mathematics

MDPI

Article

ADMM-Based Differential Privacy Learning for Penalized Quantile Regression on Distributed Functional Data

Xingcai Zhou * and Yu Xiang

School of Statistics and Data Science, Nanjing Audit University, Nanjing 211085, China
* Correspondence: xczhou@nau.edu.cn

Abstract: Alternating Direction Method of Multipliers (ADMM) is a widely used machine learning tool in distributed environments. In the paper, we propose an ADMM-based differential privacy learning algorithm (FDP-ADMM) on penalized quantile regression for distributed functional data. The FDP-ADMM algorithm can resist adversary attacks to avoid the possible privacy leakage in distributed networks, which is designed by functional principal analysis, an approximate augmented Lagrange function, ADMM algorithm, and privacy policy via Gaussian mechanism with time-varying variance. It is also a noise-resilient, convergent, and computationally effective distributed learning algorithm, even if for high privacy protection. The theoretical analysis on privacy and convergence guarantees is derived and offers a privacy–utility trade-off: a weaker privacy guarantee would result in better utility. The evaluations on simulation-distributed functional datasets have demonstrated the effectiveness of the FDP-ADMM algorithm even if under high privacy guarantee.

Keywords: distributed machine learning; ADMM; quantile regression; functional principal component analysis; differential privacy

MSC: 62G05; 62G08; 62G20

Citation: Zhou, X.; Xiang, Y. ADMM-Based Differential Privacy Learning for Penalized Quantile Regression on Distributed Functional Data. *Mathematics* **2022**, *10*, 2954. https://doi.org/10.3390/math10162954

Academic Editor: Chao Huang

Received: 25 July 2022
Accepted: 12 August 2022
Published: 16 August 2022

1. Introduction

Machine learning is becoming more and more common in statistical modeling and data analysis, along with the increasing concerns about the privacy disclosure of data. Therefore, we urgently need to develop algorithms which can provide privacy protection for personal data. In turn, the demand for data privacy protection has stimulated the establishment of formal standards on data privacy and the development of privacy framework. Among them, differential privacy (DP) [1,2] is the most widely discussed and developed technique in theory [3–6], and the feasibility of adopting these theories is shown among others by [7–9]. The framework of DP makes it convenient for us to construct privacy protection algorithms. However, these privacy protection algorithms may also need to pay a price of sacrificing the rate of convergence in statistical accuracy. Therefore, we need to develop differential privacy distributed learning algorithms that do not sacrifice statistical accuracy as much as possible for large-scale distributed data.

Distributed machine learning can disassemble the original huge training task into multiple sub-tasks, that is, transforming large-scale learning that one machine can not afford during collaborative learning with multiple machines. Recently, ref. [10] gave a sparse distributed learning solution for high-dimensional problems; ref. [11] proposed a distributed learning algorithm which segments features in a high-dimensional sparse additive model and proved the consistency of the sparse patterns for each additive component; ref. [12] provided a more flexible framework using the communication-efficient surrogate likelihood (CSL) procedure, which can solve different settings such as M-estimation for low- and high-dimensional problems and Bayesian inference. Ref. [13] extended the CSL method to distributed quantile regression and then established some statistical properties

under quantile loss, which does not satisfy the smoothness condition of CSL method. In distributed learning, each sub-task is executed separately on an independent machine. It allows each machine to complete a collective learning objective, which is usually a standardized empirical risk minimization problem. The individual data that do not need to be disclosed will be calculated in a local iterative algorithm, and the parameters will be transferred between the central machine and each local machine. At present, the widely used algorithms for the decentralized distributed learning problems mainly include subgradient-based algorithms [14,15], alternating direction method of multipliers(ADMM) [16–19], and the combination algorithms of these methods [20]. Ref. [21] proved that ADMM-based algorithms converge at the rate of $O(1/L)$, while subgradient-based algorithms usually converge at the rate of $O(1/\sqrt{L})$, where L is the number of iterations. Therefore, in this paper, we adopt an ADMM-based distributed learning algorithm against privacy disclosure and keep a statistical guarantee.

We know that sensitive individual information may be leaked in optimization algorithms as a result of sharing information such as parameters and/or gradients of the model between machines, as presented in [22,23]. The same problem exists in our ADMM-based distributed algorithm: how to avoid privacy leakage. So, we need to protect privacy via a DP mechanism while maintaining statistical accuracy in our distributed learning. Ref. [24] studied a class of regularized empirical risk minimization machine learning problems via ADMM and proposed the dual variable perturbation and the primal variable perturbation methods for dynamic differential privacy. Ref. [25] proposed a privacy-preserving cooperative learning scheme, where users are allowed to train independently using their own data and only share some updated model parameters. They used an asynchronous ADMM approach to accelerate learning. In addition, their algorithm integrates secure computing and distributed noise generation to ensure the confidentiality of shared parameters during the asynchronous ADMM algorithm process. Ref. [26] applied a new privacy-preserving distributed machine learning (PS-ADMM) algorithm based on stochastic ADMM, which provides a privacy guarantee by perturbing the gradient and has a low computational cost. In the paper, we focus on a functional linear regression model for functional data analysis via ADMM-based distributing learning to keep DP and statistical efficiency.

Functional data are natural generalizations of multivariate data from finite dimensional to infinite dimensional which are obtained by observing a number of subjects over time, space, and other continua. In practice, functional data are frequently recorded by an instrument, which involves a large number of repeated measurements per subject. They can be curves, surfaces, images, or other complex objects; see some real data sets in the monographs [27–29]. All in all, a functional datum is not a single observation but rather a set of measurements along a continuum; taken together, they are regarded as a single entity, curve, or image. In recent decades, functional data analysis has drawn considerable attention because advanced technology makes functional data easier to collect in applied fields such as medical studies, speech recognition, biological growth, climatology, online auctions, and so on. Time series data are treated as multivariate data because they are given as a finite discrete time series. In addition, longitudinal data, which are often observed in biomedical follow-up studies, are strongly linked with functional data; however, their use often involves several (few) measurements per subject taken intermittently at different time points for different subjects. Therefore, functional data and longitudinal data are also intrinsically different. In addition, some classic multivariate data analysis tools, applied to time series and longitudinal data analysis, cannot be directly applied to functional data analysis because they ignore the fact that the underlying object of the measurements of a subject is a function such as curve or surface. We know that functional data are intrinsically infinite dimensional, and our analysis methods cannot be based on the assumption that the values observed at different times for a single subject are independent because of the intra-observation dependence. The high intrinsic dimensionality and the intra-observation dependence of functional data pose challenges both for theory and computation.

Recently, various approaches and statistical models for the analysis of functional data have been developed. For an introduction and summary, see [27–30]. Ref. [31] firstly proposed a linear regression model and analyzed the effects of functional independent variables on the scalar response variables through the inner product of functional independent variables and unknown nonparametric coefficient function. Ref. [32] gave a functional linear semiparametric quantile regression model, which has been used to analyze ADHD-200 patients data. Ref. [33] studied the estimation problem of a functional partial quantile regression model, and proved the asymptotic normality of the finite dimensional parameter estimation. The conventional method of functional data analysis is principal component analysis (PCA), such as [34,35]. Ref. [35] gave the optimal convergence rates of PCA. We will consider functional principal components analysis (FPCA) for our functional linear regression model and investigate the distributed learning with privacy.

In this paper, we propose a new ADMM-based distributed learning algorithm with differential privacy to handle large amounts of functional data. We call it the FDP-ADMM algorithm. Our proposed FDP-ADMM algorithm has good properties such as a faster rate of convergence, lower communication and computation costs, and better utility–privacy tradeoffs. In the FDP-ADMM algorithm, we consider a more robust quantile loss function, combine an approximate augmented Lagrange function, and integrate time-varying Gaussian noise into local learning on each machine. These techniques allow the FDP-ADMM algorithm to be adversarial while protecting privacy.

The main contributions of this paper are summarized as follows:

- We propose a distributed learning algorithm (FDP-ADMM) that can process large-scale distributed functional data and protect privacy. For the large-scale functional data, we adopt functional principal component analysis to reduce the dimensions of the data, improve the quality of data information, and promote the efficiency of functional data analysis using distributed learning.
- We introduce a quantile loss function for functional linear model such that our models are adaptive to heavy-tail data or outliers. Thus, our ADMM-based distributed learning algorithm is more robust compared with ordinary least square procedure.
- The privacy and theoretical convergence guarantees of the FDP-ADMM algorithm are derived, and a privacy–utility trade-off is demonstrated: a weaker privacy guarantee would result in better utility.
- We conduct numerical experiments to illustrate the effectiveness of FDP-ADMM in the framework of distributed learning. The results of experiments are consistent with our theoretical analysis.

The rest of this paper, is organized as follows. In Section 2, we state our problem formulation by introducing the functional linear regression model, the penalized quantile regression, the ADMM algorithm, and DP. In Section 3, we propose an ADMM-based distributed learning algorithm with privacy protection for distributed functional data analysis. In Section 4, we present the utility analysis of our algorithm, FDP-ADMM, including the convergence and privacy guarantee. In Section 5, we give some numerical experiments to verify our theoretical results. Some conclusions are given in Section 6. The proofs of the main results are collected in Appendix A.

Notations

For any positive integer n, we define $[n] := \{1, 2, \ldots, n\}$. $\|\cdot\|$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ are denoted as the Euclidean norm, ℓ_2-norm, and ℓ_∞-norm, respectively. $\rho_\tau(u) = u(\tau - I\{u \leq 0\})$ is the quantile loss function for a scalar $u \in \mathbb{R}$. and $I\{\cdot\}$ denotes the indicator function. For a vector $\boldsymbol{u} = (u_1, \cdots, u_n)^T$, we define $\rho_\tau(\boldsymbol{u}) = \sum_{i=1}^n \rho_\tau(u_i)$. Throughout this paper, the constant C denotes positive constant whose value may change from line to line. For any function f and a positive function ϕ, $f \asymp \phi$ means $a\phi < f < b\phi$ for some positive constants a and b.

2. Problem Formulation

In this section, we present the functional data model, quantile regression, ADMM algorithm, and difference privacy mechanisms to be studied.

2.1. Functional Data Analysis

Functional data consist of functions that are basically smooth but are usually corrupted with noise, such as curves, images, and so on. For simplicity, we assume the functional predictor $X(t)$ on the finite time interval $I = [0, T]$. Let $\{x_i(t) : t \in I\}_{1 \le i \le n}$ be observation variables; that is, for each $t \in I$, there exists an observed value $x_i(t) \in \mathbb{R}$. A typical functional data set is:

$$\left\{x_i(t_{j,i}) \in \mathbb{R} : t_{j,i} \in I, 1 \le i \le n, 1 \le j \le J_i\right\},$$

where J_i is the observation number, and n is the number of individuals. If J_i is small, then the data are called sparse; otherwise they are called dense. FDA pays attention to the shape of the potential function or curve of the data via some statistical models and estimation procedures.

Functional linear regression is a standard method in functional data analysis for incorporating functional predictors, which focuses on modeling the relationship between a functional or continuous response Y and a functional predictor $X(t)$, in which t varies in a compact set I. It usually has the form:

$$Y = \beta_0 + \int_I \beta(t)X(t)dt + \epsilon,$$

where β_0 is a intercept term, ϵ is the random noise independent of $X(t)$, and $\beta(t)$ is an unknown function of interest. Without loss of generality, we assume $E(Y) = 0$ and $E(X(t)) = 0$. Based on data $\{(X_i(t), Y_i), i = 1, \cdots, n\}$, the model becomes:

$$Y_i = \int_I \beta(t)X_i(t)dt + \epsilon. \tag{1}$$

FPCA is commonly used for analyzing such models (1) with the purpose of dimension reduction. The main idea is to summarize the data variation and information via some dimensional loadings. Dimension reduction in FPCA is performed through an expansion of basis, which consists of the eigenfunctions formed by the covariance operator $\Sigma(\cdot, \cdot)$ of the process $X(t) : t \in I$. By Mercer's theorem, the spectral decomposition is

$$\Sigma(s, t) = Cov(X(s), X(t)) = \sum_{k=1}^{\infty} \lambda_k \phi_k(s)\phi_k(t),$$

where $s, t \in I$, λ_k are the ordered eigenvalues such that $\lambda_1 \ge \lambda_2 \ge \ldots$, and the function ϕ_k forms the orthogonal basis corresponding to λ_k, $k = 1, 2, \cdots$.

By Karhunen and Loève [36], in the classical functional principal component analysis, the ith random curve $X(t)$ and functional coefficient $\beta(t)$ can be expressed as:

$$X_i(t) = \sum_{k=1}^{\infty} A_{ik}\phi_k(t), \;\; \beta(t) = \sum_{k=1}^{\infty} w_k\phi_k(t),$$

where the coefficients $A_{ik} = \int_I X_i(t)\phi_k(t)dt$ and $w_k = \int_I \beta(t)\phi_k(t)dt$ are the functional principal components. In addition, $E(A_{ik}) = 0$ and var $(A_{ik}) = \lambda_k$ for $k = 1, 2, \cdots$. We have the top K of λ_k and taking their corresponding $\phi_k(t)$, we have an approximation to $X_i(t)$ truncated as:

$$X_i^K(t) = \sum_{k=1}^{K} A_{ik}\phi_k(t).$$

Then we consider the functional principal component regression into the functional linear model:

$$Y_i = \int_I X_i(t)\beta(t)\mathrm{d}t + \epsilon_i \approx \sum_{k=1}^{K} A_{ik}w_k + \epsilon, \; for \; i = 1, \cdots, n, \tag{2}$$

We regard w_k $(k = 1, \cdots, K)$ as unknown parameters. We can select a proper K FPCA basis to represent the functional data $X_i(t)$; that is, the most important information of data can be refined by FPCA.

