Recycled ADMM: Improve Privacy and Accuracy with Less Computation in Distributed Algorithms

Xueru Zhang, Mohammad Mahdi Khalili, Mingyan Liu

***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-utility tradeoff. In this study 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. We obtain a sufficient condition for the convergence of R-ADMM and provide the privacy analysis based on objective perturbation.**

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. 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 [1].

Existing approaches to decentralizing the above problem primarily consist of subgradient-based algorithms [2]–[4] and ADMM-based algorithms [5]–[12]. 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 [8]. 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 [13]–[16], 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. To the best of our knowledge, only [17], [18] apply differential privacy to ADMM. In particular, [17] proposed 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. In [18] we address this issue by inspecting the total privacy loss over the entire process and the whole network; we proposed a penalty perturbation method which improves the privacy-utility tradeoff significantly.

In the present study we present Recycled ADMM (R-ADMM), a modified version of ADMM where the privacy leakage only happens during half of the updates. Specifically, we adopt a linearized approximated optimization in every even iteration, whose solution can actually be calculated directly from results in the previous, odd iteration, and is used for updating primal variable. We establish a sufficient condition for convergence and provide a privacy analysis using the objective perturbation method. Our numerical results show that the privacy-utility tradeoff can be improved significantly.

The remainder of the paper is organized as follows. We present problem formulation and definition of differential privacy and ADMM in Section II and the Recycled ADMM algorithm along with its convergence analysis in Section III. A private version of this ADMM algorithm is then introduced in Section IV and numerical results in Section V. Section VI concludes the paper.

II. Preliminaries

A. Problem Formulation

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

This work is supported by the NSF under grants CNS-1422211, CNS-1646019, CNS-1739517.

X. Zhang, M. Khalili and M. Liu are with the Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105, {xueru, khalili, mingyan}@umich.edu

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

Consider the regularized empirical risk minimization (ERM) problem for binary classification defined as follows:

$$\min_{f_c} O_{ERM}(f_c, D_{all}) = \sum_{i=1}^{N} \frac{C}{B_i} \sum_{n=1}^{B_i} \mathscr{L}(y_i^n f_c^T x_i^n) + \rho R(f_c) \tag{1}$$

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

B. Differential Privacy [1]

A randomized algorithm $\mathscr{A}(\cdot)$ 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 $S \subseteq \text{range}(\mathscr{A})$, $\Pr(\mathscr{A}(D) \in S) \leq e^{\varepsilon} \Pr(\mathscr{A}(\hat{D}) \in S)$ holds. We call two datasets differing in at most one data point as neighboring datasets. 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.

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 $\{w_{ij} | i \in \mathscr{N}, j \in \mathscr{V}_i\}$ are introduced for every pair of connected nodes. As a result, (1) is reformulated equivalently as:

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

where $O(f_i, D_i) = \frac{C}{B_i} \sum_{n=1}^{B_i} \mathscr{L}(y_i^n f_i^T x_i^n) + \frac{\rho}{N} R(f_i)$. $\{f_i\}$ (resp. $\{w_{ij}\}$) is the shorthand for $\{f_i\}_{i \in \mathscr{N}}$ (resp. $\{w_{ij}\}_{i \in \mathscr{N}, j \in \mathscr{V}_i}$). Let $\{w_{ij}, \lambda_{ij}^k\}$ be the shorthand for $\{w_{ij}, \lambda_{ij}^k\}_{i \in \mathscr{N}, j \in \mathscr{V}_i, k \in \{a,b\}}$, 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:

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

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 [19], 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 [18] 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 [17] & Private M-ADMM [18]

In private ADMM [17], the 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, both methods cannot balance the privacy-utility tradeoff very well if consider the total privacy loss. In [18] 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-utility tradeoff. Specifically, it explores the rule of step-size (penalty parameter) in stabilizing the algorithm. M-ADMM allows each node to independently determine its penalty parameter; by perturbing the algorithm with noise correlated to penalty parameter and at the same time increasing the penalty parameters, the privacy and accuracy are shown to improve simultaneously.

E. Main idea

Fundamentally, the accumulation of privacy loss over iterations stems from the fact that the raw data is used in every primal update. If the updates can be made without using the raw 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 be reduced significantly. Based on this idea, we start with modifying ADMM such that we can repeatedly use some computational results to make updates.

III. Recycled ADMM (R-ADMM)

A. 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 [9], [20], [21], 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, raw 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 the raw data, the primal variable is updated solely based 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 will be referred to as Recycled ADMM (R-ADMM).

Specifically, in the $2k$-th (even) iteration, we approximate $O(f_i, D_i)$ (Eqn. (8), primal update optimization) 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 update only the primal variables. Using the first-order condition, the updates in the $2k$-th iteration become:

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

$$\lambda_i(2k) = \lambda_i(2k-1) \; . \quad (11)$$

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

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

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

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 can 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 and the gradient of the objective function is corrected by the primal difference and dual variable. 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 $\frac{\eta}{2} \sum_{j \in \mathscr{V}_i} (f_i(2k-1) - f_j(2k-1))$;
    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 $\frac{\eta}{2} \sum_{j \in \mathscr{V}_i} (f_i(2k-1) - f_j(2k-1))$ 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. Convergence Analysis

