Differentially Private Robust ADMM for Distributed Machine Learning

Jiahao Ding*, Xinyue Zhang*, Mingsong Chen†, Kaiping Xue‡, Chi Zhang§, and Miao Pan*
*Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204
†Shanghai Key Lab of Trustworthy Computing, East China Normal University, Shanghai 200062, China
‡ Department of EEIS, University of Science and Technology of China, Hefei 230027, China
§School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China

***Abstract*—To embrace the era of big data, there has been growing interest in designing distributed machine learning to exploit the collective computing power of the local computing nodes. Alternating Direction Method of Multipliers (ADMM) is one of the most popular methods. This method applies iterative local computations over local datasets at each agent and computation results exchange between the neighbors. During this iterative process, data privacy leakage arises when performing local computation over sensitive data. Although many differentially private ADMM algorithms have been proposed to deal with such privacy leakage, they still have to face many challenging issues such as low model accuracy over strict privacy constraints and requiring strong assumptions of convexity of the objective function. To address those issues, in this paper, we propose a differentially private robust ADMM algorithm (PR-ADMM) with Gaussian mechanism. We employ two kinds of noise variance decay schemes to carefully adjust the noise addition in the iterative process and utilize a threshold to eliminate the too noisy results from neighbors. We also prove that PR-ADMM satisfies dynamic zero-concentrated differential privacy (dynamic zCDP) and a total privacy loss is given by (ϵ, δ)-differential privacy. From a theoretical point of view, we analyze the convergence rate of PR-ADMM for general convex objectives, which is $\mathcal{O}(1/K)$ with K being the number of iterations. The performance of the proposed algorithm is evaluated on real-world datasets. The experimental results show that the proposed algorithm outperforms other differentially private ADMM based algorithms under the same total privacy loss.**

***Index Terms*—differential privacy, distributed machine learning, ADMM, decentralized optimization**

I. Introduction

With the rapid development of sensing technologies, the past decade has witnessed an explosive growth in size of generated data. For instance, the Cisco Visual Networking Index predicts that the number of mobile devices will be 11.6 billion by the year 2020 and the data will be generated at each smart phone with an average size of 4.4 gigabytes per month [1]. Because of the ability to exploit the collective computing power of the local computing nodes, distributed machine learning is a promising tool to accommodate such deluge data, especially when data is produced from different locations [2]. Several distributed optimization approaches have been developed to design distributed machine learning architectures such as distributed subgradient descent algorithm [3], [4] and alternating direction method of multipliers (ADMM) [5], [6], among which the ADMM typically achieves a fast convergence rate $\mathcal{O}(1/K)$, where K is the number of iterations [7]. Thus, in this paper, we aim to design distributed machine learning algorithm with ADMM.

Under the framework of ADMM, a large scale machine learning problem is divided into several sub-problems solved by a connected network of agents locally over local training data, and the local machine learning models are exchanged among the neighbors. However, as many recent works [8], [9], [10] indicate, the local machine learning models exchanged during the iterative process may result in privacy leakage of the sensitive training data such as medical records or financial data.

To prevent such information leakage, differential privacy [11], [12] has been exploited as a well-defined framework for performing machine learning over sensitive data. Intuitively, it works by injecting random noise to the model parameters so that an adversary with arbitrary background knowledge cannot confidently make any conclusions about whether a data sample is utilized in training a model or not. Many pioneering works have focused on integrating differential privacy with ADMM [8], [9], [10], [13]. In [8], Zhang and Zhu proposed a dual variable perturbation approach, where the dual variable of each agent at each ADMM iteration is perturbed. This approach can provide dynamic differential privacy, a new privacy framework capturing the distributed and iterative nature of ADMM. However, Zhang and Zhu only imposed a privacy constraint on each iteration and did not give a total privacy loss bound over the entire iterative procedure, which makes it hard to balance the tradeoff between the utility of the proposed algorithm and privacy guarantees. Later, in [9], Zhang et al. developed a penalty perturbation method and gave the total privacy loss of all agents during the entire process. Moreover, in [10], Zhang et al. employed the penalty perturbation method and modified the original ADMM to repeatedly use the existing computational results in order to further reduce the privacy loss. However, privacy analysis provided in above works [8], [9], [10] requires the objective functions of the learning problem are strongly convex. Our privacy analysis only needs to assume that the gradient of the loss function is bounded. Huang et al. in [13] and Ding et al. in [14] also performed privacy analysis under mild conditions of objective functions,

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

whereas their approaches need a central server to average all shared primal variables. Instead of requiring a central server, our approach is implemented in a fully decentralized manner.

In this paper, we propose a differentially private robust ADMM algorithm (PR-ADMM) by adding Gaussian noise with decaying variance to perturb exchanged variables at each iteration. To reduce the negative effects of noise addition, we propose two noise variance decay schemes: *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme*, and we utilize a threshold U to examine whether the results from neighbors are too noisy. The dynamic zCDP framework proposed in [15] is used to analyze the privacy guarantee of PR-ADMM, which achieves tight bound on privacy loss. Furthermore, we rigorously prove the convergence rate of PR-ADMM for general convex objectives. Our salient contributions are listed as follows.

- We propose a differentially private robust ADMM algorithm (PR-ADMM) by adding Gaussian noise with decay variance to address privacy concerns in distributed machine learning.
- To mitigate the effects of noise addition, we propose two noise variance decay schemes and set a threshold to eliminate the too noisy primal variables from neighboring agents. Moreover, we present the privacy analysis of PR-ADMM based on dynamic zCDP framework.
- We provide convergence analysis of PR-ADMM under above two variance decay schemes for general convex objectives. Note that in both cases, PR-ADMM exhibits a $\mathcal{O}(1/K)$ convergence rate, where K is the iteration number.
- We evaluate the efficacy of PR-ADMM by conducting extensive simulations using real-world datasets.

The rest of paper is organized as follows. Section II gives the problem formulation, and describes preliminaries. Then, the differentially private robust ADMM algorithm and privacy analysis are presented in Section III. In Section IV, we provide the convergence analysis of the proposed algorithm. The numerical experiments based on real-world datasets are shown in Section V. Finally, we draw conclusion remarks in Section VI.

Notations: In this paper, we denote $\|x\|_2$ as the Euclidean norm of a vector x and $\langle x, y\rangle$ as the inner product of two vectors x and y. Further, given a semidefinite matrix G, $\sqrt{x^TGx}$ represents the G-norm of x, i.e., $\|x\|_G$. We also denote $\phi_{max}(G)$ as the nonzero largest of G and $\phi_{min}(G)$ as the smallest nonzero singular value of G.

II. Problem Setting and Preliminaries

A. Problem Setting

Let a symmetric directed $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ represents a connected network containing N agents bidirectionally connected by E edges, where $\mathcal{N}$ is the set of vertexes with cardinality $|\mathcal{N}| = N$, and $\mathcal{E}$ is the set of arcs with $|\mathcal{E}| = 2E$. A connected network is a network that every agent can reach from every other agent. For any agent i, we let $\mathcal{V}_i$ denote the set of its neighboring agents, and the information can only be exchanged among its neighbors. Suppose that each agent i has a dataset $D_i = \{(y_i^n, z_i^n)\}_{n=1}^{|D_i|}$, where $y_i^n \in \mathcal{C}^*$ is a feature vector and $z_i^n \in \mathbb{R}^z$ is the label. The goal of the agents is to collectively train a classifier $w \in \mathbb{R}^d$ over the union dataset $\hat{D} = \cup_{i\in\mathcal{N}} D_i$ in a decentralized manner (i.e., no centralized controller/fusion center) while protecting the privacy of each data sample. Thus, we formulate the problem by regularized empirical risk minimization (ERM), which refers to find the model variable w by minimizing an objective function given by a loss function $\mathcal{L}(\cdot) : \mathbb{R}^z \times \mathcal{C}^* \times \mathbb{R}^d \to \mathbb{R}$ averaged over data samples plus a regularizer $\mathcal{R}(w) : \mathbb{R}^d \to \mathbb{R}$ to prevent overfitting. Specifically, we focus on the following problem

$$\min_{w\in\mathbb{R}^d} \sum_{i=1}^{N} \frac{1}{|D_i|} \sum_{n=1}^{|D_i|} \mathcal{L}(z_i^n, y_i^n, w) + \mathcal{R}(w). \tag{1}$$

We assume that each feature vector $\|y_i^n\|_2$ is normalized to $\|y_i^n\|_2 \le 1$. We also assume the loss function $\mathcal{L}(\cdot)$ is convex and V-Lipschitz. For example, if the loss function is binary logistic regression, i.e., $z_i^n \in \{-1, 1\}$, we have

$$\mathcal{L}(z_i^n, y_i^n, w) = \log(1 + \exp(-z_i^n w^T y_i^n)).$$

According to the literature [16], the Lipschitz constant V of logistic loss is 1, i.e., the tight upper bound of $\nabla\mathcal{L}(\cdot)$ is $\|\nabla\mathcal{L}(\cdot)\|_2 \le 1$.