2.2. Quantile Regression with Penalties

We have i.i.d observations $(\boldsymbol{A}_i, y_i), i = 1, 2, \cdots, n$, with $\boldsymbol{A}_i = (A_{i1}, \cdots, A_{iK})$. Let $Q_{y_i|\boldsymbol{A}_i}(\tau) = \boldsymbol{A}_i\boldsymbol{w}_\tau$ be the conditional quantile of y_i on $\boldsymbol{A}_i \in \mathbb{R}^K$, for a give the quantile level τth, $\tau \in$(0,1). Let $\boldsymbol{w}_\tau = (w_{1,\tau}, \cdots, w_{K,\tau})^T$. The quantile regression estimate of $\boldsymbol{w}_\tau$ is defined as:

$$\hat{\boldsymbol{w}}_\tau = \arg\min_{\boldsymbol{w}\in\mathbb{R}^K} \rho_\tau(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}), \tag{3}$$

where $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots \boldsymbol{A}_n]^T \in \mathbb{R}^{n\times K}$ is a matrix, $\rho_\tau(u) = u(\tau - I\{u \le 0\})$ is the quantile loss function for a scalar $u \in \mathbb{R}$ and $I\{\cdot\}$ denotes the indicator function.

Penalized quantile regression (PQR) is formulated as

$$\min_{\boldsymbol{w}} \rho_\tau(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}) + P_\lambda(\boldsymbol{w}), \tag{4}$$

where $P_\lambda(\cdot)$ is a penalty, such as lasso penalty [37],

$$P_\lambda(\boldsymbol{w}) = \lambda\|\boldsymbol{w}\|_1; \tag{5}$$

or elastic net [38],

$$P_\lambda(\boldsymbol{w}) = \lambda\Big(\lambda_2\|\boldsymbol{w}\|_2^2 + \lambda_1\|\boldsymbol{w}\|_1\Big), \lambda_1, \lambda_2 \ge 0. \tag{6}$$

Penalized QR, such as (5) and (6), leads to biased estimators. To obtain an unbiased estimator, refs. [39,40] proposed a non-convex penalty, for instance, the MCP penalty [39], or the SCAD penalty [40].

Our learning empirical loss is:

$$\widehat{L}(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - \boldsymbol{A}_i\boldsymbol{w}) \tag{7}$$

based on all data coming from all machines. As the data are distributed on local machines, it is difficult to collect all data into one machine, and additionally, there are privacy issues. Therefore, we apply the technique of distributed learning via the ADMM algorithm for the following distributed empirical loss (11). ADMM is a computational framework to solving optimization problems.

2.3. ADMM Algorithm

ADMM algorithm was first proposed by [41,42] in 1975 and 1976, respectively. Then, ADMM was reviewed and proven to be suitable for large-scale distributed optimization by [16]. In this section, we give the basic ADMM formulation.

Assume that our optimization problem is expressed as:

$$\min_{\boldsymbol{x},\boldsymbol{z}}\{f(\boldsymbol{x}) + g(\boldsymbol{z})\} \quad \text{s.t. } B\boldsymbol{x} + C\boldsymbol{z} = \boldsymbol{d} \tag{8}$$

where $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{z} \in \mathbb{R}^s$, matrices $B \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{m\times s}$, vector $\boldsymbol{d} \in \mathbb{R}^m$, and functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^s \to \mathbb{R}$. $\boldsymbol{x}$ and $\boldsymbol{z}$ are the variables needing to be optimized. The

optimization problem (8) consists of two parts: $f(\boldsymbol{x})$ related to variable x and $g(\boldsymbol{z})$ related to variable z. This structure can easily be dealt with via ADMM as follows: First, we have the augmented Lagrangian function:

$$L_\rho((\boldsymbol{x},\boldsymbol{z}),\boldsymbol{u}) := f(\boldsymbol{x}) + g(\boldsymbol{z}) + \boldsymbol{u}^T(B\boldsymbol{x} + C\boldsymbol{z} - \boldsymbol{d}) + \frac{\rho}{2}\|B\boldsymbol{x} + C\boldsymbol{z} - \boldsymbol{d}\|_2^2, \tag{9}$$

where $\boldsymbol{u}$ is the dual variable (or called the Lagrange multiplier), and $\rho > 0$ is a penalty parameter. The name 'augmented' in the L_ρ refers to the quadratic penalty term $\frac{\rho}{2}\|B\boldsymbol{x} + C\boldsymbol{z} - \boldsymbol{d}\|_2^2$, which is added for better convergence properties of algorithm. Then, the ADMM iterative solution of the optimization problem (9) is:

$$\begin{aligned} \boldsymbol{x}^{l+1} &:= \arg\min_{\boldsymbol{x}} L_\rho\left(\boldsymbol{x}, \boldsymbol{z}^l, \boldsymbol{u}^l\right), \\ \boldsymbol{z}^{l+1} &:= \arg\min_{\boldsymbol{z}} L_\rho\left(\boldsymbol{x}^{l+1}, \boldsymbol{z}, \boldsymbol{u}^l\right), \\ \boldsymbol{u}^{l+1} &:= \boldsymbol{u}^l + \rho\left(B\boldsymbol{x}^{l+1} + C\boldsymbol{z}^{l+1} - \boldsymbol{d}\right). \end{aligned} \tag{10}$$

The 'multiplier method' in ADMM refers to a dual ascent using augmented Lagrange function (with quadratic penalty term), and the 'alternating direction' refers to variables $\boldsymbol{x}$ and $\boldsymbol{z}$ be updated alternately. For more theories and applications about ADMM, refer to [16].

2.4. Differential Privacy

Differential privacy technology was originally designed to confront differential attacks problem. The traditional protecting method is to anonymize or encrypt to the datasets. However, some individual information can still be recovered from these anonymous data, based on certain algorithms, such as the recommendation algorithm. Therefore, Ref. [3] proposed the mechanism of differential privacy to protect privacy, which adds a designed noise in the algorithm so that attackers can not recover data information. Moreover, it has been proven that as long as the noise satisfies the differential privacy mechanism, no matter how much prior information the attacker has, the anonymous data cannot be reconstructed.

This paper mainly studies differential privacy for ADMM against adversarial attacks. Intuitively speaking, if an adversary can not tell whether a individual datum x belongs to the special data set X or not, when we output results from algorithm $\mathcal{M}(\mathcal{X})$. We call that DP. Now, we give the definition of the (ϵ, δ)-differential privacy from Dwork's work [2].

Definition 1 ((ϵ, δ)-Differential Privacy)**.** *A randomized algorithm $\mathcal{M} : \mathcal{X}^n \to R$ is (ϵ, δ)-differential private if for any two adjacent datasets (differing in only one tuple) X, $X' \in \mathcal{X}^n$, and for any measurable output subset $\mathcal{S} \subseteq$ range$(\mathcal{M})$:*

$$\mathbb{P}[\mathcal{M}(X) \in \mathcal{S}] \leq e^\epsilon \cdot \mathbb{P}\left[\mathcal{M}\left(X'\right) \in \mathcal{S}\right] + \delta,$$

where probability measure $\mathbb{P}$, which is bounded, only depends on the randomness of algorithm $\mathcal{M}$.

In Definition 1, δ and ϵ measure the protection strength of the privacy. It implies that a smaller δ or a smaller ϵ gives better privacy protection. The Laplace and Gaussian mechanisms are two typical methods that are widely used in (ϵ, δ)-differential privacy. They offer calibrated noise sampled from Laplace or Gaussian distribution, and add this noise into the algorithm. Now we consider a class of deferentially private algorithms via compositions, termed 'k-fold adaptive composition' in the literature. The advanced composition stated below, where the auxiliary inputs of the k-th algorithm are the outputs of all previous algorithms, shows how privacy parameters degrade as private algorithms are composited.

Lemma 1 (Theorem 4 in [43] (Advanced Composition)). *Let $\epsilon, \delta \geq 0$. The class of (ϵ, δ)-differentially private algorithms satisfies (ϵ', δ)-differential privacy under k-fold adaptive composition, where $\epsilon' = c_0\sqrt{k}\epsilon$ for some constant c_0.*

3. Distributed Learning with DP for Functional Data via ADMM

In this section, we propose the ADMM-based distributed learning algorithm with DP for functional data, which is called FDP-ADMM. We will transfer the functional regression model (1) into a linear regression model (2) by using FPCA. Because quantile regression has better performance in estimation and prediction for non-Gaussian distribution error, such as heavy-tailed distribution or outliers, we consider quantile regression for the models (1) and (2). First, based on data $\{(X_i(t), y_i), i = 1, \cdots, n\}$, the functional quantile linear regression model we consider is

$$y_i = \int_I \beta_\tau(t) X_i(t) dt + \epsilon_{\tau,i},$$

where $\tau \in (0, 1)$ is a given quantile level, and $\epsilon_{\tau,i}$ are random errors. Without loss of generality, we assume that the τth quantile of $\epsilon_{\tau,i}$ is equal to zero. The model can be written as $Q_{y_i|X_i}(\tau) = \int_I \beta_\tau(t) X_i(t) dt$. Our goal is to learn the functional coefficient $\beta_\tau(t)$ by

$$\min_{\beta_\tau(t)} \sum_{i=1}^{n} \rho_\tau \left(y_i - \int_I \beta_\tau(t) X_i(t) dt \right).$$

The problem is difficult because of the term $\int_I \beta_\tau(t) X_i(t) dt$. By the FPCA introduced in Section 2.1, we have the model (2):

$$y_i = \int_I X_i(t) \beta_\tau(t) \mathrm{d}t + \epsilon_{\tau,i} \approx \sum_{k=1}^{K} A_{ik} w_{\tau,k} + \epsilon_{\tau,i}, \ for\ i = 1, \cdots, n.$$

That is, functional quantile linear regression is transformed as an ordinary quantile linear regression. We suppress the dependency of $w_{\tau,k}$ on τ for simplicity. Then, for the quantile linear regression, we propose penalized quantile regression learning, which is formulated as

$$\min_{\boldsymbol{w}} \rho_\tau(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}) + P_\lambda(\boldsymbol{w}).$$

It has been introduced in Section 2.2.

However, our dataset $\{(X_i(t), y_i), i = 1, \cdots, n\}$ cannot be collected on one machine, but distributed over M machines. That is, we have the distributed data $\{(X_{i,j}(t), y_{i,j}), i = 1, \cdots, M, j = 1, \cdots, m_i\}$, where M is the number of worker machines, and m_i is the size of sample on the ith machine. Thus, based on FPCA, $\{(X_{i,j}(t), y_{i,j}), i = 1, \cdots, M, j = 1, \cdots, m_i\}$ is transformed as $\{(y_{ij}, \boldsymbol{A}_{i,j}), i = 1, \cdots, M, j = 1, \cdots, m_i\}$, where $\boldsymbol{A}_{i,j} = (A_{i,j1}, \cdots, A_{i,jK})$ is the score of the jth sample on the ith worker machine. So, based on the distributed data $\{(y_{ij}, \boldsymbol{A}_{i,j}), i = 1, \cdots, M, j = 1, \cdots, m_i\}$ and the model (2), we have the following QR estimation in the distributed framework:

$$\hat{\boldsymbol{w}}_\tau = \operatorname{argmin} \sum_{i=1}^{M} \left(\sum_{j=1}^{m_i} \frac{1}{m_i} \rho_\tau \left(y_{ij} - \boldsymbol{A}_{i,j} \boldsymbol{w} \right) \right), \tag{11}$$

where $\rho_\tau(\cdot)$ is the loss function of quantile regression, and $\boldsymbol{w}$ is the unknown coefficient. Furthermore, we modify QR as penalized QR estimator for achieving faster shrinking, that is:

$$\hat{\boldsymbol{w}}_\tau = \operatorname{argmin} \sum_{i=1}^{M} \left(\sum_{j=1}^{m_i} \frac{1}{m_i} \rho_\tau \left(y_{ij} - \boldsymbol{A}_{i,j} \boldsymbol{w} \right) \right) + P_\lambda(\boldsymbol{w}). \tag{12}$$

Note that in (12), there exist two type of tuning parameters, K and λ, where K controls the number of scores to characterize the decomposition level of FPCA and λ controls the fitness of the model.

When facing big data, that is, when n is very large, it is hard for one machine to learn w in (12). So, it is necessary to distributed storage and learning. Next, we will demonstrate the ADMM-based distributed learning for penalized QR. We provide a sketch of our FDP-ADMM algorithm based on the distributed data.

3.1. ADMM-Based Distributed Learning Algorithm

Assume we have M machines, and the ith machine has m_i local data samples. Applying the ADMM algorithm, we re-formulate the problem (12) as:

$$\begin{aligned} \min_{\{\boldsymbol{w}_i\}_{i\in[M]}} & \sum_{i=1}^{M}\left(\sum_{j=1}^{m_i}\frac{1}{m_i}\rho_\tau\left(y_{ij}-\boldsymbol{A}_{i,j}\boldsymbol{w}_i\right)+\frac{\lambda}{M}P(\boldsymbol{w}_i)\right), \\ & \text{s.t. } \boldsymbol{w}_i=\boldsymbol{w}, i=1,\ldots,M, \end{aligned} \tag{13}$$

where $\boldsymbol{w}_i \in \mathbb{R}^K$ is the local model parameters, and $\boldsymbol{w} \in \mathbb{R}^K$ is the global ones. Then, the augmented Lagrangian function for the ith machine is:

$$\mathcal{L}_{\rho,i}(\boldsymbol{w}_i,\boldsymbol{w},\boldsymbol{\gamma}_i)=\sum_{j=1}^{m_i}\frac{1}{m_i}\rho_\tau\left(y_{ij}-\boldsymbol{A}_{i,j}\boldsymbol{w}_i\right)+\frac{\lambda}{M}P(\boldsymbol{w}_i)-\langle\boldsymbol{\gamma}_i,\boldsymbol{w}_i-\boldsymbol{w}\rangle+\frac{\rho}{2}\|\boldsymbol{w}_i-\boldsymbol{w}\|^2 \tag{14}$$

$$\text{s.t. } \boldsymbol{w}_i=\boldsymbol{w}, i=1,2,\ldots,M. \tag{15}$$

The objective (14) is decoupled and each worker only needs to minimize the subproblem based on its local data set. Constraints (15) enforce all the local models to consensus. It results in the following iteration:

$$\boldsymbol{w}_i^l=\underset{\boldsymbol{w}_i}{\operatorname{argmin}}\hat{\mathcal{L}}_{\rho,i}\left(\boldsymbol{w}_i,\boldsymbol{w}^{l-1},\boldsymbol{\gamma}_i^{l-1}\right), \tag{16}$$

$$\boldsymbol{w}^l=\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{w}_i^l-\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{\gamma}_i^{l-1}/\rho, \tag{17}$$

$$\boldsymbol{\gamma}_i^l=\boldsymbol{\gamma}_i^{l-1}-\rho\left(\boldsymbol{w}_i^l-\boldsymbol{w}^l\right). \tag{18}$$

Note that each machine transfers its $\left(\boldsymbol{w}_i^l,\boldsymbol{\gamma}_i^l\right)$ to a central machine. The central machine gathers them to update $\boldsymbol{w}^l$ and then broadcasts it to each machine. Details for the algorithm are present in Algorithm 1. Based on output $\boldsymbol{w}^L$, we obtain:

$$\hat{\beta}_{N-DP}(t)=\sum_{k=1}^{K}w_k^L\phi_k(t).$$

Algorithm 1: ADMM for PQR of Functional Data (F-ADMM)

Input: Initialize: $\boldsymbol{w}^0$, $\{\boldsymbol{w}_i^0\}_{i\in[M]}$, $\{\boldsymbol{\gamma}_i^0\}_{i\in[M]}$ and number of iteration L

for $l = 1, 2, \cdots, L$ **do**
Inherit parameters from the previous iteration

for *Worker machine* $i = 1, 2, \cdots, M$ **do**
Compute $\boldsymbol{w}_i^l$ using Equation (16);

Deliver $(\boldsymbol{w}_i^l, \boldsymbol{\gamma}_i^{l-1})$ to central machine.
end

Central machine computes $\boldsymbol{w}^l$ by Equation (17), then broadcasts it to all machines.

for *Worker machine* $i = 1, 2, \cdots, M$ **do**
Compute $\boldsymbol{\gamma}_i^l$ using Equation (18).

end
end

Output: $\boldsymbol{w}^L$.

