Recycled ADMM: Improving the Privacy and Accuracy of Distributed Algorithms

Xueru Zhang, Mohammad Mahdi Khalili, and Mingyan Liu, *Fellow, IEEE*

***Abstract*— Alternating direction method of multiplier (ADMM) is a powerful method to solve decentralized convex optimization problems. In distributed settings, each node performs computation with its local data and the local results are exchanged among neighboring nodes in an iterative fashion. During this iterative process the leakage of data privacy arises and can accumulate significantly over many iterations, making it difficult to balance the privacy-accuracy tradeoff. We propose Recycled ADMM (R-ADMM), where a linear approximation is applied to every even iteration, its solution directly calculated using only results from the previous, odd iteration. It turns out that under such a scheme, half of the updates incur no privacy loss and require much less computation compared to the conventional ADMM. Moreover, R-ADMM can be further modified (MR-ADMM) such that each node independently determines its own penalty parameter over iterations. We obtain a sufficient condition for the convergence of both algorithms and provide the privacy analysis based on objective perturbation. It can be shown that the privacy-accuracy tradeoff can be improved significantly compared with conventional ADMM.**

***Index Terms*— Differential privacy, distributed learning, ADMM.**

I. Introduction

DISTRIBUTED optimization and learning are crucial for many settings where the data is possessed by multiple parties or when the quantity of data prohibits processing at a central location. Many problems can be formulated as a convex optimization of the following form: $\min_{\mathbf{x}} \sum_{i=1}^{N} f_i(\mathbf{x})$. In a distributed setting, each entity/node i has its own local objective f_i, N entities/nodes collaboratively work to solve this objective through an interactive process of local computation and message passing. At the end all local results should ideally converge to the global optimum.

The information exchanged over the iterative process gives rise to privacy concerns if the local training data contains sensitive information such as medical or financial records, web search history, and so on [2]–[5]. It is therefore highly desirable to ensure such iterative processes are privacy-preserving. We adopt the ε-differential privacy to measure such privacy guarantee; it is generally achieved by perturbing the algorithm such that the probability distribution of its output is relatively insensitive to any change to a single record in the input [6].

Existing approaches to decentralizing the above problem primarily consist of subgradient-based algorithms [7]–[9] and ADMM-based algorithms [10]–[16]. It has been shown that ADMM-based algorithms can converge at the rate of $O(\frac{1}{k})$ while subgradient-based algorithms typically converge at the rate of $O(\frac{1}{\sqrt{k}})$, where k is the number of iterations [12]. In this study, we will solely focus on ADMM-based algorithms. While a number of differentially private (sub)gradient-based distributed algorithms have been proposed [17]–[20], the same is much harder for ADMM-based algorithms due to its computational complexity stemming from the fact that each node is required to solve an optimization problem in each iteration. Differentially private ADMM has been studied in [21]–[23]. In particular, Zhang and Zhu [21] proposes the dual/primal variable perturbation method to inspect the privacy loss of one node in every single iteration; this, however, is not sufficient for guaranteeing privacy as an adversary can potentially use the revealed results from all iterations to perform inference. Zhang *et al.* [22] addresses this issue by inspecting the total privacy loss over the entire process and the entire network; A penalty perturbation method is proposed which may improve the privacy-accuracy tradeoff significantly. Huang *et al.* [23] applies the first-order approximation to the augmented Lagrangian in all iterations; however, this method requires a central server to average all updated primal variables over the network in each iteration.

Since privacy leakage accumulates over iterations, the total privacy loss over the entire process can be substantial, making it difficult to balance the privacy-accuracy tradeoff. In our prior work [22] we introduced a penalty perturbation method to achieve a better tradeoff. While the method shows significant improvement with the right choice of penalty parameters, this improvement is heavily dependent on such choices and is not guaranteed. It is therefore important to seek guaranteed improvement in the privacy-accuracy tradeoff for ADMM-based algorithms, which is the subject of the present paper.

In this study, we present Recycled ADMM (R-ADMM), a modified version of ADMM where the privacy leakage only happens during half of the updates (Algorithm 1). Specifically, we adopt a linearized approximated optimization in every even iteration, whose solution is calculated directly using results from the previous, odd iteration; this solution is also used

Manuscript received April 3, 2019; revised August 6, 2019; accepted October 1, 2019. Date of publication October 16, 2019; date of current version January 22, 2020. This work was supported by the NSF under Grant CNS-1422211, Grant CNS-1646019, and Grant CNS-1739517. This article was presented at the 2018 Allerton Conference on Communication, Control and Computing [1]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Aris Gkoulalas Divanis. *(Corresponding author: Xueru Zhang.)*

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105 USA (e-mail: xueru@umich.edu; khalili@umich.edu; mingyan@umich.edu).

Digital Object Identifier 10.1109/TIFS.2019.2947867

for updating the primal variable. These approximated updates incur no privacy loss and require much less computation. Compared with conventional ADMM, R-ADMM requires much less perturbation to provide the same level of privacy protection, thereby improving the privacy-accuracy trade-off.

We then further generalize R-ADMM and present a modified R-ADMM, referred to as MR-ADMM, which employs ideas proposed in [22] and can accommodate non-constant penalty parameters which are also entity's own private information (Algorithm 2). Since the penalty parameter controls the updating step size, the algorithm can be more robust by decreasing the step size. It allows the algorithm to tolerate more noise, i.e., be more private, without jeopardizing too much accuracy. As a result the privacy-accuracy trade-off is further improved.

Both of these algorithms are essentially modifications of the original distributed ADMM algorithm; privacy in these algorithms are provided by introducing noise. Accordingly, the private versions of these algorithms are developed using the objective perturbation method [24] (Algorithm 3). We establish a sufficient condition for the convergence of both algorithms and characterize their corresponding total privacy loss for private algorithms. Both analysis and experiments on real-world datasets show that as compared with conventional ADMM algorithm, R-ADMM can improve the privacy-accuracy tradeoff significantly with much less computation. Moreover, by controlling the penalty parameters in MR-ADMM, this privacy-accuracy tradeoff is further improved.

The remainder of the paper is organized as follows. We present problem formulation and the definition of differential privacy and ADMM in Section II. Three algorithms are introduced in Section III including R-ADMM, MR-ADMM and the private MR-ADMM. The convergence analysis of non-private MR-ADMM, privacy analysis and generalization performance analysis of (non)-private MR-ADMM are presented in Section IV, V and VI, respectively. Discussion is given in Section VII. Numerical results are illustrated in Section VIII and Section IX concludes the paper. All proofs can be found in full version [25].

II. Preliminaries

A. Problem Formulation

Consider a connected network[1] given by an undirected graph $G(\mathcal{N}, \mathcal{E})$, which consists of a set of nodes $\mathcal{N} = \{1, 2, \cdots, N\}$ and a set of edges $\mathcal{E} = \{1, 2, \cdots, E\}$. Two nodes can exchange information if and only if they are connected by an edge. Let $\mathcal{V}_i$ denote node i's set of neighbors, excluding itself. Let D_i be node i's dataset.

Consider an optimization problem over this network of N nodes, where the overall objective function can be decomposed into N sub-objective functions and each depends on a node's local dataset, i.e.,

$$\min_{f_c} \text{Obj}(f_c, D_{all}) = \sum_{i=1}^{N} O(f_c, D_i) \tag{1}$$

[1]A connected network is one in which every node is reachable (via a path) from every other node.

The goal is to find a (centralized) optimal solution $f_c \in \mathbb{R}^d$ over the union of all local datasets $D_{all} = \cup_{i\in\mathcal{N}} D_i$ in a distributed manner using ADMM, while providing privacy guarantee for each data sample.

B. Differential Privacy [6]

A randomized algorithm $\mathcal{A}(\cdot)$ taking a dataset as input satisfies ε-differential privacy if for any two datasets D, $\hat{D}$ differing in at most one data point, and for any set of possible outputs $S \subseteq \text{range}(\mathcal{A})$, $\Pr(\mathcal{A}(D) \in S) \leq e^{\varepsilon}\Pr(\mathcal{A}(\hat{D}) \in S)$ holds. We call two datasets differing in at most one data point as neighboring datasets. $\varepsilon \in [0, \infty)$ can be used to quantify the privacy loss/guarantee. The above definition suggests that for a sufficiently small ε, an adversary will observe almost the same output regardless of the presence (or value change) of any one individual in the dataset; this is what provides privacy protection for that individual, the smaller ε, the smaller privacy loss, the stronger privacy guarantee.

Differential privacy is a worse-case measure; i.e., the bound is over all possible random outputs and all possible inputs. It is a strong guarantee, as it can protect against attackers with any side information. Moreover, it is immune to post-processing [26]; i.e., given only the differentially private output without additional information about the true data, it is impossible for attackers to make it less differentially private.

For an optimization problem over a dataset, there are many approaches to randomizing the output to preserve differential privacy and some of the most commonly used are as follows. (1) Output perturbation: solve the optimization problem first and then add zero-mean noise (e.g., Laplace, Gaussian) to the optimal solution. (2) Objective perturbation: add a noisy term to the objective function first and then solve the perturbed optimization problem. Because of this randomness, the accuracy of the output also decreases accordingly. The more perturbation, the output will be less accurate but it also provides the stronger privacy for individuals. Therefore, there is a privacy-accuracy tradeoff, and an important issue is how to improve this tradeoff so that the output can be more accurate under the same privacy guarantee.

