Stochastic ADMM Based Distributed Machine Learning with Differential Privacy

Jiahao Ding[1], Sai Mounika Errapotu[1], Haijun Zhang[2], Yanmin Gong[3], Miao Pan[1(✉)], and Zhu Han[1]

[1] Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204, USA
{jding7,serrapotu,mpan2,zhan2}@uh.edu
[2] Department of Communications Engineering, University of Science and Technology Beijing, Beijing 100083, China
haijunzhang@ieee.org
[3] Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249, USA
yanmin.gong@utsa.edu

Abstract. While embracing various machine learning techniques to make effective decisions in the big data era, preserving the privacy of sensitive data poses significant challenges. In this paper, we develop a privacy-preserving distributed machine learning algorithm to address this issue. Given the assumption that each data provider owns a dataset with different sample size, our goal is to learn a common classifier over the union of all the local datasets in a distributed way without leaking any sensitive information of the data samples. Such an algorithm needs to jointly consider efficient distributed learning and effective privacy preservation. In the proposed algorithm, we extend stochastic alternating direction method of multipliers (ADMM) in a distributed setting to do distributed learning. For preserving privacy during the iterative process, we combine differential privacy and stochastic ADMM together. In particular, we propose a novel stochastic ADMM based privacy-preserving distributed machine learning (PS-ADMM) algorithm by perturbing the updating gradients, that provide differential privacy guarantee and have a low computational cost. We theoretically demonstrate the convergence rate and utility bound of our proposed PS-ADMM under strongly convex objective. Through our experiments performed on real-world datasets, we show that PS-ADMM outperforms other differentially private ADMM algorithms under the same differential privacy guarantee.

Keywords: Differential privacy · Distributed machine learning · Stochastic ADMM · Moments accountant · Distributed optimization · Privacy

S. Chen et al. (Eds.): SecureComm 2019, LNICST 304, pp. 257–277, 2019.
https://doi.org/10.1007/978-3-030-37228-6_13

1 Introduction

Recently, with rapid advances in sensing technologies, we are witnessing a deluge of data [20,21]. Statistical analysis of this data has paved the way for the development of machine learning that brings valuable benefits to society, such as more intelligent autopilot technology and higher medical quality, among others. The enormous data generated from such various applications is scattered around different places, and it increasingly becomes difficult for a single machine to process such giant data. Hence, the centralized model can no longer efficiently process this data [2,3]. Apart from the limitations on processing, this data draws detailed pictures of people's lives and involves highly sensitive information. So the data owners might be reluctant to share their data for analysis. Therefore, with the rise in the volume of data being generated, there is a critical need of privacy-preserving machine learning algorithms that can both cater the processing and privacy needs.

To address the above issues, we develop a privacy-preserving machine learning algorithm that processes data in a distributed manner while providing privacy guarantee for each training sample. One of the promise applications is the health domain. For example, in health monitoring applications multiple hospitals collaborate to provide constructive diagnosis to the patients. Hospitals have a large number of cases, and the data analysis of these cases helps doctors to make an accurate diagnosis and offer early treatment plans. Thus, multiple hospitals could collaboratively train a classifier through a central server that can help in prognosis and diagnosing diseases early. However, such medical cases may contains sensitive information about the patients and each hospital cannot share its patients' cases with other hospitals. Hence, the key challenge is to effectively conduct medical research while preserving the privacy of the patients in the analysis. Concisely, this problem is a distributed machine learning problem where data is collected from multiple data providers and each data sample's privacy needs to be guaranteed during the optimization.

One of the promising solutions for such distributed machine learning problems is alternating direction method of multipliers (ADMM) [4,14,15]. ADMM enables distributed learning by decomposing a large-scale optimization into smaller subproblems and each subproblem is easy to solve in a distributed and parallel way. Each data provider uses its own private data to train a local classifier and the central server averages all of the local classifiers and broadcasts the result to the data providers. These steps iterate several times until the server and users have a high-performance model. Through this decomposition and coordination procedure of ADMM, the distributed learning problem achieves effective results. However, when both the number of features and the size of dataset are large, the computational burden of using ADMM is heavy [17]. Recently Zhang et al. propose a novel ADMM algorithm called SCAS-ADMM, which achieves lower computational burden by employing stochastic variance reduced gradient (SVRG) [13] as an inexact solver for subproblems. Zhang et al. considered the SCAS-ADMM in the centralized scenario [23]. We investigate on the SCAS-ADMM in distributed machine learning scenario to obtain a low computational cost per iteration, without compromising the privacy of data samples.

Beyond effectively solving the distributed machine learning problem, the data privacy is the critical concern in such analyses since the private information pertaining to the datasets should not to be shared and kept private. But the privacy concerns are still inherent during the communication between the data providers and the central server. As each data provider needs to share the local model trained over the sensitive raw data at each iteration, an adversary could infer the sensitive information from the shared model as described in [8]. Therefore, we use differential privacy [6,7], a de facto notion for privacy that offers strong privacy guarantees, to tackle the privacy concerns and protect disclosed privacy from the model parameters during the iterative procedure. Differential privacy guarantees privacy by measuring the change in the outcome of the algorithm as the presence or absence of a single data entry in the original dataset does not explicitly change the outcome. In this work we investigate on collectively considering differential privacy and distributed machine learning to get effective results in the analysis, with low computation burden and without compromising the privacy of the data owners. In the existing literature, there are some research efforts integrating ADMM into private distributed learning. Zhang et al. developed a dual perturbation based on ADMM [22], in which they add noise to the dual variables of decentralized ADMM and only provide privacy guarantee of a single data provider per iteration, but their decentralized algorithm needs robust network topology and does not guarantee utility and privacy when considering all nodes during the whole training procedure. Guo et al. proposed another approach for preserving privacy in ADMM in [10], which incorporate secure computation and distributed noise generation in the asynchronous ADMM algorithm. Though privacy during communication can be preserved, their scheme suffers from poor communication and computation costs because of the encryption and decryption over huge datasets.

To address these challenges, we propose a novel stochastic ADMM based privacy-preserving distributed machine learning (PS-ADMM) algorithm in this paper, which jointly considers the distributed learning setting and differential privacy. In PS-ADMM, we employ differential privacy to stochastic ADMM algorithm with the objective of protecting the privacy of data samples and achieving distributed learning over multiple data providers. Different from the approach proposed in [22], we propose to extend the stochastic ADMM in a distributed setting to deal with the computational burden of local computation at each data provider, and add differential privacy based noise to the updating gradients during local computation procedure. We utilize the moments accountant method [1] to analyze the privacy guarantee of PS-ADMM, and we also provide the convergence rate and utility bound of PS-ADMM. The major contributions of this paper are listed as follows.

– We design a novel stochastic ADMM based privacy-preserving distributed machine learning algorithm called PS-ADMM, where we investigate the SCAS-ADMM algorithm in a distributed setting and perturb the gradient updates with Gaussian noise to further improve the computational efficiency and provide differential privacy guarantee.

– Compared to the existing research in [22] that only considers privacy guarantee at each iteration, we consider the entire iterative procedure and adopt moments accountant method to provide a tighter differential privacy guarantee for PS-ADMM.
– We theoretically analyze and prove the convergence and utility bound of the proposed algorithm PS-ADMM.
– We show that the proposed PS-ADMM outperforms other differentially private ADMM algorithms under the same differential privacy guarantee by conducting PS-ADMM over real-world data.

The remainder of this paper is organized as follows. Section 2 presents the problem statement, preliminaries and associated privacy concerns. We propose our differentially private algorithm PS-ADMM in Sect. 3. This is followed by our theoretical analysis of convergence and utility bound in Sect. 4. Detailed simulations and comparisons are presented in Sect. 5. Section 6 concludes the whole paper.

2 Problem Statement and Preliminaries

In this section, we describe the problem statement in Sect. 2.1, introduce the preliminaries of ADMM and differential privacy in Sect. 2.2. The overview of the distributed SCAS-ADMM algorithm is presented in Sect. 2.3 and the privacy concerns of the distributed ADMM based solution are presented in Sect. 2.4.

2.1 Problem Statement

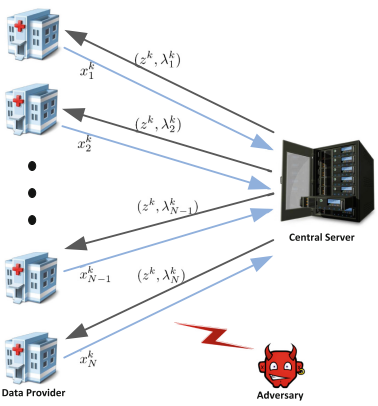


Fig. 1. System architecture

As shown in Fig. 1, we consider a star network topology consisting of a set of data providers $\mathcal{N} = \{1, \cdots, N\}$ and a central server, where multiple data providers have the ability to communicate with the server and the server is

responsible for aggregation and message passing. Here, each data provider possesses a private dataset $D_i = \{(\boldsymbol{a}_{im}, y_{im})\}_{m=1}^{M}$ consisting of feature vector $\boldsymbol{a}_{im}$ from a data universe $\mathcal{X}$, and $y_{im} \in \mathcal{Y}$ that is a label we aim to predict from $\boldsymbol{a}_{im}$. The objective of our problem is to build a classifier over the aggregated sensitive dataset $\cup_{i\in\mathcal{N}}\{D_i\}$ from data providers through a distributed manner, where the classifier can be obtained by minimizing a regularized empirical risk minimization problem (ERM) [9]. The regularized empirical risk minimization problem is to learn a classifier $\boldsymbol{x}$ over a convex set $\mathcal{C} \subseteq \mathbb{R}^p$, which can be formulated as

$$\min_{\boldsymbol{x}\in\mathcal{C}} \sum_{i=1}^{N} \frac{1}{M} \sum_{m=1}^{M} l_{im}[\boldsymbol{x}, (\boldsymbol{a}_{im}, y_{im})] + g(\boldsymbol{x}), \tag{1}$$

where $l_{im}(\cdot) : \mathcal{X} \times \mathcal{Y} \times \mathcal{C} \to \mathbb{R}$ is a loss function of provider i for each data sample $(\boldsymbol{a}_{im}, y_{im})$ and $g(\boldsymbol{x})$ is a convex regularizer to prevent overfitting. In this paper, we assume the loss function $l_{im}(\cdot)$ is convex, G-Lipschitz and has L_m-Lipschitz continuous gradient. Note that our algorithm is not limited to the classification problem since the convergence and privacy analysis are still valid.