ADMM, as a popular optimization method, can be used to solve ERM problem (1) in a decentralized fashion. In the following subsection, we will first introduce preliminaries about how to solve ERM problem (1) by ADMM.

B. Preliminaries

1) Conventional ADMM: To decentralize ERM problem (1), we introduce local classifiers $x_i \in \mathbb{R}^d$ for agent i, where x_i is a local copy of the common classifier w in (1). Additionally, a set of auxiliary variables $\{p_{ij} | i \in \mathcal{N}, j \in \mathcal{V}_i\}$ are introduced to enforce all the local classifiers reach consensus, i.e., $x_1 = x_2 = \cdots = x_N$. Then, the ERM problem (1) can be reformulated as follows

$$\begin{aligned} \min_{\{x_i\},\{p_{ij}\}} \quad & \textstyle\sum_{i=1}^{N} f_i(x_i) \\ \text{s.t.} \quad & x_i = p_{ij}, x_j = p_{ij},\ i \in \mathcal{N}, j \in \mathcal{V}_i, \end{aligned} \tag{2}$$

where $f_i(x_i) = \frac{1}{|D_i|}\sum_{n=1}^{|D_i|} \mathcal{L}(z_i^n, y_i^n, x_i) + \frac{1}{N}\mathcal{R}(x_i)$ [6]. Since agents are connected in the network, then problems (2) and (1) are equivalent [17]. Moreover, the objective function of (2) can be easily solved using ADMM in a decentralized and collaborative manner, where each agent i obtains a local classifier x_i by only minimizing objective $f_i(x_i)$ over its own dataset $D_i = \{(y_i^n, z_i^n)\}_{n=1}^{|D_i|}$.

For a clear presentation, according to [17], problem (2) can be written in a matrix form as follows

$$\begin{aligned} \min_{x,p} \quad & f(x) + g(p), \\ \text{s.t.} \quad & Ax + Bp = 0, \end{aligned} \tag{3}$$

where $x := [x_1, x_2, \cdots, x_N]^T \in \mathbb{R}^{Nd}$, p is a vector concatenating all $\{p_{ij}\}$, $g(p) = 0$ and $A := [A_1; A_2]$ with $A_1, A_2 \in$

$\mathbb{R}^{2Ed\times Nd}$ whose (q,i)-th element $(A_1)_{qi}=1,(A_2)_{qi}=1$ and all other elements are zeros if the q-th element of p is p_{ij}. Moreover, $B:=[-I_{2Ed};-I_{2Ed}]$, and aggregated function $f:\mathbb{R}^{Nd}\to\mathbb{R}$ is defined as $f(x)=\sum_{i=1}^{N}f_i(x_i)$.

The augmented Lagrangian function of (3) is given by

$$L_c(x,p,\lambda)=f(x)+\langle Ax+Bp,\lambda\rangle+\frac{\eta}{2}\|Ax+Bp\|_2^2, \quad (4)$$

where $\lambda\in\mathbb{R}^{4Ed}$ is Lagrangian multiplier and η is a positive penalty parameter.

With ADMM algorithm, alternatively, $L_c(x,p,\lambda)$ is minimized in terms of variables x, p and λ. At iteration $k+1$, the updates of ADMM are

$$\nabla f(x^{k+1})+A^T\lambda^k+\eta A^T(Ax^{k+1}+Bp^k)=0, \quad (5)$$

$$B^T\lambda^k+\eta B^T(Ax^{k+1}+Bp^{k+1})=0, \quad (6)$$

$$\lambda^{k+1}-\lambda^k-\eta(Ax^{k+1}+Bp^{k+1})=0. \quad (7)$$

If we let $\lambda=[\beta;\gamma]$ with $\beta,\gamma\in\mathbb{R}^{2EN}$, $H_+=A_1^T+A_2^T$ and $H_-=A_1^T-A_2^T$, the above ADMM updates can be simplified as

$$\nabla f(x^{k+1})+\alpha^k+2\eta Mx^{k+1}-\eta L_+x^k=0, \quad (8)$$

$$\alpha^{k+1}-\alpha^k-\eta L_-x^{k+1}=0, \quad (9)$$

where $\alpha=H_-\beta\in\mathbb{R}^{Nd}$ is a new Lagrange multiplier, and $M=\frac{1}{2}(L_++L_-)$ with $L_+=\frac{1}{2}H_+H_+^T$ and $L_-=\frac{1}{2}H_-H_-^T$. Note that L_+ and L_- are the extended signless and signed Laplacian matrices of the network.

Remember that $x:=[x_1,x_2,\cdots,x_N]^T\in\mathbb{R}^{Nd}$, where x_i is the local classifier of agent i. After simple manipulations, the matrix form of ADMM updates (8) and (9) are translated to the updates of agent i by

$$\nabla f_i(x_i^{k+1})+\alpha_i^k+2\eta|\mathcal{V}_i|x_i^{k+1}=\eta\left(|\mathcal{V}_i|x_i^k+\sum_{j\in\mathcal{V}_i}x_j^k\right), \quad (10)$$

$$\alpha_i^{k+1}=\alpha_i^k+\eta\left(|\mathcal{V}_i|x_i^{k+1}-\sum_{j\in\mathcal{V}_i}x_j^{k+1}\right), \quad (11)$$

where $\alpha_i\in\mathbb{R}^d$ is the local Lagrange multiplier of agent i and α is the concatenated form of all α_i. At iteration $k+1$, every agent i updates the local x_i^{k+1} through (10) using its previous x_i^k, α_i^k and its neighbors' previous result x_j^k with $j\in\mathcal{V}_i$, and then broadcasts x_i^{k+1} to all its neighboring agents $j\in\mathcal{V}_i$. After collecting all x_j^{k+1} from its neighbors, agent i updates its local multiplier α_i through (11).

2) Differential Privacy: Differential privacy (DP) as an important privacy paradigm is first presented in [12], which provides robust protection against a wide range of attacks by injecting random noise to perturb the released statistical results obtained from sensitive datasets. The definition of differential privacy is defined as follows.

Definition 1 (**Differential Privacy**). *A randomized algorithm $\mathcal{M}$ satisfies (ϵ,δ)-differential privacy if for any two adjacent datasets D and $\hat{D}$ that differ in only a single record, the absolute value of the privacy loss random variable of an output $o\in Range(\mathcal{M})$*

$$Z(o)=\ln\frac{\Pr[\mathcal{M}(D)=o]}{\Pr[\mathcal{M}(\hat{D})=o]}$$

is bounded by ϵ, with probability at least $1-\delta$.

A generic method of achieving (ϵ,δ)-differential privacy is Gaussian mechanism [11] that adds Gaussian noise, calibrated to the query function's sensitivity, to the output. The sensitivity captures the maximum difference of the query function by a single record in the worst case. We define the sensitivity as follows.

Definition 2 (**Sensitivity**). *The sensitivity of a query function $f(\cdot)$ that takes as input a dataset D is defined as*

$$\Delta_f=\max_{D,\hat{D}}\|f(D)-f(\hat{D})\|_2,$$

where D and $\hat{D}$ are any two neighboring datasets differing in at most one record.