C. Conventional ADMM

To decentralize (1), let f_i be the local classifier of each node i. To achieve consensus, i.e., $f_1 = f_2 = \cdots = f_N$, a set of auxiliary variables $\{w_{ij} | i \in \mathcal{N}, j \in \mathcal{V}_i\}$ are introduced for every pair of connected nodes. As a result, (1) is reformulated equivalently as:

$$\begin{aligned} \min_{\{f_i\},\{w_{ij}\}} \ & \widetilde{\text{Obj}}(\{f_i\}_{i=1}^N, D_{all}) = \sum_{i=1}^{N} O(f_i, D_i) \\ \text{s.t. } & f_i = w_{ij}, \quad w_{ij} = f_j, \ i \in \mathcal{N}, j \in \mathcal{V}_i \end{aligned} \tag{2}$$

Let $\{f_i\}$ and $\{w_{ij}\}$ be the shorthand for $\{f_i\}_{i\in\mathcal{N}}$ and $\{w_{ij}\}_{i\in\mathcal{N}, j\in\mathcal{V}_i}$, respectively. Let $\{w_{ij}, \lambda_{ij}^k\}$ be the shorthand for $\{w_{ij}, \lambda_{ij}^k\}_{i\in\mathcal{N}, j\in\mathcal{V}_i, k\in\{a,b\}}$, where λ_{ij}^a, λ_{ij}^b are dual variables corresponding to equality constraints $f_i = w_{ij}$ and $w_{ij} = f_j$ respectively. The objective in (2) can be solved using ADMM

with the augmented Lagrangian:

$$\begin{aligned} &L_\eta(\{f_i\},\{w_{ij},\lambda_{ij}^k\}) \\ &= \sum_{i=1}^{N} O(f_i,D_i) \\ &\quad + \sum_{i=1}^{N}\sum_{j\in\mathscr{V}_i}(\lambda_{ij}^a)^T(f_i-w_{ij}) + \sum_{i=1}^{N}\sum_{j\in\mathscr{V}_i}(\lambda_{ij}^b)^T(w_{ij}-f_j) \\ &\quad + \sum_{i=1}^{N}\sum_{j\in\mathscr{V}_i}\frac{\eta}{2}(||f_i-w_{ij}||_2^2+||w_{ij}-f_j||_2^2). \end{aligned} \tag{3}$$

where η is called the penalty parameter. In the $(t+1)$-th iteration, the ADMM updates consist of the following:

$$f_i(t+1) = \underset{f_i}{\text{argmin}}\ L_\eta(\{f_i\},\{w_{ij}(t),\lambda_{ij}^k(t)\}); \tag{4}$$

$$w_{ij}(t+1) = \underset{w_{ij}}{\text{argmin}}\ L_\eta(\{f_i(t+1)\},\{w_{ij},\lambda_{ij}^k(t)\}); \tag{5}$$

$$\lambda_{ij}^a(t+1) = \lambda_{ij}^a(t) + \eta(f_i(t+1) - w_{ij}(t+1)); \tag{6}$$

$$\lambda_{ij}^b(t+1) = \lambda_{ij}^b(t) + \eta(w_{ij}(t+1) - f_j(t+1)). \tag{7}$$

Using Lemma 3 in [27], if dual variables $\lambda_{ij}^a(t)$ and $\lambda_{ij}^b(t)$ are initialized to zero for all node pairs (i,j), then $\lambda_{ij}^a(t) = \lambda_{ij}^b(t)$ and $\lambda_{ij}^k(t) = -\lambda_{ji}^k(t)$ will hold for all iterations with $k\in\{a,b\}, i\in\mathscr{N}, j\in\mathscr{V}_i$. Let $\lambda_i(t) = \sum_{j\in\mathscr{V}_i}\lambda_{ij}^a(t) = \sum_{j\in\mathscr{V}_i}\lambda_{ij}^b(t)$, then the ADMM iterations (4)-(7) can be simplified as (Refer to Appendix A in [22] for proof):

$$\begin{aligned} f_i(t+1) = \underset{f_i}{\text{argmin}}\{&O(f_i,D_i) + 2\lambda_i(t)^T f_i \\ &+\eta\sum_{j\in\mathscr{V}_i}||\frac{1}{2}(f_i(t)+f_j(t)) - f_i||_2^2\ \}; \end{aligned} \tag{8}$$

$$\lambda_i(t+1) = \lambda_i(t) + \frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(t+1) - f_j(t+1)). \tag{9}$$

D. Private ADMM [21] & Private M-ADMM [22]

In private ADMM [21], noise is added either to the updated primal variable before broadcasting to its neighbors (primal variable perturbation), or to the dual variable before updating its primal variable using (8) (dual variable perturbation). The privacy property is only evaluated for a single node and a single iteration, but neither method can effectively balance the privacy-accuracy tradeoff if the total privacy loss is considered. In our prior work [22], the total privacy loss of the whole network over the entire iterative process is considered. A modified ADMM (M-ADMM) was proposed to improve the privacy-accuracy tradeoff. Specifically, it explores the use of the penalty parameter η in stabilizing the algorithm. M-ADMM allows each node to independently determine its penalty parameter and randomizes the objective function in primal update (8) by adding a linear noise term correlated to the penalty parameter while at the same time increasing the penalty over time. By doing so it is shown that the privacy and accuracy can be improved simultaneously.

III. ALGORITHMS

A. Recycled ADMM (R-ADMM)

1) Main Idea: Fundamentally, the accumulation of privacy loss over iterations stems from the fact that the individual data D_{all} is used in every primal update. If the updates can be made without directly using this original data, but only from computational results that already exist, then the privacy loss originating from these updates will be zero, while at the same time the computational cost may be reduced significantly. This idea of "recycling information" is also supported by the immunity to post-processing that differential privacy possesses [26], i.e., any computation over an output that is already differentially private cannot incur additional privacy loss. Toward this end, R-ADMM modifies the ADMM algorithm such that we repeatedly use earlier computational results to make updates.

2) Making Information Recyclable: ADMM can outperform gradient-based methods in terms of requiring fewer number of iterations for convergence; this however comes at the price of high computational cost in every iteration. In particular, the primal variable is updated by performing an optimization in each iteration. In [13], [28], [29], either a linear or quadratic approximation of the objective function is used to obtain an inexact solution in each iteration in lieu of solving the original optimization problem. While this clearly lowers the computational cost, the approximate computation is performed using the local, individual data in every iteration, which means that privacy loss inevitably accumulates over the iterations.

We begin by modifying ADMM in such a way that in every even iteration, without using data D_{all}, the primal variable is updated solely based on the existing computational results from the previous, odd iteration. Compared with conventional ADMM, these updates incur no privacy loss and less computation. Since the computational results are repeatedly used, this method is referred to as Recycled ADMM (R-ADMM).

Specifically, in the $2k$-th (even) iteration, $O(f_i,D_i)$ (Eqn. (8), primal update optimization) is approximated by $O(f_i,D_i)\approx O(f_i(2k-1),D_i) + \nabla O(f_i(2k-1),D_i)^T(f_i - f_i(2k-1)) + \frac{\gamma}{2}||f_i - f_i(2k-1)||_2^2$ ($\gamma\geq 0$) and only the primal variables are updated. Using the first-order condition, the updates in the $2k$-th iteration become:

$$\begin{aligned} f_i(2k) = f_i(2k-1) - \frac{1}{2\eta V_i+\gamma}\{&\nabla O(f_i(2k-1),D_i) \\ &+2\lambda_i(2k-1)+\eta\sum_{j\in\mathscr{V}_i}(f_i(2k-1)-f_j(2k-1))\}; \end{aligned} \tag{10}$$

$$\lambda_i(2k) = \lambda_i(2k-1). \tag{11}$$

In the $(2k-1)$-th (odd) iteration, the updates are kept the same as (8)(9):

$$\begin{aligned} f_i(2k-1) = \underset{f_i}{\text{argmin}}\{&O(f_i,D_i) + 2\lambda_i(2k-2)^T f_i \\ &+\eta\sum_{j\in\mathscr{V}_i}||\frac{1}{2}(f_i(2k-2)+f_j(2k-2))-f_i||_2^2\ \}; \end{aligned} \tag{12}$$

$$\lambda_i(2k-1) = \lambda_i(2k-2) + \frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1)). \quad (13)$$

Note that in the $(2k)$-th (even) iteration, we need the gradient $\nabla O(f_i(2k-1), D_i)$ and primal difference $\frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$ for the updates; these are available directly from the previous, $(2k-1)$-th (odd) iteration, i.e., this information can be recycled. In this sense, R-ADMM may be viewed as alternating between conventional ADMM (odd iterations) and a variant of gradient descent (even iterations), where $\frac{1}{2\eta V_i+\gamma}$ is the step-size with a slightly modified gradient term. The complete procedure is shown in Algorithm 1.

Algorithm 1 Recycled-ADMM (R-ADMM)

Input: $\{D_i\}_{i=1}^N$
Initialize: $\forall i$, generate $f_i(0)$ randomly, $\lambda_i(0) = \mathbf{0}_{d\times 1}$
for $k = 1$ **to** K **do**
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Update primal variable $f_i(2k-1)$ via (12);
    Calculate the gradient $\nabla O(f_i(2k-1), D_i)$;
    Broadcast $f_i(2k-1)$ to all neighbors $j \in \mathscr{V}_i$.
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Calculate $\eta\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$;
    Update dual variable $\lambda_i(2k-1)$ via (13).
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Use the stored $\nabla O(f_i(2k-1), D_i)$ and $\eta\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$ to update primal variable $f_i(2k)$ via (10);
    Keep the dual variable $\lambda_i(2k) = \lambda_i(2k-1)$;
    Broadcast $f_i(2k)$ to all neighbors $j \in \mathscr{V}_i$.
Output: primal $\{f_i(2K)\}_{i=1}^N$ and dual $\{\lambda_i(2K)\}_{i=1}^N$

B. Modified R-ADMM (MR-ADMM)

1) Making η a Node's Private Information: R-ADMM requires that the penalty parameter η be fixed for all nodes in all iterations. Inspired by M-ADMM in [22], we modify R-ADMM such that each node can independently determine its penalty parameter in each iteration. Specifically, replace η in (10), (12) and (13) with $\eta_i(2k-1)$. The updating formula is then given in (14)-(17), as shown at the bottom of this page. The complete procedure is shown in Algorithm 2.