3.2. ADMM-Based Distributed Learning with DP

For achieving faster optimization, we make use of the first-order approximation to the penalized objective function. Then, $\mathcal{L}_{\rho,i}(\boldsymbol{w}_i, \boldsymbol{w}, \boldsymbol{\gamma}_i)$ in (14) becomes:

$$\begin{aligned}\hat{\mathcal{L}}_{\rho,i}\left(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{l-1}, \boldsymbol{w}, \boldsymbol{\gamma}_i\right) &= \sum_{j=1}^{m_i} \frac{1}{m_i}\rho_\tau\left(y_{ij} - \boldsymbol{A}_{i,j}\tilde{\boldsymbol{w}}_i^{l-1}\right) + \frac{\lambda}{M}P\left(\tilde{\boldsymbol{w}}_i^{l-1}\right) \\ &\quad + \left\langle \sum_{j=1}^{m_i} \frac{1}{m_i}\rho'_\tau\left(y_{ij} - \boldsymbol{A}_{i,j}\tilde{\boldsymbol{w}}_i^{l-1}\right) + \frac{\lambda}{M}P'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right), \boldsymbol{w}_i - \tilde{\boldsymbol{w}}_i^{l-1} \right\rangle \\ &\quad - \langle \boldsymbol{\gamma}_i, \boldsymbol{w}_i - \boldsymbol{w}\rangle + \frac{\rho}{2}\|\boldsymbol{w}_i - \boldsymbol{w}\|^2 + \frac{\left\|\boldsymbol{w}_i - \tilde{\boldsymbol{w}}_i^{l-1}\right\|^2}{2\eta_i^l}, \end{aligned} \tag{19}$$

where $\eta_i^l \in \mathbb{R}$ is the time-varying step size which decreases as the iteration l grows, ρ'_τ is the subgradient of the quantile loss function, and P' is the subgradient of the penalty. So, we have the following optimization problem:

$$\min_{\boldsymbol{w}_i} \sum_{i=1}^{M} \hat{\mathcal{L}}_{\rho,i}\left(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{l-1}, \boldsymbol{w}, \boldsymbol{\gamma}_i\right) \quad \text{s.t. } \boldsymbol{w}_i = \boldsymbol{w}, i = 1, 2, \ldots, M. \tag{20}$$

Here, we give the ADMM-based distributed learning algorithm with DF (FDP-ADMM) as follows:

$$\boldsymbol{w}_i^l = \underset{\boldsymbol{w}_i}{\operatorname{argmin}} \hat{\mathcal{L}}_{\rho,i}\left(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{l-1}, \boldsymbol{w}^{l-1}, \boldsymbol{\gamma}_i^{l-1}\right), \tag{21}$$

$$\tilde{\boldsymbol{w}}_i^l = \boldsymbol{w}_i^l + \xi_i^l, \tag{22}$$

$$\boldsymbol{w}^l = \frac{1}{M}\sum_{i=1}^{M} \tilde{\boldsymbol{w}}_i^l - \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\gamma}_i^{l-1}/\rho, \tag{23}$$

$$\boldsymbol{\gamma}_i^l = \boldsymbol{\gamma}_i^{l-1} - \rho\left(\tilde{\boldsymbol{w}}_i^l - \boldsymbol{w}^l\right), \tag{24}$$

where ξ_i^l in (22) are sampled from $\mathcal{N}\left(0, \sigma_{i,l}^2 \mathbf{I}_K\right)$, and $\boldsymbol{w}^l$ in (23) is computed on the central machine. The rest are processed at each local machine. Details on FDP-ADMM algorithm are presented in Algorithm 2. Based on output $\boldsymbol{w}^L$ of Algorithm 2, we obtain:

$$\hat{\beta}_{DP}(t) = \sum_{k=1}^{K} w_k^L \phi_k(t).$$

Note that the central machine initializes the global $\boldsymbol{w}^0$, while each worker machine initializes their own variables: the noisy primal variables $\{\tilde{\boldsymbol{w}}_i^0\}$ and the dual variables $\{\gamma_i^0\}$ for $i \in [M]$. ξ_i^l is Gaussian noise with zero-mean and variance $\sigma_{i,l}^2$, where $\sigma_{i,l}^2$ is obtained based on the Gaussian mechanism of DP, which is given in Theorem 1. Each worker machine updates its noisy primal variable $\tilde{\boldsymbol{w}}_i^l$ based on (22). Then, the central machine receives all noisy primal variables $\left\{\tilde{\boldsymbol{w}}_i^l\right\}_{i\in[M]}$ and the dual variables $\left\{\gamma_i^l\right\}_{i\in[M]}$ from the worker machine, and updates a global variable w^l. In addition, w^l on central machine broadcasts to every worker machine to update the final dual variables $\left\{\gamma_i^l\right\}_{i\in[M]}$ using (24). It is an iterative cycle.

We set the variance $\sigma_{i,l} = \frac{2c_1\sqrt{2\ln(1.25/\delta)}}{m_i\epsilon\left(\rho+1/\eta_i^l\right)}$ for obtaining the (δ, ϵ)-DP of the FDP-ADMM algorithm, which is set based on the Gaussian mechanism of DP. $\sigma_{i,l}^2$ is time-varying; that is, it decreases as iteration l increases. The motivation of using time-varying variance in the Gaussian mechanism is to reduce the negative impact of noise and ensure the convergence of the algorithm. We find that the negative impact will be mitigated by the method of decreasing noise and can achieve a stable solution.

For the communication and computation costs of our algorithms 1 and 2, here are some remarks. We know that it is unrealistic to send the estimator of $\beta(t)$, $t \in [0, 1]$ on a worker machine to the central machine because it is infinite dimensional. In Algorithms 1 and 2, we only transmit the K-dimensional $\boldsymbol{w}$ in each round of communication, so the communication complexity is only $O(K)$. In practice of functional data analysis, K is usually a small number such as 5, 10, etc. Therefore, our algorithms are communication-efficient because of the low communication costs. In addition, in each round of learning, each worker machine learns its own low-dimensional parameter $\boldsymbol{w}$ based on its local data, then the central machine is responsible for summarizing these parameters from worker machines. This working mechanism greatly reduces the computational costs.

Algorithm 2: ADMM-based distributed learning with DP for PQR of Functional Data (FDP-ADMM)

Input: Initialize: $\boldsymbol{w}^0$, $\boldsymbol{\gamma}^0$, $\tilde{\boldsymbol{w}}_i^0$ and number of iteration L

for $l = 1, 2, \cdots, L$ **do**
  Inherit parameters from the previous iteration

  for *Worker machine* $i = 1, 2, \cdots, M$ **do**
    Compute $\boldsymbol{w}_i^l$ using Equation (21);

    Sample $\xi_i^l \sim \mathcal{N}(0, \sigma_{i,l}^2 \mathbf{I}_K)$, where $\sigma_{i,l}^2$ is given in Therorem 1;

    Compute $\tilde{\boldsymbol{w}}_i^l$ using Equation (22);

    Deliver $(\tilde{\boldsymbol{w}}_i^l, \boldsymbol{\gamma}_i^{l-1})$ to central machine.
  end

  Central machine computes $\boldsymbol{w}^l$ using Equation (23) , then broadcasts it to all machine.

  for *Worker machine* $i = 1, 2, \cdots, M$ **do**
    Compute $\boldsymbol{\gamma}_i^l$ using Equation (24).

  end
end

Output: $\boldsymbol{w}^L$.

4. Utility Analysis

4.1. Privacy Guarantee

In the section, we will analyze the privacy guarantee of the proposed FDP-ADMM algorithm. During traditional parameter transmission, the shared information $\left\{\boldsymbol{w}_i^l\right\}_{l\in[L]}$ can divulge the sensitive messages of original data. So, it is necessary to show outputs $\left\{\tilde{\boldsymbol{w}}_i^l\right\}_{l\in[L]}$ with differential privacy.

Denote the two neighboring datasets $\mathcal{A}_i$ and $\mathcal{A}_i'$. So, the $\boldsymbol{w}_{i,\mathcal{A}_i}^l$ and $\boldsymbol{w}_{i,\mathcal{A}_i'}^l$ are the primal variables obtained from every local worker machine. From the FDP-ADMM algorithm, we add noise to $\boldsymbol{w}_i^l$ by Gaussian mechanism. A fundamental tool used in DP is sensitivity. We use l_2-norm sensitivity. Due to the application of first-order approximation in the augmented Lagrange function, the proposed algorithm does not require the smoothness and strong convexity assumptions to the objective function for proving the sensitivity.

First, we give a lemma, which gives an l_2-norm sensitivity of $\boldsymbol{w}_i^l$ under the sub-gradient ℓ' of loss function ℓ, bounded.

Lemma 2. *Assume that* $\|\ell'(\cdot)\|_2 \leq c_1$. *The* l_2*-norm sensitivity of the local primal variable* $\boldsymbol{w}_i^l$ *update function is given by:*

$$\max_{\mathcal{A}_1,\mathcal{A}_1'} \left\| \boldsymbol{w}_{i,\mathcal{A}_i}^l - \boldsymbol{w}_{i,\mathcal{A}_i'}^l \right\| = \frac{2c_1}{m_i^2(\rho + 1/\eta_i^l)}.$$

Its proof is given in Appendix A. Lemma 2 shows that the l_2 sensitivity of $\boldsymbol{w}_i^l$ is affected by the time-varying η_i^l. We set η_i^l as a decreasing function of l, so the l_2 sensitivity decreases with increasing l. That is, if ϵ and δ is fixed, the added noise in the proposed algorithm will

become smaller as the l increases. Therefore, the algorithm will be stably convergent in spite of adding the noise. Then, we show that Algorithm 2 guarantees (ϵ, δ)-differential privacy.

Theorem 1. *Assume that* $\|\boldsymbol{A}_{ij}\|_2 \leq c_1$, $i = 1, \cdots, M$ *and* $j = 1, \cdots, m_i$ *in the model (11). Let* $\epsilon \in (0, 1]$ *be arbitrary and* ξ_i^k *be the noise sampled from Gaussian mechanism with variance* $\sigma_{i,k}^2$, *where*

$$\sigma_{i,k} = \frac{2c_1\sqrt{2\ln(1.25/\delta)}}{m_i\epsilon(\rho + 1/\eta_i^k)}.$$

The FDP-ADMM guarantees (ϵ, δ)*-differential privacy. Specifically, for any neighboring datasets* $\mathcal{A}_i$ *and* $\mathcal{A}_i'$, *for any output* $\tilde{\boldsymbol{w}}_i^k$, *the following inequality always holds:*

$$\Pr\left[\tilde{\boldsymbol{w}}_i^k \mid \mathcal{A}_i\right] \leq e^{\epsilon} \cdot \Pr\left[\tilde{\boldsymbol{w}}_i^k \mid \mathcal{A}_i'\right] + \delta.$$

Its proof is given in Appendix A.

4.2. Convergence of the FDP-ADMM Algorithm

The convergence of ADMM for convex problems has been widely studied in recent years. Under the requirement of high precision, the convergence of ADMM goes very slowly. However, under the requirement of medium precision, the convergence speed of ADMM is acceptable, and the global solution can be achieved by dozens of iterations. Furthermore, Ref. [44] showed that ADMM could attain a global linear convergence on strict convexity and Lipschitz gradient, especially when matrix B and C in (8) are full column rank. The ADMM framework is suitable for large-scale statistical learning problems. More convergence analysis of ADMM under convexity were studied by [45–52], and so on. Ref. [53] proposed an approximate ADMM algorithm to make it converge to the stable point with a large enough penalty parameter. Ref. [54] gave the convergence of quantile regression using ADMM for convex and non-convex penalties. We refer to [54] for the convergence of our FDP-ADMM algorithm.