Based on the definition of sensitivity, we show the Gaussian mechanism in the following theorem.

Theorem 1 (**Gaussian Mechanism [11]**). *For a query function $f:\mathcal{D}\to\mathcal{R}^d$ with sensitivity Δ_f, the Gaussian Mechanism that adds noise generated from the Gaussian distribution $\mathcal{N}(0,\sigma^2I_d)$ to the output of f satisfies (ϵ,δ)-differential privacy, where $\epsilon,\delta\in(0,1)$ and $\sigma\geq\frac{\sqrt{2\ln(1.25/\delta)}\Delta_f}{\epsilon}$.*

Concentrated differential privacy (CDP) is a recently proposed relaxation of differential privacy which aims to make privacy-preserving iterative algorithms more practical than for DP while still providing strong privacy guarantees

Zero-concentrated differential privacy (zCDP) [18] is a newly relaxation of differential privacy, which aims to obtain significantly tighter privacy bounds for privacy preserving iterative algorithms. The definition of ρ-zCDP is defined as follows.

Definition 3 (ρ-**zCDP**). *For all $\tau\in(1,\infty)$, a randomized algorithm $\mathcal{M}$ is ρ-zCDP if for any neighboring datasets D and $\hat{D}$, we have*

$$\mathbb{E}[e^{(\tau-1)Z(o)}]\leq e^{(\tau-1)\tau\rho},$$

where $Z(o)$ is the privacy loss variable of an outcome of $\mathcal{M}$.

The following three lemmas [18] show that the Gaussian mechanism satisfies zCDP, the composition theorem of ρ-zCDP and the relationship between ρ-zCDP and (ϵ,δ)-DP.

Lemma 1. *The Gaussian mechanism in Theorem 1, satisfies $\Delta_f^2/(2\sigma^2)$-zCDP.*

Lemma 2. *If randomized mechanisms $\mathcal{M}_1,\cdots,\mathcal{M}_k$ satisfies ρ_1-zCDP,$\cdots,\rho_k$-zCDP, their composition defined as $(\mathcal{M}_1,\cdots,\mathcal{M}_k)$ is $\sum_{i=1}^k\rho_k$-zCDP.*

Lemma 3. *If a randomized mechanism $\mathcal{M}$ satisfies ρ-zCDP, then for any $\delta \in (0,1)$ $\mathcal{M}$ satisfies $(\rho + 2\sqrt{\rho \ln(1/\delta)}, \delta)$-differential privacy.*

III. Differentially Private Robust ADMM

In this section, we propose a novel differentially private robust ADMM algorithm (PR-ADMM). Specifically, to provide differential privacy of each training data point, we let individual agent adds Gaussian noise to the local classifiers before sharing to neighboring agents. Moreover, we propose two techniques to mitigate the effects of noise addition and guarantee convergence property. The first technique is that we design two kinds of noise variance decay schemes: *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme*, to carefully adjust the scale of noise in the iterative process. The second technique is to set a threshold U to decide whether the noisy classifiers from neighboring agents introduce too much noise or not. If it is, this agent would not use these noisy classifiers to do ADMM updates. In addition, the privacy framework, dynamic zero-concentrated differential privacy, is utilized to measure the privacy guarantee of PR-ADMM.

The details of PR-ADMM are given in Algorithm 1. First of all, we choose a Gaussian noise variance decay scheme from *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme* to determine the relationship between $(\sigma^2)_i^{k+1}$ and $(\sigma^2)_i^k$. In each iteration, each agent i computes the primal variable x_i^{k+1} by solving the subproblem in (12) over its own dataset D_i (Line 15). Then, each agent i computes the value of $\sum_{t=0}^k \|\tilde{x}_i^t - \tilde{x}_j^t\|_2$ for every neighboring agent $j \in \mathcal{V}_i$, which is a good criterion of measuring the deviation from a consensus at current iteration [19]. When $\sum_{t=0}^k \|\tilde{x}_i^t - \tilde{x}_j^t\|_2$ is greater than a threshold U, it means that agent j's variable $\tilde{x}_j^k$ is too noisy. Then each agent i will replace $\tilde{x}_j^k$ with its own variable $\tilde{x}_i^k$ to do the primal variable update (Line 15). After that, each agent adds a noise ξ_i^{k+1} drawn from a Gaussian distribution $\mathcal{N}(0, (\sigma^2)_i^{k+1} I_d)$ to perturb the local variable x_i^{k+1}, according to the chosen noise variance decay scheme. Then, the perturbed local variable $\tilde{x}_i^{k+1}$ is sent by agent i to all its neighboring agents $j \in \mathcal{V}_i$. At last, each agent updates the dual variable α_i^{k+1} through (13). The corresponding ADMM iterations are as follows.

$$\nabla f_i(x_i^{k+1}) + \alpha_i^k + 2\eta|\mathcal{V}_i|x_i^{k+1} = \eta\left(|\mathcal{V}_i|\tilde{x}_i^k + \sum_{j\in\mathcal{V}_i}\tilde{x}_j^k\right), \tag{12}$$

$$\alpha_i^{k+1} = \alpha_i^k + \eta\left(|\mathcal{V}_i|\tilde{x}_i^{k+1} - \sum_{j\in\mathcal{V}_i}\tilde{x}_j^{k+1}\right). \tag{13}$$

Now how to set the value of threshold U is important. Since $\sum_{t=0}^k \|\tilde{x}_i^t - \tilde{x}_j^t\|_2 \le U$ and $\sum_{t=0}^k \|Q\tilde{x}^t\|_2 = \frac{1}{\sqrt{2}}\sum_{t=0}^k \sum_{(i\in\mathcal{N}, j\in\mathcal{V}_i)} \|\tilde{x}_i^t - \tilde{x}_j^t\|_2$ with $Q = \sqrt{L_-/2}$, then the value of noisy local deviation statistics $\sum_{k=0}^K \|Q\tilde{x}^k\|_2$ is upper bounded by $\sqrt{2}EU$. Note that if we set the threshold U as $U = \hat{U}/(\sqrt{2}E)$, we have $\sum_{k=0}^K \|Q\tilde{x}^k\|_2$ is upper bounded by $\hat{U}$, where $\hat{U}$ is the upper bound of the noise-free local deviation statistics $\sum_{k=0}^K \|Qx^k\|_2$. The upper bound $\hat{U}$ can be obtained from the following Lemma.

Algorithm 1 Differentially Private Robust ADMM

1: **Input:** datasets $\{D_i\}_{i=1}^N$; initial variables $x_i^0 \in \mathbb{R}^d$ and $\alpha_i^0 = 0_d$; threshold U; time period K_p, decay rate $R_P \in (0,1)$ and $R_T > 0$; initial variances $(\sigma^2)_i^1$ for all agents i;
2: Choose noise variance decay scheme (Line 3-7).
3: **if** *Periodic Linear Decay Scheme* is chosen **then**
4: $\quad(\sigma^2)_i^{k+1} = (\sigma^2)_i^1 \times R_P^{\lfloor k/K_p \rfloor}$.
5: **else**
6: $\quad(\sigma^2)_i^{k+1} = (\sigma^2)_i^1 \times \frac{1}{R_T k(k+1)}$. *//Iteration-Based Decay Scheme*
7: **end if**
8: **for** $k = 0, \ldots, K-1$ **do**
9: $\quad$**for** $i = 1, \cdots, N$ **do**
10: $\quad\quad$**if** $\sum_{t=0}^k \|\tilde{x}_i^t - \tilde{x}_j^t\|_2 > U$ with $j \in \mathcal{V}_i$ **then**
11: $\quad\quad\quad$Replace $\tilde{x}_j^k$ with $\tilde{x}_i^k$.
12: $\quad\quad$**else**
13: $\quad\quad\quad$Keep $\tilde{x}_j^k$.
14: $\quad\quad$**end if**
15: $\quad\quad$Compute x_i^{k+1} by solving $\nabla f_i(x_i^{k+1}) + \alpha_i^k + 2\eta|\mathcal{V}_i|x_i^{k+1} - \eta\left(|\mathcal{V}_i|\tilde{x}_i^k + \sum_{j\in\mathcal{V}_i}\tilde{x}_j^k\right) = 0$.
16:
17: $\quad\quad$Generate noise $\xi_i^{k+1} \sim \mathcal{N}(0, (\sigma^2)_i^{k+1} I_d)$.
18: $\quad\quad$Perturb x_i^{k+1}: $\tilde{x}_i^{k+1} = x_i^{k+1} + \xi_i^{k+1}$.
19: $\quad$**end for**
20: $\quad$**for** $i = 1, \cdots, N$ **do**
21: $\quad\quad$Broadcast $\tilde{x}_i^{k+1}$ to all neighbors $j \in \mathcal{V}_i$.
22: $\quad$**end for**
23: $\quad$**for** $i = 1, \cdots, N$ **do**
24: $\quad\quad$Compute α_i^{k+1} from $\alpha_i^{k+1} = \alpha_i^k + \eta\left(|\mathcal{V}_i|\tilde{x}_i^{k+1} - \sum_{j\in\mathcal{V}_i}\tilde{x}_j^{k+1}\right)$.
25: $\quad$**end for**
26: **end for**
27: **Output:** $\{\tilde{x}_i^K\}_{i=1}^N$ for any $i \in \mathcal{N}$.