We next show that R-ADMM (Eqn. (10)-(13)) 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 ERM 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:

$$||\nabla O(f_i^1, D_i) - \nabla O(f_i^2, D_i)||_2 \leq M_i ||f_i^1 - f_i^2||_2 \quad (14)$$

By the KKT condition of the primal update (12):

$$0 = \nabla O(f_i(2k-1), D_i) + 2\lambda_i(2k-2) + \eta \sum_{j \in \mathscr{V}_i} (2f_i(2k-1) - (f_i(2k-2) + f_j(2k-2))) \; . \quad (15)$$

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 $i \in \mathscr{N}$ 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 $V_i = |\mathscr{V}_i|$ be the number of neighbors of node i, and define the degree matrix $D = \mathbf{diag}([V_1; V_2; \cdots; V_N]) \in$

$\mathbb{R}^{N\times N}$ and the diagonal matrix $\tilde{D}$ with $\tilde{D}_{ii} = 2\eta V_i + \gamma$. Then for each k, the matrix form of (10)(11)(15)(13) are:

$$\hat{f}(2k) = \hat{f}(2k-1) - \tilde{D}^{-1}\{\nabla\hat{O}(\hat{f}(2k-1), D_{all}) + 2\Lambda(2k-1) + \eta(D-A)\hat{f}(2k-1)\} \; ; \quad (16)$$

$$2\Lambda(2k) = 2\Lambda(2k-1) \; ; \quad (17)$$

$$\mathbf{0}_{N\times d} = \nabla\hat{O}(\hat{f}(2k-1), D_{all}) + 2\Lambda(2k-2) + 2\eta D\hat{f}(2k-1) - \eta(D+A)\hat{f}(2k-2) \; ; \quad (18)$$

$$2\Lambda(2k-1) = 2\Lambda(2k-2) + \eta(D-A)\hat{f}(2k-1) \; . \quad (19)$$

Writing $\hat{f}(2k-2)$ and $\Lambda(2k-2)$ in (18)(19) as functions of $\hat{f}(2k-3)$, $\Lambda(2k-3)$ using (16)(17), we obtain:

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

$$2\Lambda(2k-1) = 2\Lambda(2k-3) + \eta(D-A)\hat{f}(2k-1) \; .$$

The convergence of R-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 $t+1$:

$$\nabla\hat{O}(\hat{f}(t+1), D_{all}) + \eta(D+A)\tilde{D}^{-1}\nabla\hat{O}(\hat{f}(t), D_{all}) + \eta(D+A)((\hat{f}(t+1) - \hat{f}(t)) + \tilde{D}^{-1}\eta(D-A)\hat{f}(t)) + 2\Lambda(t+1) + \eta(D+A)\tilde{D}^{-1}2\Lambda(t) = \mathbf{0}_{N\times d} \; ; \quad (20)$$

$$2\Lambda(t+1) = 2\Lambda(t) + \eta(D-A)\hat{f}(t+1) \; . \quad (21)$$

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) 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 [22].

Lemma III.1. *[First-order Optimality Condition [12]] 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*

$$\nabla\hat{O}(\hat{f}^*, D_{all}) + 2\Lambda^* = \mathbf{0}_{N\times d} \; ; \quad (22)$$

$$(D-A)\hat{f}^* = \mathbf{0}_{N\times d} \; . \quad (23)$$

Lemma III.1 shows that a pair $(\hat{f}^*, \Lambda^*)$ satisfying (22)(23) is equivalent to the optimal solution of our problem, hence the convergence of R-ADMM is proved by showing that $(\hat{f}(t), \Lambda(t))$ in (20)(21) converges to a pair $(\hat{f}^*, \Lambda^*)$ satisfying (22)(23).

Theorem III.1. ***[Sufficient Condition]*** *Consider the modified ADMM defined by* (20)(21). *Let* $\{\hat{f}(t), \Lambda(t)\}$ *be outputs in each iteration and* $\{\hat{f}^*, \Lambda^*\}$ *a pair satisfying* (22)(23). *Denote* $D_M = \mathbf{diag}([M_1^2; M_2^2; \cdots; M_N^2]) \in \mathbb{R}^{N\times N}$ *with* $0 < M_i < +\infty$ *as given in Assumption 3. If the following two conditions hold for some constants* $L > 0$ *and* $\mu > 1$*:*

$$(I + \eta(D+A)\tilde{D}^{-1}) \succ \frac{L\mu}{2\sigma_{\min}(\tilde{D})}\frac{1}{\eta}D_M(D-A)^+ \; ; \quad (24)$$

$$\eta(D+A) \succ \{\eta(D+A)\tilde{D}^{-1}\eta(D-A) + \frac{2}{L}\eta(D+A)\tilde{D}^{-1}\eta(D+A) + \frac{L\mu}{2\sigma_{\min}(\tilde{D})(\mu-1)}D_M\} \; . \quad (25)$$

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

Proof. See Appendix I. □

By controlling γ, it is easy to find constants $L > 0$ and $\mu > 1$ such that conditions (24)(25) are satisfied, and they are not unique. One example is $L = 2$ and $\mu = 2$, in which case (24)(25) are reduced to:

$$(I + \eta(D+A)\tilde{D}^{-1}) \succ \frac{4}{2\sigma_{\min}(\tilde{D})}\frac{1}{\eta}D_M(D-A)^+ \; ; \quad (26)$$

$$\eta(D+A) \succ 2\eta(D+A)\tilde{D}^{-1}\eta D + \frac{2}{\sigma_{\min}(\tilde{D})}D_M \; . \quad (27)$$

(26)(27) can be easily satisfied for sufficiently large $\gamma \geq 0$. Note that the conditions are sufficient but not necessary, so in practice convergence may be attained under weaker settings.

