Differentially Private ADMM for Convex Distributed Learning: Improved Accuracy via Multi-Step Approximation

Zonghao Huang
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK 74075, USA

Yanmin Gong
Department of Electrical and Computer Engineering
University of Texas at San Antonio
San Antonio, TX 78249, USA

Editor: Zonghao Huang

Abstract

Alternating Direction Method of Multipliers (ADMM) is a popular algorithm for distributed learning, where a network of nodes collaboratively solve a regularized empirical risk minimization by iterative local computation associated with distributed data and iterate exchanges. When the training data is sensitive, the exchanged iterates will cause serious privacy concern. In this paper, we aim to propose a new differentially private distributed ADMM algorithm with improved accuracy for a wide range of convex learning problems. In our proposed algorithm, we adopt the approximation of the objective function in the local computation to introduce calibrated noise into iterate updates robustly, and allow multiple primal variable updates per node in each iteration. Our theoretical results demonstrate that our approach can obtain higher utility by such multiple approximate updates, and achieve the error bounds asymptotic to the state-of-art ones for differentially private empirical risk minimization.

1. Introduction

The advances in machine learning are due to the abundance of data, which could be collected over network but cannot be handled by a single processor. This motivates distributed learning, where data is distributed and possessed by multiple nodes. In distributed learning frameworks, a network of nodes collaboratively solve an optimization problem which is usually formulated as a regularized empirical risk minimization associated with the distributed data. Distributed learning has been widely applied in a variety of areas such as vehicle networks (Han et al., 2017) and wireless sensor networks (Predd et al., 2006; Gong et al., 2016).

There exist approaches for distributed optimization including distributed subgradient descent algorithms (Nedic et al., 2008; Nedic and Ozdaglar, 2009; Lobel and Ozdaglar, 2010), dual averaging methods (Duchi et al., 2011; Tsianos et al., 2012), and Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011; Ling and Ribeiro, 2014; Shi et al., 2014; Zhang and Kwok, 2014). Among these algorithms, ADMM demonstrates fast convergence by both numerical and theoretical results in many applications. Prior works (Shi et al., 2014; Makhdoumi and Ozdaglar, 2017) have proved that in distributed ADMM, the iterates can converge linearly to the optimal solution while the objective value with feasibility violation can converge to the optimum at a rate of $O(1/t)$, where t is the number of iterations. In this paper, we mainly focus on ADMM-based distributed learning.

In ADMM-based distributed learning, nodes collaboratively solve the regularized empirical risk minimization by iterative local computation and iterate exchanges. The local computation requires each node to solve a local minimization associated with its local dataset while iterate exchanges require nodes to share the updated iterates with their neighbours. When the training data is sensitive, the exchanged learning statistics would cause serious privacy concern (Fredrikson et al., 2015; Shokri et al., 2017). Therefore, additional privacy-preserving methods are required to control privacy leakage. In this paper, we consider a state-of-art privacy standard, differential privacy (Dwork et al., 2014, 2006b,a). In ADMM-based distributed learning, differential privacy can be guaranteed by introducing calibrated noise into iterate updates. Recently, there are a few of works (Zhang et al., 2019; Zhang and Zhu, 2017; Zhang et al., 2018; Huang et al., 2019) focusing on designing differentially private ADMM-based distributed algorithm. Zhang and Zhu (Zhang and Zhu, 2017) propose primal variable perturbation and dual variable perturbation to achieve dynamic differential privacy in ADMM. Zhang et al. (Zhang et al., 2018) consider adaptive penalty parameters and propose to perturb the penalty. Huang et al. (Huang et al., 2019) propose DP-ADMM by adopting first-order approximation with time-varying Gaussian noise addition, and theoretically demonstrate that their approach can converge at a rate of $O(1/\sqrt{t})$, where t is the number of iterations. Zhang et al. (Zhang et al., 2019) propose Recycled ADMM where the information from odd iterations can be re-utilized in even iterations to save privacy budget. Hu et al. (Hu et al., 2019) propose an ADMM-based algorithm with differential privacy in a distributed data feature setting. However, it is still a challenge to design differentially private ADMM-based distributed algorithms with good privacy-utility trade-off, which has motivated our work.

In this paper, we propose an improved differentially private distributed ADMM algorithm. The key algorithmic features of our approach are to adopt the approximation of the objective function in iterate updates, and to allow multiple primal variable updates per node in each iteration. Our approach adopts the approximation in order to combine the calibrated noise ensuring differential privacy and the ADMM-based learning process robustly, and allows multiple approximate updates per node in each iteration to improve the privacy-utility trade-off. We demonstrate that by adding calibrated noise our approach can achieve differential privacy, and we analyze the utility of our proposed algorithm by the excess empirical risk with feasibility violation. Our theoretical results demonstrate that our approach can obtain higher utility by such multiple iterate updates with approximation, and achieve the error bounds asymptotic to the state-of-art ones for differentially private empirical risk minimization.

The main contributions of this paper are summarized as follows:

1. We propose a new differentially private ADMM-based distributed learning algorithm, where multiple approximate iterate updates with calibrated noise are performed per

node in each ADMM iteration to achieve differential privacy and improve the privacy-utility trade-off.

2. We analyze the utility of our approach theoretically by the excess empirical risk with feasibility violation. Our theoretical results show that our approach can obtain higher accuracy by multiple approximate iterate updates per node in each iteration and achieve the error bounds asymptotic to the state-of-art ones for differentially private empirical risk minimization.
3. We conduct numerical experiments based on real-world datasets to show the improved privacy-utility trade-off of our approach by comparing with previous works.

In the remainder of this paper is organized as follows. In Section 2, we introduce our problem statement. In Section 3, we introduce our proposed algorithm and provide the privacy analysis. In Section 4, we give the theoretical utility analysis of our approach. In Section 5, we show our numerical results. In Section 6 and Section 7, we introduce the related work and conclude our work.

2. Problem Statement

In this section, we first introduce our problem setting. Then, we describe the ADMM-based distributed learning algorithm, and discuss the associated privacy concern.

2.1 Problem Setting

We consider a connected network given by a undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, which consists of a set of nodes $\mathcal{V} = \{1, 2, \ldots, n\}$, and a set of edges $\mathcal{E}$. In this connected network, each node can only exchange information with its connected neighbours, and we use $\mathcal{N}_i$ to denote the neighbour set of node i. Each node $i \in \mathcal{V}$ possesses a private training dataset with size of m_i: $\big\{(\boldsymbol{a}_{i,j}, b_{i,j}), j \in \mathcal{D}_i\big\}$, where $\boldsymbol{a}_{i,j} \in \mathbb{R}^d$ represents the data feature vector of the j-th training sample belonging to i, and $b_{i,j} \in \{+1, -1\}$ is the corresponding data label.

The goal of our problem is to train a supervised learning model on the aggregated dataset $\{\mathcal{D}_i\}_{i\in\mathcal{V}}$, which enables predicting a label for any new data feature vector. The learning objective can be formulated as the following regularized empirical risk minimization problem:

$$\min_{\boldsymbol{w}} \quad \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{D}_i} \frac{1}{m_i} \ell(\boldsymbol{a}_{i,j}, b_{i,j}, \boldsymbol{w}) + \lambda\phi(\boldsymbol{w}), \tag{1}$$

where $\boldsymbol{w} \in \mathcal{W} \subseteq \mathbb{R}^d$ is the trained machine learning model, $\ell(\cdot)$ is the loss function used to measure the quality of the trained model, e.g., the loss function $\ell(\boldsymbol{a}, b, \boldsymbol{w})$ can be defined by $\log\big(1 + \exp(-b\boldsymbol{w}^\top \boldsymbol{a})\big)$ when we consider logistic regression, $\phi(\cdot)$ refers to the regularizer function introduced to prevent overfitting, and $\lambda > 0$ is the regularizer parameter controlling the impact of regularizer.

In this paper, we assume that the loss function $\ell(\cdot)$ and the regularizer function $\phi(\cdot)$ are both convex and Lipschitz. We use $\nabla\ell(\cdot)$ and $\nabla\phi(\cdot)$ to denote their gradient if they are differentiable or subgradient if not differentiable. We use $\|\cdot\|$ to denote the Euclidean norm.