The above ERM problem (1) can be minimized by ADMM, which is a practical distributed scheme that can be applied to large-scale machine learning algorithms. Since the goal is to build a classifier with sensitive data, privacy concerns inherent in the training procedure need to be addressed while solving the ERM problem.

2.2 Preliminaries

Distributed Machine Learning with ADMM. In order to solve the problem in (1) with ADMM method [4], the ERM problem in (1) can be reformulated as consensus formulation [18] by introducing a global variable $\boldsymbol{z} \in \mathbb{R}^p$ as

$$\min_{\substack{\boldsymbol{x}_i, \boldsymbol{z}\in\mathcal{C} \\ i=1,\ldots,N}} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) + g(\boldsymbol{z}) \tag{2}$$

$$\text{s.t.} \quad \boldsymbol{x}_i - \boldsymbol{z} = 0, \quad \forall i = 1, ..., N. \tag{3}$$

In (2), $f_i(\boldsymbol{x}_i) = \frac{1}{M}\sum_{m=1}^{M} l_{im}[\boldsymbol{x}_i, (\boldsymbol{a}_{im}, y_{im})]$ is the i-th data provider's loss function due to dataset D_i, and $\boldsymbol{x}_i$ is the local classifier of the i-th data provider. Since the objective function in (2) is already decoupled, each data provider only needs to optimize a subproblem, i.e., empirical risk minimization problem over its local dataset. The constraints (3) enforce that all the local classifiers reach consensus finally. Apparently, the problem above is equivalent to the problem in (1).

Let $\boldsymbol{\lambda} \in \mathbb{R}^p$ denote the Lagrange dual variable, and $\rho > 0$ be a pre-defined penalty parameter. The standard ADMM consists of the following iterations

$$\boldsymbol{x}_i^{k+1} = \arg\min_{\boldsymbol{x}_i} \; f_i(\boldsymbol{x}_i) + (\boldsymbol{x}_i - \boldsymbol{z}^k)^T \boldsymbol{\lambda}_i^k + \frac{\rho}{2}\|\boldsymbol{x}_i - \boldsymbol{z}^k\|^2, \tag{4}$$

$$\boldsymbol{z}^{k+1} = \arg\min_{\boldsymbol{z}} \; g(\boldsymbol{z}) + \sum_{i=1}^{N}(-\boldsymbol{z}^T \boldsymbol{\lambda}_i^k + \frac{\rho}{2}\|\boldsymbol{x}_i^{k+1} - \boldsymbol{z}\|^2), \tag{5}$$

$$\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{x}_i^{k+1} - \boldsymbol{z}^{k+1}), \tag{6}$$

where $\|\cdot\|$ denotes l_2 norm.

The entire procedure illustrates the exchange of information between data providers and the central server. It is obvious that the classifier $\boldsymbol{x}_i^{k+1}$ can be locally updated for each party. This is because the whole problem has been divided into N subproblems which can be solved in parallel. Each party broadcasts $\boldsymbol{x}_i^{k+1}$ it owns to the central server. Then, the central server solves subproblems (5) and (6) and gets $\boldsymbol{z}^{k+1}$ and then dual variable $\boldsymbol{\lambda}_i^{k+1}$. Finally, the optimal parameter can be obtained after several iterations.

Differential Privacy. Differential privacy [7] is a widely-adopted privacy notion, which can be used to quantify the privacy risk of each individual record in a dataset. Mathematically, differential privacy is defined as follows

Definition 1. *A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all datasets $D, D' \in \mathbb{D}$ that differ in a single element and for all $s \in \Omega$, where Ω is the output space of $\mathcal{A}$, we have*

$$Pr(\mathcal{A}(D) = s) \leq e^{\epsilon} Pr(\mathcal{A}(D') = s) + \delta.$$

Differential privacy concentrates on the output distribution of a mechanism when there exists the participation of an individual. Smaller values of ϵ mean stronger privacy guarantees of $\mathcal{A}$. The most common mechanism for achieving differential privacy is Gaussian Mechanism [6].

Definition 2. *Consider a function $q: \mathbb{D} \to \mathcal{R}^p$ whose l_2-sensitivity is $\Delta_2(q) = sup_{D \sim D'}\|q(D) - q(D')\|$. The Gaussian Mechanism is defined as: $\mathcal{M}(D, q, \epsilon) = q(D) + \mathcal{N}(0, \sigma^2 I_p)$ where $\mathcal{N}(0, \sigma^2 I_p)$ is a zero mean isotropic Gaussian Distribution with $\sigma \geq \frac{\sqrt{2\ln(1.25/\delta)}\Delta_2(q)}{\epsilon}$. Then, the Gaussian mechanism preserves (ϵ, δ)-differential privacy.*

2.3 Distributed Stochastic ADMM

Traditional ADMM [4,22] is quite computationally expensive when we have large size of the dataset, since solving subproblem (4) needs to visit all the M data points at each iteration. In this paper, we extend the stochastic ADMM (SCAS-ADMM [23]) into a distributed setting to highly reduce the computation cost.

Algorithm 1. Distributed Stochastic ADMM

Algorithm of the i-th data provider:
Input: Dataset $D_i = \{(\boldsymbol{a}_{im}, y_{im})\}_{m=1}^{M}$, initialize x_i^0 for all agents i, $\zeta = 2\eta - \frac{4\eta}{1-2\eta v_L}$, $\xi = \frac{4\eta}{1-2\eta v_L}$.
for $k = 0, 1, \cdots, K-1$ **do**
  Compute $\hat{\boldsymbol{u}}_i = \nabla f_i(\boldsymbol{x}_i^k) = \frac{1}{M}\sum_{m=1}^{M} \nabla l_{im}(\boldsymbol{x}_i^k)$;
  $\tilde{\boldsymbol{v}}_i = \boldsymbol{x}_i^k$;
  $\boldsymbol{v}_i^0 = \tilde{\boldsymbol{v}}_i$;
  for $s = 1, \cdots, S-1$ **do**
    Randomly pick a data point $(\boldsymbol{a}_{im^s}, y_{im^s}) \in D_i$;
    $\boldsymbol{g}_i^s = \nabla l_{im^s}(\boldsymbol{v}_i^s) - \nabla l_{im^s}(\tilde{\boldsymbol{v}}_i) + \hat{\boldsymbol{u}}_i + \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{v}_i^s - \boldsymbol{z}^k)$
    $\boldsymbol{v}_i^{s+1} = \boldsymbol{v}_i^s - \eta \boldsymbol{g}_i^s$;
    $\hat{\boldsymbol{v}}_i^{s+1} = \frac{\zeta \boldsymbol{v}_i^s + \xi \boldsymbol{v}_i^{s+1}}{2\eta}$;
  end for
  $\boldsymbol{x}_i^{k+1} = \frac{1}{s}\sum_{s=0}^{S-1} \hat{\boldsymbol{v}}_i^{s+1}$;
  Send $\boldsymbol{x}_i^{k+1}$ to the central server;
end for
Algorithm of the central server:
Initialize $\boldsymbol{z}^0, \boldsymbol{\lambda}_i^0$ and broadcast them to the providers;
for $k = 0, 1, \cdots, K-1$ **do**
  $\boldsymbol{z}^{k+1} = \arg\min_z g(\boldsymbol{z}) + \sum_{i=1}^{N}(-\boldsymbol{z}^T\boldsymbol{\lambda}_i^k + \frac{\rho}{2}\|\boldsymbol{x}_i^{k+1} - \boldsymbol{z}\|^2)$;
  $\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{x}_i^{k+1} - \boldsymbol{z}^{k+1})$;
end for
Output: $\{\boldsymbol{x}_i^K\}_{i=1}^N$, $\boldsymbol{z}^K$;

Since distributed SCAS-ADMM just needs to utilize several data points at each iteration to achieve distributed learning, it is quite computation efficient[1].

Before stating the details of distributed stochastic ADMM, we first define the following functions

$$L_i(\boldsymbol{x}_i) = f_i(\boldsymbol{x}_i) + g(\boldsymbol{z}^k) + (\boldsymbol{x}_i - \boldsymbol{z}^k)^T\boldsymbol{\lambda}_i^k + \frac{\rho}{2}\|\boldsymbol{x}_i - \boldsymbol{z}^k\|^2,$$
$$\hat{L}_{im}(\boldsymbol{x}_i) = l_{im}(\boldsymbol{x}_i) + g(\boldsymbol{z}^k) + (\boldsymbol{x}_i - \boldsymbol{z}^k)^T\boldsymbol{\lambda}_i^k + \frac{\rho}{2}\|\boldsymbol{x}_i - \boldsymbol{z}^k\|^2.$$

The following Lemma shows the convexity of above functions.