We define $\boldsymbol{w}^*$ as the optimal solution of (13) and $c_w = \|\boldsymbol{w}^*\|_2$. The convergence of the algorithm is based on the fact that the quantile loss function is convex and non-smooth. For simplicity of analysis, we define some notations as follows:

$$f_i(\boldsymbol{w}_i) = \sum_{j=1}^{m_i} \frac{1}{m_i}\rho_\tau(y_{ij} - \boldsymbol{A}_{i,j}\boldsymbol{w}_i) + \frac{\lambda}{M}P(\boldsymbol{w}_i),$$

$$\overline{\boldsymbol{w}}^L = \frac{1}{L}\sum_{l=1}^{L}\boldsymbol{w}^l, \quad \bar{\boldsymbol{\gamma}}_i^L = \frac{1}{L}\sum_{l=1}^{L}\boldsymbol{\gamma}_i^l, \quad \overline{\boldsymbol{w}}_i^l = \frac{1}{L}\sum_{l=0}^{L-1}\tilde{\boldsymbol{w}}_i^l,$$

$$\boldsymbol{u}_i^l = \begin{bmatrix} \tilde{\boldsymbol{w}}_i^l \\ \boldsymbol{w}^l \\ \boldsymbol{\gamma}_i^l \end{bmatrix}, \quad \boldsymbol{u}_i = \begin{bmatrix} \boldsymbol{w}_i \\ \boldsymbol{w} \\ \boldsymbol{\gamma}_i \end{bmatrix}, \quad F\left(\boldsymbol{u}_i^l\right) = \begin{bmatrix} -\boldsymbol{\gamma}_i^l \\ \boldsymbol{\gamma}_i^l \\ \tilde{\boldsymbol{w}}_i^l - \boldsymbol{w}^l \end{bmatrix}.$$

We analyze the convergence of our proposed algorithm in terms of both the objective value and the constraint violation as [55]:

$$\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right) - f_i(\boldsymbol{w}^*) + g\left\|\overline{\boldsymbol{w}}_i^L - \overline{\boldsymbol{w}}^L\right\|\right),$$

where $\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right) - f_i(\boldsymbol{w}^*)\right)$ is used to measure the distance between the current objective value and the optimal value, and $\sum_{i=1}^{M} g\left\|\overline{\boldsymbol{w}}_i^L - \overline{\boldsymbol{w}}^L\right\|$ depicts the difference between the local model and the global one. If the training result of our FDP-ADMM algorithm achieves optimal and local models, it obtains consensus.

Lemma 3 ([55], lemma 2). *Assume $\rho_\tau(\cdot)$ and $P(\cdot)$ are convex. For any $l \geq 1$, we have:*

$$\begin{aligned}
&\sum_{i=1}^{M}\left(f_i\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-f_i(\boldsymbol{w}_i)+\left(\boldsymbol{u}_i^l-\boldsymbol{u}_i\right)^\top F\left(\boldsymbol{u}_i^l\right)\right)\\
\leq&\sum_{i=1}^{M}\left(\frac{\eta_i^l}{2}\left\|f_i'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-\left(\rho+1/\eta_i^l\right)\xi_i^l\right\|^2-\frac{\rho}{2}\left\|\boldsymbol{w}_i-\boldsymbol{w}^l\right\|^2+\frac{\rho}{2}\left\|\boldsymbol{w}_i-\boldsymbol{w}^{l-1}\right\|^2\right.\\
&-\left(\rho+1/\eta_i^l\right)\left\langle\xi_i^l,\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^{l-1}\right\rangle+\frac{1}{2\eta_i^l}\left\|\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^{l-1}\right\|^2-\frac{1}{2\eta_i^l}\left\|\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^{l}\right\|^2\\
&\left.+\frac{1}{2\rho}\left\|\gamma_i-\gamma_i^{l-1}\right\|^2-\frac{1}{2\rho}\left\|\gamma_i-\gamma_i^{l}\right\|^2\right).
\end{aligned}$$

Based on Lemma 3, we have:

Theorem 2. *Assume that $\|\boldsymbol{A}_{ij}\|_2 \leq c_1$, $i = 1, \cdots, M$, and $j = 1, \cdots, m_i$ in the model (11); $P(\cdot)$ are convex; and $\|P'(\cdot)\| \leq c_2$. The domain of the dual variable is bounded, namely, $\|\gamma_i\| \leq g$. We set the learning rate as:*

$$\eta_i^l=\frac{c_w}{\sqrt{2l}}\left(\left(c_1+\lambda c_2/M\right)^2+\frac{8Kc_1^2\ln(1.25/\delta)}{m_i^2\epsilon^2}\right)^{-\frac{1}{2}}.$$

For any $L \geq 1$ and g, we have:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i^*)\right.\right.+g\left\|\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right\|\\
\leq&\sum_{i=1}^{M}\frac{c_w}{\sqrt{L}}\sqrt{2\left(c_1+\lambda c_2/M\right)^2+\frac{16Kc_1^2\ln(1.25/\delta)}{m_i^2\epsilon^2}}+\frac{M\left(\rho c_w^2+g^2/\rho\right)}{2L}.
\end{aligned}$$

Its proof is given in Appendix A. Theorem 2 shows our approach achieves the rate of convergence at $O(1/\sqrt{L})$, and gives an explicit utility-privacy trade-off of our FDP-ADMM algorithm. For the larger ϵ or δ, our algorithm has better utility. Note that the larger ϵ or δ means the weaker privacy-preserving ability.

5. Simulation Study

In this section, we illustrate the performance of the proposed privacy-protection FDP-ADMM algorithm using a simulated study.

The simulation design is described as follows:

$$y_i=\int_I\beta(t)X_i(t)dt+\epsilon_{\tau i},i=1,\cdots,n,$$

where n is the sample size on all worker machines;

$$\beta(t)=\sum_{k=1}^{50}w_k\phi_k(t),$$

with $w_1 = 0.3$, $w_k = 4(-1)^{k+1}k^{-2}$ for $k \geq 2$, $\phi_1(t) \equiv 1$ and $\phi_k(t) = 2^{1/2}\cos((k-1)\pi t)$ for $k \geq 2$;

$$X_i(t)=\sum_{k=1}^{50}A_{ik}\phi_k(t),$$

where A_{ik}'s are independent and normal $N(0, k^{-2})$; and the errors

$$\epsilon_{\tau i} = \epsilon_i - F_{\epsilon}^{-1}(\tau),$$

where F_ϵ is the distribution function of ϵ_i, take $\epsilon \sim t(3)$. Note that F_ϵ^{-1} is subtracted from ϵ_i to make the τth quantile of $\epsilon_{\tau i}$ zero for identifiability. The datasets $(y_i, X_i(t))_{i=1}^n$ are distributed on M worker machines, and each machine has the same sample size m. So, $n = Mm$.

In the simulation, we set $n = 100{,}000$ samples, and randomly split the dataset into $M = 10$, 20, and 50 groups to simulate the distributed learning condition. We take the penalty parameter $\rho = 0.1$, and the regularized parameter $\lambda = 0.05$. We set the level of quantile $\tau = \{0.1, 0.25, 0.5, 0.75, 0.9\}$, and set privacy budget per iteration $\epsilon = \{0.01, 0.05, 0.1, 0.2\}$ and $\delta = \left\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\right\}$. For each scenario, we run the algorithm 100 times. We consider our FDP-ADMM algorithm with typical l_1-norm and l_2-norm penalties and then assess it in terms of convergence and accuracy.

First, we report the mean integrated squared error (MISE) of the estimator $\hat{\beta}_{DP}(t)$ computed on a grid of 100 equally spaced points on $I = [0, 1]$, that is:

$$\text{MISE} = \mathbb{E}\left(\int_0^1 (\hat{\beta}_{DP}(t) - \beta(t))^T (\hat{\beta}_{DP}(t) - \beta(t)) dt\right).$$

Second, based on $\bar{\boldsymbol{w}}_i^l$ and $\bar{\boldsymbol{w}}^l$, we evaluate the convergence properties of the FDP-ADMM algorithm with respect to the augmented objective value, which measures the loss as well as the constraint penalty and is defined as:

$$\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^l\right) + \rho\left\|\overline{\boldsymbol{w}}_i^l - \overline{\boldsymbol{w}}^l\right\|\right).$$

Third, we evaluate the accuracy by empirical loss:

$$\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{m_i}\frac{1}{m_i}\rho_\tau\left(y_{ij} - \boldsymbol{A}_{ij}\tilde{\boldsymbol{w}}_i^k\right).$$

5.1. L1-Regularized Quantile Regression

We obtain the FDP-ADMM steps for the l_1-norm quantile regression by:

$$\begin{aligned}
\boldsymbol{w}_i^l &= \left(\frac{1}{m_i}\sum_{j=1}^{m_i} A_{i,j}^T(\tau - I_{\{y_{i,j} - A_{i,j}\tilde{\boldsymbol{w}}_i^{l-1} \le 0\}}) - \frac{\lambda}{M}\operatorname{sgn}\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)\right.\\
&\quad \left. + \boldsymbol{\gamma}_i^{l-1} + \rho\boldsymbol{w}^{l-1} + \tilde{\boldsymbol{w}}_i^{l-1}/\eta_i^l\right)\left(\rho + 1/\eta_i^l\right)^{-1},\\
\tilde{\boldsymbol{w}}_i^l &= \boldsymbol{w}_i^l + \mathcal{N}\left(0, \sigma_{i,l}^2\mathbf{I}_K\right),\\
\boldsymbol{w}^l &= \frac{1}{M}\sum_{i=1}^{M}\tilde{\boldsymbol{w}}_i^l - \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{\gamma}_i^{l-1}/\rho,\\
\boldsymbol{\gamma}_i^l &= \boldsymbol{\gamma}_i^{l-1} - \rho\left(\tilde{\boldsymbol{w}}_i^l - \boldsymbol{w}^l\right),
\end{aligned}$$

where $\operatorname{sgn}(\cdot)$ is the sign function. Since the objective function is convex but non-smooth, we use Theorem 2 to set η_i^l and apply Theorem 1 to set $\sigma_{i,l}$.

First, we list MISEs for the number of local machines, $M = 10$, 20, 50, in Tables 1–3, respectively. From Tables 1–3, we observe that (i) Our approach with larger ϵ and larger δ has better convergence for all quantile levels because their MISEs are smaller, which also implies weaker privacy protection. When $\epsilon = 0.8$ and $\delta = 0.001$, the MISEs of our FDP-ADMM algorithm with privacy policy are comparable to the ones of non-DP algorithm

($\delta = \infty$), that is, our FDP-ADMM algorithm does not sacrifice the estimation accuracy under weak privacy protection. (ii) For strong privacy protection, such as $\epsilon = 0.1$, the accuracy of our training model decreases as the number of machines increases. Because the size of the local dataset is smaller for a larger number of working machines, more noise should be added into the FDP-ADMM algorithm to obtain a higher privacy guarantee. For the large number of machines, a high estimation accuracy can be achieved by reducing privacy protection, for example, $M = 50$. (iii) Our FDP-ADMM algorithm has a trade off between privacy and accuracy, i.e., the stronger the privacy protection, the lower the estimation accuracy. (iv) When $\tau = 0.5$, this FDP-ADMM is a robust distributed learning algorithm for all parameters of privacy and number of machines we set. Because τ is farther from 0.5, its MISE is worse.

Table 1. MISEs of FDP-ADMM algorithm for l_1-regularized quantile regression when $M = 10$.

		M = 10				
ϵ	δ	$\tau = 0.1$	$\tau = 0.25$	$\tau = 0.5$	$\tau = 0.75$	$\tau = 0.9$
0.1	10^{-6}	1.10935	1.08272	1.08042	1.06772	1.15352
	10^{-5}	1.01472	0.88899	0.63939	0.72965	0.81495
	10^{-4}	1.00092	0.50874	0.61940	0.42669	1.20186
	10^{-3}	0.76701	0.51994	0.43749	0.45714	0.83579
0.2	10^{-6}	0.88293	0.49352	0.37903	0.48573	1.13445
	10^{-5}	0.86012	0.49398	0.35214	0.43233	0.93748
	10^{-4}	0.85513	0.47709	0.40259	0.45489	0.89320
	10^{-3}	0.97630	0.45829	0.36365	0.42839	0.95078
0.3	10^{-6}	0.75955	0.40747	0.37103	0.43872	0.99364
	10^{-5}	0.80688	0.40563	0.38785	0.43834	1.16647
	10^{-4}	0.82462	0.43617	0.37086	0.42679	1.08880
	10^{-3}	0.88212	0.43003	0.36211	0.43141	0.88249
0.5	10^{-6}	0.94682	0.46687	0.37158	0.42812	0.96098
	10^{-5}	0.92709	0.44007	0.38863	0.42447	0.89955
	10^{-4}	0.93776	0.43648	0.36595	0.44708	0.89621
	10^{-3}	0.95611	0.42511	0.37481	0.45199	0.98540
0.8	10^{-6}	0.87232	0.44024	0.38308	0.44378	0.96025
	10^{-5}	0.90369	0.44478	0.37863	0.44728	0.99718
	10^{-4}	0.91670	0.43855	0.37769	0.45473	0.92953
	10^{-3}	0.93121	0.43090	0.37537	0.43751	0.94244
∞	1	0.92588	0.43536	0.38291	0.44820	0.98044

Table 2. MISEs of FDP-ADMM algorithm for l_1-regularized quantile regression when $M = 20$.

		M = 20				
ϵ	δ	$\tau = 0.1$	$\tau = 0.25$	$\tau = 0.5$	$\tau = 0.75$	$\tau = 0.9$
0.1	10^{-6}	9.14206	7.57861	8.63208	5.99993	7.25853
	10^{-5}	4.91313	3.39025	3.84567	4.72525	2.98801
	10^{-4}	4.07548	1.99489	2.15743	1.73728	3.75287
	10^{-3}	1.19830	0.92899	0.89924	1.38734	1.72083
0.2	10^{-6}	1.27889	1.07583	0.91118	1.38400	1.39675
	10^{-5}	1.12262	0.64832	0.68967	0.60605	0.97722
	10^{-4}	0.79600	0.48621	0.40788	0.32599	0.59192
	10^{-3}	0.58956	0.35490	0.28023	0.36716	0.79028
0.3	10^{-6}	0.67998	0.48955	0.39535	0.42677	0.69698
	10^{-5}	0.62656	0.34088	0.31510	0.36501	0.82977
	10^{-4}	0.64047	0.35214	0.23091	0.42389	0.52364
	10^{-3}	0.64256	0.33683	0.31104	0.37006	0.65798
0.5	10^{-6}	0.48045	0.37946	0.28304	0.38171	0.60391
	10^{-5}	0.58620	0.35572	0.32657	0.35470	0.53452

Table 2. *Cont.*

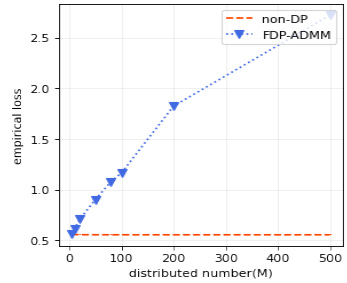
		$M = 20$				
ϵ	δ	$\tau = 0.1$	$\tau = 0.25$	$\tau = 0.5$	$\tau = 0.75$	$\tau = 0.9$
	10^{-4}	0.60473	0.32019	0.29540	0.35935	0.61225
	10^{-3}	0.55453	0.35411	0.31989	0.36962	0.61693
0.8	10^{-6}	0.62139	0.34965	0.28970	0.36232	0.61739
	10^{-5}	0.56993	0.33311	0.32059	0.35161	0.52785
	10^{-4}	0.59024	0.36645	0.31904	0.34726	0.60428
	10^{-3}	0.54107	0.35753	0.32211	0.35959	0.61891
∞	1	0.57159	0.36387	0.34485	0.36757	0.60352

Table 3. MISEs of FDP-ADMM algorithm for l_1-regularized quantile regression when $M = 50$.