IV. Private R-ADMM

In this section we present a privacy preserving version of R-ADMM. In odd iterations, we adopt the objective perturbation [23] where a random linear term $\epsilon_i(2k-1)^T f_i$ is added to the objective function in (12)[2] [3], where $\epsilon_i(2k-1)$ follows the probability density proportional to $\exp\{-\alpha_i(k)||\epsilon_i(2k-1)||_2\}$ and is stored.

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

To generate this noisy vector, 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 is obtained by finding the optimal solution to the private objective function:

$$f_i(2k-1) = \underset{f_i}{\text{argmin}}\ L_i^{priv}(2k-1), \quad i \in \mathscr{N} \; . \quad (28)$$

In even iterations, use the stored gradient $\nabla O(f_i(2k-1), D_i)$, primal difference $\frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$

[2]Other perturbation methods can also be adopted such as output perturbation, random sampling, etc.

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

and noise $\epsilon_i(2k-1)$ to update primal variables:

$$f_i(2k) = f_i(2k-1) - \frac{1}{2\eta V_i + \gamma}\{2\lambda_i(2k-1) + \underbrace{\epsilon_i(2k-1) + \nabla O(f_i(2k-1), D_i)}_{\text{the existing stored information}} + \eta \underbrace{\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))\}}_{\text{the existing stored information}} \quad (29)$$

Algorithm 2 shows the complete procedure, where the condition used to generate η helps to bound the worst-case privacy loss but is not necessary in guaranteeing convergence.

Algorithm 2: Private R-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: Select η s.t. $2c_1 < \min_i\{\frac{B_i}{C}(\frac{\rho}{N} + 2\eta V_i)\}$
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 (28);
    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 $\frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$;
    Update dual variable $\lambda_i(2k-1)$ via (13).
  for $i = 1$ **to** $\mathscr{N}$ **do**
    Use the stored $\epsilon_i(2k-1)$, $\nabla O(f_i(2k-1), D_i)$ and $\frac{\eta}{2}\sum_{j\in\mathscr{V}_i}(f_i(2k-1) - f_j(2k-1))$ to update primal variable $f_i(2k)$ via (29);
    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 $\{\lambda_i(2K)\}_{i=1}^N$

In the 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. For this reason, we adopt the differential privacy definition proposed in [18] as follows.

Definition IV.1. *Consider a connected network $G(\mathscr{N}, \mathscr{E})$ with a set of nodes $\mathscr{N} = \{1, 2, \cdots, N\}$. 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:*

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

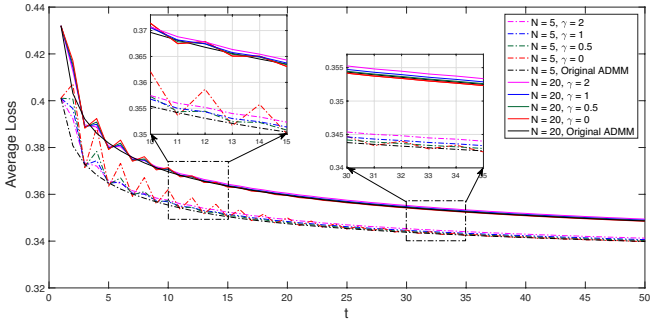


Fig. 1. Convergence properties of R-ADMM.

We now state another result of this paper, on the privacy property of the private R-ADMM (Algorithm 2) using the above definition. Additional assumptions on $\mathscr{L}(\cdot)$ and $R(\cdot)$ are used.

***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 IV.1. *Consider the private R-ADMM (Algorithm 2), $\forall k = 1, \cdots K$, 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 II. □

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

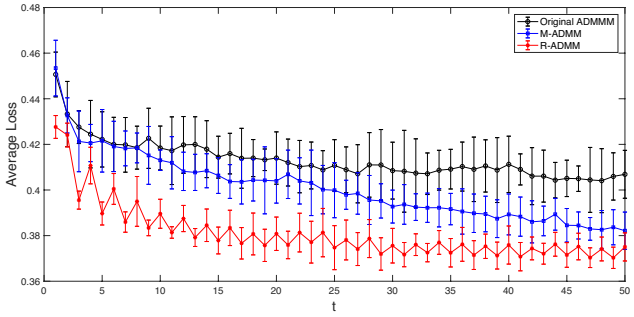
$$\beta \geq \max_{i\in\mathscr{N}}\{\sum_{k=1}^K \frac{2C}{B_i}(\frac{1.4c_1}{(\frac{\rho}{N} + 2\eta V_i)} + \alpha_i(k))\} . \quad (30)$$

Proof. See Appendix III. □

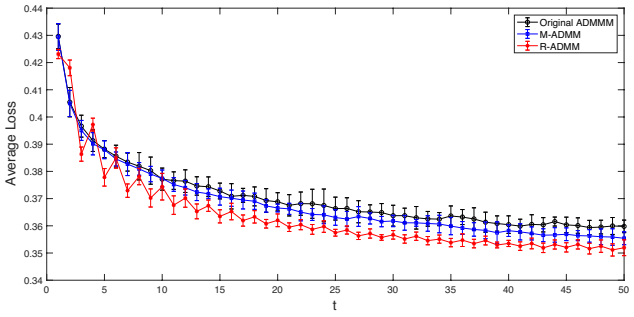
V. Numerical Experiments

We use the *Adult* dataset from the UCI Machine Learning Repository [24]. 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.