Algorithm 2 Modified R-ADMM (MR-ADMM)

Input: $\{D_i\}_{i=1}^N$
Initialize: $\forall i$, generate $f_i(0)$ randomly, $\lambda_i(0) = \mathbf{0}_{d\times 1}$
Parameter: $\forall i$, select $\{\eta_i(2k-1)\}_{k=1}^K$ s.t. $0 < \eta_i(2k-1) \le \eta_i(2k+1) < +\infty$, $\forall k$
for $k = 1$ **to** K **do**
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Update primal variable $f_i(2k-1)$ via (14);
    Calculate the gradient $\nabla O(f_i(2k-1), D_i)$;
    Broadcast $f_i(2k-1)$ to all neighbors $j \in \mathscr{V}_i$.
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Calculate $\eta_i(2k-1)\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$;
    Update dual variable $\lambda_i(2k-1)$ via (15).
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Use the stored $\nabla O(f_i(2k-1), D_i)$ and $\eta_i(2k-1)\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$ to update primal variable $f_i(2k)$ via (16);
    Keep the dual variable $\lambda_i(2k) = \lambda_i(2k-1)$;
    Broadcast $f_i(2k)$ to all neighbors $j \in \mathscr{V}_i$.
Output: primal $\{f_i(2K)\}_{i=1}^N$ and dual $\{\lambda_i(2K)\}_{i=1}^N$

2) Relationship Between R-ADMM and MR-ADMM: MR-ADMM is a generalized version of R-ADMM. If fix $\eta_i(2k-1) = \eta$, $\forall k$, then MR-ADMM will be reduced to R-ADMM.

3) Role of $\eta_i(2k-1)$ in Stabilizing the Algorithm: The penalty parameter $\eta_i(2k-1)$ directly controls the step size of the algorithm. Since the goal is to minimize the objective in (14), if $\eta_i(2k-1)$ is larger, the solution $f_i(2k-1)$ will be closer to the primal variable in the previous iteration so that the penalty term $\sum_{j\in\mathscr{V}_i}||\frac{1}{2}(f_i(2k-2)+f_j(2k-2)) - f_i||_2^2$ will be small. In other words, larger $\eta_i(2k-1)$ results in smaller update of the primal variable $f_i(2k-1)$. In even updates (16), $\frac{1}{2\eta_i(2k-1)V_i+\gamma}$ can also be regarded as step size as mentioned earlier. Therefore, increasing $\eta_i(2k-1)$ decreases the step size in both even and odd iterations.

Without perturbation, decreasing step size might slow down the convergence. However, when the algorithm is

$$f_i(2k-1) = \underset{f_i}{\operatorname{argmin}}\{O(f_i, D_i) + 2\lambda_i(2k-2)^T f_i + \eta_i(2k-1)\sum_{j\in\mathscr{V}_i}||\frac{1}{2}(f_i(2k-2) + f_j(2k-2)) - f_i||_2^2\}; \quad (14)$$

$$\lambda_i(2k-1) = \lambda_i(2k-2) + \frac{\eta_i(2k-1)}{2}\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1)). \quad (15)$$

$$f_i(2k) = f_i(2k-1) - \frac{1}{2\eta_i(2k-1)V_i+\gamma}\{\nabla O(f_i(2k-1), D_i) + 2\lambda_i(2k-1) + \eta_i(2k-1)\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))\}; \quad (16)$$

$$\lambda_i(2k) = \lambda_i(2k-1). \quad (17)$$

perturbed with added noise, a smaller step size could prevent the variable from deviating too much from the optimal solution in each update, which in turn stabilizes the algorithm. In the rest of paper, we will introduce a private algorithm by perturbing MR-ADMM and illustrate how we can use our ability to control stability via $\eta_i(2k-1)$ to improve the accuracy of algorithm without jeopardizing privacy.

C. Private MR-ADMM

In this section we present a privacy preserving version of MR-ADMM. Since MR-ADMM is a generalized version of R-ADMM, the private version of R-ADMM can be built in a similar way. In odd iterations, we adopt the objective perturbation [24] where a random linear term $\epsilon_i(2k-1)^T f_i$ is added to the objective function in (12),[2] where $\epsilon_i(2k-1)$ follows the probability density proportional to $\exp\{-\alpha_i(k)||\epsilon_i(2k-1)||_2\}$. Consequently the objective function for updating the primal variable $f_i(2k-1)$ becomes $L_i^{priv}(2k-1)$ given as follows:

$$\begin{aligned} &L_i^{priv}(2k-1) \\ &= O(f_i, D_i) + (2\lambda_i(2k-2) + \epsilon_i(2k-1))^T f_i \\ &\quad + \eta_i(2k-1) \sum_{j\in\mathscr{V}_i} ||\frac{1}{2}(f_i(2k-2) + f_j(2k-2)) - f_i||_2^2 \end{aligned}$$

To generate this noisy vector $\epsilon_i(2k-1)$, choose the norm from the gamma distribution with shape d and scale $\frac{1}{\alpha_i(k)}$ and the direction uniformly, where d is the dimension of the feature space. Node i's local result (primal variable) is obtained by finding the optimal solution to the private objective function:

$$f_i(2k-1) = \underset{f_i}{\operatorname{argmin}}\ L_i^{priv}(2k-1), \quad i \in \mathscr{N}. \tag{18}$$

In the $2k$-th iteration, use the stored results $\epsilon_i(2k-1) + \nabla O(f_i(2k-1), D_i)$ and $\eta_i(2k-1)\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$ to update primal variables, where the latter can be obtained from the dual update in the $(2k-1)$-th update, and the former can be obtained directly from the KKT condition in the $(2k-1)$-th iteration:

$$\begin{aligned} &\epsilon_i(2k-1) + \nabla O(f_i(2k-1), D_i) \\ &= -2\lambda_i(2k-2) - \eta_i(2k-1) \\ &\quad \times \sum_{j\in\mathscr{V}_i}(2f_i(2k-1)) - f_i(2k-2) - f_j(2k-2)). \end{aligned}$$

Then the even update is given by:

$$\begin{aligned} f_i(2k) = f_i(2k-1) - \frac{1}{2\eta_i(2k-1)V_i + \gamma}\{&2\lambda_i(2k-1) \\ &+ \underbrace{\epsilon_i(2k-1) + \nabla O(f_i(2k-1), D_i)}_{\text{the existing result by KKT}} \\ &+ \underbrace{\eta_i(2k-1)\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))}_{\text{the existing result by the previous dual update}}\}. \end{aligned} \tag{19}$$

[2]Pure differential privacy was adopted in this work, but the weaker (ϵ, δ)-differential privacy can be applied as well.

Algorithm 3 Private MR-ADMM

Input: $\{D_i\}_{i=1}^N$, $\{\alpha_i(1), \cdots, \alpha_i(K)\}_{i=1}^N$
Initialize: $\forall i$, generate $f_i(0)$ randomly, $\lambda_i(0) = \mathbf{0}_{d\times 1}$
Parameter: $\forall i$, select $\{\eta_i(2k-1)\}_{k=1}^K$ s.t. $0 < \eta_i(2k-1) \le \eta_i(2k+1) < +\infty$, $\forall k$ and $\eta_i(1)$ satisfies $2c_1 < \min_i\{\frac{B_i}{C}(\frac{\rho}{N} + 2\eta_i(1)V_i)\}$
for $k = 1$ **to** K **do**
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Generate noise $\epsilon_i(2k-1) \sim \exp(-\alpha_i(k)||\epsilon||_2)$;
    Update primal variable $f_i(2k-1)$ via (18);
    Broadcast $f_i(2k-1)$ to all neighbors $j \in \mathscr{V}_i$.
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Calculate $\eta_i(2k-1)\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$;
    Update dual variable $\lambda_i(2k-1)$ via (15).
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Use the stored information $\epsilon_i(2k-1) + \nabla O(f_i(2k-1), D_i)$ and $\eta_i(2k-1)\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$ to update primal variable $f_i(2k)$ via (19);
    Keep the dual variable $\lambda_i(2k) = \lambda_i(2k-1)$;
    Broadcast $f_i(2k)$ to all neighbors $j \in \mathscr{V}_i$.
Output: Upper bound of the total privacy loss β; primal $\{f_i(2K)\}_{i=1}^N$ and dual $\{\lambda_i(2K)\}_{i=1}^N$

Algorithm 3 shows the complete procedure, where the condition used to generate $\eta_i(1)$ helps to bound the worst-case privacy loss but is not necessary in guaranteeing convergence.

IV. Convergence of Non-Private MR-ADMM

Since MR-ADMM is a generalized version of R-ADMM, we focus on the convergence analysis of MR-ADMM in this section while the results immediately apply to R-ADMM by fixing $\eta_i(2k-1) = \eta$, $\forall k$.

We next show that the MR-ADMM (Eqn. (14)-(17)) converges to the optimal solution under a set of common technical assumptions.