Lemma 4. *If we randomly initialize x^0 and the gradient $\nabla f(x)$ is bounded as $\|\nabla f(x)\|_2 \le V_2$ and the feasible x is bounded as $\|x\|_2 \le V_1$, in conventional ADMM (8-9), we have*

$$\sum_{k=0}^K \|Qx^k\|_2 \le \hat{U}$$
$$= (\phi_{max}(L_+) + 2\phi_{max}(Q))V_1^2 + \frac{2V_2^2}{\eta^2\phi_{min}(L_-)} + 1,$$

Proof. The upper bound of $\sum_{k=0}^K \|Qx^k\|_2$ can be directly derived from Lemma 8 in [20] if we use the inequality $\|a+b\|_2^2 \le \|a\|_2^2 + \|b\|_2^2$ for all $a, b \in \mathbb{R}^n$ and $\|Qx^0\| \le \phi_{max}(Q)\|x^0\|$. □

In distributed settings, the output of the algorithm includes all intermediate results generated at every stage of the learning and final result. For this reason, we present the dynamic zero concentrated differential privacy framework to quantify the privacy leakage of ADMM-based algorithms.

Definition 4 (Dynamic ρ^k-zCDP [15]). *Consider a connected network $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ that contains a set of agents/nodes $\mathcal{N} = \{1, \cdots, N\}$ and each agent possesses a training dataset D_i, and $\tilde{D} = \cup_{i\in\mathcal{N}} D_i$. We denote $\mathcal{K}$ is a randomized version of ADMM algorithm with updates (10) and (11). Let $\mathcal{K}_i^k$ be the agent-i-dependent sub-algorithm of $\mathcal{K}$, which corresponds to ADMM update (10) at k-iteration that outputs x_i^k. A randomized algorithm $\mathcal{K}$ gives dynamic ρ_i^k-zCDP if for all datasets D_i and $\hat{D}_i$ differing at most a single record, and for all agents $i \in \mathcal{N}$, and for all k during a learning process, the privacy loss variable of an outcome $o \in Range(\mathcal{K})$*

$$Z_i^k(o) = \ln \frac{\Pr[x_{i,D_i}^k = o]}{\Pr[x_{i,\hat{D}_i}^k = o]}$$

satisfies

$$\mathbb{E}[e^{(\tau-1)Z_i^k(o)}] \leq e^{(\tau-1)\tau\rho_i^k},$$

$\forall\tau \in (1, \infty)$.

For dynamic zCDP algorithms, the adversaries cannot obtain additional information by observing the intermediate results and final results at each step. Since the added noise may destroy the convergence behavior and lead to poor model performance. It is vital to carefully design and adjust privacy budget allocation for each iteration, i.e., dynamically reducing the noise variance in the iterative process, instead of just adding a noise ξ_i^{k+1} for agent i in iteration $k+1$ [8].

Here we propose two kinds of noise variance decay schemes, which effectively reduce the bad impact of noise and stabilize the convergence property.

Periodic Linear Decay Scheme In a period of time K_p, there is a decay rate $R_P \in (0, 1)$ to describe the decrease of noise variance. The mathematical form is

$$(\sigma^2)_i^{k+1} = (\sigma^2)_i^1 \times R_P^{\lfloor k/K_p \rfloor}, \tag{14}$$

where $(\sigma^2)_i^1$ is the initial noise variance determined by agent i and the value of K_p decides how often to reduce noise variance regards the number of iterations. Without loss of generality, suppose the total iteration number K is divisible by K_p.

Iteration-Based Decay Scheme In the iteration $k+1$, the noise variance $(\sigma^2)_i^{k+1}$ can be obtained based on the previous noise variance. It has the mathematical form

$$(\sigma^2)_i^{k+1} = (\sigma^2)_i^1 \times \frac{1}{R_T k(k+1)}, \tag{15}$$

where $R_T > 0$ is decay rate.

Before showing PR-ADMM satisfies dynamic zCDP, we first estimate the sensitivity of the local primal variable x_i^{k+1} as shown in the following lemma.

Lemma 5. *The sensitivity of local primal variable x_i^{k+1}, denoted by Δ_i, is $\frac{V}{\eta|\mathcal{V}_i|}$, where V is the Lipschitz constant of the loss function $\mathcal{L}(\cdot)$, $|\mathcal{V}_i|$ is the number of neighboring agents of agent i, and η is a positive penalty parameter.*

Proof. According to subproblem (10) and Definition 2, we have

$$x_{i,D_i}^{k+1} = -\frac{1}{2\eta|\mathcal{V}_i|}\nabla f_i(x_i^{k+1}, D_i) + \frac{1}{2|\mathcal{V}_i|}\left(|\mathcal{V}_i|\tilde{x}_i^k + \sum_{j\in\mathcal{V}_i}\tilde{x}_j^k\right) - \frac{1}{2\eta|\mathcal{V}_i|}\alpha_i^k,$$

$$x_{i,\hat{D}_i}^{k+1} = -\frac{1}{2\eta|\mathcal{V}_i|}\nabla f_i(x_i^{k+1}, \hat{D}_i) + \frac{1}{2|\mathcal{V}_i|}\left(|\mathcal{V}_i|\tilde{x}_i^k + \sum_{j\in\mathcal{V}_i}\tilde{x}_j^k\right) - \frac{1}{2\eta|\mathcal{V}_i|}\alpha_i^k,$$

where D_i and $\hat{D}_i$ are two neighboring datasets. Without loss of generality, suppose only the first data sample in D_i and $\hat{D}_i$ is different, say (y_i^1, z_i^1) and $(\hat{y}_i^1, \hat{z}_i^1)$ respectively. Then by the definitions of sensitivity, we have

$$\begin{aligned}
\Delta_i &= \|x_{i,D_i}^{k+1} - x_{i,\hat{D}_i}^{k+1}\|_2 \\
&= \frac{1}{2\eta|\mathcal{V}_i|}\|\nabla f_i(x_i^{k+1}, D_i) - \nabla f_i(x_i^{k+1}, \hat{D}_i)\|_2 \\
&= \frac{1}{2\eta|\mathcal{V}_i|}\|\frac{1}{|D_i|}\sum_{n=1}^{|D_i|}\nabla\mathcal{L}(y_i^n, z_i^n, x_i^{k+1}) + \frac{1}{N}\nabla\mathcal{R}(x_i^{k+1}) \\
&\quad - \frac{1}{|\hat{D}_i|}\sum_{n=1}^{|\hat{D}_i|}\nabla\mathcal{L}(\hat{y}_i^n, \hat{z}_i^n, x_i^{k+1}) - \frac{1}{N}\nabla\mathcal{R}(x_i^{k+1})\|_2 \\
&= \frac{1}{2\eta|\mathcal{V}_i|}\|\nabla\mathcal{L}(y_i^1, z_i^1, x_i^{k+1}) - \nabla\mathcal{L}(\hat{y}_i^1, \hat{z}_i^1, x_i^{k+1})\|_2 \\
&\leq \frac{V}{\eta|\mathcal{V}_i|}.
\end{aligned}$$

In the last inequality, we use the fact the Lipschitz constant V is an upper bound of $\|\nabla\mathcal{L}(\cdot)\|_2$. □

The following theorems show the privacy guarantee of PR-ADMM.