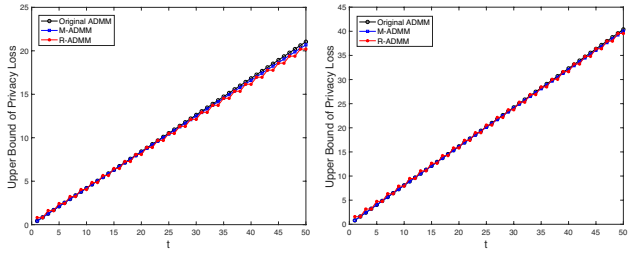
Following the same pre-processing steps as in [18], the final data includes 45,223 individuals, each represented as a 105-dimensional vector of norm at most 1. We will use as loss function the logistic loss $\mathscr{L}(z) = \log(1 + \exp(-z))$, with $|\mathscr{L}'| \leq 1$ and $\mathscr{L}'' \leq c_1 = \frac{1}{4}$. The regularizer is $R(f_i) = \frac{1}{2}||f_i||_2^2$. We will measure the accuracy of the algorithm by the average loss $L(t) := \frac{1}{N}\sum_{i=1}^N \frac{1}{B_i}\sum_{n=1}^{B_i}\mathscr{L}(y_i^n f_i(t)^T x_i^n)$ over the training set. We will measure the privacy of the algorithm by the upper bound



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



(b) Accuracy comparison: $\alpha = 4$



(c) Privacy comparison: $\alpha = 2$ (d) Privacy comparison: $\alpha = 4$

Fig. 2. Comparison of accuracy and privacy.

$P(t) := \max_{i\in\mathcal{N}}\{\sum_{k=1}^{K}\frac{2C}{B_i}(\frac{1.4c_1}{(\frac{\rho}{N}+2\eta V_i)}+\alpha_i(k))\}$. The smaller $L(t)$ and $P(t)$, the higher accuracy and stronger privacy guarantee.

A. Convergence of non-private R-ADMM

Figure 1 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.

B. Private R-ADMM

We next inspect the accuracy and privacy of the private R-ADMM (Algorithm 2) and compare it with the private (conventional) ADMM using dual variable perturbation (DVP) [17] and the private M-ADMM using penalty perturbation (PP) [18]. In the set of experiments, we fix $\gamma = 0.2$, $\eta = 1$ in private R-ADMM and set the noise parameter $\alpha_i(k) = \alpha, \forall i, k$. The noise parameters of conventional ADMM and M-ADMM are also chosen respectively such that they have almost the same total privacy loss bounds.

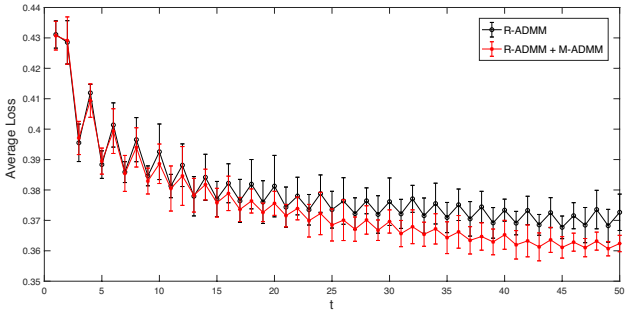


Fig. 3. Accuracy comparison: $\eta(t) = 1.01^t$, $\gamma(t) = 0.2 * 1.01^t$

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 then 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 every two experiments is larger. Each parameter setting also has a corresponding upper bound on the privacy loss denoted by $P(t)$. Figures 2(a)-2(b) show both $L_{mean}(t)$ and $L_{range}(t)$ as vertical bars centered at $L_{mean}(t)$. Their corresponding privacy upper bound is given in Figures 2(c)-2(d). The pair 2(a), 2(c) (resp. 2(b), 2(d)) is for the same parameter setting. We see that the private R-ADMM has higher accuracy than both the private ADMM and M-ADMM, and the improvement is more significant with the smaller total privacy loss.

We also incorporate the idea from [18] into private R-ADMM, where we decrease the step-size, i.e., increase η and γ, over iterations to stabilize the algorithm and improve the algorithmic performance. The result is shown in Figure 3 where the privacy loss bound is controlled to be the same during the whole period. It shows that by varying the step-size, the privacy-utility tradeoff can be further improved.