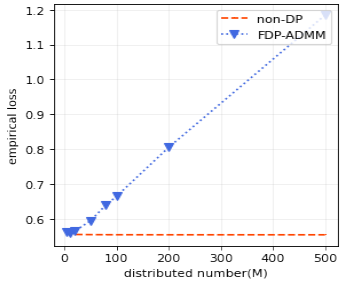
		$M = 50$				
ϵ	δ	$\tau = 0.1$	$\tau = 0.25$	$\tau = 0.5$	$\tau = 0.75$	$\tau = 0.9$
0.1	10^{-6}	32.69092	45.61741	18.57951	42.89526	26.78259
	10^{-5}	19.54404	18.28042	24.98649	14.56024	23.62843
	10^{-4}	9.25744	15.35268	14.21395	16.51154	15.62429
	10^{-3}	8.48043	8.02662	9.36117	8.54005	10.24176
0.2	10^{-6}	7.30838	10.65070	6.65439	8.11806	8.28644
	10^{-5}	3.28129	7.68274	5.46142	5.37228	6.88005
	10^{-4}	3.69134	3.30237	2.53601	2.68905	4.13535
	10^{-3}	1.23577	1.46382	0.89365	1.12239	1.55549
0.3	10^{-6}	2.77899	2.96351	2.07031	2.92310	2.93987
	10^{-5}	2.79987	2.22634	1.83420	1.72759	1.82370
	10^{-4}	1.21077	0.99900	1.30057	0.93872	1.55615
	10^{-3}	0.78847	0.53879	0.45738	0.51968	0.78215
0.5	10^{-6}	0.95185	0.79408	0.94353	0.85746	1.39231
	10^{-5}	0.91010	0.90503	0.54837	0.61216	0.83725
	10^{-4}	0.69861	0.37440	0.44130	0.41332	0.55033
	10^{-3}	0.47677	0.29099	0.26389	0.28654	0.50744
0.8	10^{-6}	0.43862	0.42597	0.28514	0.41843	0.57740
	10^{-5}	0.69838	0.32106	0.24453	0.33619	0.55950
	10^{-4}	0.51937	0.39685	0.22664	0.28052	0.42013
	10^{-3}	0.46601	0.26187	0.20617	0.24769	0.38190
∞	1	0.43715	0.30922	0.229919	0.29383	0.45229

Second, we study the training performance (empirical loss) v.s. different number of distributed data sources under different levels of privacy protection when $\tau = 0.5$. See Figure 1. Figure 1 shows that the accuracy of our training model will be reduced if more local machines are used. Since the number of agents is larger the smaller the size of the local dataset is, more noise should be added to guarantee the same level of differential privacy. Thus, it results in reducing the performance of the trained model. This is consistent with Theorem 1 that the standard deviation of noises is scaled by $1/mi$. From another perspective, when more local machines participate, a weaker privacy protection can obtain a higher estimation accuracy.

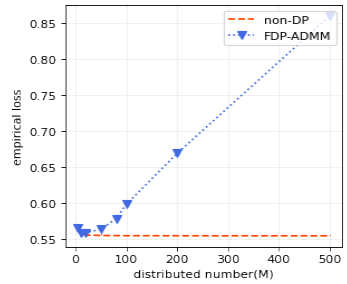
Third, we illustrate the convergence of the FDP-ADMM algorithm by demonstrating how the augmented objective value converges for different values of ϵ and δ. See Figure 2. Figure 2 shows our algorithm with larger ϵ and δ (which implies the weaker privacy protection) has better convergence. This result is consistent with Theorem 2.



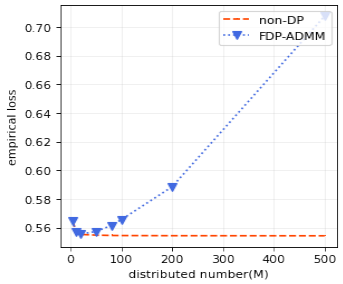
(**a**) $\epsilon = 0.01, \delta = 10^{-3}$



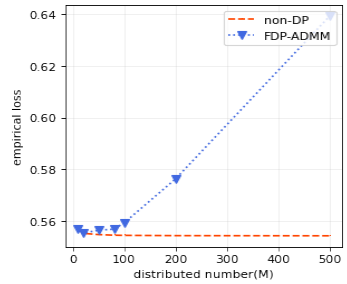
(**b**) $\epsilon = 0.05, \delta = 10^{-3}$



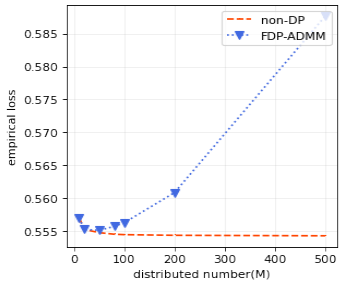
(**c**) $\epsilon = 0.1, \delta = 10^{-3}$



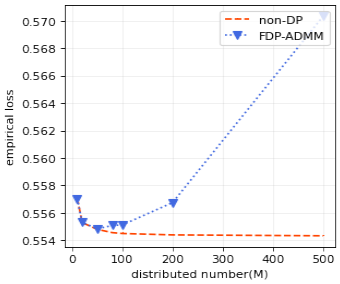
(**d**) $\epsilon = 0.2, \delta = 10^{-3}$



(**e**) $\epsilon = 0.3, \delta = 10^{-3}$



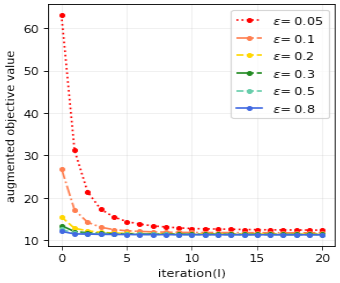
(**f**) $\epsilon = 0.5, \delta = 10^{-3}$



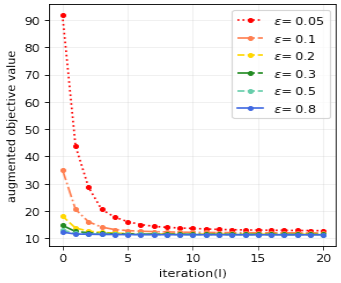
(**g**) $\epsilon = 0.8, \delta = 10^{-3}$

Figure 1. Impact of the number of distributed data sources on FDP-ADMM for l_1-regularized quantile regression.

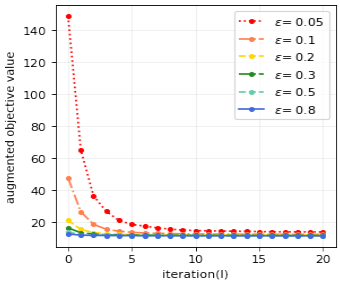
Finally, we evaluate the performance of FDP-ADMM by empirical loss for different levels of privacy protection. See Figure 3. Figure 3 shows our approach has fast convergence property for all privacy policies. In addition, all results we obtained show the privacy–utility trade-off of our FDP-ADMM: better utility is achieved when privacy leakage increases.



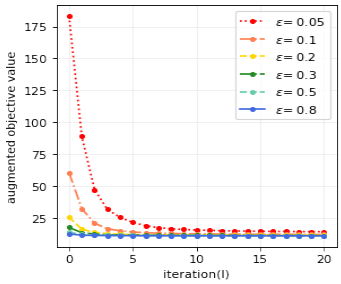
(**a**) $\delta = 0.001$



(**b**) $\delta = 0.0001$

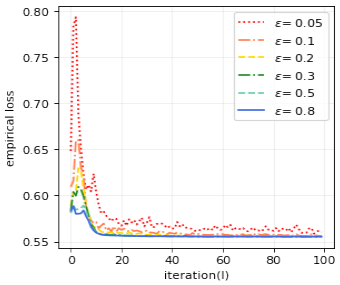


(**c**) $\delta = 0.00001$

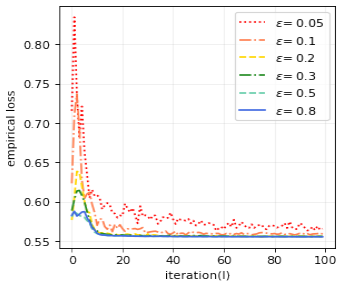


(**d**) $\delta = 0.000001$

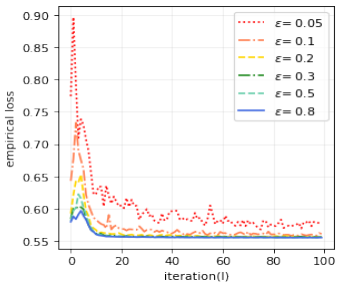
Figure 2. Convergence properties of FDP-ADMM via augmented objective value for l_1-regularized quantile regression with $\tau = 0.5$.



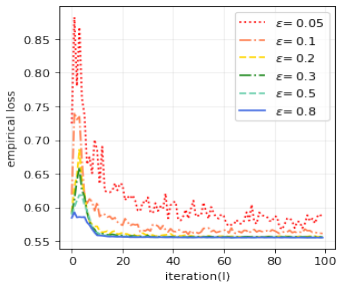
(**a**) $\delta = 0.001$



(**b**) $\delta = 0.0001$



(**c**) $\delta = 0.00001$



(**d**) $\delta = 0.000001$

Figure 3. Convergence properties of FDP-ADMM via empirical loss for l_1-regularized quantile regression $\tau = 0.5$.

5.2. L2-Regularized Quantile Regression

We obtain the FDP-ADMM steps for l_2-norm quantile regression as follows:

$$
\begin{aligned}
\boldsymbol{w}_i^l &= \left(\frac{1}{m_i}\sum_{j=1}^{m_i} A_{i,j}^T(\tau - I_{\{y_{i,j}-A_{i,j}\tilde{\boldsymbol{w}}_i^{l-1}\le 0\}}) - \frac{\lambda}{M}\tilde{\boldsymbol{w}}_i^{l-1} + \right. \\
&\quad \left. \gamma_i^{l-1} + \rho \boldsymbol{w}^{l-1} + \tilde{\boldsymbol{w}}_i^{l-1}/\eta_i^l\right)\left(\rho + 1/\eta_i^l\right)^{-1}, \\
\tilde{\boldsymbol{w}}_i^l &= \boldsymbol{w}_i^l + \mathcal{N}\left(0, \sigma_{i,l}^2 \mathbf{I}_K\right), \\
\boldsymbol{w}^l &= \frac{1}{M}\sum_{i=1}^{M}\tilde{\boldsymbol{w}}_i^l - \frac{1}{M}\sum_{i=1}^{M}\gamma_i^{l-1}/\rho, \\
\gamma_i^l &= \gamma_i^{l-1} - \rho\left(\tilde{\boldsymbol{w}}_i^l - \boldsymbol{w}^l\right).
\end{aligned}
$$

Similar to the setting of Section 5.1, we present results in Tables 4–6 and Figures 4–6. We also obtain the same conclusion as in Section 5.1.

Table 4. MISEs of FDP-ADMM algorithm for l_2-regularized quantile regression when $M = 10$.

		$M = 10$				
ϵ	δ	$\tau = 0.1$	$\tau = 0.25$	$\tau = 0.5$	$\tau = 0.75$	$\tau = 0.9$
0.1	10^{-6}	2.89210	2.75353	2.46941	2.54780	3.81696
	10^{-5}	2.92584	2.27235	2.09434	2.27390	2.86815
	10^{-4}	1.37434	1.52857	1.63519	1.01153	1.76823
	10^{-3}	0.99495	0.95007	0.90349	1.12417	1.11879
0.2	10^{-6}	1.14263	1.07720	0.69340	1.03047	1.25381
	10^{-5}	0.89339	0.60944	0.76508	0.70292	1.22899
	10^{-4}	0.76141	0.60331	0.43044	0.56315	0.87222
	10^{-3}	0.61730	0.58038	0.38235	0.42983	0.67101
0.3	10^{-6}	0.89291	0.52495	0.53238	0.50471	0.90524
	10^{-5}	0.82594	0.52562	0.34734	0.48837	0.83485
	10^{-4}	0.61622	0.43849	0.34253	0.47389	0.52728
	10^{-3}	0.50265	0.34256	0.24852	0.36576	0.58658
0.5	10^{-6}	0.50657	0.35774	0.33148	0.36034	0.61492
	10^{-5}	0.62303	0.31707	0.25926	0.33131	0.56412
	10^{-4}	0.44178	0.32944	0.27809	0.27520	0.56905
	10^{-3}	0.55427	0.31399	0.22924	0.27581	0.53565
0.8	10^{-6}	0.48112	0.30366	0.26190	0.29383	0.54639
	10^{-5}	0.52303	0.31519	0.21492	0.30934	0.49795
	10^{-4}	0.51200	0.29494	0.25546	0.26651	0.51737
	10^{-3}	0.49710	0.32842	0.21990	0.26114	0.49119
∞	1	0.47430	0.27901	0.20853	0.26926	0.49421

Table 5. MISEs of FDP-ADMM algorithm for l_2-regularized quantile regression when $M = 20$.