Theorem 2. *The PR-ADMM algorithm satisfies the dynamic ρ_i^{k+1}-zCDP, where $\rho_i^{k+1} = \frac{V^2}{2\eta^2|\mathcal{V}_i|^2(\sigma^2)_i^{k+1}}$.*

Proof. The privacy loss variable of $\tilde{x}_i^{k+1}$ on an output o over two neighboring datasets D_i and $\hat{D}_i$ is

$$Z_i^{k+1}(o) = \ln \frac{\Pr[\tilde{x}_{i,D_i}^{k+1} = o]}{\Pr[\tilde{x}_{i,\hat{D}_i}^{k+1} = o]}.$$

Since $\tilde{x}_i^{k+1} = x_i^{k+1} + \xi_i^{k+1}$ and $\xi_i^{k+1} \sim \mathcal{N}(0, (\sigma^2)_i^{k+1} I_d)$, the probability distribution $\tilde{x}_{i,D_i}^{k+1}$ is $\mathcal{N}(x_{i,D_i}^{k+1}, (\sigma^2)_i^{k+1} I_d)$, and the probability distribution of $\tilde{x}_{i,\hat{D}_i}^{k+1}$ is $\mathcal{N}(x_{i,\hat{D}_i}^{k+1}, (\sigma^2)_i^{k+1} I_d)$.

According to Lemma 2.5 in [18] and $\forall \tau \in (1, \infty)$, the Rényi divergence is given by

$$\begin{aligned} &D_\tau(\mathcal{N}(x_{i,D_i}^{k+1}, (\sigma^2)_i^{k+1} I_d) \| \mathcal{N}(x_{i,\hat{D}_i}^{k+1}, (\sigma^2)_i^{k+1} I_d)) \\ &= \frac{\tau \| x_{i,D_i}^{k+1} - x_{i,\hat{D}_i}^{k+1} \|_2^2}{2(\sigma^2)_i^{k+1}} \\ &= \frac{\tau \Delta_i^2}{2(\sigma^2)_i^{k+1}}. \end{aligned}$$

Then,

$$\begin{aligned} &\mathbb{E}[e^{(\tau-1)Z_i^{k+1}(o)}] \\ &\leq e^{(\tau-1)D_\tau(\mathcal{N}(x_{i,D_i}^{k+1}, (\sigma^2)_i^{k+1} I_d) \| \mathcal{N}(x_{i,\hat{D}_i}^{k+1}, (\sigma^2)_i^{k+1} I_d))} \\ &= e^{(\tau-1)\tau \Delta_i^2 / [2(\sigma^2)_i^{k+1}]} \\ &\leq e^{(\tau-1)\tau \frac{V^2}{2\eta^2 |\mathcal{V}_i|^2 (\sigma^2)_i^{k+1}}} \\ &= e^{(\tau-1)\tau \rho_i^{k+1}}. \end{aligned}$$

Therefore, PR-ADMM provides the dynamic ρ_i^{k+1}-zCDP at each agent i with $\rho_i^{k+1} = \frac{V^2}{2\eta^2 |\mathcal{V}_i|^2 (\sigma^2)_i^{k+1}}$. □

The parameter ρ_i^{k+1} in Theorem 2 only inspects the privacy loss of one agent in each iteration. However, it does not show the privacy guarantee when an adversary uses the revealed results from all iterations to perform inference. Therefore, the total privacy loss over the entire computational process and the entire network should be calculated. For two kinds of noise variance decay schemes: *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme*, we leverage (ϵ, δ)-differential privacy to derive the total privacy loss as shown in the Theorem 3 and Theorem 4, respectively.

Theorem 3. *For any $R_P \in (0,1)$ and $\delta \in (0,1)$, if Periodic Linear Decay Scheme is chosen, the PR-ADMM algorithm is (ϵ, δ)-differential privacy with $\epsilon = \max_{i \in \mathcal{N}} \rho_i^{total} + 2\sqrt{\rho_i^{total} \ln 1/\delta}$, where*

$$\rho_i^{total} = \frac{V^2}{2\eta^2 |\mathcal{V}_i|^2 (\sigma^2)_i^1} \left(\frac{K_p (1 - R_P^{K/K_p})}{R_P^{K/K_p - 1} - R_P^{K/K_p}} - 1 \right),$$

and K_p is time period, and $R_P \in (0,1)$ is the decay rate, and K is the total number of iterations.

Proof. According to Theorem 2, PR-ADMM satisfies dynamic ρ_i^{k+1}-zCDP. It ensures that each primal variable x_i^{k+1} perturbed by noise drawn from the Gaussian distribution $\mathcal{N}(0, (\sigma^2)_i^{k+1} I_d)$ is ρ_i^{k+1}-zCDP at $k+1$ iteration. By the composition theorem in Lemma 2 and for each agent, PR-ADMM provides $\sum_{k=0}^{K-1} \rho_i^{k+1}$-zCDP. Since *Periodic Linear Decay Scheme* is chosen, together with the result in Theorem 2, we have $\rho_i^{k+1} = \rho_i^1 / R_P^{\lfloor k/K_p \rfloor}$ and PR-ADMM is ρ_i^{total}-zCDP for each agent with

$$\rho_i^{total} = \rho_i^1 \left(\frac{K_p (1 - R_P^{K/K_p})}{R_P^{K/K_p - 1} - R_P^{K/K_p}} - 1 \right).$$

By Lemma 3 and $\forall \delta \in (0,1)$, PR-ADMM satisfies $(\epsilon_i^{total}, \delta)$-differential privacy with $\epsilon_i^{total} = \rho_i^{total} + 2\sqrt{\rho_i^{total} \ln 1/\delta}$. Therefore, considering all of agents, the total privacy loss of PR-ADMM is bounded by (ϵ, δ)-differential privacy with $\epsilon = \max_{i \in \mathcal{N}} \epsilon_i^{total}$. □

Theorem 4. *For any $R_T > 0$ and $\delta \in (0,1)$, if Iteration-Based Decay Scheme is chosen, the PR-ADMM algorithm is (ϵ, δ)-differential privacy with*

$$\begin{aligned} \epsilon = \max_{i \in \mathcal{N}} \Bigg(&\frac{\rho_i^1 (R_T K (K^2 - 1) + 3)}{3} \\ &+ 2\sqrt{\frac{\rho_i^1 (R_T K (K^2 - 1) + 3) \ln 1/\delta}{3}} \Bigg) \end{aligned}$$

where $\rho_i^1 = \frac{V^2}{2\eta^2 |\mathcal{V}_i|^2 (\sigma^2)_i^1}$ and K is the total number of iterations.