For completeness, we also give the following additional definitions related to convex functions used in this paper:

Definition 1 (Convex Function) *A function $f(\cdot)$: $\mathcal{C} \to \mathbb{R}$ is convex if, for all pairs $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$, we have:*

$$f(\boldsymbol{x}) \leq f(\boldsymbol{y}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{y} \rangle. \tag{2}$$

Definition 2 (Lipschitz Function) *A function $f(\cdot)$: $\mathcal{C} \to \mathbb{R}$ is c-Lipschitz if, for all pairs $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$, we have:*

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq c\|\boldsymbol{x} - \boldsymbol{y}\|. \tag{3}$$

2.2 ADMM-Based Distributed Learning Algorithm

To solve problem (1) with ADMM in distributed manner, we need to reformulate it as:

$$\min_{\{\boldsymbol{w}_i\}} \quad \sum_{i \in \mathcal{V}} \Bigg(\sum_{j \in \mathcal{D}_i} \frac{1}{m_i} \ell(\boldsymbol{a}_{i,j}, b_{i,j}, \boldsymbol{w}_i) + \frac{\lambda}{n} \phi(\boldsymbol{w}_i) \Bigg), \tag{4a}$$

$$\text{s.t.} \quad \boldsymbol{w}_i = \boldsymbol{z}_{i,j}, \boldsymbol{z}_{i,j} = \boldsymbol{w}_j, \forall i \in \mathcal{V}, \forall j \in \mathcal{N}_i \tag{4b}$$

where $\boldsymbol{w}_i \in \mathcal{W} \subseteq \mathbb{R}^d$ is the local model solved by node i, and $\boldsymbol{z}_{i,j} \in \mathbb{R}^d$ is an auxiliary variable imposing the consensus constraint on neighboring nodes. The objective function (4a) is decoupled and constraints (4b) enforce that all the local models reach consensus finally.

Let $\{\boldsymbol{w}_i\}$, $\{\boldsymbol{z}_{i,j}\}$, and $\{\boldsymbol{\gamma}^a_{i,j}\}$ be the shorthand for $\{\boldsymbol{w}_i\}_{i \in \mathcal{V}}$, $\{\boldsymbol{z}_{i,j}\}_{i \in \mathcal{V}, j \in \mathcal{N}_i}$, and $\{\boldsymbol{\gamma}^a_{i,j}\}_{i \in \mathcal{V}, j \in \mathcal{N}_i, a=1,2}$, respectively. We define:

$$L_{\mathcal{D}_i}(\boldsymbol{w}_i) = \sum_{j \in \mathcal{D}_i} \frac{1}{m_i} \ell(\boldsymbol{a}_{i,j}, b_{i,j}, \boldsymbol{w}_i) + \frac{\lambda}{n} \phi(\boldsymbol{w}_i). \tag{5}$$

The augmented Lagrangian function associated with the problem (4) is:

$$\begin{aligned} \mathcal{L}\big(\{\boldsymbol{w}_i\}, \{\boldsymbol{z}_{i,j}\}, \{\boldsymbol{\gamma}^a_{i,j}\}\big) = \sum_{i \in \mathcal{V}} \Bigg(& L_{\mathcal{D}_i}(\boldsymbol{w}_i) - \sum_{j \in \mathcal{N}_i} \big\langle \boldsymbol{\gamma}^1_{i,j}, \boldsymbol{w}_i - \boldsymbol{z}_{i,j} \big\rangle \\ & - \sum_{j \in \mathcal{N}_i} \big\langle \boldsymbol{\gamma}^2_{i,j}, \boldsymbol{z}_{i,j} - \boldsymbol{w}_j \big\rangle + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \big(\|\boldsymbol{w}_i - \boldsymbol{z}_{i,j}\|^2 + \|\boldsymbol{w}_j - \boldsymbol{z}_{i,j}\|^2 \big) \Bigg), \end{aligned} \tag{6}$$

where $\{\boldsymbol{\gamma}^a_{i,j}\}$ are the dual variables associated with constraints (4b) and $\rho > 0$ is the penalty parameter. The ADMM solves the problem (4) in a Gauss-Seidel manner by minimizing (6) w.r.t. $\{\boldsymbol{w}_i\}$ and $\{\boldsymbol{z}_{i,j}\}$ alternatively followed by dual updates of $\{\boldsymbol{\gamma}^a_{i,j}\}$:

$$\boldsymbol{w}^k_i = \operatorname*{argmin}_{\boldsymbol{w}_i} \mathcal{L}\big(\{\boldsymbol{w}_i\}, \{\boldsymbol{z}^{k-1}_{i,j}\}, \{\boldsymbol{\gamma}^{a,k-1}_{i,j}\}\big), \tag{7a}$$

$$\boldsymbol{z}^k_{i,j} = \operatorname*{argmin}_{\boldsymbol{z}_{i,j}} \mathcal{L}\big(\{\boldsymbol{w}^k_i\}, \{\boldsymbol{z}_{i,j}\}, \{\boldsymbol{\gamma}^{a,k-1}_{i,j}\}\big), \tag{7b}$$

$$\boldsymbol{\gamma}^{1,k}_{i,j} = \boldsymbol{\gamma}^{1,k-1}_{i,j} - \rho(\boldsymbol{w}^k_i - \boldsymbol{z}^k_{i,j}), \tag{7c}$$

$$\boldsymbol{\gamma}^{2,k}_{i,j} = \boldsymbol{\gamma}^{2,k-1}_{i,j} - \rho(\boldsymbol{z}^k_{i,j} - \boldsymbol{w}^k_j). \tag{7d}$$

According to the previous works (Forero et al., 2010), the above iterate updates could be simplified by initializing $\boldsymbol{\gamma}_{i,j}^{1,0} = \boldsymbol{\gamma}_{i,j}^{2,0} = \mathbf{0}$, which can enforce $\boldsymbol{\gamma}_{i,j}^{1,k} = \boldsymbol{\gamma}_{i,j}^{2,k}$ and $\boldsymbol{z}_{i,j}^{k} = \frac{1}{2}(\boldsymbol{w}_i^k + \boldsymbol{w}_j^k)$. Let $\boldsymbol{\gamma}_i^k = \sum_{j\in\mathcal{N}_i} \boldsymbol{\gamma}_{i,j}^{1,k} = \sum_{j\in\mathcal{N}_i} \boldsymbol{\gamma}_{i,j}^{2,k}$, let $\{\boldsymbol{w}_j^k\}$ be the shorthand of $\{\boldsymbol{w}_j^k\}_{j\in\mathcal{N}_i}$, and define $\mathcal{L}_i^s\big(\boldsymbol{w}_i, \boldsymbol{w}_i^k, \{\boldsymbol{w}_j^k\}, \boldsymbol{\gamma}_i^k\big)$ as:

$$\mathcal{L}_i^s\big(\boldsymbol{w}_i, \boldsymbol{w}_i^k, \{\boldsymbol{w}_j^k\}, \boldsymbol{\gamma}_i^k\big) = L_{\mathcal{D}_i}(\boldsymbol{w}_i) - 2\big\langle \boldsymbol{\gamma}_i^k, \boldsymbol{w}_i \big\rangle + \rho \sum_{j\in\mathcal{N}_i} \|\boldsymbol{w}_i - \frac{1}{2}(\boldsymbol{w}_i^k + \boldsymbol{w}_j^k)\|^2. \tag{8}$$

Iterate updates (7) can be simplified as:

$$\boldsymbol{w}_i^k = \operatorname*{argmin}_{\boldsymbol{w}_i} \mathcal{L}_i^s\big(\boldsymbol{w}_i, \boldsymbol{w}_i^{k-1}, \{\boldsymbol{w}_j^{k-1}\}, \boldsymbol{\gamma}_i^{k-1}\big), \tag{9a}$$

$$\boldsymbol{\gamma}_i^k = \boldsymbol{\gamma}_i^{k-1} - \frac{\rho}{2} \sum_{j\in\mathcal{N}_i} \big(\boldsymbol{w}_i^k - \boldsymbol{w}_j^k\big), \tag{9b}$$

where Eq. (9a) is regarded as the primal varibale update while Eq. (9b) is known as the dual variable update.

