Divide-and-conquer for debiased l1-norm support vector machine in ultra-high dimensions. *J. Mach. Learn. Res.* **2018**, *18*, 1–26.
15. Huang, C.; Huo, X. A distributed one-step estimator. *Math. Program.* **2019**, *174*, 41–76. [CrossRef]

16. Shamir, O.; Srebro, N.; Zhang, T. Communication-efficient distributed optimization using an approximate Newton-type method. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; Volume 32, pp. 1000–1008.
17. Jordan, M.I.; Lee, J.D.; Yang, Y. Communication-efficient distributed statistical inference. *J. Am. Stat. Assoc.* **2018**, *114*, 668–681. [CrossRef]
18. Zou, H.; Yuan, M. Composite quantile regression and the oracle model selection theory. *Ann. Stat.* **2008**, *36*, 1108–1126. [CrossRef]
19. Wang, H.J.; Li, D. Estimation of extreme conditional quantiles through power transformation. *J. Am. Stat. Assoc.* **2013**, *108*, 1062–1074. [CrossRef]
20. Shi, W.; Ling, Q.; Wu, G.; Yin, W. Extra: An exact first-order algorithm for decentralized consensus optimization. *Siam J. Optim.* **2015**, *25*, 944–966. [CrossRef]
21. Ling, Q.; Shi, W.; Wu, G.; Ribeiro, A. DLM: Decentralized linearized alternating direction method of multipliers. *IEEE Trans. Signal Process.* **2015**, *63*, 4051–4064. [CrossRef]
22. Ma, B.; Feng, Y.; Chen, G.; Li, C.; Xia, Y. Federated adaptive reweighting for medical image classification. *Pattern Recognit.* **2023**, *144*, 109880. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.