Lemma 1. *If $f_i(\cdot)$ is μ_f-strongly convex, and $l_{im}(\cdot)$ is convex, G-Lipschitz and has L_m-Lipschitz continuous gradient, then we have $f_i(\boldsymbol{x})$ is v_f-smooth, where $v_f = \max_m L_m$, and $L_i(\boldsymbol{x})$ is both v_L-smooth and μ_L-strongly convex. Moreover, $\hat{L}_{im}(\boldsymbol{x}_i)$ is v_L-smooth.*

Proof. See Appendix A.1.

The details of distributed stochastic ADMM are summarized in Algorithm 1. To be specific, after receiving updated variable $\boldsymbol{z}^k$ and $\boldsymbol{\lambda}_i^k$ from the server,

[1] The specific results of computation cost and memory cost refer to [23].

each data provider updates its local variable $\boldsymbol{x}_i^{k+1}$ at iteration k by optimizing subproblem (4) through the SVRG method [13]. At the beginning of each iteration k, the gradient $\hat{\boldsymbol{u}}_i = \nabla f_i(\boldsymbol{x}_i^k) = \frac{1}{M}\sum_{m=1}^{M} \nabla l_{im}(\boldsymbol{x}_i^k)$ is computed using a past parameter estimate $\boldsymbol{x}_i^k$. For each inner iteration s, the approximate gradient $\boldsymbol{g}_i^s = \nabla l_{im^s}(\boldsymbol{v}_i^s) - \nabla l_{im^s}(\tilde{\boldsymbol{v}}_i) + \hat{\boldsymbol{u}}_i + \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{v}_i^s - \boldsymbol{z}^k)$ is used to iteratively update $\boldsymbol{v}_i^{s+1}$ with a step size η. And then, we adopt the convex combination to improve the convergence rate. Hence, the subproblem (4) reduces to $\boldsymbol{x}_i^{k+1} = \frac{1}{s}\sum_{s=0}^{S-1} \hat{\boldsymbol{v}}_i^{s+1}$. Then, all the data providers broadcast their $\boldsymbol{x}_i^{k+1}$ to the central server which computes $\boldsymbol{z}^{k+1}$ and $\boldsymbol{\lambda}_i^{k+1}$. The whole procedure ends when the number of iterations exceeds a maximum value K. However, while there is no direct exchange of data among data providers, the sequence of iterations broadcasted by a provider may reveal sensitive information through the output of the local learning.

2.4 Privacy Concerns

In our problem setting, there is no need to send the dataset stored at each data provider to the central server directly, while the risk of information leakage still exists. We assume that an adversary can eavesdrop all communications between data providers and the server. In some cases, the adversary using model inversion attack [8] may be able to obtain sensitive information about the private data points of the training dataset by observing the local learning parameter from the provider at iteration k and the final output model parameters of the distributed algorithm. To mitigate this risk, we develop a differentially private algorithm that provides differential privacy for all of the intermediate parameters. If the adversary collects all the intermediate computational results of a provider during communications with the server and the final output of the algorithm, the privacy of local data points at each data provider is still protected.

3 Distributed Stochastic ADMM with Differential Privacy

In this section, we propose our novel algorithm PS-ADMM, which integrates differential privacy into distributed stochastic ADMM. In order to provide differential privacy in distributed stochastic ADMM algorithm, we use the noisy gradient that adds Gaussian noise to the gradient updates of subproblem (4). To analyze the privacy guarantee of PS-ADMM, we consider the moments accountant method [1] of computing privacy loss during a iterative process, which is shown in Theorem 1.

Theorem 1. *There exist constants c_1 and c_2 such that given the sampling probability $q = l/M$ and the number of steps K, for any $\epsilon < c_1 q^2 K$ and for the G-Lipschitz loss function, a differentially private stochastic gradient algorithm with batch size l that injects Gaussian Noise with standard deviation $G\sigma$ to the*

Algorithm 2. Differentially Private Stochastic ADMM (PS-ADMM)

Algorithm of the i-th data provider:
Input: Dataset $D_i = \{(\boldsymbol{a}_{im}, y_{im})\}_{m=1}^{M}$, initialize x_i^0 for all agents i, $\zeta = 2\eta - \frac{4\eta}{1-2\eta v_L}$, $\xi = \frac{4\eta}{1-2\eta v_L}$.
for $k = 0, 1, \cdots, K-1$ **do**
  Compute $\hat{\boldsymbol{u}}_i = \nabla f_i(\boldsymbol{x}_i^k) = \frac{1}{M}\sum_{m=1}^{M} \nabla l_{im}(\boldsymbol{x}_i^k)$;
  $\tilde{\boldsymbol{v}}_i = \boldsymbol{x}_i^k$;
  $\boldsymbol{v}_i^0 = \tilde{\boldsymbol{v}}_i$;
  for $s = 1, \cdots, S-1$ **do**
    Generate Gaussian noise: $\theta_k^s \sim \mathcal{N}(0, (\sigma^2)_k^s \boldsymbol{I}_p)$;
    Randomly pick a data point $(\boldsymbol{a}_{im^s}, y_{im^s}) \in D_i$;
    $\boldsymbol{g}_i^s = \nabla l_{im^s}(\boldsymbol{v}_i^s) - \nabla l_{im^s}(\tilde{\boldsymbol{v}}_i) + \hat{\boldsymbol{u}}_i + \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{v}_i^s - \boldsymbol{z}^k) + \theta_k^s$
    $\boldsymbol{v}_i^{s+1} = \boldsymbol{v}_i^s - \eta \boldsymbol{g}_i^s$;
    $\hat{\boldsymbol{v}}_i^{s+1} = \frac{\zeta \boldsymbol{v}_i^s + \xi \boldsymbol{v}_i^{s+1}}{2\eta}$;
  end for
  $\boldsymbol{x}_i^{k+1} = \frac{1}{s}\sum_{s=0}^{S-1} \hat{\boldsymbol{v}}_i^{s+1}$;
  Send $\boldsymbol{x}_i^{k+1}$ to the central server;
end for
Algorithm of the central server:
Initialize $\boldsymbol{z}^0, \boldsymbol{\lambda}_i^0$ and broadcast them to the providers;
for $k = 0, 1, \cdots, K-1$ **do**
  $\boldsymbol{z}^{k+1} = \arg\min_z g(\boldsymbol{z}) + \sum_{i=1}^{N}(-\boldsymbol{z}^T\boldsymbol{\lambda}_i^k + \frac{\rho}{2}\|\boldsymbol{x}_i^{k+1} - \boldsymbol{z}\|^2)$;
  $\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{x}_i^{k+1} - \boldsymbol{z}^{k+1})$;
end for
Output: $\{\boldsymbol{x}_i^K\}_{i=1}^N$, $\boldsymbol{z}^K$;

gradients, is (ϵ, δ)*-differentially private for any* $\delta > 0$*, if we choose*

$$\sigma \geq c_2 \frac{q\sqrt{K\log(1/\delta)}}{\epsilon}. \tag{8}$$

The differentially private stochastic ADMM (PS-ADMM) is shown in Algorithm 2. Details of PS-ADMM are summarized as follows: At iteration k, each data provider utilizes the SVRG method to solve subproblem (4) in order to obtain the local classifier $\boldsymbol{x}_i^{k+1}$. For the inner iteration s at this iteration, the i-th data provider generates a zero mean Gaussian noise θ_k^s with variance $(\sigma^2)_k^s$ to perturb the approximate gradient $\boldsymbol{g}_i^s$, and by averaging $\hat{\boldsymbol{v}}_i^{s+1}$ of all S inner iterations, the i-th data provider gets a differential private local classifier $\boldsymbol{x}_i^{k+1}$. During the iteration of SVRG method, we adopt the convex combination to increase the convergence rate. In addition, we employ the iteration average to improve the convergence of ADMM. And then data providers send all differentially private $\{\boldsymbol{x}_i^{k+1}\}_{i=1}^N$ to the server. The server will update $\boldsymbol{z}^{k+1}$ and $\{\boldsymbol{\lambda}_i^{k+1}\}_{i=1}^N$ by solving subproblems (5) and (6) after receiving all of the local parameters$\{\boldsymbol{x}_i^{k+1}\}_{i=1}^N$. Next, each data provider updates its private local parameter by using updated variable $\boldsymbol{z}^{k+1}$ and $\{\boldsymbol{\lambda}_i^{k+1}\}_{i=1}^N$ from the central server. The iterative process will

continue until reaching K rounds of communication between server and data provider.

During this iterative process, the shared local classifiers $\{\boldsymbol{x}_i^{k+1}\}_{i=1}^N$ may reveal sensitive information about local dataset D_i of data provider i. Thus, we need to show that PS-ADMM guarantees differential privacy with local classifiers $\{\boldsymbol{x}_i^{k+1}\}_{i=1}^N$. Since we use Gaussian mechanism to add noise, we should give the l_2 sensitivity estimation of the approximate gradient $\boldsymbol{g}_i^s$ at first. According to [19], the sensitivity of $\boldsymbol{g}_i^s$ is $\Delta_2 \leq 3G$, where G is the lipschitz constant of loss function $l_{im}(\cdot)$. The following theorem shows that our algorithm provides (ϵ, δ)-differential privacy[2].