2.3 Privacy Concern

In ADMM, iterates are updated by solving a minimization associated with the local dataset (Eq. (9a)), and they are needed to shared with neighbours. According to the previous works (Fredrikson et al., 2015; Shokri et al., 2017), adversary can infer the data information from the released learning statistics. If the local training data is sensitive, the shared iterates would cause privacy leakage.

The main goal of this paper is to provide privacy protection in ADMM against inference attacks from an adversary, who tries to infer sensitive information about the nodes' private datasets from the shared messages.

In order to provide privacy guarantee against such attacks, we define our privacy model formally by the notion of differential privacy (Dwork et al., 2006b, 2014). Specifically, we adopt the (ϵ, δ)-differential privacy defined as follows:

Definition 3 ((ϵ, δ)-Differential Privacy) *A randomized mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if for any two neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in only one tuple, and for any output subset $\mathcal{O} \subseteq$ range($\mathcal{M}$):*

$$\Pr\big[\mathcal{M}(\mathcal{D}) \in \mathcal{O}\big] \le e^{\epsilon} \cdot \Pr\big[\mathcal{M}(\mathcal{D}') \in \mathcal{O}\big] + \delta, \tag{10}$$

which means, with probability of at least $1-\delta$, the ratio of the probability distributions for two neighboring datasets is bounded by e^{ϵ}.

In Definition 3, δ and ϵ indicate the strength of privacy protection from the mechanism (a smaller ϵ or a smaller δ gives better privacy protection). Gaussian mechanism is a widely used randomization method used to guarantee (ϵ, δ)-differential privacy, where calibrated noise sampled from normal (Gaussian) distribution is added to the output.

When we consider a class of differetially private algorithms under t-fold adaptive composition where the auxiliary inputs of the k-th algorithm are the outputs of all previous algorithms, we use the following moments accoutant-based advanced composition theorem to analyze the privacy guarantee.

Theorem 4 (Advanced Composition) *Let $\epsilon, \delta \geq 0$. The class of (ϵ, δ)-differentially private algorithms satisfies*
$(\epsilon^{'}, \delta)$-differential privacy under t-fold adaptive composition, where $\epsilon^{'} = c_0\sqrt{t}\epsilon$ for some constant c_0.

Proof The proof of the advanced composition theorem is based on the moments accountant method proposed in (Abadi et al., 2016). In moments accountant method, the τ-th log moments of privacy loss from each (ϵ, δ)-differentially private algorithm can be given by $\frac{t\tau(\tau+1)\epsilon^2}{4\ln(1.25/\delta)}$. According to the linear composability of the log moments, we obtain the log moment of the total privacy loss from the class of private algorithms by $\frac{t\tau(\tau+1)\epsilon^2}{4\ln(1.25/\delta)}$. By using the tail bound property of the log moment, we can obtain the relationship between ϵ and $\epsilon^{'}$, which is $\epsilon^{'} \geq \sqrt{\frac{t\ln(1/\delta)}{\ln(1.25/\delta)}}\epsilon$. Thus, there exists a constant c_0 so that $\epsilon^{'} = c_0\sqrt{t}\epsilon$. Due to the limited space here, we suggest readers to refer to the previous works (Abadi et al., 2016; Huang et al., 2019) for the details. ■

3. Improved Differentially Private Distributed ADMM

3.1 Main Idea

As discussed in the last section, the privacy concern in ADMM-based distributed learning comes from the exchanged iterates. Calibrated noise is added to the iterates to control the privacy leakage and guarantee differential privacy. In order to introduce the calibrated noise into ADMM robustly, we adopt the approximation of the objective function when updating the primal variables. Such approximation is used in linearized ADMM (Ling and Ribeiro, 2014) to reduce computation cost, stochasitc ADMM (Ouyang et al., 2013), and some previous works on differentially private ADMM (Huang et al., 2019; Zhang et al., 2019). In addition, inspired by the federated learning framework proposed by (McMahan et al., 2016), our approach allows to perform multiple primal variable updates based on the approximate function per node in each iteration, in order to improve the accuracy.

3.2 Our Approach

Our proposed algorithm adopts the approximation of the objective when updating the primal variable, and allows performing l updates with calibrated noise per node in each iteration. Here we use $\tilde{\boldsymbol{w}}_i^{k,r}$ to denote the r-th noisy primal variable from node i in the k-th ADMM iteration.

In each iteration, each node does not perform the exact minimization to update the primal variable by (9a). Instead, each node performs the inexact minimization by adopting the approximation of the objective at $\tilde{\boldsymbol{w}}_i^{k,r}$:

$$L_{\mathcal{D}_i}(\boldsymbol{w}_i) \approx L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \langle \nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}), \tilde{\boldsymbol{w}}_i^{k,r} - \boldsymbol{w}_i \rangle + \frac{\eta_i^{k,r+1}}{2}\|\boldsymbol{w}_i - \tilde{\boldsymbol{w}}_i^{k,r}\|^2, \tag{11}$$

where $\eta_i^{k,r+1}$ is an approximation parameter to control the distance between the updated variable and the previous one. By approximation in Eq. (8), we define $\hat{\mathcal{L}}_i^s(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{k,r}, \tilde{\boldsymbol{w}}_i^{k-1}, \{\tilde{\boldsymbol{w}}_j^{k-1}\}, \boldsymbol{\gamma}_i^{k-1})$

by:

$$\begin{aligned}\hat{\mathcal{L}}_i^s\big(\boldsymbol{w}_i,\tilde{\boldsymbol{w}}_i^{k,r}, \tilde{\boldsymbol{w}}_i^{k-1}, \{\tilde{\boldsymbol{w}}_j^{k-1}\}, \boldsymbol{\gamma}_i^{k-1}\big) &= L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \langle \nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}), \tilde{\boldsymbol{w}}_i^{k,r} - \boldsymbol{w}_i\rangle \\ &+ \frac{\eta_i^{k,r+1}}{2}\|\boldsymbol{w}_i - \tilde{\boldsymbol{w}}_i^{k,r}\|^2 - 2\big\langle\boldsymbol{\gamma}_i^{k-1}, \boldsymbol{w}_i\big\rangle + \rho\sum_{j\in\mathcal{N}_i}\Big\|\boldsymbol{w}_i - \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1}+\tilde{\boldsymbol{w}}_j^{k-1})\Big\|^2.\end{aligned} \tag{12}$$

Our approach allows l primal variable updates per node in each iteration based on the approximate function. Thus, step (9a) is replaced by an inner l-iterative process:

$$\boldsymbol{w}_i^{k,r+1} = \min_{\boldsymbol{w}_i} \hat{\mathcal{L}}_i^s\big(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{k,r}, \tilde{\boldsymbol{w}}_i^{k-1}, \{\tilde{\boldsymbol{w}}_j^{k-1}\}, \boldsymbol{\gamma}_i^{k-1}\big), \tag{13a}$$

$$\tilde{\boldsymbol{w}}^{k,r+1} = \boldsymbol{w}_i^{k,r+1} + \boldsymbol{\xi}_i^{k,r+1}, \tag{13b}$$

where $\boldsymbol{\xi}_i^{k,r+1}$ is the sampled noise from normal (Gaussian) distribution to ensure differential privacy:

$$\boldsymbol{\xi}_i^{k,r+1} \sim \mathcal{N}(0, s_i^{k,r+1^2}\sigma^2\mathrm{I}_d). \tag{14}$$

After the l-iterative primal variable updates, the dual variable update follows as:

$$\boldsymbol{\gamma}_i^k = \boldsymbol{\gamma}_i^{k-1} - \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\big(\tilde{\boldsymbol{w}}_i^k - \tilde{\boldsymbol{w}}_j^k\big), \tag{15}$$

where $\tilde{\boldsymbol{w}}_i^k = \frac{1}{l}\sum_{r=1}^{l}\tilde{\boldsymbol{w}}_i^{k,r}$.