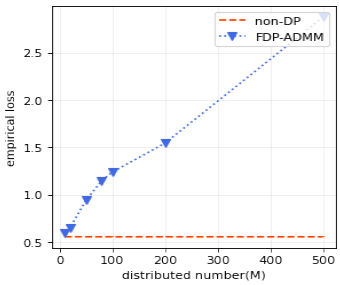
		$M = 20$				
ϵ	δ	$\tau = 0.1$	$\tau = 0.25$	$\tau = 0.5$	$\tau = 0.75$	$\tau = 0.9$
0.1	10^{-6}	12.09882	7.53753	8.99489	9.28500	10.41200
	10^{-5}	6.36908	4.83324	9.44384	5.38515	7.07413
	10^{-4}	5.40212	3.78114	2.92258	5.80128	3.78069
	10^{-3}	3.56865	1.67093	2.02714	3.12041	2.73505
0.2	10^{-6}	2.35128	2.40928	2.47491	3.13811	2.81266
	10^{-5}	2.44151	1.86417	2.14190	1.79286	1.69856
	10^{-4}	1.25333	1.12828	1.06031	1.29433	1.47163
	10^{-3}	1.00380	0.67944	0.75994	0.89221	1.20207
0.3	10^{-6}	1.44773	1.21530	1.82965	1.16274	1.67580
	10^{-5}	0.92305	0.84215	0.98673	0.87402	0.83162

Table 5. *Cont.*

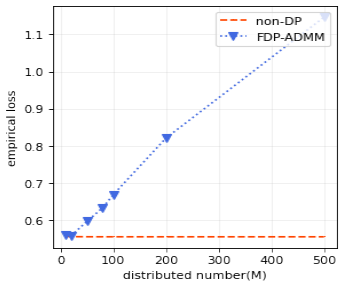
		$M = 20$				
ϵ	δ	$\tau = 0.1$	$\tau = 0.25$	$\tau = 0.5$	$\tau = 0.75$	$\tau = 0.9$
	10^{-4}	0.61151	0.50056	0.62719	0.75269	1.11462
	10^{-3}	0.79422	0.43239	0.46819	0.43958	0.65150
0.5	10^{-6}	0.65224	0.49834	0.69298	0.50219	0.72637
	10^{-5}	0.52623	0.48679	0.39195	0.41469	0.81617
	10^{-4}	0.45604	0.44620	0.39356	0.35684	0.54628
	10^{-3}	0.48589	0.35496	0.26164	0.28309	0.44583
0.8	10^{-6}	0.58554	0.36776	0.36210	0.35751	0.44102
	10^{-5}	0.56310	0.37512	0.24621	0.38069	0.41426
	10^{-4}	0.46934	0.30216	0.22564	0.27554	0.43864
	10^{-3}	0.39110	0.22099	0.22896	0.22580	0.42480
∞	1	0.37468	0.21974	0.16201	0.20135	0.38618

Table 6. MISEs of FDP-ADMM algorithm for l_2-regularized quantile regression when $M = 50$.

		$M = 50$				
ϵ	δ	$\tau = 0.1$	$\tau = 0.25$	$\tau = 0.5$	$\tau = 0.75$	$\tau = 0.9$
0.1	10^{-6}	28.61208	40.72216	29.56713	19.82800	22.75443
	10^{-5}	24.29355	10.61405	23.48655	25.28936	16.24641
	10^{-4}	18.61518	10.56809	15.63120	18.87195	14.77067
	10^{-3}	8.23236	11.80948	6.46954	8.75901	7.39285
0.2	10^{-6}	9.21639	7.30661	10.83166	6.50947	6.74332
	10^{-5}	5.33237	7.20520	6.48928	4.99331	6.98322
	10^{-4}	3.69557	4.25720	5.23905	4.25916	4.26556
	10^{-3}	2.64944	3.40341	2.10149	1.80372	2.19432
0.3	10^{-6}	3.72135	5.26765	4.62668	2.82433	2.56538
	10^{-5}	3.52995	3.49432	3.82902	2.46528	2.24935
	10^{-4}	2.10102	2.30679	1.91362	2.48765	2.40823
	10^{-3}	1.34345	1.65347	1.23016	1.02293	1.63715
0.5	10^{-6}	1.76621	1.91785	1.38861	1.41912	2.15178
	10^{-5}	1.14848	0.91837	1.25444	1.12365	0.96906
	10^{-4}	0.82513	1.11108	0.79060	0.87009	1.03410
	10^{-3}	0.63068	0.47302	0.41747	0.48054	0.59396
0.8	10^{-6}	0.75022	0.86022	0.72234	0.83085	0.91919
	10^{-5}	0.88104	0.91287	0.59528	0.52652	0.47342
	10^{-4}	0.52061	0.44782	0.33773	0.43406	0.61304
	10^{-3}	0.39031	0.37708	0.26519	0.28573	0.42317
∞	1	0.31062	0.18729	0.13762	0.16653	0.31931

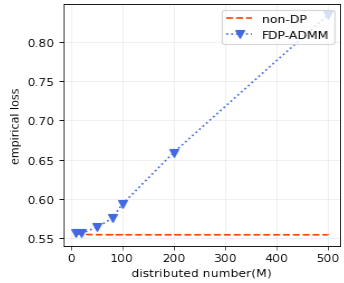


(**a**) $\epsilon = 0.01, \delta = 10^{-3}$

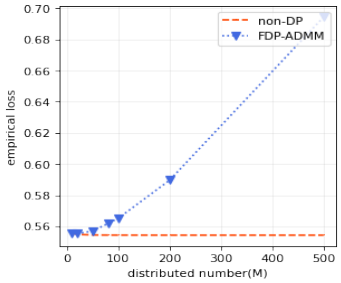


(**b**) $\epsilon = 0.05, \delta = 10^{-3}$

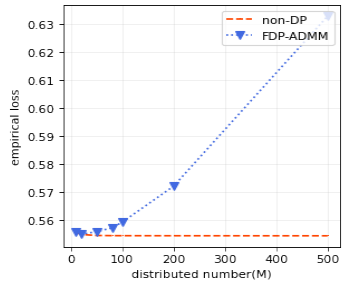
Figure 4. *Cont.*



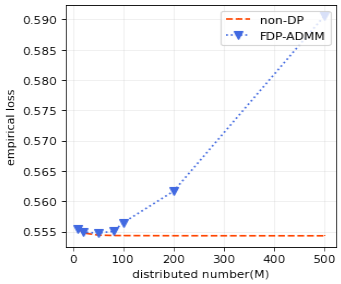
(**c**) $\epsilon = 0.1, \delta = 10^{-3}$



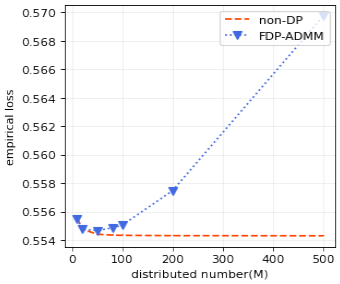
(**d**) $\epsilon = 0.2, \delta = 10^{-3}$



(**e**) $\epsilon = 0.3, \delta = 10^{-3}$

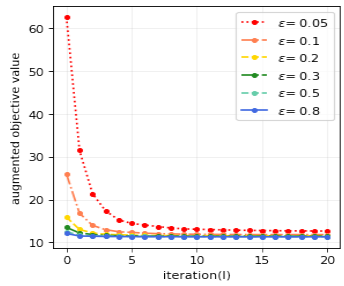


(**f**) $\epsilon = 0.5, \delta = 10^{-3}$

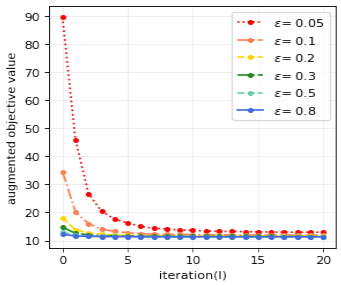


(**g**) $\epsilon = 0.8, \delta = 10^{-3}$

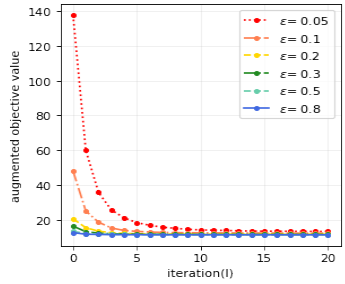
Figure 4. Impact of the number of distributed data sources on FDP-ADMM for l_2-regularized quantile regression.



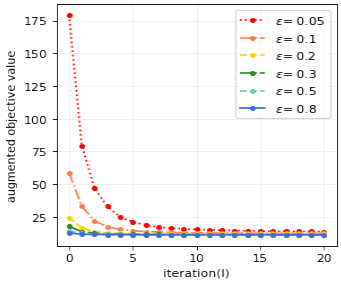
(**a**) $\delta = 0.001$



(**b**) $\delta = 0.0001$

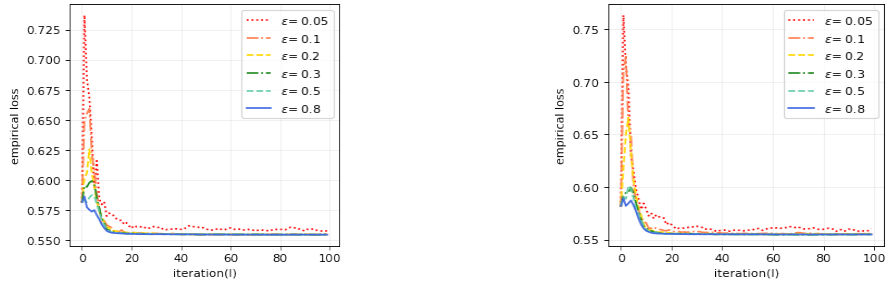


(**c**) $\delta = 0.00001$

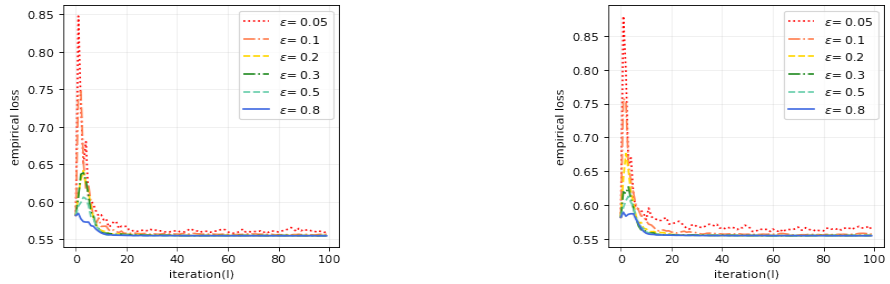


(**d**) $\delta = 0.000001$

Figure 5. Convergence properties of FDP-ADMM via augmented objective value for l_2-regularized quantile regression with $\tau = 0.5$.



(**a**) $\delta = 0.001$ (**b**) $\delta = 0.0001$



(**c**) $\delta = 0.00001$ (**d**) $\delta = 0.000001$

Figure 6. Convergence properties of FDP-ADMM via empirical loss for l_2-regularized quantile regression $\tau = 0.5$.

6. Conclusions

In the paper, we proposed an ADMM-based differential privacy learning algorithm on penalized quantile regression for functional data: FDP-ADMM. We first transform functional quantile regression into an ordinary linear regression model by functional principal analysis, and then design the FDP-ADMM algorithm by an approximate augmented Lagrange function, ADMM algorithm, and Gaussian mechanism with time-varying variance. The FDP-ADMM is a noise-resilient, convergent, and computationally effective distributed learning algorithm, even if for high privacy guarantee. Lastly, we obtain the estimation of coefficient function with privacy protection for functional quantile regression distributed model by the Karhunen and Loève expression. We also derived the privacy guarantee and theoretical convergence by the objective value and the constraint violation. The evaluations on simulation datasets have demonstrated the effectiveness of the FDP-ADMM algorithm, even if under high privacy protection, and have shown its privacy–utility trade-off: larger ϵ and larger δ, indicating weaker privacy guarantee, results in better utility.

Author Contributions: Conceptualization, X.Z.; methodology, Y.X.; validation, X.Z. and Y.X.; investigation, X.Z.; writing—original draft preparation, Y.X.; writing—review and editing, X.Z. and Y.X.; supervision, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Chinese National Social Science Fund (19BTJ034), the National Natural Science Foundation of China (12171242, 11971235, 12071220), and the Postgraduate Research and Practice Innovation Program of Jiangsu Province (KYCX21_1940).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In the Appendix, we give the proofs of Lemma 2 and Theorems 1 and 2.

Appendix A.1. Proof of Lemma 2

Proof. First, we have that $\hat{\mathcal{L}}_{\rho,i}\left(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{l-1}, \boldsymbol{w}^{l-1}, \boldsymbol{\gamma}_i^{l-1}\right)$ is convex because it is a quadratic function of $\boldsymbol{w}_i$. Thus, we have a closed-form solution:

$$\boldsymbol{w}_{i,\mathcal{A}_i}^l = \left(\rho + 1/\eta_i^l\right)^{-1}\left(-\sum_{j=1}^{m_i}\frac{1}{m_i}\ell'\left(\tilde{\boldsymbol{w}}_i^{l-1}, \mathcal{A}_i\right) - \frac{\lambda}{M}P'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right) + \boldsymbol{\gamma}_i^{l-1} + \rho\boldsymbol{w}^{l-1} + \frac{\tilde{\boldsymbol{w}}_i^{l-1}}{\eta_i^l}\right),$$

$$\boldsymbol{w}_{i,\mathcal{A}_i'}^l = \left(\rho + 1/\eta_i^l\right)^{-1}\left(-\sum_{j=1}^{m_i}\frac{1}{m_i}\ell_i'\left(\tilde{\boldsymbol{w}}_i^{l-1}, \mathcal{A}_i'\right) - \frac{\lambda}{M}P'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right) + \boldsymbol{\gamma}_i^{l-1} + \rho\boldsymbol{w}^{l-1} + \frac{\tilde{\boldsymbol{w}}_i^{l-1}}{\eta_i^l}\right).$$

Then, the l_2-norm sensitivity of primal variable $\boldsymbol{w}_i^l$ is:

$$\begin{aligned}\max_{\mathcal{A}_i,\mathcal{A}_i'}\left\|\boldsymbol{w}_{i,\mathcal{A}_i}^l - \boldsymbol{w}_{i,\mathcal{A}_i'}^l\right\|_2 &= \max_{\mathcal{A}_i,\mathcal{A}_i'}\frac{\left\|\ell'\left(\tilde{\boldsymbol{w}}_i^{l-1}, \mathcal{A}_i\right) - \ell'\left(\tilde{\boldsymbol{w}}_i^{l-1}, \mathcal{A}_i'\right)\right\|}{m_i\left(\rho + 1/\eta_i^l\right)} \\ &\leq \frac{2\|\ell'(\cdot)\|_2}{m_i\left(\rho + 1/\eta_i^l\right)}.\end{aligned}$$