Proof. Since *Iteration-Based Decay Scheme* is chosen, together with the result in Theorem 2, we have $\rho_i^{k+1} = R_T k(k+1)\rho_i^1$. Then PR-ADMM is ρ_i^{total}-zCDP for each agent with

$$\rho_i^{total} = \rho_i^1 \frac{R_T K (K^2 - 1) + 3}{3}.$$

By Lemma 3 and $\forall \delta \in (0,1)$, PR-ADMM satisfies $(\epsilon_i^{total}, \delta)$-differential privacy, where

$$\begin{aligned} \epsilon_i^{total} = \rho_i^{total} + 2\sqrt{\rho_i^{total} \ln 1/\delta} = &\frac{\rho_i^1 (R_T K (K^2 - 1) + 3)}{3} \\ &+ 2\sqrt{\frac{\rho_i^1 (R_T K (K^2 - 1) + 3) \ln 1/\delta}{3}}. \end{aligned}$$

Therefore, considering all of agents, the total privacy loss of PR-ADMM is bounded by (ϵ, δ)-differential privacy with

$$\begin{aligned} \epsilon &= \max_{i \in \mathcal{N}} \epsilon_i^{total} \\ &= \max_{i \in \mathcal{N}} \Bigg(\frac{\rho_i^1 (R_T K (K^2 - 1) + 3)}{3} \\ &\quad + 2\sqrt{\frac{\rho_i^1 (R_T K (K^2 - 1) + 3) \ln 1/\delta}{3}} \Bigg) \end{aligned}$$

□

IV. Convergence Analysis

In this section, we present the convergence analysis of proposed PR-ADMM algorithm for general convex objective functions. We also write the updates of PR-ADMM in matrix forms.

$$\nabla f(x^{k+1}) + \alpha^k + 2\eta M x^{k+1} - \eta L_+ \tilde{x}^k = 0, \quad (16)$$

$$\alpha^{k+1} - \alpha^k - \eta L_- \tilde{x}^{k+1} = 0, \quad (17)$$

where $\tilde{x}^k = x^k + \xi^k$ and $\xi^k \in \mathbb{R}^{Nd}$ is a vector concatenating all noise variables $\{\xi_i^k\}$.

Given the perturbed primal variable $\tilde{x}^k$, two auxiliary sequences r^k and q^k, and a matrix G are defined as follows

$$r^k = \sum_{s=0}^{k} Q \tilde{x}^s, \; q^k = \begin{pmatrix} r^k \\ x^k \end{pmatrix}, \; G = \begin{pmatrix} \eta I & 0 \\ 0 & \eta L_+ / 2 \end{pmatrix},$$

where $Q = \sqrt{L_-/2}$. Since the network is connected, the Laplcaian matrix L_- is positive semi-definite.

Substituting (17) into (16), we obtain $x^{k+1} = -\frac{M^{-1}\nabla f(x^{k+1})}{2\eta} + \frac{M^{-1}L_+\tilde{x}^k}{2} - \frac{M^{-1}L_-}{2}\sum_{s=0}^{k}\tilde{x}^s$. Based on the auxiliary sequence r^k and the fact $M = (L_- + L_+)/2$, we further have $\frac{\nabla f(x^{k+1})}{\eta} + 2Qr^{k+1} + L_+(\tilde{x}^{k+1} - \tilde{x}^k) = 2M^{-1}\xi^{k+1}$.

Proposition 1. *For any $r \in \mathbb{R}^{Nd}$ and $k > 0$, we have*

$$\begin{aligned}&\frac{f(x^{k+1}) - f(x^*)}{\eta} + \left\langle 2r, Qx^{k+1}\right\rangle \\ &\le \frac{1}{\eta}\left(\|q^k - q^*\|_G^2 - \|q^{k+1} - q^*\|_G^2\right) + \frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^k\|_2^2 \\ &\quad + \left\langle \xi^{k+1}, 2Q(r^{k+1} - r)\right\rangle,\end{aligned}$$

where x^* is the optimal solution of (3) and $q^* = \begin{pmatrix} r \\ x^* \end{pmatrix}$.

We can now prove the following convergence results of PR-ADMM for the general convex problem.

Theorem 5. *Suppose the objective function $f(x)$ is general convex. In PR-ADMM, if Periodic Linear Decay Scheme is chosen, we have*

$$\begin{aligned}\mathbb{E}[f(\hat{x}^K) - f(x^*)] \le \frac{\eta}{K}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) \\ + \frac{\eta}{K}\underbrace{\frac{d\phi_{max}^2(L_+)K_p\sum_{i=1}^{N}(\sigma^2)_i^1}{2\phi_{min}(L_-)(1-R_P)}}_{\text{Accumulated noise term}}\end{aligned}$$

with $0 < R_P < 1$ and time period K_p, where the expectation is taking with respect to the noise and $\hat{x}^K = \frac{1}{K}\sum_{k=1}^{K}x^k$.

Proof. Summing Proposition 1 from $k = 0$ to $k = K - 1$, we have

$$\begin{aligned}&\frac{1}{\eta}\left(\sum_{k=1}^{K} f(x^k) - f(x^*)\right) + \left\langle 2r, Qx^k\right\rangle \\ &\le \sum_{k=1}^{K}\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{k-1}\|_2^2 + \left\langle \xi^k, 2Q(r^k - r)\right\rangle\right) \\ &\quad + \frac{1}{\eta}\|q^0 - q^*\|_G^2 \\ &\le \sum_{k=1}^{K}\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{k-1}\|_2^2 + \|2Q\xi^k\|_2\left(\hat{U} + \|r\|_2\right)\right) \\ &\quad + \frac{1}{\eta}\|q^0 - q^*\|_G^2 \\ &\le \sum_{k=1}^{K}\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{k-1}\|_2^2\right) + \frac{1}{\eta}\|q^0 - q^*\|_G^2 \\ &\quad + \sum_{k=1}^{K}\|2Q\xi^k\|_2\left(\hat{U} + \|r\|_2\right) \\ &\le \sum_{k=1}^{K}\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{k-1}\|_2^2\right) + 2\hat{U}\left(\hat{U} + \|r\|_2\right) \\ &\quad + \frac{1}{\eta}\|q^0 - q^*\|_G^2,\end{aligned}$$

where we use the fact that $\sum_{k=1}^{K}\|Q\xi^k\|_2 \le \hat{U}$. Letting $r = 0$, there is

$$\begin{aligned}\frac{1}{\eta}\left(\sum_{k=0}^{K} f(x^k) - f(x^*)\right) \le \sum_{k=1}^{K}\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{k-1}\|_2^2\right) \\ + 2\hat{U}^2 + \|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2.\end{aligned}$$

By taking expectation of above function and using Jensen's inequality and convexity of the functions, we have

$$\begin{aligned}&\mathbb{E}[f(\hat{x}^K) - f(x^*)] \\ &\le \frac{\eta}{K}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) \\ &\quad + \frac{\eta}{K}\sum_{k=1}^{K}\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\mathbb{E}\|\xi^{k-1}\|_2^2 \\ &\le \frac{\eta}{K}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) \\ &\quad + \frac{\eta}{K}\frac{d\phi_{max}^2(L_+)K_p\sum_{i=1}^{N}(\sigma^2)_i^1}{2\phi_{min}(L_-)(1-R_P)},\end{aligned}$$

where $\hat{x}^K = \frac{1}{K}\sum_{k=1}^{K}x^k$. In the second inequality, we use the sum of infinity terms of geometric sequence. □

Theorem 6. *Suppose the objective function $f(x)$ is general convex. In PR-ADMM, if Iteration-Based Decay Scheme is chosen, we have*

$$\begin{aligned}\mathbb{E}[f(\hat{x}^K) - f(x^*)] \le \frac{\eta}{K}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) \\ + \frac{\eta}{K}\underbrace{\frac{d\phi_{max}^2(L_+)\sum_{i=1}^{N}(\sigma^2)_i^1}{2\phi_{min}(L_-)R_T}}_{\text{Accumulated noise term}}\end{aligned}$$

with $R_T > 0$, where the expectation is taking with respect to the noise and $\hat{x}^K = \frac{1}{K}\sum_{k=1}^{K}x^k$.