The details of our approach are given in Algorithm 1. Each node i firstly initializes its noisy primal variables $\tilde{\boldsymbol{w}}_i^{0,l}$ and $\tilde{\boldsymbol{w}}_i^0$, and dual variables $\boldsymbol{\gamma}_i^0$. Then each node i updates its noisy primal variables by an inner l-iterative process, where $\tilde{\boldsymbol{w}}_i^{k,r+1}$ is updated by (13a) and (13b) in each inner iteration. After l iterations of the inner process, node i obtain a noisy primal variable $\tilde{\boldsymbol{w}}_i^{k,l}$ and $\tilde{\boldsymbol{w}}_i^k$, and broadcast $\tilde{\boldsymbol{w}}_i^k$ to its neighbours $j \in \mathcal{N}_i$. After receiving the noisy primal variables $\{\tilde{\boldsymbol{w}}_j^k\}_{j\in\mathcal{N}_i}$ from its neighbours, node i continues to update its dual variable $\boldsymbol{\gamma}_i^k$ by (15). The iterative process will continue until reaching t iterations.

Note 1: In Algorithm 1, The inner iterative process leads to higher computation cost with a choice of larger l. In order to release the computation burden, we can use the stochastic variant of $\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r})$ when updating the primal variable in Step 7 by sampling a batch of data with replacement. This stochastic variant also leads to the same utility guarantee in Theorem 7.

Note 2: In Algorithm 1, each node shares the averaged result $\tilde{\boldsymbol{w}}_i^k$ instead of the latest one $\tilde{\boldsymbol{w}}_i^{k,l}$. And the dual variable is updated based on the shared parameters $\{\tilde{\boldsymbol{w}}_j^k\}_{j\in\mathcal{N}_i}$.

3.3 Privacy Analysis

In this section, we define the l_2 norm sensitivity $s_i^{k,r+1}$ and noise magnitude σ to achieve (ϵ, δ)-differential privacy in Algorithm 1.

Lemma 5 (L_2-Norm Sensitivity) *Assume that the loss function $\ell(\cdot)$ is c_1-Lipschitz. The l_2 norm sensitivity of the primal variable update function (Eq.* (13a)*) is given by:*

$$s_i^{k,r+1} = \frac{2c_1}{(2\rho|\mathcal{N}_i| + \eta_i^{k,r+1})m_i}. \tag{16}$$

Algorithm 1 Improved Differentially Private Distributed ADMM

```
Initialize $\{\tilde{\boldsymbol{w}}_i^{0,l}\}_{i\in\mathcal{V}}$, $\{\tilde{\boldsymbol{w}}_i^{0}\}_{i\in\mathcal{V}}$ and $\{\boldsymbol{\gamma}_i^{0}\}_{i\in\mathcal{V}}$;
for $k = 1, 2, \ldots, t$ do
  for $i \in \mathcal{V}$ do
    Let $\tilde{\boldsymbol{w}}_i^{k,0} = \tilde{\boldsymbol{w}}_i^{k-1,l}$.
    for $r = 0, 1, \ldots, l-1$ do
      Sample $\boldsymbol{\xi}_i^{k,r+1} \sim \mathcal{N}(0, {s_i^{k,r+1}}^2 \sigma^2 \mathrm{I}_d)$;
      Compute $\boldsymbol{w}_i^{k,r+1}$ by Eq. (13a);
      Compute $\tilde{\boldsymbol{w}}_i^{k,r+1}$ by Eq. (13b);
    end for
    Compute $\tilde{\boldsymbol{w}}_i^{k} = \frac{1}{l}\sum_{r=1}^{l} \tilde{\boldsymbol{w}}_i^{k,r}$.
  end for
  for $i \in \mathcal{V}$ do
    Broadcast $\tilde{\boldsymbol{w}}_i^{k}$ to all neighbours $j \in \mathcal{N}_i$;
  end for
  for $i \in \mathcal{V}$ do
    Compute $\boldsymbol{\gamma}_i^{k}$ by Eq. (15).
  end for
end for
```

Proof The l_2 norm sensitivity of the primal variable update function (Eq. (13a)) is defined by:

$$s_i^{k,r+1} = \max_{\mathcal{D}_i, \mathcal{D}_i'} \big\| \boldsymbol{w}_{i,\mathcal{D}_i}^{k,r+1} - \boldsymbol{w}_{i,\mathcal{D}_i'}^{k,r+1} \big\|. \tag{17}$$

According to Eq. (12) and Eq. (13a), we obtain a closed-form solution to $\boldsymbol{w}_i^{k,r+1}$:

$$\boldsymbol{w}_i^{k,r+1} = \big(- L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + 2\boldsymbol{\gamma}_i^{k-1} + \rho \sum_{j\in\mathcal{N}_i} \tilde{\boldsymbol{w}}_j^{k-1} + \rho|\mathcal{N}_i|\tilde{\boldsymbol{w}}_i^{k-1} + \eta_i^{k,r+1}\tilde{\boldsymbol{w}}_i^{k,r}\big)/(2\rho|\mathcal{N}_i| + \eta_i^{k,r}). \tag{18}$$

Then, we have:

$$\begin{aligned} \big\| \boldsymbol{w}_{i,\mathcal{D}_i}^{k,r+1} - \boldsymbol{w}_{i,\mathcal{D}_i'}^{k,r+1} \big\| =& \frac{\big\| \nabla\ell(\boldsymbol{a}_{i,j}, b_{i,j}, \tilde{\boldsymbol{w}}_i^{k,r}) - \nabla\ell(\boldsymbol{a}_{i,j}', b_{i,j}', \tilde{\boldsymbol{w}}_i^{k,r}) \big\|}{(2\rho|\mathcal{N}_i| + \eta_i^{k,r+1})m_i} \\ \le& \frac{2\big\|\nabla\ell(\cdot)\big\|}{(2\rho|\mathcal{N}_i| + \eta_i^{k,r+1})m_i}. \end{aligned} \tag{19}$$

Since function $\ell(\cdot)$ is c_1-Lipschitz, we obtain the result: $s_i^{k,r} = \frac{2c_1}{(2\rho|\mathcal{N}_i|+\eta_i^{k,r})m_i}$. ■

Theorem 6 (Privacy Guarantee) *Let $\epsilon, \delta \ge 0$ be arbitrary. There exists constant c_0 so that Algorithm 1 achieves (ϵ, δ)-differential privacy if we set the noise magnitude σ in Gaussian distribution $\mathcal{N}(0, {s_i^{k,r+1}}^2\sigma^2\mathrm{I}_d)$ by:*

$$\sigma = \frac{c_0\sqrt{t \cdot l \cdot 2\ln(1.25/\delta)}}{\epsilon}. \tag{20}$$

Proof Due to the limited space, we only provide the proof sketch here. We first follow Theorem A.1. in (Dwork et al., 2014) to demonstrate that by setting $\sigma = \frac{c_0\sqrt{t\cdot l\cdot 2\ln(1.25/\delta)}}{\epsilon}$, each primal variable update function with Gaussian noise satisfies $(\frac{1}{c_0\sqrt{t\cdot l}}\epsilon, \delta)$-differential privacy. Since our approach includes a class of $t \times l$ differentially private gradient functions. By adopting the advanced composition (Theorem 4), we prove that Algorithm 1 achieves (ϵ, δ)-differential privacy. ■

4. Utility Analysis

In this section, we theoretically analyze the utility of Algorithm 1, which can be measured by the expected excess empirical risk with feasibility violation, namely:

$$\mathbb{E}\big[\sum_{i\in\mathcal{V}} L_{\mathcal{D}_i}(\hat{\boldsymbol{w}}_i) - L_{\mathcal{D}_i}(\boldsymbol{w}^*)\big] + \beta\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\|, \tag{21}$$

where $\{\hat{\boldsymbol{w}}_i\}_{i\in\mathcal{V}}$ and $\{\hat{\boldsymbol{w}}_i\}_{i\in\mathcal{V}}$ are the outputs of our proposed algorithm, and $\boldsymbol{w}^*$ is the true minimizer of problem (1). The excess empirical risk measures the accuracy of the trained model by our approach while the feasibility violation measures the differences of local models.

4.1 Main Results

We analyze the excess empirical risk of our approach under the assumption: the objective function is convex and Lipschitz.