VI. Conclusion

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 raw data to make updates. We also established a sufficient condition for convergence and privacy analysis using objective perturbation.

$$\langle \hat{f}(t+1)-\hat{f}^*, -\eta(D+A)\tilde{D}^{-1}(\nabla\hat{O}(\hat{f}(t),D_{all})-\nabla\hat{O}(\hat{f}^*,D_{all}))+(I+\eta(D+A)\tilde{D}^{-1})(2\Lambda^*-2\Lambda(t+1))$$
$$+\eta(D+A)\tilde{D}^{-1}(2\Lambda(t+1)-2\Lambda(t))-\eta(D+A)(\hat{f}(t+1)-\hat{f}(t))-\eta(D+A)\tilde{D}^{-1}\eta(D-A)\hat{f}(t)\rangle_F \geq 0\ . \quad (31)$$

$$\begin{aligned}
&\langle \hat{f}(t+1)-\hat{f}^*, \eta(D+A)\tilde{D}^{-1}(2\Lambda(t+1)-2\Lambda(t))-\eta(D+A)\tilde{D}^{-1}\eta(D-A)\hat{f}(t)\rangle_F \\
&= \langle \hat{f}(t+1)-\hat{f}^*, \eta(D+A)\tilde{D}^{-1}\eta(D-A)(\hat{f}(t+1)-\hat{f}(t))\rangle_F \quad (32)\\
&= \frac{1}{2}||\hat{f}(t+1)-\hat{f}^*||^2_{G_1}+\frac{1}{2}||\hat{f}(t+1)-\hat{f}(t)||^2_{G_1}-\frac{1}{2}||\hat{f}(t)-\hat{f}^*||^2_{G_1}\ ; \\
&\langle \hat{f}(t+1)-\hat{f}^*, (I+\eta(D+A)\tilde{D}^{-1})(2\Lambda^*-2\Lambda(t+1))\rangle_F \\
&= \langle \frac{1}{\eta}(D-A)^+(2\Lambda(t+1)-2\Lambda(t)), (I+\eta(D+A)\tilde{D}^{-1})(2\Lambda^*-2\Lambda(t+1))\rangle_F \quad (33)\\
&= \frac{1}{2}||2\Lambda^*-2\Lambda(t)||^2_{G_2}-\frac{1}{2}||2\Lambda^*-2\Lambda(t+1)||^2_{G_2}-\frac{1}{2}||2\Lambda(t+1)-2\Lambda(t)||^2_{G_2}\ ; \\
&\langle \hat{f}(t+1)-\hat{f}^*, -\eta(D+A)(\hat{f}(t+1)-\hat{f}(t))\rangle_F \\
&= \frac{1}{2}||\hat{f}(t)-\hat{f}^*||^2_{\eta(D+A)}-\frac{1}{2}||\hat{f}(t+1)-\hat{f}^*||^2_{\eta(D+A)}-\frac{1}{2}||\hat{f}(t)-\hat{f}(t+1)||^2_{\eta(D+A)}\ . \quad (34)
\end{aligned}$$

$$\begin{aligned}
&\langle \hat{f}(t+1)-\hat{f}^*, -\eta(D+A)\tilde{D}^{-1}(\nabla\hat{O}(\hat{f}(t),D_{all})-\nabla\hat{O}(\hat{f}^*,D_{all}))\rangle_F \\
&= \langle \hat{f}(t+1)-\hat{f}(t)+\hat{f}(t)-\hat{f}^*, -\eta(D+A)\tilde{D}^{-1}(\nabla\hat{O}(\hat{f}(t),D_{all})-\nabla\hat{O}(\hat{f}^*,D_{all}))\rangle_F \\
&\leq \langle \hat{f}(t)-\hat{f}(t+1), \eta(D+A)\tilde{D}^{-1}(\nabla\hat{O}(\hat{f}(t),D_{all})-\nabla\hat{O}(\hat{f}^*,D_{all}))\rangle_F \\
&= \langle \eta(D+A)\sqrt{\tilde{D}^{-1}}(\hat{f}(t)-\hat{f}(t+1)), \sqrt{\tilde{D}^{-1}}(\nabla\hat{O}(\hat{f}(t),D_{all})-\nabla\hat{O}(\hat{f}^*,D_{all}))\rangle_F\ . \quad (35)
\end{aligned}$$

$$\begin{aligned}
(35) &\leq \frac{1}{L}||(\hat{f}(t)-\hat{f}(t+1))||^2_{\eta(D+A)\tilde{D}^{-1}\eta(D+A)}+\frac{L}{4\sigma_{\min}(\tilde{D})}(\mu||\hat{f}^*-\hat{f}(t+1)||^2_{D_M}+\frac{\mu}{\mu-1}||\hat{f}(t+1)-\hat{f}(t)||^2_{D_M}) \\
&= \frac{1}{2}||(\hat{f}(t)-\hat{f}(t+1))||^2_{\frac{2}{L}\eta(D+A)\tilde{D}^{-1}\eta(D+A)+\frac{L\mu}{2\sigma_{\min}(\tilde{D})(\mu-1)}D_M}+\frac{1}{2}||2\Lambda(t+1)-2\Lambda(t)||^2_{\frac{L\mu}{2\sigma_{\min}(\tilde{D})}(\frac{1}{\eta}(D-A)^+)^2D_M} \quad (36)
\end{aligned}$$