So, Lemma 2 holds. □

Appendix A.2. Proof of Theorem 1

Proof. In our quantile loss, we have a subgradient of $\rho_\tau(u)$, $\rho_\tau'(u) = \tau - I_{\{u\leq 0\}}$, which is bounded. Based on $\|\boldsymbol{A}_{ij}\|_2$, uniformly bounded, we have the subgradient of $\rho_\tau(y_{ij} - \boldsymbol{A}_{ij}\boldsymbol{w})$ with regard to $\boldsymbol{w}$, bounded. The privacy loss from $\tilde{\boldsymbol{w}}_i^l$ is calculated as:

$$\left|\ln\frac{\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i\right]}{\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i'\right]}\right| = \left|\ln\frac{\Pr\left[\tilde{w}_i^{l(h)} \mid \mathcal{A}_i\right]}{\Pr\left[\tilde{w}_i^{l(h)} \mid \mathcal{A}_i'\right]}\right| = \left|\ln\frac{\Pr\left[\xi_i^{l(h)}\right]}{\Pr\left[\xi_i^{l,\prime(h)}\right]}\right|,$$

where $\xi_i^{l(h)}$ and $\xi_i^{l\prime(h)}$ are the h-entry of $\boldsymbol{\xi}_i^l$ and $\boldsymbol{\xi}_i^{l,\prime}$ and are sampled from $\mathcal{N}\left(0, \sigma_{i,l}^2\right)$. The numerator in the ratio above describes the probability of seeing $\tilde{\boldsymbol{w}}_i^l$ when the database is $\mathcal{A}_i$, the denominator corresponds the probability of seeing this same value when the database is $\mathcal{A}_i'$. This leads to:

$$\begin{aligned}\left|\ln\frac{\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i\right]}{\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i'\right]}\right| &= \left|\frac{1}{2\sigma_{i,l}^2}\left(\left\|\xi_i^{l(h)}\right\|^2 - \left\|\xi_i^{l,\prime(h)}\right\|^2\right)\right| \\ &= \left|\frac{1}{2\sigma_{i,l}^2}\left(\left\|\xi_i^{l(h)}\right\|^2 - \left\|\xi_i^{l(h)} + \left(w_{i,\mathcal{A}_i}^{l(h)} - w_{i,\mathcal{A}_i'}^{l(h)}\right)\right\|^2\right)\right| \\ &= \left|\frac{1}{2\sigma_{i,l}^2}\left(2\xi_i^{l(h)}\left\|w_{i,\mathcal{A}_i}^{l(h)} - w_{i,\mathcal{A}_i'}^{l(h)}\right\| + \left\|w_{i,\mathcal{A}_i}^{l(h)} - w_{i,\mathcal{A}_i'}^{l(h)}\right\|^2\right)\right|.\end{aligned}$$

Since $\|\ell'(\cdot)\| \leq c_1$, according to Lemma 2, we have:

$$\|w_{i,\mathcal{A}_i}^{l(h)} - w_{i,\mathcal{A}_i'}^{l(h)}\| < \|\boldsymbol{w}_{i,\mathcal{A}_i}^l - \boldsymbol{w}_{i,\mathcal{A}_i'}^l\| \leq 2c_1/\left(m_i\left(\rho + 1/\eta_i^l\right)\right).$$

Thus, by letting $\sigma_{i,l} = 2c_1\sqrt{2\ln(1.25/\delta)}/\left(m_i\epsilon\left(\rho + 1/\eta_i^l\right)\right)$, we have:

$$\left|\ln\frac{\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i\right]}{\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i'\right]}\right| \leq \left|\frac{\xi_i^{l(h)}m_i\left(\rho + 1/\eta_i^l\right) + c_1}{4\ln(1.25/\delta)c_1/\epsilon^2}\right|.$$

When $\left|\xi_i^{l^{(h)}}\right| \leq (4\ln(1.25/\delta)c_1/\epsilon - c_1)/\left(\epsilon m_i\left(\rho + 1/\eta_i^l\right)\right)$, $\left|\ln\left(\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i\right]/\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i'\right]\right)\right|$ is bounded by ϵ. Next, we need to prove that

$$\Pr\left[\left|\xi_i^{l^{(h)}}\right| > (4\ln(1.25/\delta)c_1/\epsilon - \ c_1)/\left(\epsilon m_i\left(\rho + 1/\eta_i^l\right)\right)\right] \leq \delta,$$

which requires $\Pr\left[\xi_i^{l^{(h)}} > \ (4\ln(1.25/\delta)c_1/\epsilon - c_1)/\left(\epsilon m_i\left(\rho + 1/\eta_i^l\right)\right)\right] \leq \delta/2$. According to the tail bound of normal distribution $\mathcal{N}\left(0, \sigma_{i,l}^2\right)$, we have

$$\Pr\left[\xi_i^{l^{(h)}} > r\right] \leq \frac{\sigma_{i,l}}{r\sqrt{2\pi}} e^{-r^2/2\sigma_{i,l}^2}.$$

By letting $r = (4\ln(1.25/\delta)c_1/\epsilon - c_1)/\left(\epsilon m_i\left(\rho + 1/\eta_i^l\right)\right)$ in the above inequality, we have:

$$\begin{aligned}&\Pr\left[\xi_i^{l^{(h)}} > \frac{4\ln(1.25/\delta)c_1/\epsilon - c_1}{m_i(\rho + 1/\eta_i^l)}\right]\\ \leq &\frac{2\sqrt{2\ln(1.25/\delta)}}{(4\ln(1.25/\delta) - \epsilon)\sqrt{2\pi}} \exp\left(-\frac{(4\ln(1.25/\delta) - \epsilon)^2}{8\ln(1.25/\delta)}\right).\end{aligned}$$

When δ is small (≤ 0.01) and $\epsilon \leq 1$, we have:

$$\frac{2\sqrt{2\ln(1.25/\delta)}}{(4\ln(1.25/\delta) - \epsilon)\sqrt{2\pi}} < \frac{1}{\sqrt{2\pi}}$$

and

$$-\frac{(4\ln(1.25/\delta) - \epsilon)^2}{8\ln(1.25/\delta)} < \ln\left(\sqrt{2\pi}\frac{\delta}{2}\right).$$

As a result, we have:

$$\Pr\left[\xi_i^{l^{(h)}} > \frac{4\ln(1.25/\delta)c_1/\epsilon - c_1}{m_i(\rho + 1/\eta_i^l)}\right] < \frac{\delta}{2}.$$

So far, we have proved that $\Pr\left[\xi_i^{l^{(h)}} > (4\ln(1.25/\delta)c_1/\epsilon - \ c_1)/\left(\epsilon m_i\left(\rho + 1/\eta_i^l\right)\right)\right] \leq \delta/2$; thus, we can prove that $\Pr\left[\left|\xi_i^{l^{(h)}}\right| > (4\ln(1.25/\delta)c_1/\epsilon - c_1)/\left(\epsilon m_i\left(\rho + 1/\eta_i^l\right)\right)\right] \leq \delta$. Define:

$$\mathbb{D}_1 = \left\{\xi_i^{l^{(h)}} : \left|\xi_i^{l^{(h)}}\right| \leq \frac{4\ln(1.25/\delta)c_1/\epsilon - c_1}{m_i(\rho + 1/\eta_i^l)}\right\},$$

$$\mathbb{D}_2 = \left\{\xi_i^{l^{(h)}} : \left|\xi_i^{l^{(h)}}\right| > \frac{4\ln(1.25/\delta)c_1/\epsilon - c_1}{m_i(\rho + 1/\eta_i^l)}\right\}.$$

Therefore, we obtain:

$$\begin{aligned}\Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i\right] = &\Pr\left[w_{i,\mathcal{A}_i}^{l^{(h)}} + \xi_i^{l^{(h)}} : \xi_i^{l^{(h)}} \in \mathbb{D}_1\right]\\ &+ \Pr\left[w_{i,\mathcal{A}_i}^{l^{(h)}} + \xi_i^{l^{(h)}} : \xi_i^{l^{(h)}} \in \mathbb{D}_2\right]\\ <&e^{\epsilon} \cdot \Pr\left[\tilde{\boldsymbol{w}}_i^l \mid \mathcal{A}_i'\right] + \delta,\end{aligned}$$

which proves that each iteration of DP-ADMM guarantees (ϵ, δ)-differential privacy. □

Appendix A.3. Proof of Theorem 2

Proof. According to the convexity of $f_i(\cdot)$, the monotonicity of the operator $F(\cdot)$, and applying Lemma 3, we have:

$$\begin{aligned}
&\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i)+\left(\overline{\boldsymbol{u}}_i^L-\boldsymbol{u}_i\right)^\top F\left(\overline{\boldsymbol{u}}_i^L\right)\right)\\
=&\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i)+\left\langle-\overline{\boldsymbol{\gamma}}_i^L,\overline{\boldsymbol{w}}_i^L-\boldsymbol{w}_i\right\rangle+\left\langle\overline{\boldsymbol{\gamma}}_i^L,\overline{\boldsymbol{w}}^L-\boldsymbol{w}\right\rangle+\left\langle\overline{\boldsymbol{\gamma}}_i^L-\boldsymbol{\gamma}_i,\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right\rangle\right)\\
\le&\frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{M}\left(f_i\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-f_i(\boldsymbol{w}_i)+\left(\boldsymbol{u}_i^l-\boldsymbol{u}_i\right)^\top F\left(\boldsymbol{u}_i^l\right)\right)\\
=&\frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{M}\left(f_i\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-f_i(\boldsymbol{w}_i)+\left\langle-\boldsymbol{\gamma}_i^l,\tilde{\boldsymbol{w}}_i^l-\boldsymbol{w}_i\right\rangle+\left\langle\boldsymbol{\gamma}_i^l,\boldsymbol{w}^l-\boldsymbol{w}\right\rangle+\left\langle\boldsymbol{\gamma}_i^l-\boldsymbol{\gamma}_i,\tilde{\boldsymbol{w}}_i^l-\boldsymbol{w}^l\right\rangle\right)\\
\le&\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\left(\frac{\eta_i^l}{2}\left\|f_i'\left(\tilde{\boldsymbol{w}}_i^l\right)-\left(\rho+1/\eta_i^l\right)\boldsymbol{\xi}_i^l\right\|^2-\left(\rho+1/\eta_i^l\right)\left\langle\boldsymbol{\xi}_i^l,\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^{l-1}\right\rangle\right)\\
&+\frac{1}{L}\sum_{i=1}^{M}\left(\frac{1}{2\eta_i^L}\left\|\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^0\right\|^2+\frac{\rho}{2}\left\|\boldsymbol{w}_i-\boldsymbol{w}^0\right\|^2+\frac{1}{2\rho}\left\|\boldsymbol{\gamma}_i-\boldsymbol{\gamma}_i^0\right\|^2\right).
\end{aligned}$$

Let $(\boldsymbol{w}_i,\boldsymbol{w})$ be the optimal solution $\left(\boldsymbol{w}_i^*,\boldsymbol{w}^*\right)$ in the above inequality. We obtain:

$$\begin{aligned}
&\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i^*)+\left\langle-\bar{\boldsymbol{\gamma}}_i^L,\overline{\boldsymbol{w}}_i^L-\boldsymbol{w}_i^*\right\rangle+\left\langle\bar{\boldsymbol{\gamma}}_i^L,\overline{\boldsymbol{w}}^L-\boldsymbol{w}^*\right\rangle+\left\langle\bar{\boldsymbol{\gamma}}_i^L-\boldsymbol{\gamma}_i,\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right\rangle\right)\\
\le&\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\frac{\eta_i^l}{2}\left\|f_i'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-\left(\rho+1/\eta_i^l\right)\xi_i^l\right\|^2-\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\left(\rho+1/\eta_i^l\right)\left\langle\xi_i^l,\boldsymbol{w}_i^*-\tilde{\boldsymbol{w}}_i^{l-1}\right\rangle\\
&+\frac{1}{L}\sum_{i=1}^{M}\frac{c_w^2}{2\eta_i^L}+\frac{M}{L}\frac{\rho}{2}c_w^2+\frac{1}{L}\sum_{i=1}^{M}\frac{1}{2\rho}\left\|\boldsymbol{\gamma}_i-\boldsymbol{\gamma}_i^0\right\|^2.
\end{aligned}$$

The above inequality holds for all $\boldsymbol{\gamma}_i$; thus, it also holds for $\boldsymbol{\gamma}_i\in\{\boldsymbol{\gamma}_i:\|\boldsymbol{\gamma}_i\|\le g\}$. By letting $\boldsymbol{\gamma}_i$ be the optimal solution, we have the maximum of the left side of the above inequality:

$$\begin{aligned}
&\max_{\{\boldsymbol{\gamma}_i:\|\boldsymbol{\gamma}_i\|\le g\}}\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i^*)+\left\langle-\bar{\boldsymbol{\gamma}}_i^L,\overline{\boldsymbol{w}}_i^L-\boldsymbol{w}_i^*\right\rangle+\left\langle\bar{\boldsymbol{\gamma}}_i^L,\overline{\boldsymbol{w}}^L-\boldsymbol{w}^*\right\rangle+\left\langle\bar{\boldsymbol{\gamma}}_i^L-\boldsymbol{\gamma}_i,\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right\rangle\right)\\
=&\max_{\{\boldsymbol{\gamma}_i:\|\boldsymbol{\gamma}_i\|\le g\}}\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i)-\boldsymbol{\gamma}_i\left(\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right)\right)\\
=&\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i)-\max_{\{\boldsymbol{\gamma}_i:\|\boldsymbol{\gamma}_i\|\le g\}}\boldsymbol{\gamma}_i\left(\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right)\right)\\
=&\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i)+g\left(\left\|\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right\|\right)\right);
\end{aligned}$$