Theorem 7 (Utility Analysis) *Assume the objective function $L(\cdot)$ is c_2-Lipschitz, the diameter of the $\mathcal{W}$ is bounded by D, namely $\sup_{\boldsymbol{w},\boldsymbol{w}'\in\mathcal{W}}\|\boldsymbol{w} - \boldsymbol{w}^{'}\| \leq D$, and the domain of dual variable is bounded, namely $\|\boldsymbol{\gamma}_{i,j}\| \leq \beta$ Let*

$$\hat{\boldsymbol{w}}_i = \frac{1}{t}\frac{1}{l}\sum_{k=1}^{t}\sum_{r=0}^{l-1}\tilde{\boldsymbol{w}}_i^{k,r}. \tag{22}$$

If we set the learning rate:

$$\eta_i^{k,r} = \frac{\sqrt{2kr}}{D}\sqrt{\frac{c_2^2}{n^2} + \frac{dc_0^2c_1^2tl8\ln(1.25/\delta)}{\epsilon^2m_i^2}}, \tag{23}$$

we have the following expected error bound:

$$\begin{aligned}&\mathbb{E}\Big[\sum_{i\in\mathcal{V}} L_{\mathcal{D}_i}(\hat{\boldsymbol{w}}_i) - L_{\mathcal{D}_i}(\boldsymbol{w}^*)\Big] + \beta\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\|\\ \leq&\sum_{i\in\mathcal{V}}\Big(\frac{\sqrt{2}D}{\sqrt{t\cdot l}}\big(\frac{c_2^2}{n^2} + \frac{dc_0^2c_1^2tl8\ln(1.25/\delta)}{\epsilon^2m_i^2}\big)^{\frac{1}{2}} + \frac{\rho|\mathcal{N}_i|D^2 + |\mathcal{N}_i|\beta^2/\rho}{t}\Big).\end{aligned} \tag{24}$$

Proof See the proof in Appendix. ■

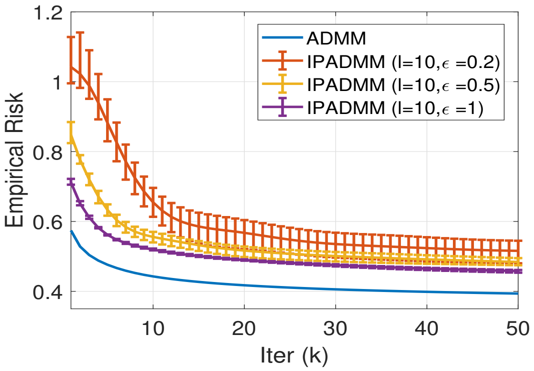


Figure 1: Utility and Privacy Trade-off.

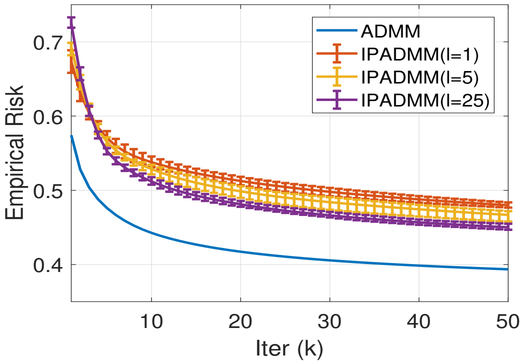


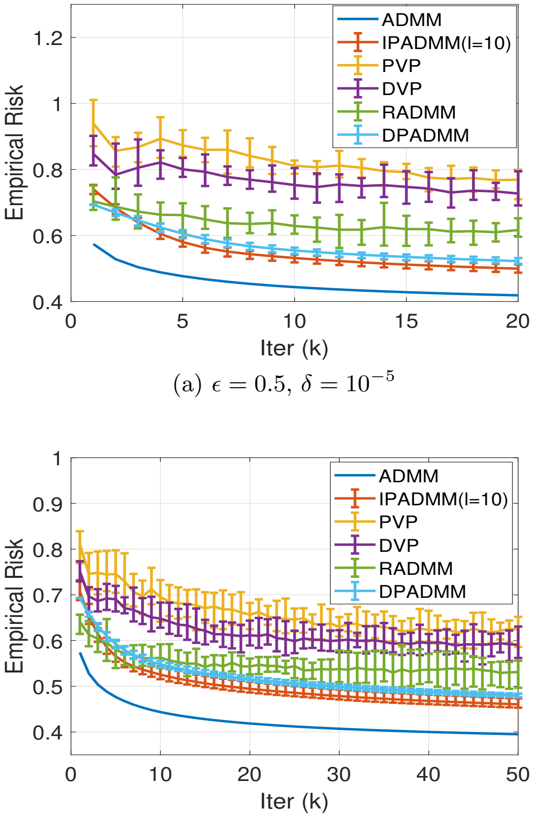
Figure 2: Impact of l on Accuracy.

4.2 Discussion

Our main result (Theorem 7) demonstrates the privacy-utility trade-off of our approach. When we want to ensure smaller privacy leakage from our output by setting a smaller ϵ, the utility of our approach will decrease. When ϵ is set to be extremely large leading to no privacy protection, our approach achieves a sublinear convergence rate of $O(1/t)$, which matches the result from the conventional ADMM method.

Theorem 7 demonstrates that our approach can be improved by allowing multiple approximate primal variable updates. When we set a larger l in our proposed algorithm, our approach can achieves better accuracy but this also leads to higher computation cost in each iteration.

From Theorem 7, if the iteration number t is sufficiently large enough our approach achieve the error bound $O\bigg(\sum_{i\in\mathcal{V}}\frac{\sqrt{d\ln(1/\delta)}}{m_i\epsilon}\bigg)$, which is comparable to the state-of-art error bound $O(\frac{\sqrt{d\ln(1/\delta)}}{N\epsilon})$ for differentially private empirical risk minimization under the assumption that the objective is Lischipz and convex, here N is the total number of training data.



(a) $\epsilon = 0.5$, $\delta = 10^{-5}$

(b) $\epsilon = 1$, $\delta = 10^{-5}$

Figure 3: Accuracy Comparison in Empirical Risk.

5. Numerical Results

In this section, we give the numerical results of our approach and compare our proposed algorithm (Algorithm 1) with algorithms proposed by prior works by the simulation in MATLAB.

5.1 Regularized Logistic Regression

We evaluate our approach by l_2 regularized logistic regression. Logistic regression is a widely used statistical model for classification, and its loss function is described as:

$$\ell(\boldsymbol{a}, b, \boldsymbol{w}) = \log\big(1 + \exp(-b\boldsymbol{w}^\top \boldsymbol{a})\big). \tag{25}$$

Thus by l_2 regularized logistic regression, the objective of our regularized empirical risk minimization problem can be formulated as:

$$L_{\mathcal{D}_i}(\boldsymbol{w}_i) = \sum_{j \in \mathcal{D}_i} \frac{1}{m_i} \log\big(1 + \exp(-b_{i,j}\boldsymbol{w}_i^\top \boldsymbol{a}_{i,j})\big) + \frac{\lambda}{2n}\|\boldsymbol{w}_i\|^2. \tag{26}$$

5.2 Dataset

The dataset used in our simulation is Adult dataset (Asuncion and Newman, 2007) from UCI Machine Learning Repository. Adult dataset includes $48,842$ instances, each of which has 14 personal attributes with a label representing whether the income is above $\$50,000$ or not. We follow the previous works (Zhang et al., 2018; Huang et al., 2019) to preprocess the data by removing all the instances with missing values, converting the categorical attributes into binary vectors, normalizing columns to guarantee the maximum value of each column is 1, normalizing rows to enforce their l_2 norm to be less than 1, and converting the labels $\{> 50k, < 50k\}$ into $\{+1, -1\}$. After the data preprocessing, we obtain $45,222$ data instances each with a 104-dimensional feature vector and a label belonging to $\{+1, -1\}$.

5.3 Baseline Algorithms

We compare our approach: Improved differentially Private Distributed ADMM (IPADMM) with four baseline algorithms: (1) non-private distributed ADMM algorithm, (2) ADMM algorithm with PVP in (Zhang and Zhu, 2017), (3) ADMM with dual variable perturbation (DVP) in (Zhang and Zhu, 2017), (4) Recycled ADMM (RADMM) proposed in (Zhang et al., 2019), and (5) DPADMM proposed in (Huang et al., 2019).