Assumption 1: Function $O(f_i, D_i)$ is convex and differentiable in f_i, $\forall i$.

Assumption 2: The solution set to the original problem (1) is nonempty and there exists at least one bounded element.

Assumption 3: For all $i \in \mathscr{N}$, $O(f_i, D_i)$ has Lipschitz continuous gradients, i.e., for any f_i^1 and f_i^2, we have:

$$||\nabla O(f_i^1, D_i) - \nabla O(f_i^2, D_i)||_2 \le M_i||f_i^1 - f_i^2||_2 \tag{28}$$

By the KKT condition of the primal update (14):

$$\begin{aligned} 0 = \nabla O(f_i(2k-1), D_i) + 2\lambda_i(2k-2) + \eta_i(2k-1) \\ \cdot \sum_{j\in\mathscr{V}_i}(2f_i(2k-1) - (f_i(2k-2) + f_j(2k-2))). \end{aligned} \tag{29}$$

Define the adjacency matrix $A \in \mathbb{R}^{N\times N}$ as:

$$a_{ij} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are connected} \\ 0, & \text{otherwise .} \end{cases}$$

Stack the variables $f_i(t)$, $\lambda_i(t)$ and $\nabla O(f_i(t), D_i)$ for $i \in \mathcal{N}$ into matrices, i.e.,

$$\hat{f}(t) = \begin{bmatrix} f_1(t)^T \\ f_2(t)^T \\ \vdots \\ f_N(t)^T \end{bmatrix} \in \mathbb{R}^{N\times d}, \quad \Lambda(t) = \begin{bmatrix} \lambda_1(t)^T \\ \lambda_2(t)^T \\ \vdots \\ \lambda_N(t)^T \end{bmatrix} \in \mathbb{R}^{N\times d}$$

$$\nabla \hat{O}(\hat{f}(t), D_{all}) = \begin{bmatrix} \nabla O(f_1(t), D_1)^T \\ \nabla O(f_2(t), D_2)^T \\ \vdots \\ \nabla O(f_N(t), D_N)^T \end{bmatrix} \in \mathbb{R}^{N\times d}$$

Let $V_i = |\mathcal{V}_i|$ be the number of neighbors of node i, and define the degree matrix $D = \mathbf{diag}([V_1; V_2; \cdots; V_N]) \in \mathbb{R}^{N\times N}$, the diagonal matrix $\tilde{D}(2k-1)$ with $\tilde{D}(2k-1)_{ii} = 2\eta_i(2k-1)V_i + \gamma$, and the weight matrix $W(2k-1) = \mathbf{diag}([\eta_1(2k-1); \eta_2(2k-1); \cdots; \eta_N(2k-1)]) \in \mathbb{R}^{N\times N}$. Then for each k, the matrix form of (16)(17)(29)(15) are given in (20)-(23), as shown at the bottom of this page.

Writing $\hat{f}(2k-2)$ and $\Lambda(2k-2)$ in (22)(23) as functions of $\hat{f}(2k-3)$, $\Lambda(2k-3)$ using (20)(21), we obtain Eqn. (24)(25), as shown at the bottom of this page.

The convergence of the MR-ADMM is proved by showing that the pair $(\hat{f}(2k-1), \Lambda(2k-1))$ from odd iterations converges to the optimal solution. To simplify the notation, we will re-index every two consecutive odd iterations $2k-3$ and $2k-1$ using t and $t+1$, it results in Eqn. (26)(27), as shown at the bottom of this page.

Note that $D - A$ is the laplacian and $D + A$ is the signless Laplacian matrix of the network, with the following properties if the network is connected: (i) $D \pm A \succeq 0$ is positive semi-definite; (ii) $\text{Null}(D-A) = c\mathbf{1}$, i.e., every member in the null space of $D - A$ is a scalar multiple of $\mathbf{1}$ with $\mathbf{1}$ being the vector of all 1's [30].

Lemma 1 (First-Order Optimality Condition [16]): Under Assumptions 1 and 2, the following two statements are equivalent:

- $\hat{f}^* = [(f_1^*)^T; (f_2^*)^T; \cdots; (f_N^*)^T] \in \mathbb{R}^{N\times d}$ *is consensual, i.e.,* $f_1^* = f_2^* = \cdots = f_N^* = f_c^*$ *where* f_c^* *is the optimal solution to* (1).
- *There exists a pair* $(\hat{f}^*, \Lambda^*)$ *with* $2\Lambda^* = (D-A)X$ *for some* $X \in \mathbb{R}^{N\times d}$ *such that*

$$\nabla \hat{O}(\hat{f}^*, D_{all}) + 2\Lambda^* = \mathbf{0}_{N\times d}; \tag{30}$$

$$(D-A)\hat{f}^* = \mathbf{0}_{N\times d}. \tag{31}$$

Lemma 1 shows that a pair $(\hat{f}^*, \Lambda^*)$ satisfying (30)(31) is equivalent to the optimal solution of our problem, hence the convergence of the MR-ADMM is proved by showing that $(\hat{f}(t), \Lambda(t))$ in (26)(27) converges to a pair $(\hat{f}^*, \Lambda^*)$ satisfying (30)(31).

Theorem 1 (Sufficient Condition): Consider the modified ADMM defined by (26)(27)*. Let* $\{\hat{f}(t), \Lambda(t)\}$ *be outputs in each iteration and* $\{\hat{f}^*, \Lambda^*\}$ *a pair satisfying* (30)(31)*. Denote* $D_M = \boldsymbol{diag}([M_1^2; M_2^2; \cdots; M_N^2]) \in \mathbb{R}^{N\times N}$ *with* $0 < M_i < +\infty$ *as given in Assumption 3. If* $\eta_i(t+1) \geq \eta_i(t) > 0$ *and* $\eta_i(t) < +\infty$ *hold and the following two conditions can also be satisfied for some constants* $L > 0$ *and* $\mu > 1$*:*

$$(i)\ I + W(t+1)(D+A)\tilde{D}(t)^{-1} \succ \frac{L\mu}{2\sigma_{\min}(\tilde{D}(t))}(W(t+1)(D-A))^+ D_M;$$

$$(ii)\ W(t+1)(D+A) \succ W(t+1)(D+A)\tilde{D}(t)^{-1}\Big(W(t)(D-A) + \frac{2}{L}W(t+1)(D+A)\Big) + \frac{L\mu}{2\sigma_{\min}(\tilde{D}(t))(\mu-1)}D_M.$$

where $\sigma_{\min}(\tilde{D}(t)) = \min_i\{2\eta_i(t)V_i + \gamma\}$ *is the smallest singular value of* $\tilde{D}(t)$*, then* $(\hat{f}(t), \Lambda(t))$ *converges to* $(\hat{f}^*, \Lambda^*)$.

Proof: See Appendix A in [25]. □

By controlling γ to be sufficiently large, $\tilde{D}(t)_{ii} = 2\eta_i(t)V_i + \gamma$ will be large and conditions *(i)(ii)* can always be satisfied under some constants $L > 0$ and $\mu > 1$. Note that the conditions *(i)(ii)* are sufficient but not necessary, so in practice convergence may be attained under weaker settings.

$$\hat{f}(2k) = \hat{f}(2k-1) - \tilde{D}(2k-1)^{-1}\{\nabla \hat{O}(\hat{f}(2k-1), D_{all}) + 2\Lambda(2k-1) + W(2k-1)(D-A)\hat{f}(2k-1)\}; \tag{20}$$

$$2\Lambda(2k) = 2\Lambda(2k-1); \tag{21}$$

$$\mathbf{0}_{N\times d} = \nabla \hat{O}(\hat{f}(2k-1), D_{all}) + 2\Lambda(2k-2) + W(2k-1)(2D\hat{f}(2k-1) - (D+A)\hat{f}(2k-2)); \tag{22}$$

$$2\Lambda(2k-1) = 2\Lambda(2k-2) + W(2k-1)(D-A)\hat{f}(2k-1). \tag{23}$$

$$\begin{aligned}\mathbf{0}_{N\times d} = {} & \nabla \hat{O}(\hat{f}(2k-1), D_{all}) + W(2k-1)(D+A)\tilde{D}(2k-3)^{-1}\nabla \hat{O}(\hat{f}(2k-3), D_{all}) \\ & + W(2k-1)(D+A)(\hat{f}(2k-1) - \hat{f}(2k-3)) \\ & + W(2k-1)(D+A)\tilde{D}(2k-3)^{-1}W(2k-3)(D-A)\hat{f}(2k-3) \\ & + 2\Lambda(2k-1) + W(2k-1)(D+A)\tilde{D}(2k-3)^{-1}2\Lambda(2k-3);\end{aligned} \tag{24}$$

$$2\Lambda(2k-1) = 2\Lambda(2k-3) + W(2k-1)(D-A)\hat{f}(2k-1). \tag{25}$$

$$\begin{aligned}\mathbf{0}_{N\times d} = {} & \nabla \hat{O}(\hat{f}(t+1), D_{all}) + W(t+1)(D+A)\tilde{D}(t)^{-1}\nabla \hat{O}(\hat{f}(t), D_{all}) + W(t+1)(D+A)(\hat{f}(t+1) - \hat{f}(t)) \\ & + W(t+1)(D+A)\tilde{D}(t)^{-1}W(t)(D-A)\hat{f}(t) + 2\Lambda(t+1) + W(t+1)(D+A)\tilde{D}(t)^{-1}2\Lambda(t);\end{aligned} \tag{26}$$

$$2\Lambda(t+1) = 2\Lambda(t) + W(t+1)(D-A)\hat{f}(t+1). \tag{27}$$

For R-ADMM, take $L = 2$ and $\mu = 2$, then condition *(i)(ii)* are reduced to:

$$(iii)\ I + \eta(D+A)\tilde{D}^{-1} \succ \frac{2}{\eta\sigma_{\min}(\tilde{D})}((D-A))^{+}D_M;$$

$$(iv)\ \eta(D+A) \succ 2\eta(D+A)\tilde{D}^{-1}\eta D + \frac{2}{\sigma_{\min}(\tilde{D})}D_M.$$

Again for a sufficiently large $\gamma \geq 0$, *(iii)(iv)* can be easily satisfied.