Proof. Similar to the proof of Theorem 5, we have

$$\begin{aligned}&\mathbb{E}[f(\hat{x}^K) - f(x^*)] \\ &\le \frac{\eta}{K}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) \\ &\quad + \frac{\eta}{K}\sum_{k=1}^{K}\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\mathbb{E}\|\xi^{k-1}\|_2^2 \\ &\le \frac{\eta}{K}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) \\ &\quad + \frac{\eta}{K}\frac{d\phi_{max}^2(L_+)\sum_{i=1}^{N}(\sigma^2)_i^1}{2\phi_{min}(L_-)R_T},\end{aligned}$$

where $\hat{x}^K = \frac{1}{K}\sum_{k=1}^{K}x^k$. □

Remark 1. *As we can see from the above two theorems, if the variance of Gaussian noise decays according to Periodic Linear Decay Scheme and Iteration-Based Decay Scheme, then the averaged function value approaches the minimum function value with a convergence rate of $\mathcal{O}(1/K)$. Note that the non-private decentralized ADMM in [19] also achieves a $\mathcal{O}(1/K)$ rate for a general convex problem.*

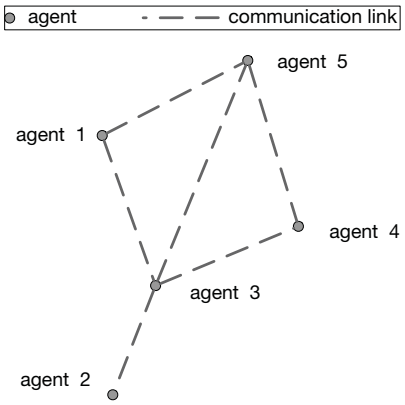


Fig. 1. A network with five agents ($N = 5$).

V. Numerical Experiments

In this section, we experimentally evaluate the performance of PR-ADMM under two noise variance decay schemes: *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme* on binary classification tasks. Specifically, our evaluation considers logistic regression as the loss function.

Logistic regression The logistic regression loss function on a data sample (y, z) with $z \in \{+1, -1\}$ is defined as $\mathcal{L}(z, y, w) = \log(1 + \exp(-zw^T y))$, and the regularizer $\mathcal{R}(w) = \Lambda \|w\|_2^2$.

Data preprocessing We also use the Adult dataset from UCI Machine Learning Repository, as in [8], [9], [10]. The dataset consists of 48,842 personal records with, including age, work-class, sex, race, income, etc. Our goal is to predict whether the annual income an individual is more than \$50k or not. We preprocess the data by removing all individuals with missing values. We also normalize the feature vectors such that its l_2 norm is at most 1 while transforming labels of Adult $\{> 50\text{k}, \leq 50\text{k}\}$ to $\{+1, -1\}$.

Baseline algorithms In our experiments, we compare our PR-ADMM algorithm against four benchmark algorithms, namely, DVP, M-ADMM, and R-ADMM and Non-private. The private ADMM algorithm using dual variable perturbation is called DVP [8]. ADMM with a penalty perturbation, proposed in [9], is referred to M-ADMM. Based on the penalty perturbation, R-ADMM with repeatedly using the existing computational results to make updates is proposed in [10]. Furthermore, we denote the non-private ADMM algorithm [17] as Non-private baseline. Finally, we denote our PR-ADMM with *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme* as PR-ADMM (Per) and PR-ADMM (Iter), respectively.

Setup As shown in Figure 1, we consider a bidirectionally connected network with $N = 5$ agents, and each agent is randomly assigned $|D_i| = 8000$ data samples for training. In the testing process, we random sample 1000 instance from the remaining dataset. We set $\eta = 0.5$ and the total iteration number $K = 50$. For privacy parameters, we consider the total privacy loss $\epsilon = \{0.1, 0.5, 1, 5, 10\}$ and $\delta = 0.0001$.

Evaluation We evaluate the convergence of the algorithms with respect to the average loss defined by $\mathcal{L}_k = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{|D_i|} \sum_{n=1}^{|D_i|} \mathcal{L}(y_i^n, z_i^n, x_i^k)$. Moreover, the accuracy is measured by classification accuracy defined as follows

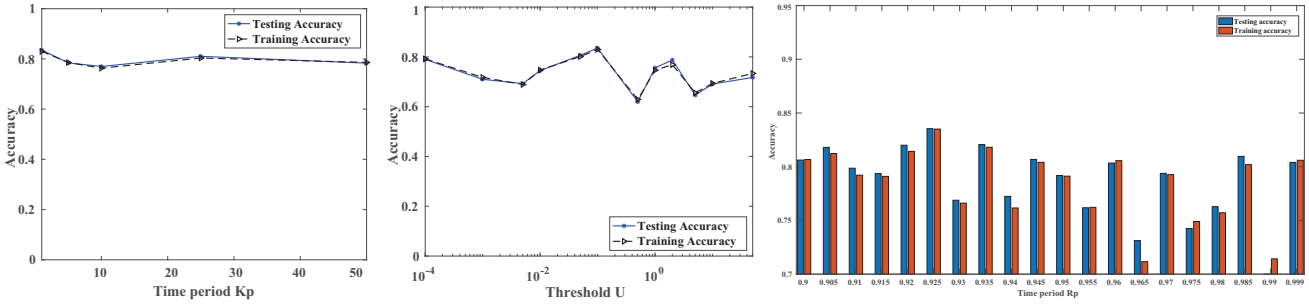
$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}.$$

Since each of the baseline algorithms introduces randomness due to noise, we perform 10 independent runs of algorithms and report the mean of accuracy (Testing and Training). Moreover, we also record both the mean and standard deviation of the average loss. The smaller the standard deviation, the more stable of the algorithm.

As we can see from Algorithm 1, there are some hyperparameters for tuning, such as threshold U, time period K_p, decay rate R_P and R_T. Given the total privacy loss $\epsilon = 10$ and the total iteration number $K = 50$, we manipulate different hyperparameters separately, while keeping the rest unchanged to show their impact on testing/training accuracy[1]. Figure 2(a) shows the impacts of time period K_p on the performance of PR-ADMM with *Periodic Linear Decay Scheme*. The value of time period K_p represents how often to reduce noise variance regards the number of iterations. From the figure, we see that $K_p = 1$ achieves the highest testing/training accuracy. Figure 2(b) illustrates how classification accuracy changes with varying values of U, the threshold U is to eliminate the too noisy results from neighbors. As it was shown in the figure, $U = 0.1$ achieves the best testing/training accuracy. Figure 2(c) describes how classification accuracy changes with varying values of R_P. The parameter R_P controls how fast the noise variance decreases. As it can be seen from the figure, $R_P = 0.925$ is best. To see the impact of decay rate R_T and threshold U on performance of PR-ADMM with *Iteration-Based Decay Scheme*, Figure 3(b) and Figure 3(a) describe how R_T and U affect the testing/training accuracy, respectively. From these figures, we see the testing/training accuracy are highest when $R_T = 0.015$ and $U = 1$.

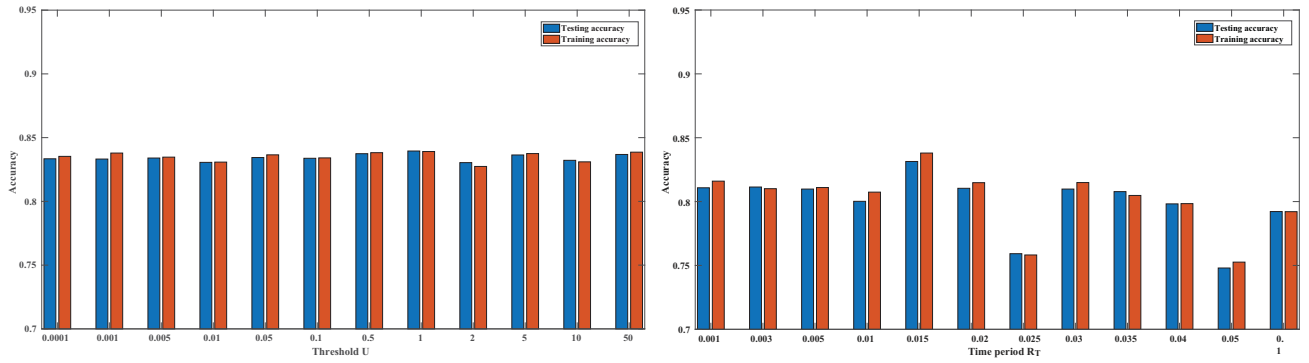
Figure 4 compares the convergence performance of PR-ADMM algorithm with other baseline algorithms. Compared to DVP and M-ADMM, R-ADMM indeed improves the privacy-utility tradeoff significantly, i.e., R-ADMM has the low value of average loss, with repeatedly using the existing computational results. However, R-ADMM performs many iterations that do not help decrease the average loss value. As it was shown in the figure, both PR-ADMM (Per) and PR-ADMM (Iter) significantly outperform all other algorithms and get close to the best achievable average loss (Non-private) during the entire iterative process. This is because, with carefully adjusting privacy budgets and setting a threshold to eliminate the too noisy intermediate results, the negative effects of noise addition have been reduced and the convergence behavior of ADMM has maintained.