Theorem 2. *For $\epsilon \leq c_1 \frac{KS}{M^2}$ and $\delta \in (0,1)$, and the noise θ_k^s is sampled from zero mean Gaussian distribution with variance*

$$(\sigma^2)_k^s = c\frac{G^2 KS \ln(1/\delta)}{M^2\epsilon^2},$$

then, PS-ADMM algorithm satisfies (ϵ, δ)-differential privacy, where c_1 and c are some constants.

In a distributed and iterative algorithm, the output of the algorithm includes all of exchanged intermediate results and the end result. Since the adversary may perform inference by using all intermediate results, the privacy leakage accumulates over time through the iterative process. Different from the prior study in [22], where the privacy leakage is only bounded at a single iteration, our proposed differentially private algorithm PS-ADMM provides (ϵ, δ)-differential privacy guarantee for all of the intermediate results exchanged during the iterative procedure and the end result.

4 Convergence Analysis

In this section, we discuss the convergence and the utility bound of the proposed PS-ADMM algorithm. To define the convergence and utility bound, we will use the following criterion

$$\mathbb{E}[P(\boldsymbol{u}) - P(\boldsymbol{u}^*) + \sum_{i=1}^{N} \tau_i \|\boldsymbol{x}_i - \boldsymbol{z}\|] \quad \forall \tau_i > 0, \tag{9}$$

which is the same as the variational inequality used in [16] and [12]. In criterion (9), $\boldsymbol{u} = \{\boldsymbol{x}_1; \boldsymbol{x}_2; ...; \boldsymbol{x}_N; \boldsymbol{z}\}$ and $P(\boldsymbol{u}) = \sum_{i=1}^N f_i(\boldsymbol{x}_i) + g(\boldsymbol{z})$, and $\boldsymbol{u}^*$ is the optimal solution of problem 2.

Similar to most iterative distributed optimization algorithms [11], distributed stochastic ADMM only converges in a probabilistic sense when the number of iterations $K \to \infty$. Therefore, we can now prove the following expected suboptimality of the proposed algorithm according to the criterion (9).

[2] The proof of Theorem 2 is very similar to the B.2 in [19]. Due to the space limitation, we omit the detail of it.

Theorem 3 (Convergence). *If $f_i(\boldsymbol{x})$ is μ_f-strongly convex, $\tau_i > 0$ and η satisfies $0 < \eta \leq \frac{1}{2v_L}$, $0 < \eta \leq \frac{4\mu_L - 4\rho - 3\mu_f}{8v_L^2 + 2\mu_f v_L}$, $1 - \frac{\rho\xi}{2} - \frac{\mu_f \xi}{4} + \frac{4\eta^2 v_L^2 S}{1-2\eta v_L} \leq \frac{S\eta\mu_f}{2}$, the expected suboptimality of PS-ADMM is bounded after K iterations*

$$\begin{aligned}
&\mathbb{E}\left\{ P(\boldsymbol{u}^{k+1}) - P(\boldsymbol{u}^*) + \sum_{i=1}^{N} \tau_i \|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{z}}\| \right\} \\
&\leq \sum_{i=1}^{N} \left\{ \frac{\mu_f}{4K} \|\boldsymbol{x}_i^0 - \boldsymbol{x}_i^*\|^2 + \frac{\rho}{2K} \|\boldsymbol{z}^0 - \boldsymbol{z}^*\|^2 + \frac{1}{2\rho K} (\|\boldsymbol{\lambda}_i^0\|^2 + \tau_i^2) \right\} \\
&\quad + \frac{2\eta}{1 - 2\eta v_L} \frac{pKS \ln(1/\delta)}{M^2 \epsilon^2},
\end{aligned} \tag{10}$$

where $P(\boldsymbol{u}) = \sum_{i=1}^{N} f_i(\hat{\boldsymbol{x}}_i) + g(\hat{\boldsymbol{z}})$, $\hat{\boldsymbol{x}}_i = \frac{1}{K} \sum_{k=0}^{K-1} \boldsymbol{x}_i^{k+1}$ and $\hat{\boldsymbol{z}} = \frac{1}{K} \sum_{k=0}^{K-1} \boldsymbol{z}^{k+1}$ and $(\boldsymbol{x}_i^*, \boldsymbol{z}^*)$ is the optimal solution.

Proof. See Appendix A.3.

As K increases, the first term in (10) decreases, though the second term in (10) increases. Then, the minimized suboptimality of the proposed algorithm exists as we choose an optimal K. Hence, the following theorem gives the utility bound when choosing an optimal K.

Theorem 4 (Utility Bound). *If $f_i(\boldsymbol{x})$ is μ_f-strongly convex, $\tau_i > 0$ and $S = O(\frac{v_f}{\mu_f})$ is sufficiently large, and η satisfies condition in Theorem 3, then the utility bound of PS-ADMM is bounded if we choose $K = O\left(\frac{M\epsilon}{G} \sqrt{\frac{\mu_f}{v_f p \ln(1/\delta)}}\right)$,*

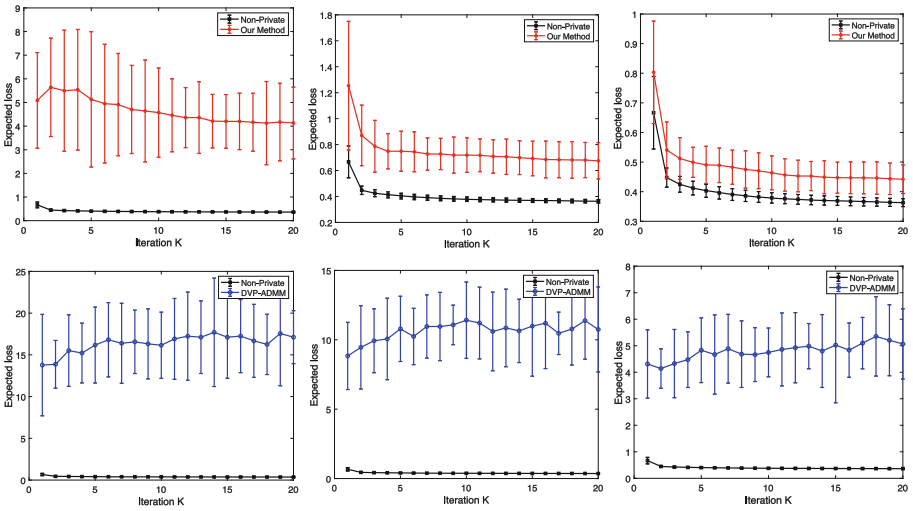
$$\mathbb{E}\left\{ P(\hat{\boldsymbol{u}}) - P(\boldsymbol{u}^*) + \sum_{i=1}^{N} \tau_i \|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{z}}\| \right\} \leq O\left(\frac{NG}{M\epsilon} \sqrt{\frac{p \ln(1/\delta) v_f}{\mu_f}} \right),$$

where $P(\boldsymbol{u}) = \sum_{i=1}^{N} f_i(\hat{\boldsymbol{x}}_i) + g(\hat{\boldsymbol{z}})$, $\hat{\boldsymbol{x}}_i = \frac{1}{K} \sum_{k=0}^{K-1} \boldsymbol{x}_i^{k+1}$ and $\hat{\boldsymbol{z}} = \frac{1}{K} \sum_{k=0}^{K-1} \boldsymbol{z}^{k+1}$ and $(\boldsymbol{x}_i^*, \boldsymbol{z}^*)$ is the optimal solution.

Proof. See Appendix A.4.

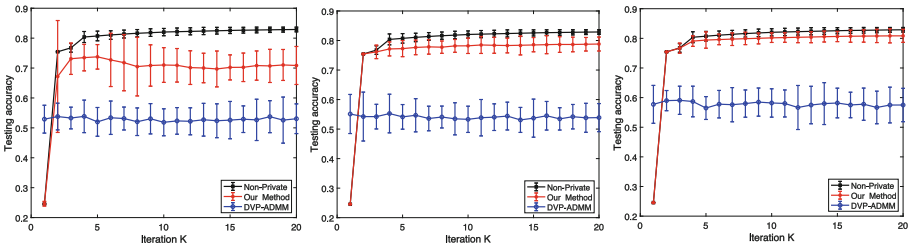
5 Performance Analysis

We conduct simulations on the same dataset as [22], i.e, the Adult dataset from UCI Machine Learning Repository [5], which contains 48,842 samples with 14 features like age, sex, education, etc. The goal is to predict whether the annual income is more than 50k or not. Before the simulation, we preprocess the data by normalizing all numerical attributes such that l_2-norm is at most 1 and transform the label $\{>50\text{k}, \leq 50\text{k}\}$ to $\{+1, -1\}$. We separate the whole dataset for training and testing (the ratio is around 70%:30%). And for training samples, we separate them into five parts representing five data providers ($N = 5$). Consistent with



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.5$ (c) $\epsilon = 1$

Fig. 2. The convergence comparison for different privacy budgets ϵ. (Color figure online)



(a) $\epsilon = 0.1$ (b) $\epsilon = 0.5$ (c) $\epsilon = 1$

Fig. 3. The accuracy comparison for different privacy budgets ϵ. (Color figure online)

[22], we use the logistic loss $l(z) = log(1 + exp(z))$. And N data providers collaboratively solve the following regularized logistic regression

$$\min_{\boldsymbol{x}} \sum_{i=1}^{N} \frac{1}{M} \sum_{m=1}^{M} log(1 + exp(-y_{im}\boldsymbol{x}^T \boldsymbol{a}_{im})) + R\|\boldsymbol{x}\|^2.$$

We inspect the convergence and accuracy of our approach by comparing with the dual variable perturbation (DVP-ADMM) method adopted in [22]. The convergence is measured by expected loss defined by $\frac{1}{M}\sum_{i=1}^{N}\sum_{m=1}^{M} log(1+exp(-y_{im}\boldsymbol{x}_{i,k}^T \boldsymbol{a}_{im}))$. The accuracy is defined by classification error rate over testing dataset. For the DVP-ADMM algorithm, the parameters are the same as in settings of [22].