V. Privacy Analysis

In this section, we characterize the total privacy loss of private MR-ADMM as presented in Algorithm 3. Similar to the previous section, the results also apply to private R-ADMM by fixing $\eta_i(2k-1) = \eta$, $\forall k$.

As mentioned earlier, Zhang and Zhu [21] only quantifies the privacy loss of a single node in a single iteration, i.e., $\frac{\Pr(f_i(t)\in S_i|D_i)}{\Pr(f_i(t)\in S_i|\hat{D}_i)} \leq \exp(\alpha_i(t))$ holds $\forall t, i$, where $\alpha_i(t)$ is the bound on the privacy loss of node i at iteration t. However, in a distributed and iterative setting, the "output" of the algorithm is not merely the end result, but includes all intermediate results generated and exchanged during the iterative process; an attacker can use all such intermediate results to perform inference. For this reason, we adopt the differential privacy definition proposed in [22] as follows, which bounds the total privacy loss during the entire iterative process.

Definition 1: Consider a connected network $G(\mathcal{N}, \mathcal{E})$ with a set of nodes $\mathcal{N} = \{1, 2, \cdots, N\}$. Let $f(t) = \{f_i(t)\}_{i=1}^{N}$ denote the information exchange of all nodes in the t-th iteration. A distributed algorithm is said to satisfy β-differential privacy during T iterations if for any two datasets $D_{all} = \cup_i D_i$ and $\hat{D}_{all} = \cup_i \hat{D}_i$, differing in at most one data point, and for any set of possible outputs S during T iterations, the following holds:

$$\frac{Pr(\{f(t)\}_{t=0}^{T} \in S|D_{all})}{Pr(\{f(t)\}_{t=0}^{T} \in S|\hat{D}_{all})} \leq \exp(\beta)$$

The analysis is focused on the regularized empirical risk minimization (ERM) problem for binary classification, while its generalization is discussed in Section VII. Let node i's dataset be $D_i = \{(x_i^n, y_i^n)|n = 1, 2, \cdots, B_i\}$, where $x_i^n \in \mathbb{R}^d$ is the feature vector representing the n-th sample belonging to i, $y_i^n \in \{-1, 1\}$ the corresponding label, and B_i the size of D_i. Then the sub-objective function for each node i is defined as follows:

$$O(f_i, D_i) = \frac{C}{B_i}\sum_{n=1}^{B_i}\mathscr{L}(y_i^n f_i^T x_i^n) + \frac{\rho}{N}R(f_i)\ ,$$

where $C \leq B_i$ and $\rho > 0$ are constant parameters of the algorithm, the loss function $\mathscr{L}(\cdot)$ measures the accuracy of the classifier, and the regularizer $R(\cdot)$ helps prevent overfitting.

For this binary classification problem, we now state another result on the privacy property of the private MR-ADMM (Algorithm 3) using definition 1 above and additional assumptions on $\mathscr{L}(\cdot)$ and $R(\cdot)$ as follows.

Assumption 4: The loss function $\mathscr{L}$ is strictly convex and twice differentiable. $|\nabla\mathscr{L}| \leq 1$ and $0 < \mathscr{L}'' \leq c_1$ with c_1 being a constant.

Assumption 5: The regularizer R is 1-strongly convex and twice continuously differentiable.

Lemma 2: Consider the private MR-ADMM (Algorithm 3), $\forall k = 1, \cdots K$, assume the total privacy loss up to the $(2k-1)$-th iteration can be bounded by β_{2k-1}, then the total privacy loss up to the $2k$-th iteration can also be bounded by β_{2k-1}. In other words, given the private results in odd iterations, outputting private results in the even iterations does not release more information about the input data.

Proof: See Appendix B in [25]. □

Theorem 2: Normalize feature vectors in the training set such that $||x_i^n||_2 \leq 1$ for all $i \in \mathcal{N}$ and n. Then the private MR-ADMM algorithm (Algorithm 3) satisfies the β-differential privacy with

$$\beta \geq \max_{i\in\mathcal{N}}\{\sum_{k=1}^{K}\frac{2C}{B_i}(\frac{1.4c_1}{(\frac{\rho}{N}+2\eta_i(2k-1)V_i)} + \alpha_i(k))\}. \quad (32)$$

Proof: See Appendix C in [25]. □

VI. Sample Complexity Analysis

We next quantify the generalization performance of (non)-private MR-ADMM. The analysis is focused on the ERM problem defined in Section V and we assume samples from each node i are drawn i.i.d. from a fixed distribution P. The expected loss of node i using classifier $f_i(t)$ at time t is given as $\mathcal{L}(f_i(t)) = \mathbb{E}_{(X,Y)\sim P}(\mathscr{L}(Yf_i(t)^T X))$. Similar to the analysis in [21], [24], we introduce a reference classifier f_{ref} with expected loss $\mathcal{L}(f_{ref})$ and evaluate the generalization performance using the number of samples (B_i) required at each node to achieve $\mathcal{L}(f_i(t)) \leq \mathcal{L}(f_{ref}) + \tau$ with high probability.

A. Non-Private MR-ADMM

As shown in Section IV, the sequence of outputs $\{f_i^{non}(2k-1)\}$ from odd iterations in non-private MR-ADMM converges to $f_i^* = f_c^*$ as $k \to \infty$. Therefore, there exists a constant $\Delta_i(k)$ for each node i at the $(2k-1)$-th iteration such that $\mathcal{L}(f_i^{non}(2k-1)) \leq \mathcal{L}(f_c^*) + \Delta_i(k)$. Using the same method as [21], [24], we have the following result.

Theorem 3: Consider a regularized ERM problem with regularizer $R(f) = \frac{1}{2}||f||^2$ and let f_{ref} be a reference classifier for all nodes and $\{f_i^{non}(2k-1)\}$ be a sequence of outputs of non-private MR-ADMM in odd iterations (Eqn. (14)*). If the number of samples at node i satisfies*

$$B_i \geq w\max_k\{\frac{||f_{ref}||^2\log(1/\delta)}{(\tau-\Delta_i(k))^2}\}$$

for some constant w, then $f_i^{non}(2k-1)$ satisfies

$$Pr(\mathcal{L}(f_i^{non}(2k-1)) \leq \mathcal{L}(f_{ref}) + \tau) \geq 1-\delta$$

where $\tau > \Delta_i(k)$, $\forall i, k \in \mathbb{Z}_+$.

Proof: See Appendix D in [25]. □

As expected, the number of required samples depends on the choice of the reference classifier via its l_2 norm $||f_{ref}||^2$,

by imposing an upper bound b_{ref} on $||f_{ref}||^2$. The result shows that if B_i satisfies $B_i \geq w \max_k \{\frac{b_{ref} \log(1/\delta)}{(\tau - \Delta_i(k))^2}\}$, then the non-private intermediate classifier of each node at odd iterations will have an additional error no more than τ as compared to any classifier with $||f_{ref}||^2 \leq b_{ref}$.

B. Private MR-ADMM

We next present the result on the sample complexity of the private MR-ADMM algorithm. Similar to the analysis of non-private MR-ADMM, we bound the error of the intermediate classifier of each node at odd iterations. Since the algorithm is perturbed with different random noise in different iterations, to better analyze the effect of noise in a single iteration, we adopt a strategy similar to that used in [21], by intentionally fixing the noise in iterations after the targeted iteration. Specifically, $\forall i$, to compare the private $f_i^{priv}(2k-1)$ at the $(2k-1)$-th iteration with reference classifier f_{ref}, we slightly modify Algorithm 3 such that $\forall k' > k$, the added noise is fixed at $\epsilon_i(2k'-1) = \epsilon_i(2k-1)$, which allows us to solely study the effect of $\epsilon_i(2k-1)$. This problem can be formulated as a new MR-ADMM optimization problem where node i's sub-objective function becomes $O^{new}(f_i, D_i) = O(f_i, D_i) + \epsilon_i(2k-1)^T f_i$ and the initialization given by $f_i(0) = f_i(2k-1)$, $\lambda_i(0) = \lambda_i(2k-1)$. Let $\{f_i^{new}(2k-1)\}$ be a sequence of outputs from odd iterations of this new algorithm; it converges to a fixed point f_{new}^* as $k \to \infty$. Therefore, there exists a constant $\Delta_i^{new}(k)$ for each node i at the $(2k-1)$-th iteration such that $\mathcal{L}(f_i^{new}(2k-1)) \leq \mathcal{L}(f_{new}^*) + \Delta_i^{new}(k)$. Using this, we have the following result.