5.4 Simulation Results

We mainly evaluate our approacch on the utility-privacy trade-off and the effect of the choice of l on utility, and compare our approach with the baseline algorithms. In our simulation, we set ρ to be 0.001 and μ to be 0.0001, and by data preprocessing, we can enforce the objective function to be $(n + \frac{\mu D}{n})$-Lischiptz, where n is the number of nodes and D is the diameter of $\mathcal{W}$. Here, we consider a network consisting of 100 nodes fully connected and with evenly divided data. Our numerical results are the averaged results from 10 simulations.

Figure 1 shows the utility-privacy trade-off of our approach. Here we fix δ to be 10^{-5} and l to be 10, and only change ϵ. With increasing ϵ indicating weaker privacy guarantee, our approach has less empirical risk achieving better utility, which is consistent with Theorem 7.

Figure 2 demonstrates the impact of the choice of l on the utility of our approach. In this simulation, we fix ϵ to be 1 and change l from 1 to 25. When we set a larger l, the accuracy of our algorithm can be improved, which is consistent with Theorem 7.

Figure 3 compares our approach with the four baseline algorithms on empirical risk when we set the privacy parameter ϵ to be 0.5 and 1, and δ to be 0.00001. The results show that our approach has more stable update processing and has better utility than the other differentially private ADMM algorithms.

6. Related Work

6.1 Distributed ADMM

ADMM demonstrates fast convergence in many applications and it is widely used to solve distributed optimizations. Shi et al. (Shi et al., 2014) focus the theoretical aspects on the convergence rate of distributed ADMM, and demonstrate that the iterates from distributed

ADMM can converge to the optimal solution linearly under the assumptions that the objective function is strongly convex and Lipschitz smooth. Zhang and Kwok (Zhang and Kwok, 2014) propose an asynchronous distributed ADMM by using partial barrier and bounded delay. Ling et al. (Ling et al., 2015) design a linearized distributed ADMM where the augmented Lagrangian function is replaced by its first-order approximation to reduce the local computation cost. Song et al. (Song et al., 2016) show that distributed ADMM can converge faster by adaptively choosing the penalty parameter. Makhdoumi and Ozdaglar (Makhdoumi and Ozdaglar, 2017) demonstrate that the objective value with feasibility violation converges to the optimum at a rate of $O(1/t)$ by distributed ADMM, where t is the number of iterations.

6.2 Differentially Private Empirical Risk Minimization

There have been tremendous research efforts on differentially private empirical risk minimization (Chaudhuri et al., 2011; Bassily et al., 2014; Wang et al., 2017; Thakurta and Smith, 2013). Chaudhuri et al. (Chaudhuri et al., 2011) propose two perturbation methods: output perturbation and objective perturbation to guarantee ϵ-differential privacy. Bassily et al. (Bassily et al., 2014) provide a systematic investigation of differentially private algorithms for convex empirical risk minimization and propose efficient algorithms with tighter error bound. Wang et al. (Wang et al., 2017) focus on a more general problem: non-convex problem, and propose a faster algorithm based on a proximal stochastic gradient method. Smith and Thakurta (Thakurta and Smith, 2013) explore the stability of model selection problems, and propose two differentially private algorithms based on perturbation stability and subsampling stability respectively.

6.3 Differentially Private ADMM-based Distributed Learning

Recently, there are some works focusing on differentially private ADMM-based distributed learning algorithms. Zhang and Zhu (Zhang and Zhu, 2017) propose two perturbation methods: primal perturbation and dual perturbation to guarantee dynamic differential privacy in ADMM-based distributed learning. Zhang et al. (Zhang et al., 2018) propose to perturb the penalty parameter of ADMM to guarantee differential privacy. Huang et al. (Huang et al., 2019) propose an algorithm named DP-ADMM, where an approximate augmented Lagrangian function with time-varying Gaussian noise addition is adopted to update iterates while guaranteeing differential privacy, and theoretically analyze the convergence rate of their approach. Zhang et al. (Zhang et al., 2019) propose recycled ADMM with differential privacy guarantee where the results from odd iterations could be re-utilized by the even iterations, and thus half of updates incur no privacy leakage. Hu et al. (Hu et al., 2019) consider a setting where data features are distributed, and use ADMM with primal variable perturbation for distributed learning while guaranteeing differential privacy. Comparing with previous works on differentially private distributed ADMM, we propose an improved algorithm by performing multiple iterate updates with approximation per node in each iteration. Our theoretical analysis demonstrates the improvement of our approach.

7. Conclusion

In this paper, we have proposed a new differentially private distributed ADMM algorithm for a class of convex learning problems. In our approach, we have adopted the approximation when updating the primal variables and have allowed each node to perform such primal variable updates with differentially private noise for l times in each iteration. We have analyzed the privacy guarantee of our proposed algorithm by properly setting the noise magnitude in Gaussian distribution and using the moments accountant method. We have theoretically analyzed the utility of our approach by the excess empirical risk with feasibility violation under the setting that the objective is Lipschitz and convex. Our theoretical results have shown that our approach can obtain higher accuracy if we set a larger l and can achieve the error bounds, which are comparable to the state-of-art error bounds for differentially private empirical risk minimization.

8. Appendix

8.1 Proof of Theorem 7

In this appendix, we will give the proof of Theorem 7. Firstly, by assuming that the diameter of dual variable domain is bounded, namely $\|\boldsymbol{\gamma}_{i,j}\| \leq \beta$, we have:

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{i\in\mathcal{D}} L_{\mathcal{D}_i}(\hat{\boldsymbol{w}}_i) - L_{\mathcal{D}_i}(\boldsymbol{w}^*)\Big] + \beta\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\| \\
=&\max_{\boldsymbol{\gamma}_{i,j}:\|\boldsymbol{\gamma}_{i,j}\|\leq\beta} \mathbb{E}\Big[\sum_{i\in\mathcal{D}}\Big(L_{\mathcal{D}_i}(\hat{\boldsymbol{w}}_i) - L_{\mathcal{D}_i}(\boldsymbol{w}^*) - \sum_{j\in\mathcal{N}_i}\langle\boldsymbol{\gamma}_{i,j}, \hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\rangle\Big)\Big].
\end{aligned} \tag{27}$$

Due to the convexity of $L_{\mathcal{D}}(\cdot)$ and the definition of $\hat{\boldsymbol{w}}_i$, we have:

$$\begin{aligned}
&L_{\mathcal{D}_i}(\hat{\boldsymbol{w}}_i) - L_{\mathcal{D}_i}(\boldsymbol{w}^*) - \sum_{j\in\mathcal{N}_i}\langle\boldsymbol{\gamma}_{i,j}, \hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\rangle \\
\leq&\frac{1}{t}\sum_{k=1}^{t}\frac{1}{l}\sum_{r=0}^{l-1}\big(L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) - L_{\mathcal{D}_i}(\boldsymbol{w}^*) - \sum_{j\in\mathcal{N}_i}\langle\boldsymbol{\gamma}_{i,j}, \tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_j^{k,r}\rangle\big) \\
\leq&\frac{1}{t}\sum_{k=1}^{t}\frac{1}{l}\sum_{r=0}^{l-1}\big(\langle\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}), \tilde{\boldsymbol{w}}_i^{k,r} - \boldsymbol{w}^*\rangle - \sum_{j\in\mathcal{N}_i}\langle\boldsymbol{\gamma}_{i,j}, \tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_j^{k,r}\rangle\big).
\end{aligned} \tag{28}$$

Next, we analyze $\big\langle\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}), \tilde{\boldsymbol{w}}_i^{k,r} - \boldsymbol{w}^*\big\rangle$:

$$\begin{aligned}
\big\langle\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}), \tilde{\boldsymbol{w}}_i^{k,r} - \boldsymbol{w}^*\big\rangle =& \big\langle\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \boldsymbol{\xi}_i, \tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^*\big\rangle \\
&+ \big\langle\boldsymbol{\xi}_i, \boldsymbol{w}^* - \boldsymbol{w}_i^{k,r+1}\big\rangle + \big\langle\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \boldsymbol{\xi}_i, \tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_i^{k,r+1}\big\rangle.
\end{aligned} \tag{29}$$