For each parameter setting, we conduct 20 independent runs of the algorithm. For each time, both the mean and standard deviation of the expected loss and the accuracy are recorded. The smaller the standard deviation is, the greater is the stability of the algorithm. In all experiments, we set the regularization coefficient $R = 0.0001$, and $\delta = 0.001$.

Figures 2 and 3 compare our approach with DVP-ADMM method and the non-private algorithm for expected loss and testing accuracy under different privacy budgets. The non-private algorithm here is a stochastic ADMM without adding noise. As the number of iterations increases, we see that our approach (red) has achieved much less expected loss and higher testing accuracy than DVP-ADMM (blue) for all three cases of privacy budget ϵ. Hence, our method can outperform DVP-ADMM (blue) significantly. However, the excepted loss does not always monotonically decrease as too much noise introduced in PS-ADMM affects the convergence, especially when ϵ is small. While privacy budget ϵ is large enough (e.g., $\epsilon = 0.5$), it follows the same trend as non-private ADMM and still outperforms DVP-ADMM.

6 Conclusions

In this paper, we proposed a novel algorithm called PS-ADMM by extending SCAS-ADMM into a distributed setting and adding differentially private Gaussian noise to the gradient updates. Thus, the sensitive information stored in the training dataset at each data provider can be protected against an adversary who can eavesdrop the communications between the data provider and the server. The convergence and utility bound of PS-ADMM have been analyzed theoretically. We empirically demonstrate that PS-ADMM outperforms other differentially private ADMM algorithms under the same privacy guarantee.

Acknowledgement. This work of J. Ding, and M. Pan was supported in part by the U.S. Natural Science Foundation under grants US CNS-1613661, CNS-1646607, CNS-1702850, and CNS-1801925. This work of Y. Gong was partly supported by the US National Science Foundation under grant CNS-1850523. This work of H. Zhang was partly supported by the National Natural Science Foundation of China (Grant No. 61822104, 61771044), Beijing Natural Science Foundation (No. L172025, L172049), and 111 Project (No. B170003).

A Appendix

The approximate gradient $\boldsymbol{g}_i^s$ can be written as $\boldsymbol{g}_i^s = \boldsymbol{b}_i^s + \boldsymbol{q}_i^s$, where

$$\boldsymbol{b}_i^s = \nabla l_{im^s}(\boldsymbol{v}_i^s) - \nabla l_{im^s}(\tilde{\boldsymbol{v}}_i) + \hat{\boldsymbol{u}}_i + \theta_i^s,$$
$$\boldsymbol{q}_i^s = \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{v}_i^s - \boldsymbol{z}^k).$$

A.1 Proof of Lemma 1

Proof. Since each $l_{im}(\boldsymbol{x})$ is convex, G-Lipschitz and has L_m-Lipschitz continuous gradient, for any $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, there exists $L_m > 0$ such that

$$l_{im}(\boldsymbol{x}_1) \leq l_{im}(\boldsymbol{x}_2) + (\boldsymbol{x}_2 - \boldsymbol{x}_1)^T \nabla l_{im}(\boldsymbol{x}_2) + \frac{L_m}{2}\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|^2.$$

We can see that $f_i(\boldsymbol{x})$ is v_f-smooth, with $f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2) + (\boldsymbol{x}_2 - \boldsymbol{x}_1)^T \nabla f_i(\boldsymbol{x}_1) + \frac{v_f}{2}\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|^2$, where $v_f = \max_m L_m$. Then, we can have

$$\begin{aligned}
&\|\nabla L_i(\boldsymbol{x}_1) - \nabla L_i(\boldsymbol{x}_2)\| = \|\nabla f_i(\boldsymbol{x}_1) - \nabla f_i(\boldsymbol{x}_2) + \rho(\boldsymbol{x}_1 - \boldsymbol{x}_2)\| \\
&\leq \|\nabla f_i(\boldsymbol{x}_1) - \nabla f_i(\boldsymbol{x}_2)\| + \|\rho(\boldsymbol{x}_1 - \boldsymbol{x}_2)\| \leq (v_f + \rho)\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \leq v_L\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|,
\end{aligned}$$

where we let $v_L \geq v_f + \rho$. Thus, $L_i(\boldsymbol{x})$ and $\hat{L}_m(\boldsymbol{x})$ are v_L-smooth. Moreover, it is obvious to see that $L_i(\boldsymbol{x})$ is μ_L-strongly convex with $\mu_L \leq \mu_f + \rho$.

A.2 Basic Lemmas

Lemma 2. *The variance of $\boldsymbol{g}_i^s$ satisfies*

$$\begin{aligned}
\mathbb{E}(\|\boldsymbol{g}_i^s\|^2) &\leq 2\mathbb{E}(\|\nabla \hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla \hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2) + 4\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 + 4(\sigma^2)_k^s p \\
&\leq 4v_L^2(\|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 + \|\tilde{\boldsymbol{v}}_i - \boldsymbol{x}_i\|^2) + 4\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 + 4(\sigma^2)_k^s p.
\end{aligned}$$

Proof. Notice that

$$\begin{aligned}
\boldsymbol{g}_i^s &= \boldsymbol{b}_i^s + \boldsymbol{q}_i^s \\
&= \nabla l_{im^s}(\boldsymbol{v}_i^s) - \nabla l_{im^s}(\tilde{\boldsymbol{v}}_i) + \hat{\boldsymbol{u}}_i + \boldsymbol{q}_i^s + \theta_i^s \\
&= \nabla l_{im^s}(\boldsymbol{v}_i^s) + \boldsymbol{q}_i^s - \nabla l_{im^s}(\tilde{\boldsymbol{v}}_i) - \hat{\boldsymbol{q}}_i + \hat{\boldsymbol{u}}_i + \hat{\boldsymbol{q}}_i + \theta_i^s \\
&= \nabla \hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla \hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i) + \nabla L_i(\tilde{\boldsymbol{v}}_i) + \theta_i^s.
\end{aligned}$$

Hence, the variance of $\boldsymbol{g}_i^s$ can be bounded as

$$\begin{aligned}
&\mathbb{E}(\|\boldsymbol{g}_i^s\|^2) = \mathbb{E}\|\nabla \hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla \hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i) + \nabla L_i(\tilde{\boldsymbol{v}}_i) + \theta_i^s\|^2 \\
&\leq 2\mathbb{E}(\|\nabla \hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla \hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i) - (\nabla L_i(\boldsymbol{v}_i^s) - \nabla L_i(\tilde{\boldsymbol{v}}_i)\|^2)) + 2\mathbb{E}\|\nabla L_i(\boldsymbol{v}_i^s) + \theta_i^s\|^2 \\
&\leq 2\mathbb{E}(\|\nabla \hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla \hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2) + 2\mathbb{E}\|\nabla L_i(\boldsymbol{v}_i^s) + \theta_i^s\|^2 \\
&\leq 2\mathbb{E}(\|\nabla \hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla \hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2) + 4\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 + 4\mathbb{E}\|\theta_i^s\|^2 \\
&\leq 2v_L^2\|\boldsymbol{v}_i^s - \tilde{\boldsymbol{v}}_i\|^2 + 4\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 + 4(\sigma^2)_k^s p \\
&\leq 4v_L^2(\|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 + \|\tilde{\boldsymbol{v}}_i - \boldsymbol{x}_i\|^2) + 4\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 + 4(\sigma^2)_k^s p,
\end{aligned}$$

where the first inequality uses $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ and the second inequality uses $\mathbb{E}\|\boldsymbol{x}_i - \mathbb{E}\boldsymbol{x}_i\|^2 = \mathbb{E}\|\boldsymbol{x}_i\|^2 - \|\mathbb{E}\boldsymbol{x}_i\|^2 \leq \mathbb{E}\|\boldsymbol{x}_i\|^2$.