Theorem 4: Consider a regularized ERM problem with regularizer $R(f) = \frac{1}{2}||f||^2$, let f_{ref} be a reference classifier for all nodes and $\{f_i^{priv}(2k-1)\}$ be a sequence of outputs of private MR-ADMM in odd iterations. If the number of samples at node i satisfies

$$B_i \geq w \max_k \{\frac{CN\log(1/\delta)}{\frac{NC(\tau-\Delta_i^{new}(k))^2}{2||f_{ref}||^2} - (1+a)\frac{Nd^2}{C(\alpha_i(k))^2}(\log(d/\delta))^2}\}$$

for some constants w and $a > 0$, then $f_i^{priv}(2k-1)$ satisfies

$$Pr(\mathcal{L}(f_i^{priv}(2k-1)) \leq \mathcal{L}(f_{ref}) + \tau) \geq 1 - 2\delta$$

where $\tau > \Delta_i^{new}(k)$, $\forall i, k \in \mathbb{Z}_+$.

Proof: See Appendix E in [25]. □

Compared to Theorem 3, we see an additional term imposed by the privacy constraints, i.e., $(1+a)\frac{Nd^2}{C(\alpha_i(k))^2}(\log(d/\delta))^2$. If $\alpha_i(k) \to \infty$, the result reduces to $B_i \geq w \max_k \{\frac{2||f_{ref}||^2 \log(1/\delta)}{(\tau - \Delta_i^{new}(k))^2}\}$, the same as given in Theorem 3. The additional term shows that the higher dimension of features, the more injected noise, which would require more samples to achieve the same accuracy.

VII. Discussion

A. Improving Privacy-Accuracy Tradeoff

We now provide some intuitive explanation as to why the ideas presented in this paper work. We explored two key ideas to improve the privacy-accuracy tradeoff of a differentially private algorithm. The first is to accomplish the computational task by repeatedly using the already released differentially private outputs. Utilizing differential privacy's immunity to post-processing, this information recycling incurs no additional privacy loss. Since less information is revealed during computation, less perturbation is required to obtain the same privacy guarantee, which then improves the privacy-accuracy tradeoff. The second idea is to improve the the stability/robustness of the algorithm by directly controlling the penalty parameter. This allows the algorithm to accommodate more noise to improve privacy without sacrificing too much accuracy, which improves the privacy-accuracy tradeoff.

B. Other Perturbation Methods and Privacy Analysis Tools

While we have primarily used objective perturbation to make an algorithm differentially private and to calculate the privacy loss, it should be noted that this is done as an example to illustrate how MR-ADMM can outperform both R-ADMM and ADMM in the privacy-accuracy tradeoff. Other perturbation methods such as output perturbation to achieve differential privacy (each node perturbs its primal variable before broadcasting to its neighbors) can be used as well; our conclusion would still hold. This is because our key ideas (revealing less information and making the algorithm more robust/stable to noise via the penalty parameter) are orthogonal to the choice of the perturbation method.

Similarly, in our privacy analysis we have adopted the notion of pure ε-differential privacy to measure privacy. As a result, the bound on the total privacy loss can be fairly large. It is also possible to adopt a weaker notion, the (ε, δ)-differential privacy, to find a tighter bound on privacy loss by allowing the algorithm to violate ε-differential privacy with a small probability δ. In this case, the total privacy loss can be calculated using more advanced composition theorems such as moments accountant [31] and zero-concentrated differential privacy [32]. However, our key ideas (revealing less information and making the algorithm more robust/stable to noise via the penalty parameter) are orthogonal to the choice of the privacy definition and analysis tools used; thus the algorithmic properties will not be affected by such choices and the conclusion remains valid.

C. Privacy Analysis for a Broader Class of Optimizations

In Section V, the privacy property of the private MR-ADMM is analyzed for the ERM binary classification problem. This is so that we can easily compare with ADMM and M-ADMM in [21], [22]. This privacy analysis can be extended to more general forms of $O(f_i, D_i)$, such as multi-class settings. There have been extensive studies on the differentially private ERM with convex loss function [33], which can also be adopted for our framework.

VIII. Numerical Experiments

We use the *Adult* dataset from the UCI Machine Learning Repository [34]. It consists of personal information of around

48,842 individuals, including age, sex, race, education, occupation, income, etc. The goal is to predict whether the annual income of an individual is above \$50,000.

Following the same pre-processing steps as in [22], the final data includes 45,223 individuals, each represented as a 105-dimensional vector of norm at most 1. We then randomly partition this sample set into a training set (40,000 samples) and a testing set (5,223 samples). The training samples are then evenly distributed across nodes in a network.

We use as loss function the logistic loss $\mathscr{L}(z) = \log(1 + \exp(-z))$, with $|\mathscr{L}'| \leq 1$ and $\mathscr{L}'' \leq c_1 = \frac{1}{4}$. The regularizer is $R(f_i) = \frac{1}{2}||f_i||_2^2$. We measure the accuracy of the algorithm by the average loss over the training set:

$$L(t) := \frac{1}{N}\sum_{i=1}^{N}\frac{1}{B_i}\sum_{n=1}^{B_i}\mathscr{L}(y_i^n f_i(t)^T x_i^n),$$

and the classification error rate over the testing set $\mathcal{S}_{test}$:

$$E = \frac{\sum_{(x_j,y_j)\in\mathcal{S}_{test}} \mathbf{1}(y_j \neq \hat{y}_j)}{\sum_{(x_j,y_j)\in\mathcal{S}_{test}} 1},$$

where $\hat{y}_j$ is the prediction of sample (x_j, y_j) by using the averaged classifier $\bar{f}(t) = \frac{1}{N}\sum_{i=1}^{N} f_i(t)$, and each $f_i(t)$ is the local classifier(primal variable) of node i after t iterations.

We measure the privacy of an algorithm by the upper bound:

$$P(t) := \max_{i\in\mathcal{N}}\{\sum_{k=1}^{\lfloor\frac{t+1}{2}\rfloor}\frac{2C}{B_i}(\frac{1.4c_1}{(\frac{\rho}{N}+2\eta_i(2k-1)V_i)}+\alpha_i(k))\}.$$

The smaller $L(t)$ and $P(t)$, the higher accuracy and stronger privacy guarantee.

A. Convergence of Non-Private R-ADMM & MR-ADMM

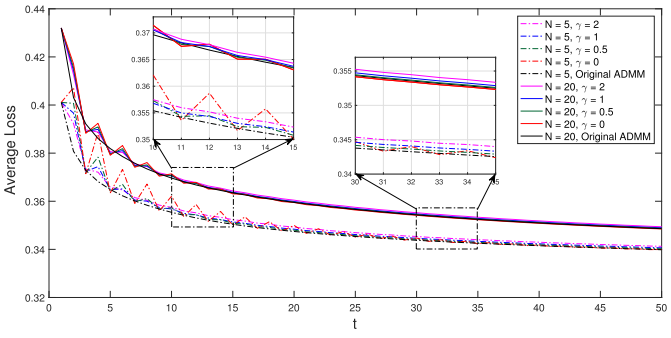
Fig. 1(a) shows the convergence of R-ADMM with different γ and fixed $\eta = 0.5$ for a small network ($N = 5$) and a large network ($N = 20$), both are randomly generated. Due to the linear approximation in even iterations, it's possible to cause an increased average loss as shown in the plot. However, the odd iterations will always compensate this increase; if we only look at the odd iterations, R-ADMM achieves a similar convergence rate as conventional ADMM. γ can also be thought of as an extra penalty parameter for each node in even iterations to punish its update, i.e., the difference between $f_i(2k)$ and $f_i(2k-1)$. Larger γ can result in smaller oscillation between even and odd iterations but will also lower the convergence rate.

Fig. 1(c)1(b) show the convergence of MR-ADMM with penalty parameters $\eta_i(2k-1)$ increasing at different speed. We see that increasing penalty slows down the convergence, and larger increase in $q_1(i)$ slows it down more. In 1(b), each node adopts different penalty parameter $\eta_i(2k-1)$ in each iteration while in 1(c), the same penalty parameter is shared among all the nodes. The convergence is attained in both cases.

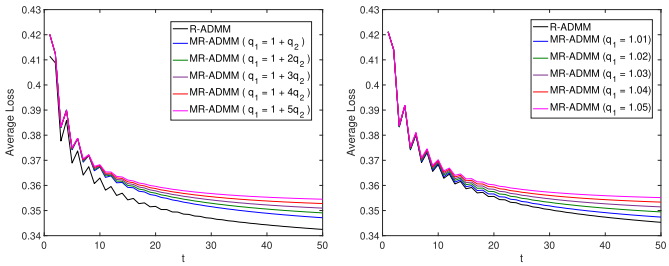
B. Private R-ADMM & MR-ADMM

1) The Effect of ρ, γ, $\eta_i(2k-1)$: We next inspect the accuracy and privacy of the private R-ADMM and MR-ADMM (Algorithm 3), and compare it with the private (conventional) ADMM using dual variable perturbation (DVP) [21], the private M-ADMM using penalty perturbation (PP) [22].

To begin, we first examine the effect of ρ in controlling overfitting. Fig. 2 shows the classification error rate over the testing set under different ρ, where the classifiers are trained with original ADMM and the algorithm runs for 50 iterations. Since the classification error rate is minimized at $\rho \approx 0.22$, we will use $\rho = 0.22$ in the following experiments.