If we define: $\boldsymbol{\xi}_i = \boldsymbol{\xi}_i^{k,r}/(2\rho|\mathcal{N}_i| + \eta_i^{k,r})$, according to the primal variable update (13a) and (13b), we have:

$$\begin{aligned}
&\bigl\langle \nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \boldsymbol{\xi}_i, \tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^* \bigr\rangle \\
=&\bigl\langle \nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) - 2\boldsymbol{\gamma}_i^{k-1} + 2\rho \sum_{j\in\mathcal{N}_i} \bigl(\tilde{\boldsymbol{w}}_i^{k,r} - \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1})\bigr) + \boldsymbol{\xi}_i, \tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^* \bigr\rangle \\
&+ 2\bigl\langle \boldsymbol{\gamma}_i^{k-1} - \rho \sum_{j\in\mathcal{N}_i} \bigl(\tilde{\boldsymbol{w}}_i^{k,r} - \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1})\bigr), \tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^* \bigr\rangle \\
=&\bigl(\eta_i^{k,r+1} + 2\rho|\mathcal{N}_i|\bigr)\bigl\langle \tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_i^{k,r+1}, \tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^* \bigr\rangle \\
&+ 2\bigl\langle \boldsymbol{\gamma}_i^{k-1} - \rho \sum_{j\in\mathcal{N}_i} \bigl(\tilde{\boldsymbol{w}}_i^{k,r} - \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1})\bigr), \tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^* \bigr\rangle.
\end{aligned} \tag{30}$$

We handle the two terms separately:

$$\bigl\langle \tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_i^{k,r+1}, \tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^* \bigr\rangle = \frac{1}{2}\|\tilde{\boldsymbol{w}}_i^{k,r} - \boldsymbol{w}^*\|^2 - \frac{1}{2}\|\tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^*\|^2 - \frac{1}{2}\|\tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_i^{k,r+1}\|^2, \tag{31}$$

Based on the dual update (15) and the definition of $\tilde{\boldsymbol{w}}_i^k$, we have:

$$\begin{aligned}
&\frac{1}{l}\sum_{r=0}^{l-1} 2\bigl\langle \boldsymbol{\gamma}_i^{k-1} - \rho \sum_{j\in\mathcal{N}_i} \bigl(\tilde{\boldsymbol{w}}_i^{k,r} - \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1})\bigr), \tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^* \bigr\rangle \\
=&\frac{1}{l}\sum_{r=0}^{l-1} 2\rho \sum_{j\in\mathcal{N}_i} \bigl\langle \tilde{\boldsymbol{w}}_i^{k,r+1} - \tilde{\boldsymbol{w}}_i^{k}, \boldsymbol{w}^* - \tilde{\boldsymbol{w}}_i^{k,r+1} \bigr\rangle + 2\sum_{j\in\mathcal{N}_i} \bigl\langle \boldsymbol{\gamma}_{i,j}^{k}, \tilde{\boldsymbol{w}}_i^{k} - \boldsymbol{w}^* \bigr\rangle \\
&+ \frac{1}{l}\sum_{r=0}^{l-1} 2\rho \sum_{j\in\mathcal{N}_i} \bigl\langle \tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_i^{k,r+1}, \boldsymbol{w}^* - \tilde{\boldsymbol{w}}_i^{k,r+1} \bigr\rangle \\
&+ 2\rho \sum_{j\in\mathcal{N}_i} \bigl\langle \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k} + \tilde{\boldsymbol{w}}_j^{k}) - \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1}), \boldsymbol{w}^* - \tilde{\boldsymbol{w}}_i^{k} \bigr\rangle.
\end{aligned} \tag{32}$$

Since we have:

$$\sum_{r=0}^{l-1} \bigl\langle \tilde{\boldsymbol{w}}_i^{k,r+1} - \tilde{\boldsymbol{w}}_i^{k}, \boldsymbol{w}^* - \tilde{\boldsymbol{w}}_i^{k,r+1} \bigr\rangle = \frac{1}{l}\sum_{r=0}^{l-1}\sum_{a=0}^{l-1} \bigl\langle \tilde{\boldsymbol{w}}_i^{k,r+1} - \tilde{\boldsymbol{w}}_i^{k,a+1}, \boldsymbol{w}^* - \tilde{\boldsymbol{w}}_i^{k,r+1} \bigr\rangle < 0, \tag{33}$$

and

$$\begin{aligned}
&\bigl\langle \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k} + \tilde{\boldsymbol{w}}_j^{k}) - \frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1}), \boldsymbol{w}^* - \tilde{\boldsymbol{w}}_i^{k} \bigr\rangle \\
\leq&\frac{1}{2}\bigl(\|\frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1}) - \boldsymbol{w}^*\|^2 - \|\frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k} + \tilde{\boldsymbol{w}}_j^{k}) - \boldsymbol{w}^*\|^2 + \|\frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k} - \tilde{\boldsymbol{w}}_j^{k})\|^2\bigr) \\
=&\frac{1}{2}\bigl(\|\frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1}) - \boldsymbol{w}^*\|^2 - \|\frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k} + \tilde{\boldsymbol{w}}_j^{k}) - \boldsymbol{w}^*\|^2\bigr) + \frac{1}{2\rho^2}\|\boldsymbol{\gamma}_{i,j}^{k-1} - \boldsymbol{\gamma}_{i,j}^{k}\|^2,
\end{aligned} \tag{34}$$

and by Young's inequality:

$$\big\langle \nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \boldsymbol{\xi}_i, \tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_i^{k,r+1} \big\rangle \leq \frac{1}{2\eta_i^{k,r+1}} \|\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \boldsymbol{\xi}_i\|^2 + \frac{\eta_i^{k,r+1}}{2} \|\tilde{\boldsymbol{w}}_i^{k,r} - \tilde{\boldsymbol{w}}_i^{k,r+1}\|^2, \tag{35}$$

we can obtain:

$$\begin{aligned}
&\frac{1}{l}\sum_{r=0}^{l-1} \big\langle \nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}), \tilde{\boldsymbol{w}}_i^{k,r} - \boldsymbol{w}^* \big\rangle \\
\leq& \frac{1}{l}\sum_{r=0}^{l-1} \big(\frac{\eta_i^{k,r+1}}{2} \big(\|\tilde{\boldsymbol{w}}_i^{k,r} - \boldsymbol{w}^*\|^2 + \|\tilde{\boldsymbol{w}}_i^{k,r+1} - \boldsymbol{w}^*\|^2 \big) \\
&+ \frac{1}{2\eta_i^{k,r+1}} \|\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \boldsymbol{\xi}_i\|^2 \big) + \frac{1}{l}\sum_{r=0}^{l-1} \big\langle \boldsymbol{\xi}_i, \boldsymbol{w}^* - \boldsymbol{w}_i^{k,r+1} \big\rangle \\
&+ \rho \sum_{j\in\mathcal{N}_i} \big(\|\frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k-1} + \tilde{\boldsymbol{w}}_j^{k-1}) - \boldsymbol{w}^*\|^2 - \|\frac{1}{2}(\tilde{\boldsymbol{w}}_i^{k} + \tilde{\boldsymbol{w}}_j^{k}) - \boldsymbol{w}^*\|^2 \big) \\
&+ \sum_{j\in\mathcal{N}_i} \frac{1}{\rho} \|\boldsymbol{\gamma}_{i,j}^{k-1} - \boldsymbol{\gamma}_{i,j}^{k}\|^2 + \frac{1}{l}\sum_{r=0}^{l-1} \sum_{j\in\mathcal{N}_i} 2\big\langle \boldsymbol{\gamma}_{i,j}^{k}, \tilde{\boldsymbol{w}}_i^{k+1} - \boldsymbol{w}^* \big\rangle.
\end{aligned} \tag{36}$$