Lemma 3. *For $0 < \eta < \frac{1}{2v_L}$, we have*

$$\begin{aligned}
\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 \leq &\frac{1}{\eta - 2\eta^2 v_L}\{L_i(\boldsymbol{v}_i^s) - \mathbb{E}[L_i(\boldsymbol{v}_i^{s+1})]\} \\
&+ \frac{\eta v_L}{1 - 2\eta v_L}\mathbb{E}(\|\nabla \hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla \hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2 + 2(\sigma^2)_k^s p).
\end{aligned}$$

Proof.

$$L_i(\boldsymbol{v}_i^{s+1}) \leq L_i(\boldsymbol{v}_i^s) + (\boldsymbol{v}_i^{s+1} - \boldsymbol{v}_i^s)^T \nabla L_i(\boldsymbol{v}_i^s) + \frac{\upsilon_L}{2}\|\boldsymbol{v}_i^{s+1} - \boldsymbol{v}_i^s\|^2.$$

Taking expectation on both sides, we obtain

$$\begin{aligned}\mathbb{E}[L_i(\boldsymbol{v}_i^{s+1})] &\leq L_i(\boldsymbol{v}_i^s) - \eta\nabla\|L_i(\boldsymbol{v}_i^s)\|^2 + \frac{\eta^2\upsilon_L}{2}\mathbb{E}(\|\boldsymbol{g}_i^s\|^2) \\ &\leq L_i(\boldsymbol{v}_i^s) - \eta\nabla\|L_i(\boldsymbol{v}_i^s)\|^2 + \eta^2\upsilon_L[\mathbb{E}(\|\nabla\hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla\hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2) \\ &\quad + 2\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 + 2(\sigma^2)_k^s p].\end{aligned}$$

Then, we have

$$\begin{aligned}(\eta - 2\eta^2\upsilon_L)\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 \leq& L_i(\boldsymbol{v}_i^s) - \mathbb{E}[L_i(\boldsymbol{v}_i^{s+1})] \\ &+ \eta^2\upsilon_L\mathbb{E}(\|\nabla\hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla\hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2 + 2(\sigma^2)_k^s p).\end{aligned}$$

By choosing $\eta < 1/(2\upsilon_L)$, we get

$$\begin{aligned}\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 \leq& \frac{1}{\eta - 2\eta^2\upsilon_L}\{L_i(\boldsymbol{v}_i^s) - \mathbb{E}[L_i(\boldsymbol{v}_i^{s+1})]\} \\ &+ \frac{\eta\upsilon_L}{1 - 2\eta\upsilon_L}\mathbb{E}(\|\nabla\hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla\hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2 + 2(\sigma^2)_k^s p).\end{aligned}$$

Lemma 4.

$$\begin{aligned}&\mathbb{E}\|\boldsymbol{v}_i^{s+1} - \boldsymbol{x}_i\|^2 + 2\eta(\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T\nabla L_i(\boldsymbol{v}_i^s) + \frac{4\eta}{1 - 2\eta\upsilon_L}(\mathbb{E}[L_i(\boldsymbol{v}_i^{s+1})] - L_i(\boldsymbol{x}_i)) \\ &\leq \|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 + \frac{2\eta^2}{1 - 2\eta\upsilon_L}\mathbb{E}\|\nabla\hat{L}_{im^s}(\boldsymbol{v}_i^s) + \nabla\hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2 \\ &\quad + \frac{4\eta}{1 - 2\eta\upsilon_L}[L_i(\boldsymbol{v}_i^s) - L_i(\boldsymbol{x}_i)] + \frac{2\eta^2}{1 - 2\eta\upsilon_L}(\sigma^2)_k^s p.\end{aligned}$$

Proof. We have $\mathbb{E}(\boldsymbol{b}_i^s) = \nabla f_i(\boldsymbol{v}_i^s)$ and this leads to

$$\begin{aligned}\mathbb{E}\|\boldsymbol{v}_i^{s+1} - \boldsymbol{x}_i\|^2 &\leq \|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 - 2\eta(\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T\mathbb{E}(\boldsymbol{g}_i^s) + \eta^2\mathbb{E}(\|\boldsymbol{g}_i^s\|^2) \\ &\leq \|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 - 2\eta(\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T\mathbb{E}(\nabla f_i(\boldsymbol{v}_i^s) + \boldsymbol{q}_i^s) + \eta^2\mathbb{E}(\|\boldsymbol{g}_i^s\|^2) \\ &\leq \|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 - 2\eta(\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T\nabla L_i(\boldsymbol{v}_i^s) + \eta^2\mathbb{E}(\|\boldsymbol{g}_i^s\|^2).\end{aligned}$$

Then, we have

$$\begin{aligned}&\mathbb{E}\|\boldsymbol{v}_i^{s+1} - \boldsymbol{x}_i\|^2 + 2\eta(\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T\nabla L_i(\boldsymbol{v}_i^s) \\ &\leq \|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 + 2\eta^2\mathbb{E}(\|\nabla\hat{L}_{im^s}(\boldsymbol{v}_i^s) - \nabla\hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2) + 4\eta^2\|\nabla L_i(\boldsymbol{v}_i^s)\|^2 \\ &\quad + 4\eta^2(\sigma^2)_k^s p.\end{aligned}$$

According to Lemma 3, we obtain

$$\mathbb{E}\|\boldsymbol{v}_i^{s+1}-\boldsymbol{x}_i\|^2+2\eta(\boldsymbol{v}_i^s-\boldsymbol{x}_i)^T\nabla L_i(\boldsymbol{v}_i^s)+\frac{4\eta}{1-2\eta v_L}(\mathbb{E}[L_i(\boldsymbol{v}_i^{s+1})]-L_i(\boldsymbol{x}_i))$$
$$\leq\|\boldsymbol{v}_i^s-\boldsymbol{x}_i\|^2+\frac{2\eta^2}{1-2\eta v_L}\mathbb{E}\|\nabla\hat{L}_{im^s}(\boldsymbol{v}_i^s)+\nabla\hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2$$
$$+\frac{4\eta}{1-2\eta v_L}[L_i(\boldsymbol{v}_i^s)-L_i(\boldsymbol{x}_i)]+\frac{2\eta^2}{1-2\eta v_L}(\sigma^2)_k^s p.$$

Lemma 5.

$$g(\boldsymbol{z}^{k+1})-g(\boldsymbol{z})-\sum_{i=1}^N(\boldsymbol{z}^{k+1}-\boldsymbol{z})^T\boldsymbol{\alpha}_i^{k+1}$$
$$\leq\frac{\rho}{2}(\|\boldsymbol{z}^k-\boldsymbol{z}\|^2-\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|^2-\|\boldsymbol{z}^{k+1}-\boldsymbol{z}\|^2)$$

where $\boldsymbol{\alpha}_i^{k+1}=\boldsymbol{\lambda}_i^k+\rho(\boldsymbol{x}_i^{k+1}-\boldsymbol{z}^k)$.

Proof. By deriving the optimal conditions of the minimization problem in (5), we have

$$g(\boldsymbol{z}^{k+1})-g(\boldsymbol{z})\leq-(\boldsymbol{z}^{k+1}-\boldsymbol{z})^T\sum_{i=1}^N[-\boldsymbol{\lambda}_i^k+\rho(\boldsymbol{x}_i^{k+1}-\boldsymbol{z}^{k+1})].$$

Then, by using the notation $\boldsymbol{\alpha}_i^{k+1}=\boldsymbol{\lambda}_i^k+\rho(\boldsymbol{x}_i^{k+1}-\boldsymbol{z}^k)$, we obtain

$$g(\boldsymbol{z}^{k+1})-g(\boldsymbol{z})-\sum_{i=1}^N(\boldsymbol{z}^{k+1}-\boldsymbol{z})^T\boldsymbol{\alpha}_i^{k+1}\leq\rho(\boldsymbol{z}^k-\boldsymbol{z}^{k+1})^T(\boldsymbol{z}^{k+1}-\boldsymbol{z}))$$
$$\leq\frac{\rho}{2}(\|\boldsymbol{z}^k-\boldsymbol{z}\|^2-\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^k\|^2-\|\boldsymbol{z}^{k+1}-\boldsymbol{z}\|^2).$$

Lemma 6.

$$(\boldsymbol{\alpha}_i^{k+1}-\boldsymbol{\alpha}_i)^T[-(\boldsymbol{x}_i^{k+1}-\boldsymbol{z}^{k+1})]$$
$$\leq\frac{1}{2\rho}(\|\boldsymbol{\lambda}_i^k-\boldsymbol{\alpha}_i\|^2-\|\boldsymbol{\lambda}_i^{k+1}-\boldsymbol{\alpha}_i\|^2)+\frac{\rho}{2}\|\boldsymbol{z}^k-\boldsymbol{z}^{k+1}\|^2$$

where $\boldsymbol{\alpha}_i^{k+1}=\boldsymbol{\lambda}_i^k+\rho(\boldsymbol{x}_i^{k+1}-\boldsymbol{z}^k)$.

Proof.

$$(\boldsymbol{\alpha}_i^{k+1}-\boldsymbol{\alpha}_i)^T[-(\boldsymbol{x}_i^{k+1}-\boldsymbol{z}^{k+1})]=\frac{1}{\rho}(\boldsymbol{\alpha}_i^{k+1}-\boldsymbol{\alpha}_i)^T(\boldsymbol{\lambda}_i^k-\boldsymbol{\lambda}_i^{k+1})$$
$$=\frac{1}{2\rho}(\|\boldsymbol{\alpha}_i^{k+1}-\boldsymbol{\lambda}_i^{k+1}\|^2-\|\boldsymbol{\alpha}_i^{k+1}-\boldsymbol{\lambda}_i^k\|^2+\|\boldsymbol{\lambda}_i^k-\boldsymbol{\alpha}_i\|^2-\|\boldsymbol{\lambda}_i^{k+1}-\boldsymbol{\alpha}_i\|^2)$$
$$\leq\frac{1}{2\rho}(\|\boldsymbol{\lambda}_i^k-\boldsymbol{\alpha}_i\|^2-\|\boldsymbol{\lambda}_i^{k+1}-\boldsymbol{\alpha}_i\|^2)+\frac{\rho}{2}\|\boldsymbol{z}^k-\boldsymbol{z}^{k+1}\|^2.$$

Lemma 7. *Assume $f_i(\cdot)$ be μ_f-strongly convex, and let $\boldsymbol{x}_i^{k+1}$, $\boldsymbol{z}^k$ and $\boldsymbol{\lambda}_i^k$ be generated by the proposed algorithm. For η satisfies $0 < \eta \leq \frac{1}{2v_L}$, $0 < \eta \leq \frac{4\mu_L - 4\rho - 3\mu_f}{8v_L^2 + 2\mu_f v_L}$, $1 - \frac{\rho\xi}{2} - \frac{\mu_f \xi}{4} + \frac{4\eta^2 v_L^2 S}{1-2\eta v_L} \leq \frac{S\eta\mu_f}{2}$, the following holds if*

$$\mathbb{E}[f_i(\boldsymbol{x}_i^{k+1}) - f_i(\boldsymbol{x}_i) + (\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i)^T \boldsymbol{\alpha}_i^{k+1}] \leq \frac{\mu_f}{4}[\|\boldsymbol{x}_i^k - \boldsymbol{x}_i\|^2 - \|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i\|^2] + \frac{2\eta}{1 - 2\eta v_L}(\sigma^2)_k^s p$$

where $\boldsymbol{\alpha}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{x}_i^{k+1} - \boldsymbol{z}^k)$.