[1]Note that tuning hyperparameters may not be private. In the future work, we can consider differentially private hyperparameter tuning algorithms proposed in [21], [22] to achieve end-to-end differential privacy.



(a) Different values of time period K_p. (b) Different values of threshold U (*Periodic Linear Decay*). (c) Different values of decay rate R_P.

Fig. 2. Hyperparameters of PR-ADMM (*Periodic Linear Decay*).



(a) Different values of threshold U (*Iteration-Based Decay*). (b) Different values of decay rate R_T.

Fig. 3. Hyperparameters of PR-ADMM (*Iteration-Based Decay*).

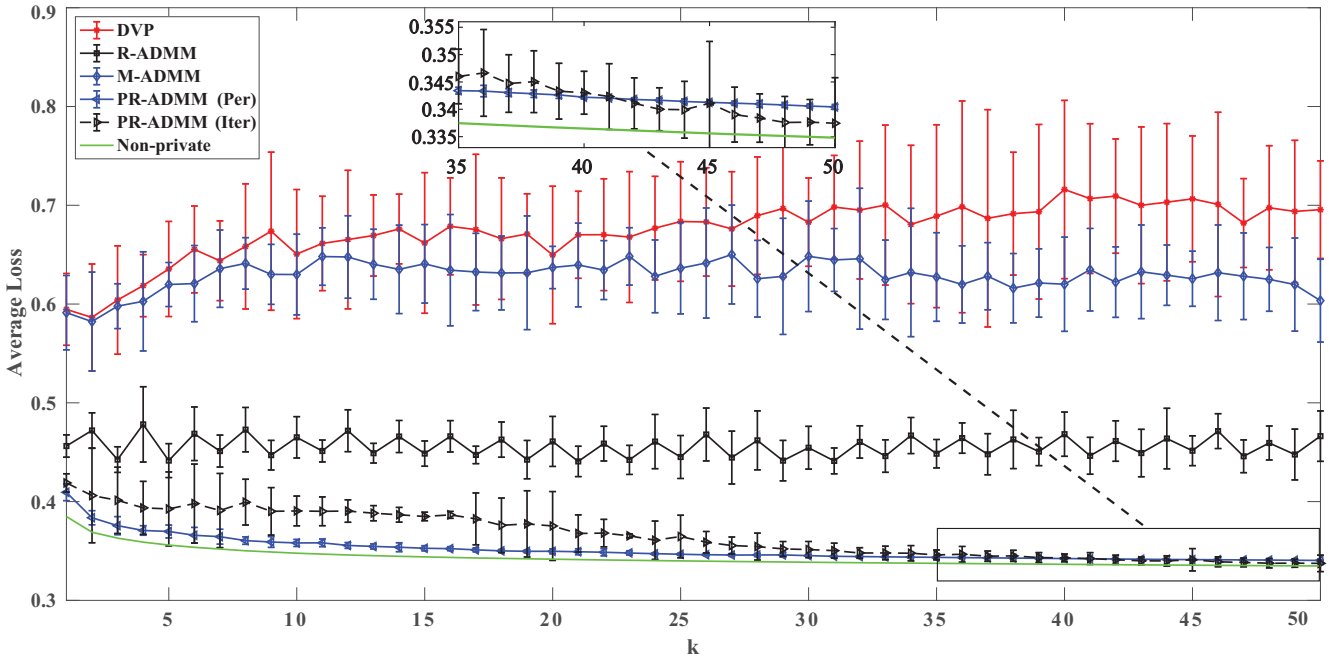


Fig. 4. Compare convergence: Total privacy loss $\epsilon = 10$.

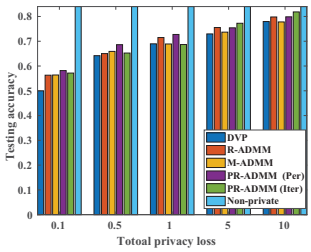


Fig. 5. Compare testing accuracy by varying total privacy loss ϵ.

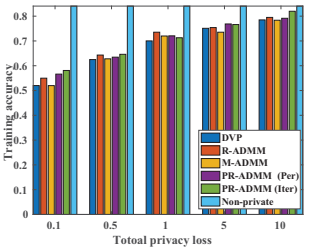


Fig. 6. Compare training accuracy by varying total privacy loss ϵ.

Figure 5 and Figure 6 illustrate the testing and training accuracy achieved by each algorithm changes as the value of ϵ increases. We can see that both PR-ADMM (Per) and PR-ADMM (Iter) achieve the competitive testing/training accuracies on a wide range of values for total privacy loss ϵ.

VI. Conclusion

In this paper, we have proposed a differentially private robust ADMM algorithm (PR-ADMM) by introducing Gaussian noise with decay variance to provide dynamic zCDP. By employing two noise variance decay schemes and setting a threshold to examine whether the primal variables from neighbors are too noisy, the negative effects of noise have been reduced. We have analyzed the convergence properties of the proposed algorithm for general convex optimization objectives. By performing extensive simulations, we have demonstrated that the proposed algorithm outperforms other differentially private ADMM algorithms.

VII. Acknowledgement

The work of J. Ding, X. Zhang, and M. Pan was supported in part by the U.S. National Science Foundation under grants US CNS-1350230 (CAREER), CNS-1646607, CNS-1702850, and CNS-1801925. The work of M. Chen was supported in part by the National Natural Science Foundation of China (61872147).

References

[1] Cisco. (2016) Cisco visual networking index: Global mobile data traffic forecast update 2015-2020. [Online]. Available: https://www.cisco.com/c/dam/m/en_in/innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf

[2] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 67, May 2016.

[3] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, January 2009.

[4] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, December 2010.

[5] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, October 2010.

[6] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links – Part I: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, January 2008.

[7] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *IEEE Conference on Decision and Control (CDC)*, Maui, HI, December 2012.

[8] T. Zhang and Q. Zhu, "A dual perturbation approach for differential private admm-based distributed empirical risk minimization," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, Vienna, Austria, October 2016.

[9] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, Sweden, July 2018.

[10] X. Zhang, M. M. Khalili, and M. Liu, "Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms," in *56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Urbana, IL, October 2018.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, March 2006, pp. 265–284.

[12] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology - EUROCRYPT 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, May 2006, pp. 486–503.

[13] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "Dp-admm: Admm-based distributed learning with differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1002–1012, July 2019.

[14] J. Ding, S. M. Errapotu, H. Zhang, M. Pan, and Z. Han, "Stochastic admm based distributed machine learning with differential privacy," in *International conference on security and privacy in communication systems*, Orlando, FL, October 2019.

[15] J. Ding, Y. Gong, C. Zhang, M. Pan, and Z. Han, "Optimal differentially private ADMM for distributed machine learning," *CoRR*, vol. abs/1901.02094, February 2019. [Online]. Available: https://arxiv.org/abs/1901.02094

[16] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[17] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, April 2014.

[18] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 635–658.

[19] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed ADMM over networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082–5095, October 2017.

[20] Q. Li, B. Kailkhura, R. Goldhahn, P. Ray, and P. K. Varshney, "Robust decentralized learning using ADMM with unreliable agents," *CoRR*, vol. abs/1710.05241, 2018. [Online]. Available: http://arxiv.org/abs/1710.05241

[21] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, October 2016.

[22] K. Chaudhuri and S. A. Vinterbo, "A stability-based validation procedure for differentially private machine learning," in *Advances in Neural Information Processing Systems*, Lake Tahoe, NV, December 2013.