Next, we analyze $\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} \langle -\boldsymbol{\gamma}_{i,j}, \tilde{\boldsymbol{w}}_i^k - \tilde{\boldsymbol{w}}_j^k \rangle$:

$$\begin{aligned}
\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} \langle -\boldsymbol{\gamma}_{i,j}, \tilde{\boldsymbol{w}}_i^k - \tilde{\boldsymbol{w}}_j^k \rangle =& \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} \big(\langle -\boldsymbol{\gamma}_{i,j}^k, \tilde{\boldsymbol{w}}_i^k - \boldsymbol{w}^* \rangle \\
&+ \langle \boldsymbol{\gamma}_{i,j}^k, \tilde{\boldsymbol{w}}_j^k - \boldsymbol{w}^* \rangle + 2\langle \boldsymbol{\gamma}_{i,j}^k - \boldsymbol{\gamma}_{i,j}, \frac{1}{2}(\tilde{\boldsymbol{w}}_i^k - \tilde{\boldsymbol{w}}_j^k) \rangle \big) \\
=& \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i} \big(\langle -2\boldsymbol{\gamma}_{i,j}^k, \tilde{\boldsymbol{w}}_i^k - \boldsymbol{w}^* \rangle + 2\langle \boldsymbol{\gamma}_{i,j}^k - \boldsymbol{\gamma}_{i,j}, \frac{1}{2}(\tilde{\boldsymbol{w}}_i^k - \tilde{\boldsymbol{w}}_j^k) \rangle \big).
\end{aligned} \tag{37}$$

Furthermore, we have:

$$\begin{aligned}
\big\langle \boldsymbol{\gamma}_{i,j}^k - \boldsymbol{\gamma}_{i,j}, \frac{1}{2}(\tilde{\boldsymbol{w}}_i^k - \tilde{\boldsymbol{w}}_j^k) \big\rangle =& \frac{1}{\rho} \big\langle \boldsymbol{\gamma}_{i,j}^k - \boldsymbol{\gamma}_{i,j}, \boldsymbol{\gamma}_{i,j}^{k-1} - \boldsymbol{\gamma}_{i,j}^k \big\rangle \\
=& \frac{1}{2\rho} \big(\|\boldsymbol{\gamma}_{i,j}^{k-1} - \boldsymbol{\gamma}_{i,j}\|^2 - \|\boldsymbol{\gamma}_{i,j}^{k} - \boldsymbol{\gamma}_{i,j}\|^2 - \|\boldsymbol{\gamma}_{i,j}^{k-1} - \boldsymbol{\gamma}_{i,j}^k\|^2 \big).
\end{aligned} \tag{38}$$

Since we assume $L_{\mathcal{D}}(\cdot)$ is c_2-Lipschitz, we have:

$$\mathbb{E}\big[\|\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \boldsymbol{\xi}_i\|^2\big] = \frac{c_2^2}{n^2} + \frac{dc_0^2c_1^2tl8\ln(1.25/\delta)}{\epsilon^2 m_i^2}. \tag{39}$$

Since we have $\mathbb{E}\big[\langle \boldsymbol{\xi}_i, \boldsymbol{w}^* - \boldsymbol{w}_i^{k,r} \rangle\big] = 0$, by assuming that the diameter of the $\mathcal{W}$ is bounded by D, and let $\eta_i^{k,r} = \frac{\sqrt{2kr}}{D}\sqrt{\frac{c_2^2}{n^2} + \frac{dc_0^2c_1^2tl8\ln(1.25/\delta)}{\epsilon^2 m_i^2}}$, according to Eq. (27), Eq. (28), Eq. (36), Eq. (37), and Eq. (38), we can obtain:

$$
\begin{aligned}
&\mathbb{E}\Big[\sum_{i\in\mathcal{D}} L_{\mathcal{D}_i}(\hat{\boldsymbol{w}}_i) - L_{\mathcal{D}_i}(\boldsymbol{w}^*)\Big] + \beta \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{N}_i} \|\hat{\boldsymbol{w}}_i - \hat{\boldsymbol{w}}_j\| \\
\le& \sum_{i\in\mathcal{V}} \Bigg(\frac{1}{t}\frac{1}{l}\sum_{k=1}^{t}\sum_{r=0}^{l-1}\mathbb{E}\big[\frac{\|\nabla L_{\mathcal{D}_i}(\tilde{\boldsymbol{w}}_i^{k,r}) + \boldsymbol{\xi}_i\|^2}{2\eta_i^{k,r+1}}\big] + \frac{1}{t}\frac{1}{l}\frac{\eta_i^{t,l}}{2}D^2 \\
&\quad + \frac{1}{t}\rho|\mathcal{N}_i|D^2 + \frac{1}{t}\frac{|\mathcal{N}_i|}{\rho}\max_{\boldsymbol{\gamma}_{i,j}:\|\boldsymbol{\gamma}_{i,j}\|\le\beta}\|\boldsymbol{\gamma}_{i,j}^0 - \boldsymbol{\gamma}_{i,j}\|^2\Bigg) \\
\le& \sum_{i\in\mathcal{V}} \Bigg(\frac{\sqrt{2}D}{\sqrt{t\cdot l}}\big(\frac{c_2^2}{n^2} + \frac{dc_0^2c_1^2tl8\ln(1.25/\delta)}{\epsilon^2 m_i^2}\big)^{\frac{1}{2}} + \frac{\rho|\mathcal{N}_i|D^2 + |\mathcal{N}_i|\beta^2/\rho}{t}\Bigg).
\end{aligned}
\tag{40}
$$

References

Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2011.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006b.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Pedro A Forero, Alfonso Cano, and Georgios B Giannakis. Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, 11(May):1663–1707, 2010.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333. ACM, 2015.

Yanmin Gong, Yuguang Fang, and Yuanxiong Guo. Private data analytics on biomedical sensing data via distributed computation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(3):431–444, 2016.

Shuo Han, Ufuk Topcu, and George J Pappas. Differentially private distributed constrained optimization. *IEEE Transactions on Automatic Control*, 62(1):50–64, 2017.

Yaochen Hu, Peng Liu, Linglong Kong, and Di Niu. Learning privately over distributed features: An admm sharing approach. *arXiv preprint arXiv:1907.07735*, 2019.

Zonghao Huang, Rui Hu, Yuanxiong Guo, Eric Chan-Tin, and Yanmin Gong. Dp-admm: Admm-based distributed learning with differential privacy. *IEEE Transactions on Information Forensics and Security*, 2019.

Qing Ling and Alejandro Ribeiro. Decentralized linearized alternating direction method of multipliers. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5447–5451. IEEE, 2014.

Qing Ling, Wei Shi, Gang Wu, and Alejandro Ribeiro. Dlm: Decentralized linearized alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 63 (15):4051–4064, 2015.

Ilan Lobel and Asuman Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *IEEE Transactions on Automatic Control*, 56(6):1291–1306, 2010.

Ali Makhdoumi and Asuman Ozdaglar. Convergence rate of distributed admm over networks. *IEEE Transactions on Automatic Control*, 62(10):5082–5095, 2017.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48, 2009.

Angelia Nedic, Alex Olshevsky, Asuman Ozdaglar, and John N Tsitsiklis. Distributed subgradient methods and quantization effects. In *2008 47th IEEE Conference on Decision and Control*, pages 4177–4184. IEEE, 2008.

Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, pages 80–88, 2013.

Joel B Predd, Sanjeev B Kulkarni, and H Vincent Poor. Distributed learning in wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):56–69, 2006.

Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the admm in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 3–18. IEEE, 2017.

Changkyu Song, Sejong Yoon, and Vladimir Pavlovic. Fast admm algorithm for distributed optimization with adaptive penalty. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, pages 819–850, 2013.

Konstantinos I Tsianos, Sean Lawlor, and Michael G Rabbat. Push-sum distributed dual averaging for convex optimization. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 5453–5458. IEEE, 2012.

Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2722–2731, 2017.

Ruiliang Zhang and James Kwok. Asynchronous distributed admm for consensus optimization. In *International Conference on Machine Learning*, pages 1701–1709, 2014.

Tao Zhang and Quanyan Zhu. Dynamic differential privacy for admm-based distributed classification learning. *IEEE Transactions on Information Forensics and Security*, 12(1): 172–187, 2017.

Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. Improving the privacy and accuracy of admm-based distributed algorithms. *arXiv preprint arXiv:1806.02246*, 2018.

Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu. Recycled admm: Improving the privacy and accuracy of distributed algorithms. *IEEE Transactions on Information Forensics and Security*, 2019.