In addition, we also obtain the maximum of the right side as:

$$\begin{aligned}
&\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\frac{\eta_i^l}{2}\left\|f_i'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-\left(\rho+1/\eta_i^l\right)\xi_i^l\right\|^2-\sum_{i=1}^{l}\frac{1}{L}\sum_{l=1}^{L}\left(\rho+1/\eta_i^l\right)\left\langle\xi_i^l,\boldsymbol{w}_i^*-\tilde{\boldsymbol{w}}_i^{l-1}\right\rangle\\
&+\frac{1}{L}\sum_{i=1}^{M}\frac{c_w^2}{2\eta_i^L}+\frac{\rho M}{2L}c_w^2+\max_{\{\gamma_i:\|\gamma_i\|\le g\}}\frac{1}{L}\sum_{i=1}^{M}\frac{1}{2\rho}\left\|\gamma_i-\gamma_i^0\right\|^2\\
=&\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\frac{\eta_i^l}{2}\left\|f_i'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-\left(\rho+1/\eta_i^l\right)\xi_i^l\right\|^2-\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\left(\rho+1/\eta_i^l\right)\left\langle\xi_i^l,\boldsymbol{w}_i^*-\tilde{\boldsymbol{w}}_i^{l-1}\right\rangle\\
&+\frac{1}{L}\sum_{i=1}^{M}\frac{c_w^2}{2\eta_i^L}+\frac{\rho M}{2L}c_w^2+\frac{M}{L}\frac{g^2}{2\rho}.
\end{aligned}$$

Thus, we obtain:

$$\begin{aligned}
&\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i)+g\left\|\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right\|\right)\\
\le&\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\frac{\eta_i^l}{2}\left\|f_i'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-\left(\rho+1/\eta_i^l\right)\xi_i^l\right\|^2-\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\left(\rho+1/\eta_i^l\right)\left\langle\xi_i^l,\boldsymbol{w}_i^*-\tilde{\boldsymbol{w}}_i^{l-1}\right\rangle \quad \text{(A1)}\\
&+\frac{1}{L}\sum_{i=1}^{M}\frac{c_w^2}{2\eta_i^L}+\frac{\rho M}{2L}c_w^2+\frac{M}{L}\frac{g^2}{2\rho}.
\end{aligned}$$

Since $\|\ell'(\cdot)\|\le c_1$ and $\|R'(\cdot)\|\le c_2$, we have:

$$\mathbb{E}\left[\left\|f_i'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-\left(\rho+1/\eta_i^l\right)\xi_i^l\right\|^2\right]=(c_1+\lambda c_2/M)^2+8pc_1^2\ln(1.25/\delta)/\left(m_i^2\epsilon^2\right).$$

With $\mathbb{E}\left[\left\langle\xi_i^l,\boldsymbol{w}_i^*-\tilde{\boldsymbol{w}}_i^{l-1}\right\rangle\right]=0$ and $\eta_i^l=c_w\left(2l(c_1+\lambda c_2/M)^2+16lpc_1^2\ln(1.25/\delta)/\left(m_i^2\epsilon^2\right)\right)^{-\frac{1}{2}}$, by taking expectation of the previous inequality (A1), we obtain:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{M}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i^*)+g\left\|\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right\|\right)\right]\\
\le&\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\mathbb{E}\left[\frac{\eta_i^l}{2}\left\|f_i'\left(\tilde{\boldsymbol{w}}_i^{l-1}\right)-\left(\rho+1/\eta_i^l\right)\xi_i^l\right\|^2\right]-\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\left(\rho+1/\eta_i^l\right)\mathbb{E}\left[\left\langle\xi_i^l,\boldsymbol{w}_i^*-\tilde{\boldsymbol{w}}_i^{l-1}\right\rangle\right]\\
&+\frac{1}{L}\sum_{i=1}^{M}\frac{c_w^2}{2\eta_i^L}+\frac{\rho M}{2L}c_w^2+\frac{M}{L}\frac{g^2}{2\rho},
\end{aligned}$$

which leads to

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n}\left(f_i\left(\overline{\boldsymbol{w}}_i^L\right)-f_i(\boldsymbol{w}_i^*)+g\left\|\overline{\boldsymbol{w}}_i^L-\overline{\boldsymbol{w}}^L\right\|\right)\right]\\
=&\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\frac{c_w}{2\sqrt{2l}}\sqrt{(c_1+\lambda c_2/M)^2+\frac{8pc_1^2\ln(1.25/\delta)}{m_i^2\epsilon^2}}\\
&+\sum_{i=1}^{M}\frac{1}{L}\sum_{l=1}^{L}\frac{c_w\sqrt{2L}}{2}\sqrt{(c_1+\lambda c_2/M)^2+\frac{8pc_1^2\ln(1.25/\delta)}{m_i^2\epsilon^2}}+\frac{M\rho}{2t}c_w^2+\frac{Mg^2}{2\rho T}\\
=&\sum_{i=1}^{M}\frac{c_w}{2\sqrt{2}L}\sqrt{(c_1+\lambda c_2/M)^2+\frac{8pc_1^2\ln(1.25/\delta)}{m_i^2\epsilon^2}}\left(\sum_{l=1}^{L}\frac{1}{\sqrt{l}}+2\sqrt{L}\right)+\frac{M\rho}{2L}c_w^2+\frac{Mg^2}{2\rho L}\\
\le&\sum_{i=1}^{M}\frac{\sqrt{2}c_w}{\sqrt{L}}\sqrt{(c_1+\lambda c_2/M)^2+\frac{8pc_1^2\ln(1.25/\delta)}{m_i^2\epsilon^2}}+\frac{M\left(\rho c_w^2+g^2/\rho\right)}{2L}.
\end{aligned}$$

Thus, we complete the proof of Theorem 2. □

References

1. Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; Naor, M. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In Proceedings of the Advances in Cryptology—EUROCRYPT 2006, St. Petersburg, Russia, 28 May–1 June 2006; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4004, pp. 486–503.
2. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3876, pp. 265–284.
3. Dwork, C.; Rothblum, G.N.; Vadhan, S. *Boosting and Differential Privacy*; IEEE Computer Society: Los Alamitos, CA, USA, 2010.
4. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends® Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
5. Dwork, C.; Smith, A.; Steinke, T.; Ullman, J.; Vadhan, S. Robust Traceability from Trace Amounts. In Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 17–20 October 2015; pp. 650–669.
6. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 308–318.
7. Differential Privacy Team. Learning with Privacy at Scale. 2017. Available online: https://machinelearning.apple.com/research/learning-with-privacy-at-scale (accessed on 1 February 2022).
8. Ding, B.; Kulkarni, J.; Yekhanin, S. Collecting Telemetry Data Privately. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 3571–3580.
9. Erlingsson, U.; Korolova, A.; Pihur, V. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 1054–1067.
10. Wang, J.; Kolar, M.; Srebro, N.; Zhang, T. Efficient Distributed Learning with Sparsity. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Voulme 70, pp. 3636–3645.
11. He, Y.; Zhou, Y.; Feng, Y. Distributed Feature Selection for High-dimensional Additive Models. *arXiv* **2022**, arXiv:2205.07932.
12. Jordan, M.I.; Lee, J.D.; Yang, Y. Communication-Efficient Distributed Statistical Inference. *J. Am. Stat. Assoc.* **2019**, *114*, 668–681. [CrossRef]
13. Hu, A.; Jiao, Y.; Liu, Y.; Shi, Y.; Wu, Y. Distributed quantile regression for massive heterogeneous data. *Neurocomputing* **2021**, *448*, 249–262. [CrossRef]
14. Nedic, A.; Olshevsky, A.O.A.; Tsitsiklis, J.N. Distributed subgradient methods and quantization effects. In Proceedings of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2008; pp. 4177–4184.
15. Nedic, A.; Ozdaglar, A. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Trans. Autom. Control.* **2009**, *54*, 48–61. [CrossRef]
16. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends® Mach. Learn.* **2011**, *3*, 1–122.
17. Ling, Q.; Ribeiro, A. Decentralized linearized alternating direction method of multipliers. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 5447–5451.
18. Shi, W.; Ling, Q.; Yuan, K.; Wu, G.; Yin, W. On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. *IEEE Trans. Signal Process.* **2014**, *62*, 1750–1761. [CrossRef]
19. Zhang, R.; Kwok, J. Asynchronous distributed ADMM for consensus optimization. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; Volume 5, pp. 3689–3697.
20. Bianchi, P.; Hachem, W.; Iutzeler, F. A stochastic primal-dual algorithm for distributed asynchronous composite optimization. In Proceedings of the 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Atlanta, GA, USA, 3–5 December 2014; pp. 732–736.
21. Wei, E.; Ozdaglar, A. Distributed Alternating Direction Method of Multipliers. In Proceedings of the 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), Maui, HI, USA, 10–13 December 2012; pp. 5445–5450.
22. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 3–18.
23. Fredrikson, M.; Jha, S.; Ristenpart, T. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1322–1333.
24. Zhang, T.; Zhu, Q. Dynamic Differential Privacy for ADMM-Based Distributed Classification Learning. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 172–187. [CrossRef]
25. Guo, Y.; Gong, Y. Practical Collaborative Learning for Crowdsensing in the Internet of Things with Differential Privacy. In Proceedings of the 2018 IEEE Conference on Communications and Network Security (CNS), Beijing, China, 30 May–1 June 2018; pp. 1–9.

26. Ding, J.; Errapotu, S.M.; Zhang, H.; Gong, Y.; Pan, M.; Han, Z. Stochastic ADMM Based Distributed Machine Learning with Differential Privacy. *Secur. Priv. Commun. Netw.* **2019**, *304*, 257–277.
27. Ferraty, F.; Vieu, P. *Nonparametric Functional Data Analysis*; Springer: New York, NY, USA, 2006.
28. Ramsay, J.O.; Silverman, B.W. *Functional Data Analysis*, 2nd ed.; Springer: New York, NY, USA, 2005.
29. Zhang, J.T. *Analysis of Variance for Functional Data*; Chapman and Hall/CRC: New York, NY, USA, 2013.
30. Ramsay, J.O.; Silverman, B.W. *Applied Functional Data Analysis: Methods and Case Studies*; Springer: New York, NY, USA, 2002.
31. Ramsay, J.O.; Dalzell, C.J. Some Tools for Functional Data Analysis. *J. R. Stat. Soc. Ser. Methodol.* **1991**, *53*, 539–561. [CrossRef]
32. Tang, Q.; Kong, L. Quantile regression in functional linear semiparametric model. *Statistics* **2017**, *51*, 1342–1358.
33. Lu, Y.; Du, J.; Sun, Z. Functional partially linear quantile regression model. *Metrika* **2014**, *77*, 317–332. [CrossRef]
34. Auton, T. Applied Functional Data Analysis: Methods and Case Studies. *J. R. Stat. Soc. Ser. Stat. Soc.* **2004**, *167*, 378–379. [CrossRef]
35. Hall, P.; Horowitz, J.L. Methodology and convergence rates for functional linear regression. *Ann. Stat.* **2007**, *35*, 70–91. [CrossRef]
36. Karhunen, K. Zur Spektraltheorie stochastischer Prozesse. *Ann. Acad. Sci. Fenn.* **1946**, *1*, 7.
37. Tibshirani, R. Regression shrinkage and selection via the lasso: a retrospective. *J. R. Stat. Soc. Ser. Stat. Methodol.* **2011**, *73*, 273–282. [CrossRef]
38. Zou, H.; Hastie, T. Regularization and variable selection via the Elastic Net. *J. R. Stat. Soc. Ser. B* **2005**, *67*, 301–320. [CrossRef]
39. Zhang, C. Nearly unbiased variable selection under minimax concave penalty. *Ann. Stat.* **2010**, *38*. [CrossRef]
40. Fan, J.; Li, R. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *J. Am. Stat. Assoc.* **2001**, *96*, 1348–1360. [CrossRef]
41. Glowinski, R.; Marroco, A. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *ESAIM Math. Model. Numer. Anal.* **1975**, *9*, 41–76. [CrossRef]
42. Gabay, D.; Mercier, B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.* **1976**, *2*, 17–40. [CrossRef]
43. Huang, Z.; Gong, Y. Differentially Private ADMM for Convex Distributed Learning: Improved Accuracy via Multi-Step Approximation. *arXiv* **2020**, arXiv:2005.07890.
44. Deng, W.; Yin, W. On the Global and Linear Convergence of the Generalized Alternating Direction Method of Multipliers. *J. Sci. Comput.* **2016**, *66*, 889–916. [CrossRef]
45. Wang, H.; Banerjee, A. Bregman Alternating Direction Method of Multipliers. *Adv. Neural Inf. Process. Syst.* **2014**, *4*, 2816–2824.
46. Hong, M.; Luo, Z. On the Linear Convergence of the Alternating Direction Method of Multipliers. *Math. Program.* **2017**, *162*, 165–199. [CrossRef]
47. Eckstein, J.; Bertsekas, D.P. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.* **1992**, *55*, 293–318. [CrossRef]
48. He, B.; Yuan, X. On the $O(1/n)$ Convergence Rate of the Douglas-Rachford Alternating Direction Method. *SIAM J. Numer. Anal.* **2012**, *50*, 700–709. [CrossRef]
49. Sun, J.; Zhang, S. A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs. *Eur. J. Oper. Res.* **2010**, *207*, 1210–1220. [CrossRef]
50. Zhang, S.; Ang, J.; Sun, J. An alternating direction method for solving convex nonlinear semidefinite programming problems. *Optimization* **2013**, *62*, 527–543. [CrossRef]
51. Goldstein, T.; O'Donoghue, B.; Setzer, S.; Baraniuk, R. Fast Alternating Direction Optimization Methods. *SIAM J. Imaging Sci.* **2014**, *7*, 1588–1623. [CrossRef]
52. Gabay, D. Chapter IX Applications of the Method of Multipliers to Variational Inequalities. *Augment. Lagrangian Methods Appl. Solut. -Bound.-Value Probl.* **1983**, *15*, 299–331.
53. Li, G.; Pong, T. Global Convergence of Splitting Methods for Nonconvex Composite Optimization. *SIAM J. Optim.* **2015**, *25*, 2434–2460. [CrossRef]
54. Yu, L.; Lin, N. ADMM for Penalized Quantile Regression in Big Data. *Int. Stat. Rev.* **2017**, *85*, 494–518. [CrossRef]
55. Huang, Z.; Hu, R.; Guo, Y.; Chan-Tin, E.; Gong, Y. DP-ADMM: ADMM-Based Distributed Learning With Differential Privacy. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1002–1012. [CrossRef]