$$\begin{aligned}
&\frac{1}{2}||\hat{f}(t)-\hat{f}(t+1)||^2_{\eta(D+A)-G_1}-\frac{1}{2}||(\hat{f}(t)-\hat{f}(t+1))||^2_{\frac{2}{L}\eta(D+A)\tilde{D}^{-1}\eta(D+A)+\frac{L\mu}{2\sigma_{\min}(\tilde{D})(\mu-1)}D_M} \\
&+\frac{1}{2}||2\Lambda(t+1)-2\Lambda(t)||^2_{G_2}-\frac{1}{2}||2\Lambda(t+1)-2\Lambda(t)||^2_{\frac{L\mu}{2\sigma_{\min}(\tilde{D})}(\frac{1}{\eta}(D-A)^+)^2D_M} \\
&\leq \frac{1}{2}||\hat{f}(t+1)-\hat{f}^*||^2_{G_1}-\frac{1}{2}||\hat{f}(t)-\hat{f}^*||^2_{G_1}+\frac{1}{2}||2\Lambda^*-2\Lambda(t)||^2_{G_2} \\
&-\frac{1}{2}||2\Lambda^*-2\Lambda(t+1)||^2_{G_2}+\frac{1}{2}||\hat{f}(t)-\hat{f}^*||^2_{\eta(D+A)}-\frac{1}{2}||\hat{f}(t+1)-\hat{f}^*||^2_{\eta(D+A)} \quad (37)
\end{aligned}$$

APPENDIX I
PROOF OF THEOREM III.1

By convexity of $O(f_i, D_i)$, $(f_i^1 - f_i^2)^T(\nabla O(f_i^1, D_i) - \nabla O(f_i^2, D_i)) \geq 0$ holds $\forall\ f_i^1, f_i^2$. Let $\langle\cdot,\cdot\rangle_F$ be frobenius inner product of two matrices, there is:

$$\langle \hat{f}(t+1)-\hat{f}^*, \nabla\hat{O}(\hat{f}(t+1),D_{all})-\nabla\hat{O}(\hat{f}^*,D_{all})\rangle_F \geq 0$$

According to (20)(22) and (21), substitute $\nabla\hat{O}(\hat{f}(t+1), D_{all}) - \nabla\hat{O}(\hat{f}^*, D_{all})$ and add an extra term $\eta(D+A)\tilde{D}^{-1}(\nabla\hat{O}(\hat{f}^*, D_{all}) + 2\Lambda^*) = \mathbf{0}_{N\times d}$, implies Eqn. (31).

To simplify the notation, for a matrix X, let $||X||^2_J = \langle X, JX\rangle_F$ and $(X)^+$ be the pseudo inverse of X. Define:

$$G_1 = \eta(D+A)\tilde{D}^{-1}\eta(D-A)\ ;$$
$$G_2 = \frac{1}{\eta}(D-A)^+(I+\eta(D+A)\tilde{D}^{-1})\ .$$

Use (21)(23) and the fact that $\langle A, JB\rangle_F = \langle J^TA, B\rangle_F$, Eqn. (32)(33)(34) hold. Let $\sqrt{X}$ denote the square root of a symmetric positive semi-definite (PSD) matrix X that is also symmetric PSD. Eqn. (35) holds, where the inequality uses the facts that $O(f_i, D_i)$ is convex for all i and that the matrix $\eta(D+A)\tilde{D}^{-1}$ is positive definite.

According to (14) in Assumption 3, define the matrix $D_M = \mathbf{diag}([M_1^2; M_2^2; \cdots; M_N^2]) \in \mathbb{R}^{N\times N}$, it implies $||\nabla\hat{O}(\hat{f}^1, D_{all}) - \nabla\hat{O}(\hat{f}^2, D_{all})||^2_F \leq \langle \hat{f}^1 - \hat{f}^2, D_M(\hat{f}^1 -$

$\hat{f}^2)\rangle_F$. Since $\langle A,B\rangle_F \le \frac{1}{L}||A||_F^2 + \frac{L}{4}||B||_F^2$ holds for any $L>0$, there is:

$$\begin{aligned}(35) &\le \frac{1}{L}||\eta(D+A)\sqrt{\tilde{D}^{-1}}(\hat{f}(t)-\hat{f}(t+1))||_F^2\\ &+\frac{L}{4}||\sqrt{\tilde{D}^{-1}}(\nabla\hat{O}(\hat{f}(t),D_{all})-\nabla\hat{O}(\hat{f}^*,D_{all}))||_F^2\\ &\le \frac{1}{L}||(\hat{f}(t)-\hat{f}(t+1))||^2_{\eta(D+A)\tilde{D}^{-1}\eta(D+A)}\\ &+\frac{L\sigma_{\max}(\tilde{D}^{-1})}{4}||\nabla\hat{O}(\hat{f}(t),D_{all})-\nabla\hat{O}(\hat{f}^*,D_{all})||_F^2\\ &= \frac{1}{L}||(\hat{f}(t)-\hat{f}(t+1))||^2_{\eta(D+A)\tilde{D}^{-1}\eta(D+A)}\\ &+\frac{L}{4\sigma_{\min}(\tilde{D})}||\hat{f}^*-\hat{f}(t)||^2_{D_M} \quad (38)\end{aligned}$$

where $\sigma_{\max}(\cdot)$, $\sigma_{\min}(\cdot)$ denote the largest and smallest singular value of a matrix respectively. Since for any $\mu>1$ and any matrices C_1, C_2, J with the same dimensions, there is $||C_1+C_2||_J^2 \le \mu||C_1||_J^2 + \frac{\mu}{\mu-1}||C_2||_J^2$. which implies:

$$\begin{aligned}||\hat{f}^*-\hat{f}(t)||^2_{D_M} &= ||\hat{f}^*-\hat{f}(t+1)+\hat{f}(t+1)-\hat{f}(t)||^2_{D_M}\\ &\le \mu||\hat{f}^*-\hat{f}(t+1)||^2_{D_M}+\frac{\mu}{\mu-1}||\hat{f}(t+1)-\hat{f}(t)||^2_{D_M}\end{aligned}$$

Plug into (38) and use (21)(23) gives Eqn. (36).