Proof. Using Lemma 4 and the strong convexity of $L_i(\boldsymbol{v}_i)$, we have

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{v}_i^{s+1} - \boldsymbol{x}_i\|^2 + \zeta(\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T \nabla L_i(\boldsymbol{v}_i^s) + \xi(\mathbb{E}[L_i(\boldsymbol{v}_i^{s+1})] - L_i(\boldsymbol{x}_i)) \\
&+ \frac{\mu_L \xi}{2}\|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 \\
&\leq \frac{2\eta^2}{1 - 2\eta v_L}[\mathbb{E}\|\nabla \hat{L}_{im^s}(\boldsymbol{v}_i^s) + \nabla \hat{L}_{im^s}(\tilde{\boldsymbol{v}}_i)\|^2 + (\sigma^2)_k^s p] + \|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2,
\end{aligned}$$

where $\zeta = 2\eta - \frac{4\eta}{1-2\eta v_L}$, $\xi = \frac{4\eta}{1-2\eta v_L}$.
Then, we obtain

$$\begin{aligned}
&(1 - \frac{\rho\xi}{2} - \frac{\mu_f \xi}{4})\mathbb{E}\|\boldsymbol{v}_i^{s+1} - \boldsymbol{x}_i\|^2 + \zeta[f_i(\boldsymbol{v}_i^s) - f_i(\boldsymbol{x}_i) + (\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T \boldsymbol{q}_i^s \\
&+ \frac{\mu_f}{4}\|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2] + \xi\mathbb{E}[f_i(\boldsymbol{v}_i^{s+1}) - f_i(\boldsymbol{x}_i) + (\boldsymbol{v}_i^{s+1} - \boldsymbol{x}_i)^T \boldsymbol{q}_i^{s+1} \\
&+ \frac{\mu_f}{4}\|\boldsymbol{v}_i^{s+1} - \boldsymbol{x}_i\|^2] \\
&\leq (1 + \frac{4\eta^2 v_L^2}{1 - 2\eta v_L} - \frac{\mu_L \xi}{2} - \frac{\mu_f \zeta}{4})\|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 + \frac{4\eta^2 v_L^2}{1 - 2\eta v_L}\|\tilde{\boldsymbol{v}}_i - \boldsymbol{x}_i\|^2 \\
&\quad + \frac{2\eta^2}{1 - 2\eta v_L}(\sigma^2)_k^s p,
\end{aligned}$$

where we apply Lemma 2 and $L_i(\boldsymbol{v}_i^s) - L_i(\boldsymbol{x}_i) = f_i(\boldsymbol{v}_i^s) - f_i(\boldsymbol{x}_i) + (\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T \boldsymbol{q}_i^s - \frac{\rho}{2}\|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2$ to obtain the inequality. Hence, we choose $\eta \leq \frac{4\mu_L - 4\rho - 3\mu_f}{8v_L^2 + 2\mu_f v_L}$ so that $1 - \frac{\rho\xi}{2} - \frac{\mu_f \xi}{4} \geq 1 + \frac{4\eta^2 v_L^2}{1-2\eta v_L} - \frac{\mu_L \xi}{2} - \frac{\mu_f \zeta}{4}$. We take $\hat{\boldsymbol{v}}_i^{s+1} = \frac{\zeta \boldsymbol{v}_i^s + \xi \boldsymbol{v}_i^{s+1}}{2\eta}$ and we know that $f_i(\boldsymbol{v}_i^s) - f_i(\boldsymbol{x}_i) + (\boldsymbol{v}_i^s - \boldsymbol{x}_i)^T \boldsymbol{q}_i^s$ is convex in $\boldsymbol{v}_i^s$. By using the Jensen's inequality, we have

$$\begin{aligned}
&(1 - \frac{\rho\xi}{2} - \frac{\mu_f \xi}{4})\mathbb{E}\|\boldsymbol{v}_i^{s+1} - \boldsymbol{x}_i\|^2 \\
&+ 2\eta\mathbb{E}[f_i(\hat{\boldsymbol{v}}_i^{s+1}) - f_i(\boldsymbol{x}_i) + (\hat{\boldsymbol{v}}_i^{s+1} - \boldsymbol{x}_i)^T \hat{\boldsymbol{q}}_i^{s+1} + \frac{\mu_f}{4}\|\hat{\boldsymbol{v}}_i^{s+1} - \boldsymbol{x}_i\|^2] \\
&\leq (1 - \frac{\rho\xi}{2} - \frac{\mu_f \xi}{4})\|\boldsymbol{v}_i^s - \boldsymbol{x}_i\|^2 + \frac{4\eta^2 v_L^2}{1 - 2\eta v_L}\|\tilde{\boldsymbol{v}}_i - \boldsymbol{x}_i\|^2 + \frac{2\eta^2}{1 - 2\eta v_L}(\sigma^2)_k^s p,
\end{aligned}$$

where $\hat{\boldsymbol{q}}_i^{s+1} = \boldsymbol{\lambda}_i^k + \rho(\hat{\boldsymbol{v}}_i^{s+1} - \boldsymbol{z}^k)$. Summing from $s = 0, 1, 2, ..., S-1$ and using $\boldsymbol{x}_i^{k+1} = \frac{1}{s}\sum_{s=0}^{S-1} \hat{\boldsymbol{v}}_i^{s+1}$, we obtain

$$
\begin{aligned}
&2S\eta\mathbb{E}[f_i(\boldsymbol{x}_i^{k+1}) - f_i(\boldsymbol{x}_i) + (\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i)^T\boldsymbol{\alpha}_i^{k+1} + \frac{\mu_f}{4}\|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i\|^2] \\
&\leq \frac{2\eta^2 S}{1-2\eta v_L}(\sigma^2)_k^s p + (1 - \frac{\rho\xi}{2} - \frac{\mu_f\xi}{4} + \frac{4\eta^2 v_L^2 S}{1-2\eta v_L})\|\boldsymbol{x}_i^k - \boldsymbol{x}_i\|^2,
\end{aligned}
$$

where $\boldsymbol{\alpha}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho(\boldsymbol{x}_i^{k+1} - \boldsymbol{z}^k)$.
Thus, we have

$$
\begin{aligned}
&\mathbb{E}[f_i(\boldsymbol{x}_i^{k+1}) - f_i(\boldsymbol{x}_i) + (\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i)^T\boldsymbol{\alpha}_i^{k+1}] \\
&\leq \frac{1}{2S\eta}(1 - \frac{\rho\xi}{2} - \frac{\mu_f\xi}{4} + \frac{4\eta^2 v_L^2 S}{1-2\eta v_L})\|\boldsymbol{x}_i^k - \boldsymbol{x}_i\|^2 - \frac{\mu_f}{4}\mathbb{E}\|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i\|^2 \\
&\quad + \frac{2\eta}{1-2\eta v_L}(\sigma^2)_k^s p \\
&\leq \frac{\mu_f}{4}[\|\boldsymbol{x}_i^k - \boldsymbol{x}_i\|^2 - \|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i\|^2] + \frac{2\eta}{1-2\eta v_L}(\sigma^2)_k^s p,
\end{aligned}
$$

where we assume $1 - \frac{\rho\xi}{2} - \frac{\mu_f\xi}{4} + \frac{4\eta^2 v_L^2 S}{1-2\eta v_L} \leq \frac{S\eta\mu_f}{2}$.