(a) R-ADMM: $\eta = 0.5$



(b) $\eta_i(2k-1) = \hat{\eta}_i q_1(i)^k$ (c) $\eta_i(2k-1) = q_1^k$

Fig. 1. Convergence properties of R-ADMM and MR-ADMM: Fig. 1(a) illustrates the average loss over iterations of R-ADMM for the network of different sizes under fixed $\eta = 0.5$ and different γ. Dashed (resp. solid) curves represent the performance over a randomly generated small (resp. large) network with $N = 5$ (resp. $N = 20$) nodes. Fig. 1(c)1(b) illustrate the average loss over iterations of MR-ADMM for a randomly generated network with $N = 5$ nodes. Black curve represents the R-ADMM where $\eta_i(t) = \eta = 1$ is fixed for all nodes and all iterations. Each colored curve represents MR-ADMM with $\eta_i(2k-1)$ increasing over iterations at different speed. In Fig. 1(b), each node i adopts $\eta_i(2k-1) = \eta_i q_1(i)^k$ as penalty parameter in $2k-1$-th iteration, where $[\eta_1, \cdots, \eta_5] = [1, 1.03, 1.02, 0.8, 1.01]$, $q_1 = [q_1(1), \cdots, q_1(5)] = \mathbf{1} + kq_2$ (each $k \in \{1, \cdots, 5\}$ corresponds to one curve in plot) and $q_2 = [q_2(1), \cdots, q_2(5)] = [0.01, 0.005, 0.003, 0.015, 0.01]$. In Fig. 1(c), each node adopts the same penalty parameter $\eta_i(2k-1) = q_1^k$ in odd iterations.

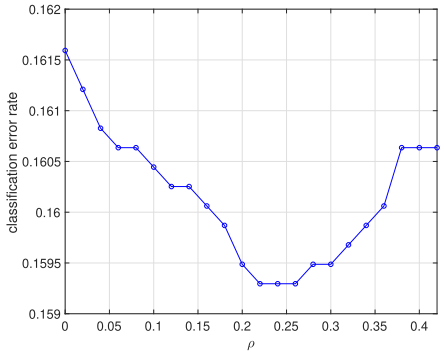
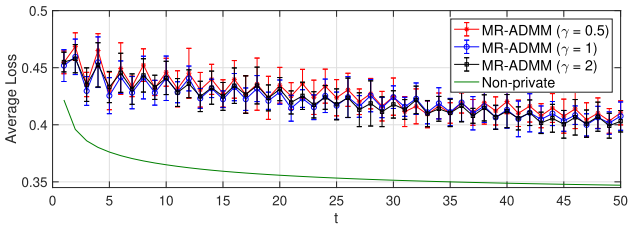
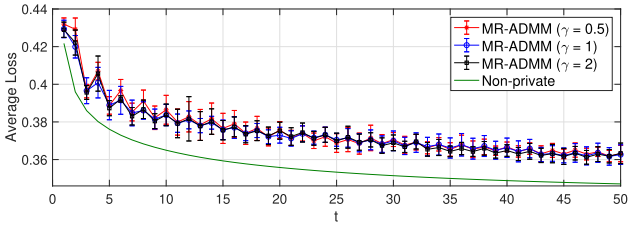


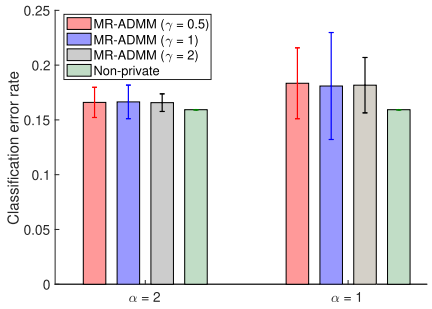
Fig. 2. The effect of ρ, fixing $C = 1750$.



(a) Accuracy comparison for different γ ($\alpha = 1$)



(b) Accuracy comparison for different γ ($\alpha = 2$)

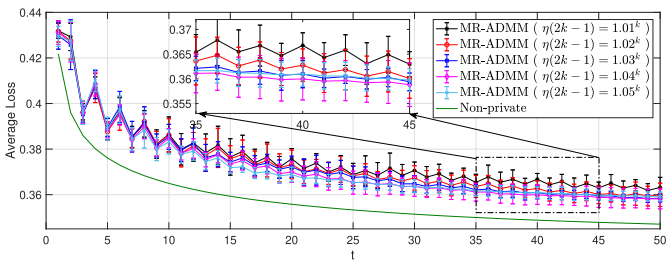


(c) Classification error rate comparison

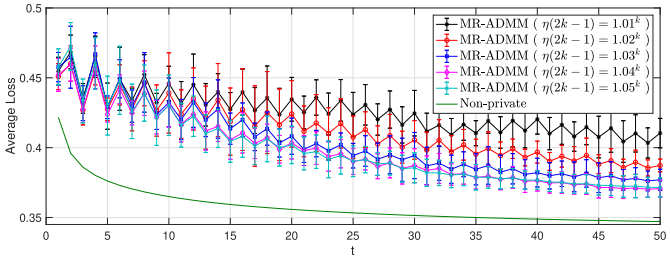
Fig. 3. The effect of γ on the performance of MR-ADMM, fixing $\eta_i(2k-1) = 1.01^k$: in Fig. 3(a)3(b), green curves represent the non-private conventional ADMM while other curves represent the private MR-ADMM with different γ and each of them illustrates the overall result summarized from 10 independent runs of experiments under the same parameter. The corresponding classification error rates are shown in Fig. 3(c). It shows that varying γ within a certain range doesn't effect the performance significantly.

For simplicity of presentation, in the next set of experiments the penalty $\eta_i(t) = \eta(t)$ in both M-ADMM and MR-ADMM and noise $\alpha_i(k) = \alpha, \forall i, k$. We observe similar results when $\alpha_i(k)$, $\eta_i(t)$ vary from node to node.

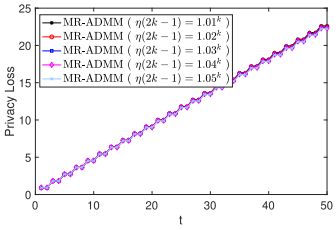
For each parameter setting, we perform 10 independent runs of the algorithm, and record both the mean and the range of their accuracy. Specifically, $L^l(t)$ denotes the average loss over the training dataset in the t-th iteration of the l-th experiment ($1 \le l \le 10$). The mean of average loss is given by $L_{mean}(t) = \frac{1}{10}\sum_{l=1}^{10} L^l(t)$ and the range $L_{range}(t) = \max_{1\le l\le 10} L^l(t) - \min_{1\le l\le 10} L^l(t)$. The larger the range $L_{range}(t)$ the less stable the algorithm, i.e., under the same parameter setting, the difference in performances (convergence curves) of two experiments is larger. In the next few plots, $L_{range}(t)$ is shown as the size of a vertical bar centered at $L_{mean}(t)$. Similarly, let E^l be the classification error rate over the testing set in the l-th experiment, with an average error rate $E_{mean} = \frac{1}{10}\sum_{l=1}^{10} E^l$ and range $E_{range} = \max_{1\le l\le 10} E^l - \min_{1\le l\le 10} E^l$ shown as the size of a vertical bar centered at E_{mean}. Each parameter setting also has a corresponding upper bound on the privacy loss denoted by $P(t)$.



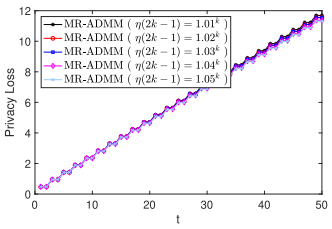
(a) Accuracy comparison for different $\eta(2k-1)$ ($\alpha = 2$)



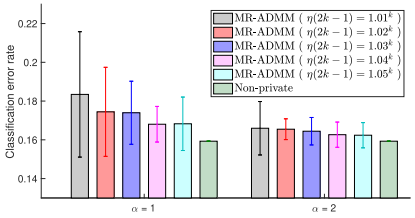
(b) Accuracy comparison for different $\eta(2k-1)$ ($\alpha = 1$)



(c) Privacy comparison ($\alpha = 2$)



(d) Privacy comparison ($\alpha = 1$)

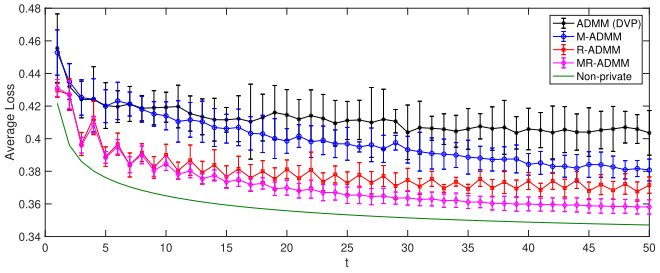


(e) Classification error rate comparison

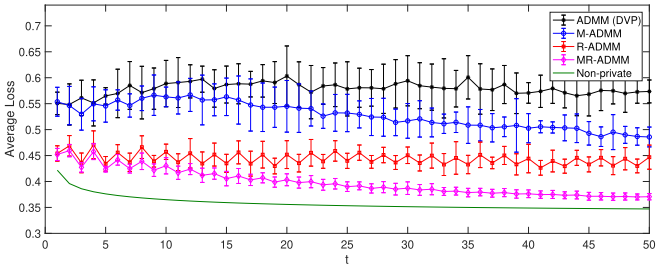
Fig. 4. The effect of $\eta_i(2k-1)$ on the performance of MR-ADMM, fixing $\gamma = 0.5$: in Fig. 4(a)4(b), green curves represent the non-private conventional ADMM while other curves represent the private MR-ADMM with different $\eta_i(2k-1) = q_1^k$ ($q_1 = 1.01, 1.02, 1.03, 1.04, 1.05$) and each of them illustrates the overall result summarized from 10 independent runs of experiments under the same parameter. Fig. 4(c)4(d) illustrate the upper bound of their privacy loss and the corresponding classification error rates are shown in Fig. 4(e).