Combine (32)(33)(34)(36), (31) becomes Eqn. (37). Suppose the following two conditions hold for some constants $L>0$ and $\mu>1$:

$$(I+\eta(D+A)\tilde{D}^{-1}) \succ \frac{L\mu}{2\sigma_{\min}(\tilde{D})}\frac{1}{\eta}D_M(D-A)^+ \ ; \quad (39)$$

$$\begin{aligned}\eta(D+A) &\succ \eta(D+A)\tilde{D}^{-1}\eta(D-A)\\ +\frac{2}{L}\eta(D+A)\tilde{D}^{-1}\eta(D+A) &+ \frac{L\mu}{2\sigma_{\min}(\tilde{D})(\mu-1)}D_M \ . \quad (40)\end{aligned}$$

Substitute $G_1=\eta(D+A)\tilde{D}^{-1}\eta(D-A)$ and $G_2=\frac{1}{\eta}(D-A)^+(I+\eta(D+A)\tilde{D}^{-1})$, define R_1 and R_2 below gives:

$$\begin{aligned}R_1 &= \eta(D+A)-G_1-\frac{L\mu}{2\sigma_{\min}(\tilde{D})(\mu-1)}D_M\\ &-\frac{2}{L}\eta(D+A)\tilde{D}^{-1}\eta(D+A) \succ \mathbf{0}_{N\times N} \ ; \quad (41)\\ R_2 &= G_2-\frac{L\mu}{2\sigma_{\min}(\tilde{D})}(\frac{1}{\eta}(D-A)^+)^2 D_M \succ \mathbf{0}_{N\times N} \ . \quad (42)\end{aligned}$$

Eqn. (37) becomes:

$$\begin{aligned}&\frac{1}{2}||\hat{f}(t)-\hat{f}(t+1)||^2_{R_1}+\frac{1}{2}||2\Lambda(t+1)-2\Lambda(t)||^2_{R_2}\\ &\le \frac{1}{2}||\hat{f}(t+1)-\hat{f}^*||^2_{G_1}-\frac{1}{2}||\hat{f}(t)-\hat{f}^*||^2_{G_1}\\ &+\frac{1}{2}||2\Lambda^*-2\Lambda(t)||^2_{G_2}-\frac{1}{2}||2\Lambda^*-2\Lambda(t+1)||^2_{G_2}\\ &+\frac{1}{2}||\hat{f}(t)-\hat{f}^*||^2_{\eta(D+A)}-\frac{1}{2}||\hat{f}(t+1)-\hat{f}^*||^2_{\eta(D+A)} \quad (43)\end{aligned}$$

Sum up (43) over t from 0 to $+\infty$ leads to:

$$\begin{aligned}&\sum_{t=0}^{\infty}\{||\hat{f}(t)-\hat{f}(t+1)||^2_{R_1}+||2\Lambda(t+1)-2\Lambda(t)||^2_{R_2}\}\\ &\le ||\hat{f}(0)-\hat{f}^*||^2_{\eta(D+A)}-||\hat{f}(+\infty)-\hat{f}^*||^2_{\eta(D+A)}\\ &+||\hat{f}(\infty)-\hat{f}^*||^2_{G_1}-||\hat{f}(0)-\hat{f}^*||^2_{G_1}\\ &+||2\Lambda^*-2\Lambda(0)||^2_{G_2}-||2\Lambda^*-2\Lambda(\infty)||^2_{G_2} \quad (44)\end{aligned}$$

The RHS of (44) is finite, implies that $\lim_{t\to\infty}\{||\hat{f}(t)-\hat{f}(t+1)||^2_{R_1}+||2\Lambda(t+1)-2\Lambda(t)||^2_{R_2}\}=0$. Since R_1, R_2 are not unique, by (41)(42), it requires $\lim_{t\to\infty}||\hat{f}(t)-\hat{f}(t+1)||^2_{R_1}=0$ and $\lim_{t\to\infty}||2\Lambda(t+1)-2\Lambda(t)||^2_{R_2}=0$ should hold for all possible R_1, R_2. Therefore, $\lim_{t\to\infty}(\hat{f}(t)-\hat{f}(t+1))=\mathbf{0}_{N\times d}$ and $\lim_{t\to\infty}(2\Lambda(t+1)-2\Lambda(t))=\mathbf{0}_{N\times d}$ should hold. $(\hat{f}(t),\Lambda(t))$ converges to the stationary point $(\hat{f}^s,\Lambda^s)$. Now show that the stationary point $(\hat{f}^s,\Lambda^s)$ is the optimal point $(\hat{f}^*,\Lambda^*)$.