A.3 Proof of Theorem 3

Proof. Combining Lemmas 7, 5 and 6 together and using the convergence criterion (9), we let $\boldsymbol{w}_i^{k+1} = (\boldsymbol{x}_i^{k+1}; \boldsymbol{z}^{k+1}; \boldsymbol{\alpha}_i^{k+1})$, and $\hat{\boldsymbol{w}}_i = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{w}_i^{k+1}$. For any $\boldsymbol{w} = (\boldsymbol{x}_i; \boldsymbol{z}; \boldsymbol{\lambda}_i)$, we have

$$
\begin{aligned}
&\mathbb{E}[P(\boldsymbol{u}^{k+1}) - P(\boldsymbol{u}) + \sum_{i=1}^N (\boldsymbol{w}^{k+1} - \boldsymbol{w})^T F(\boldsymbol{w}^{k+1})] \\
&\leq \mathbb{E}\Bigg\{\sum_{i=1}^N f_i(\boldsymbol{x}_i^{k+1}) + g(\boldsymbol{z}^{k+1}) - \sum_{i=1}^N f_i(\boldsymbol{x}_i) - g(\boldsymbol{z}) \\
&\quad + \sum_{i=1}^N \begin{pmatrix} \boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i \\ \boldsymbol{z}^{k+1} - \boldsymbol{z} \\ \boldsymbol{\alpha}_i^{k+1} - \boldsymbol{\alpha}_i \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\alpha}_i^{k+1} \\ -\boldsymbol{\alpha}_i^{k+1} \\ -(\boldsymbol{x}_i^{k+1} - \boldsymbol{z}^{k+1}) \end{pmatrix} \Bigg\} \\
&\leq \sum_{i=1}^N \Bigg\{ \frac{\mu_f}{4}[\|\boldsymbol{x}_i^k - \boldsymbol{x}_i\|^2 - \|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i\|^2] + \frac{\rho}{2}(\|\boldsymbol{z}^k - \boldsymbol{z}\|^2 - \|\boldsymbol{z}^{k+1} - \boldsymbol{z}\|^2) \\
&\quad + \frac{1}{2\rho}(\|\boldsymbol{\lambda}_i^k - \boldsymbol{\alpha}_i\|^2 - \|\boldsymbol{\lambda}_i^{k+1} - \boldsymbol{\alpha}_i\|^2) + \frac{2\eta}{1-2\eta v_L}(\sigma^2)_k^s p \Bigg\},
\end{aligned}
$$

where $F(\boldsymbol{w}) = \begin{pmatrix} \boldsymbol{\alpha}_i \\ -\boldsymbol{\alpha}_i \\ -(\boldsymbol{x}_i - \boldsymbol{z}) \end{pmatrix}$.

Summing the inequality over $k = 0, 1, 2, ..., K-1$ and using the Jensen's inequality, we get

$$\begin{aligned}
&\mathbb{E}\left\{P(\hat{\boldsymbol{u}}^{k+1}) - P(\boldsymbol{u}) + \sum_{i=1}^{N}(\hat{\boldsymbol{w}}_i - \boldsymbol{w})^T F(\hat{\boldsymbol{w}}_i)\right\} \\
&\leq \frac{1}{K}\sum_{k=0}^{K-1}\left\{\mathbb{E}[P(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1}) - P(\boldsymbol{x}, \boldsymbol{z}) + \sum_{i=1}^{N}(\boldsymbol{w}_i^{k+1} - \boldsymbol{w})^T F(\boldsymbol{w}_i^{k+1})]\right\} \\
&\leq \sum_{i=1}^{N}\left\{\frac{\mu_f}{4K}\|\boldsymbol{x}_i^0 - \boldsymbol{x}_i\|^2 + \frac{\rho}{2K}\|\boldsymbol{z}^0 - \boldsymbol{z}\|^2 + \frac{1}{2\rho K}\|\boldsymbol{\lambda}_i^0 - \boldsymbol{\alpha}_i\|^2\right\} \\
&\quad + \frac{2\eta}{1 - 2\eta v_L}\frac{pG^2KS\ln(1/\delta)}{M^2\epsilon^2},
\end{aligned}$$

where $P(\hat{\boldsymbol{u}}) = \sum_{i=1}^{N} f_i(\hat{\boldsymbol{x}}_i) + g(\hat{\boldsymbol{z}})$, $\hat{\boldsymbol{x}}_i = \frac{1}{K}\sum_{k=0}^{K-1}\boldsymbol{x}_i^{k+1}$ and $\hat{\boldsymbol{z}} = \frac{1}{K}\sum_{k=0}^{K-1}\boldsymbol{z}^{k+1}$. If we take $\boldsymbol{x} = \boldsymbol{x}^*, \boldsymbol{z} = \boldsymbol{z}^*$, and $\boldsymbol{\alpha}_i = \tau_i\frac{\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{z}}}{\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{z}}\|}$, we have

$$\begin{aligned}
&\mathbb{E}\left\{P(\hat{\boldsymbol{u}}^{k+1}) - P(\boldsymbol{u}^*) + \sum_{i=1}^{N}\tau_i\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{z}}\|\right\} \\
&\leq \sum_{i=1}^{N}\left\{\frac{\mu_f}{4K}\|\boldsymbol{x}_i^0 - \boldsymbol{x}_i^*\|^2 + \frac{\rho}{2K}\|\boldsymbol{z}^0 - \boldsymbol{z}^*\|^2 + \frac{1}{2\rho K}(\|\boldsymbol{\lambda}_i^0\|^2 + \tau_i^2)\right\} \\
&\quad + \frac{2\eta}{1 - 2\eta v_L}\frac{pG^2KS\ln(1/\delta)}{M^2\epsilon^2}.
\end{aligned}$$

A.4 Proof of Theorem 4

By choosing η, which satisfies condition in Theorem 3, and $S = O(\frac{v_f}{\mu_f})$, we can make $A = \frac{\mu_f}{4}\|\boldsymbol{x}_i^0 - \boldsymbol{x}_i^*\|^2 + \frac{\rho}{2}\|\boldsymbol{z}^0 - \boldsymbol{z}^*\|^2 + \frac{1}{2\rho}(\|\boldsymbol{\lambda}_i^0\|^2 + \tau_i^2)$ a constant.

Then, we have

$$\mathbb{E}\left\{P(\hat{\boldsymbol{u}}^{k+1}) - P(\boldsymbol{u}^*) + \sum_{i=1}^{N}\tau_i\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{z}}\|\right\} \leq \frac{NA}{K} + O\left(\frac{NpG^2K\ln(1/\delta)v_f}{M^2\epsilon^2\mu_f}\right).$$

Thus, if we choose $K = O\left(\frac{M\epsilon}{G}\sqrt{\frac{\mu_f}{v_f p\ln(1/\delta)}}\right)$, we have

$$\mathbb{E}\left\{P(\hat{\boldsymbol{u}}^{k+1}) - P(\boldsymbol{u}^*) + \sum_{i=1}^{N}\tau_i\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{z}}\|\right\} \leq O\left(\frac{NG}{M\epsilon}\sqrt{\frac{p\ln(1/\delta)v_f}{\mu_f}}\right).$$

References

1. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Vienna, October 2016, pp. 308–318 (2016)

2. Bekkerman, R., Bilenko, M., Langford, J.: Scaling Up Machine Learning: Parallel and Distributed Approaches. Cambridge University Press, Cambridge (2011)
3. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Prentice Hall, Englewood Cliffs (1989)
4. 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. **3**(1), 1–122 (2011)
5. Dheeru, D., Taniskidou, E.K.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
6. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
7. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
8. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, October 2015, pp. 1322–1333 (2015)
9. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning. Springer Series in Statistics. Springer, New York (2001). https://doi.org/10.1007/978-0-387-21606-5
10. Guo, Y., Gong, Y.: Practical collaborative learning for crowdsensing in the internet of things with differential privacy. In: IEEE Conference on Communications and Network Security (CNS), Beijing, May 2018, pp. 1–9 (2018)
11. Han, S., Topcu, U., Pappas, G.J.: Differentially private distributed constrained optimization. IEEE Trans. Autom. Control **62**(1), 50–64 (2017)
12. He, B., Yuan, X.: On the O(1/n) convergence rate of the Douglas-Rachford alternating direction method. SIAM J. Numer. Anal. **50**(2), 700–709 (2012)
13. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in Neural Information Processing Systems, Lake Tahoe, December 2013, pp. 315–323 (2013)
14. Liu, L., Han, Z.: Multi-block ADMM for big data optimization in smart grid. In: International Conference on Computing, Networking and Communications (ICNC), Anaheim, February 2015, pp. 556–561 (2015)
15. Nguyen, H., Khodaei, A., Han, Z.: A big data scale algorithm for optimal scheduling of integrated microgrids. IEEE Trans. Smart Grid **9**(1), 274–282 (2016)
16. Ouyang, H., He, N., Tran, L., Gray, A.: Stochastic alternating direction method of multipliers. In: International Conference on Machine Learning, Atlanta, June 2013, pp. 80–88 (2013)
17. Qin, Z., Goldfarb, D.: Structured sparsity via alternating direction methods. J. Mach. Learn. Res. **13**(1), 1435–1468 (2012)
18. Schizas, I.D., Ribeiro, A., Giannakis, G.B.: Consensus in ad hoc wsns with noisy links - Part I: distributed estimation of deterministic signals. IEEE Trans. Sig. Process. **56**(1), 350–364 (2008)
19. Wang, D., Ye, M., Xu, J.: Differentially private empirical risk minimization revisited: faster and more general. In: Advances in Neural Information Processing Systems, Long Beach, December 2017, pp. 2722–2731 (2017)
20. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2016)
21. Wu, X., Zhu, X., Wu, G.Q., Ding, W.: Data mining with big data. IEEE Trans. Knowl. Data Eng. **26**(1), 97–107 (2014)

22. Zhang, T., Zhu, Q.: A dual perturbation approach for differential private ADMM-based distributed empirical risk minimization. In: Proceedings of the ACM Workshop on Artificial Intelligence and Security, Vienna, October 2016, pp. 129–137 (2016)
23. Zhao, S., Li, W., Zhou, Z.: Scalable stochastic alternating direction method of multipliers. arXiv preprint arXiv:1502.03529 (2015)