In the non-private case, γ controls the oscillation between even and odd iterations, as well as the convergence rate. We now examine its effect when MR-ADMM is perturbed. Fig. 3 shows the average loss over the training set (Fig. 3(a)3(b)) and the classification error rate over the testing set (Fig. 3(c)) under different $\gamma > 0$, noting that the corresponding privacy loss of these cases are the same under the same α. It shows that varying γ (within a certain range) does not effect performance significantly. For the next set of experiments, we fix $\gamma = 0.5$.

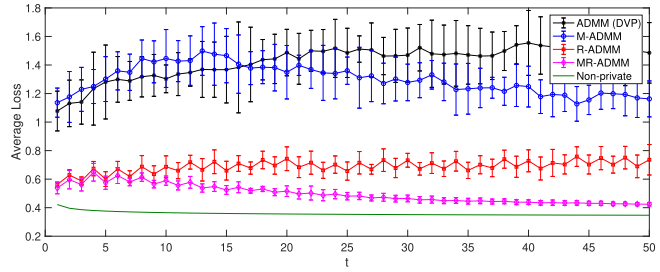
The effect of $\eta_i(2k-1)$ on the performance of private MR-ADMM is illustrated in Fig. 4, where the pair Fig. 4(a), 4(c) is for the case when noise parameter is $\alpha = 2$ (low privacy requirement) and the pair Fig. 4(b), 4(d) is for



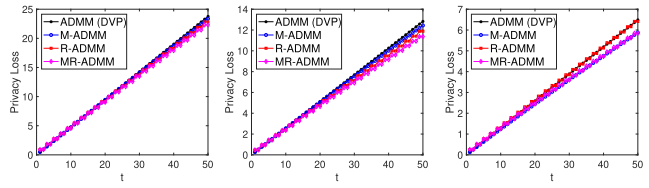
(a) Accuracy comparison ($\alpha = 2$)



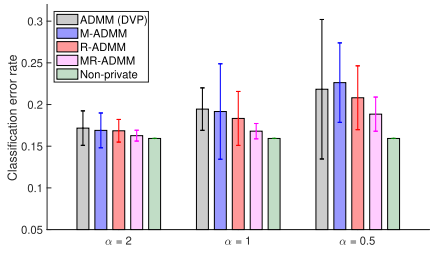
(b) Accuracy comparison ($\alpha = 1$)



(c) Accuracy comparison ($\alpha = 0.5$)



(d) $\alpha = 2$ (e) $\alpha = 1$ (f) $\alpha = 0.5$



(g) Classification error rate comparison

Fig. 5. Performance comparison: in Fig. 5(a)5(b)5(c), green curves represent the non-private conventional ADMM while other curves represent different private algorithms and each of them illustrates the overall result summarized from 10 independent runs of experiments under the same parameter. M-ADMM (blue) and MR-ADMM (magenta) adopt the varied penalty parameter while ADMM (black) and R-ADMM (red) adopt the fixed $\eta_i(t) = \eta = 1$. Fig. 5(d)5(e)5(f) illustrate the upper bound of their privacy loss and the corresponding classification error rates are shown in Fig. 5(g).

the case when $\alpha = 1$ (high privacy requirement). Although increasing $\eta_i(2k-1)$ over time can decrease the convergence rate of non-private MR-ADMM (Fig. 1(c)1(b)), it helps to stabilize the algorithm when MR-ADMM is perturbed and can improve the accuracy while maintain the privacy guarantee. Moreover, the improvement is more significant when algorithm is under higher perturbation (high privacy requirement) and when $\eta_i(2k-1)$ increases faster (within a range).

2) Performance Comparison Among Different Algorithms: Our last set of experiments is conducted to compare the performance of different algorithms with results illustrated in Fig. 5. The noise parameters of both MR-ADMM and R-ADMM are set as α shown in the plots, and the noise parameters of conventional ADMM and M-ADMM are chosen respectively such that they have approximately the same total privacy loss bounds. We set $\eta_i(2k-1) = 1.04^k$ in MR-ADMM. We see that both private R-ADMM (red) and private MR-ADMM (magenta) outperform private ADMM (black) and M-ADMM (blue) with higher accuracy and lower privacy loss. In particular, the private MR-ADMM (magenta) has the highest accuracy with the lowest privacy loss among all algorithms; the improvement is more significant with smaller total privacy loss. This improvement is also illustrated by the classification error rate over the testing set in Fig. 5(g).

IX. Conclusion

In this work, we presented Recycled ADMM (R-ADMM), a modified version of ADMM that can improve the privacy-utility tradeoff significantly with less computation. The idea is to repeatedly use the existing computational results instead of the original individuals' data to make updates. We also modify R-ADMM (MR-ADMM) by incorporating the idea from [22] to further improve the privacy-utility tradeoff of R-ADMM. The idea is to stabilize algorithm by decreasing its step-size, i.e., increasing penalty parameters, over iterations. A sufficient condition for the convergence and the privacy analysis using objective perturbation of two algorithms are established. The experiments on real-world dataset also validate the algorithm.

References

[1] X. Zhang, M. M. Khalili, and M. Liu, "Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2018, pp. 959–965.

[2] I. Vakilinia, D. K. Tosh, and S. Sengupta, "Privacy-preserving cybersecurity information exchange mechanism," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst. (SPECTS)*, Jul. 2017, pp. 1–7.

[3] M. M. Khalili, X. Zhang, and M. Liu, "Contract design for purchasing private data using a biased differentially private algorithm," in *Proc. 14th Workshop Econ. Netw., Syst. Comput.*, 2019, pp. 4:1–4:6.

[4] I. Vakilinia, J. Xin, M. Li, and L. Guo, "Privacy-preserving data aggregation over incomplete data for crowdsensing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[5] X. Zhang, C. Huang, M. Liu, A. Stefanopoulou, and T. Ersal, "Predictive cruise control with private vehicle-to-vehicle communication for improving fuel consumption and emissions," *IEEE Commun. Mag.*, vol. 57, no. 10, pp. 91–97, Oct. 2019. doi: 10.1109/MCOM.001.1900146.

[6] C. Dwork, "Differential privacy," in *Proc. 33rd Int. Conf. Automata, Lang. Program. (ICALP)*. Berlin, Germany: Springer-Verlag, 2006, pp. 1–12.

[7] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[8] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1291–1306, Jun. 2011.

[9] S. Gade and N. H. Vaidya, "Private optimization on networks," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 1402–1409.

[10] Z. Xu, G. Taylor, H. Li, M. A. Figueiredo, X. Yuan, and T. Goldstein, "Adaptive consensus ADMM for distributed optimization," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3841–3850.

[11] Z. Xu, M. A. Figueiredo, and T. Goldstein, "Adaptive ADMM with spectral penalty parameter selection," 2016, *arXiv:1605.07246*. [Online]. Available: https://arxiv.org/abs/1605.07246

[12] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. IEEE 51st Annu. Conf. Decis. Control (CDC)*, Dec. 2012, pp. 5445–5450.

[13] Q. Ling and A. Ribeiro, "Decentralized linearized alternating direction method of multipliers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 5447–5451.

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

[15] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1701–1709.

[16] Q. Ling, Y. Liu, W. Shi, and Z. Tian, "Weighted ADMM for fast decentralized network optimization," *IEEE Trans. Signal Process.*, vol. 64, no. 22, pp. 5930–5942, Aug. 2016.

[17] M. T. Hale and M. Egerstedty, "Differentially private cloud-based multi-agent optimization with constraints," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 1235–1240.

[18] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. Int. Conf. Distrib. Comput. Netw.*, 2015, p. 4.

[19] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 50–64, Jan. 2017.

[20] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, "Fast and differentially private algorithms for decentralized collaborative machine learning," Ph.D. dissertation, INRIA, Lille, France, 2017.

[21] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 172–187, Jan. 2017.

[22] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 5796–5805.

[23] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Security*, to be published.

[24] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011.

[25] X. Zhang, M. M. Khalili, and M. Liu, "Recycled ADMM: Improving the privacy and accuracy of distributed algorithms," 2019, *arXiv:1910.04581*. [Online]. Available: https://arxiv.org/abs/1910.04581

[26] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[27] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1663–1707, Jan. 2010.

[28] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "Decentralized quadratically approximated alternating direction method of multipliers," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2015, pp. 795–799.

[29] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 4051–4064, Aug. 2015.

[30] J. Kelner. (2007). *An Algorithmist's Toolkit*. [Online]. Available: http://bit.ly/2C4yRCX

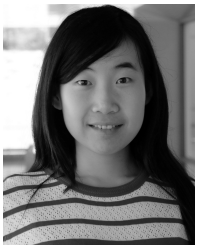
[31] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[32] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2016, pp. 635–658.

[33] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2722–2731.

[34] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[35] K. Sridharan, S. Shalev-Shwartz, and N. Srebro, "Fast rates for regularized objectives," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1545–1552.



Xueru Zhang received the B.Eng. degree in electronic and information engineering from Beihang University (BUAA), Beijing, China, in 2015, and the M.S. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, in 2016. She is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Michigan. Her research interests include fairness and privacy in machine learning, distributed optimization, and sequential decision making.

Mohammad Mahdi Khalili received the B.S. and M.S. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2013 and 2015, and the M.S. degree in applied mathematics from the University of Michigan, Ann Arbor, in 2018, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include fairness in machine learning and the applications of mathematical economics in network security and privacy.

Mingyan Liu (M'00–SM'11–F'14) received the Ph.D. degree in electrical engineering from the University of Maryland, College Park. She is currently a Professor and the Peter and Evelyn Fuss Chair of electrical and computer engineering with the University of Michigan, Ann Arbor. Her interests are in sequential decision and learning theory, game theory, and incentive mechanisms, with applications to large-scale networked systems. She is also a member of the ACM.