Take the limit of both sides of (20)(21) yield:

$$(I+\eta(D+A)\tilde{D}^{-1})(\nabla\hat{O}(\hat{f}^s,D_{all})+2\Lambda^s)=\mathbf{0}_{N\times d} \ ; \quad (45)$$

$$(D-A)\hat{f}^s=\mathbf{0}_{N\times d} \ . \quad (46)$$

Since $I+\eta(D+A)\tilde{D}^{-1} \succ \mathbf{0}_{N\times N}$, to satisfy (45), $\nabla\hat{O}(\hat{f}^s,D_{all})+2\Lambda^s=\mathbf{0}_{N\times d}$ must hold.

Compare with (22)(23) in Lemma IV.1 and observe that $(\hat{f}^s,\Lambda^s)$ satisfies the optimality condition and is thus the optimal point. Therefore, $(\hat{f}(t),\Lambda(t))$ converges to $(\hat{f}^*,\Lambda^*)$.

APPENDIX II
PROOF OF LEMMA IV.1

Consider the Private R-ADMM up to $2k$-th iteration. In $(2k-1)$-th iteration, the primal variable is updated via (28), By KKT condition:

$$\begin{aligned}&\nabla O(f_i(2k-1),D_i)+\epsilon_i(2k-1)=-2\lambda_i(2k-2)\\ &-\eta\sum_{j\in\mathcal{V}_i}(2f_i(2k-1)-f_i(2k-2)-f_j(2k-2)) \quad (47)\end{aligned}$$

Given $\{f_i(t)\}_{i=1}^N$ for $t\le 2k-2$, $\{\lambda_i(2k-2)\}_{i=1}^N$ are also given. RHS of (47) can be calculated completely after releasing $\{f_i(k-1)\}_{i=1}^N$, i.e., the information of $\nabla O(f_i(2k-1),D_i)+\epsilon_i(2k-1)$ is completely released during $(2k-1)$-th iteration. Suppose the Private R-AMDD satisfies β_{2k-1}-differential privacy during $(2k-1)$ iterations, then in $(2k)$-th iterations, by (29):

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

which is a deterministic mapping taking the outputs from $(2k-1)$-th iteration as input. Because the differential privacy is immune to post-processing [25], releasing $\{f_i(2k)\}_{i=1}^N$ doesn't increase the privacy loss, i.e., the total privacy loss up to $(2k)$-th iteration can still be bounded by β_{2k-1}.

APPENDIX III
PROOF OF THEOREM IV.1

Use the uppercase letters X and lowercase letters x to denote random variables and the corresponding realizations, and use $\mathscr{F}_X(\cdot)$ to denote its probability distribution.

For two neighboring datasets D_{all} and $\hat{D}_{all}$ of the network, by Lemma IV.1, the total privacy loss is only contributed by odd iterations. Thus, the ratio of joint probabilities (privacy loss) is given by:

$$\frac{\mathscr{F}_{F(0:2K)}(\{f(r)\}_{r=0}^{2}K|D_{all})}{\mathscr{F}_{F(0:2K)}(\{f(r)\}_{r=0}^{2}K|\hat{D}_{all})}=\frac{\mathscr{F}_{F(0)}(f(0)|D_{all})}{\mathscr{F}_{F(0)}(f(0)|\hat{D}_{all})}\cdot\prod_{k=1}^{K}\frac{\mathscr{F}_{F(2k-1)}(f(2k-1)|\{f(r)\}_{r=0}^{2k-2},D_{all})}{\mathscr{F}_{F(2k-1)}(f(2k-1)|\{f(r)\}_{r=0}^{2t-2},\hat{D}_{all})} \quad (48)$$

Since $f_i(0)$ is randomly selected for all i, which is independent of dataset, there is $\mathscr{F}_{F(0)}(f(0)|D_{all}) = \mathscr{F}_{F(0)}(f(0)|\hat{D}_{all})$. First only consider $(2k-1)$-th iteration, since the primal variable is updated according to (28), by KKT optimality condition:

$$\epsilon_i(2k-1)=-\nabla O(f_i(2k-1),D_i)-2\lambda_i(2k-2)-\eta\sum_{j\in\mathscr{V}_i}(2f_i(2k-1)-f_i(2k-2)-f_j(2k-2)) \quad (49)$$

Given $\{f(r)\}_{r=0}^{2k-2}$, $F_i(2k-1)$ and $E_i(2k-1)$ will be bijective $\forall i$, there is:

$$\frac{\mathscr{F}_{F(2k-1)}(f(2k-1)|\{f(r)\}_{r=0}^{2k-2},D_{all})}{\mathscr{F}_{F(2k-1)}(f(2k-1)|\{f(r)\}_{r=0}^{2k-2},\hat{D}_{all})}=\prod_{v=1}^{N}\frac{\mathscr{F}_{F_v(2k-1)}(f_v(2k-1)|\{f_v(r)\}_{r=0}^{2k-2},D_v)}{\mathscr{F}_{F_v(2k-1)}(f_v(2k-1)|\{f_v(r)\}_{r=0}^{2k-2},\hat{D}_v)}=\frac{\mathscr{F}_{F_i(2k-1)}(f_i(2k-1)|\{f_i(r)\}_{r=0}^{2k-2},D_i)}{\mathscr{F}_{F_i(2k-1)}(f_i(2k-1)|\{f_i(r)\}_{r=0}^{2k-2},\hat{D}_i)} \quad (50)$$

Since two neighboring datasets D_{all} and $\hat{D}_{all}$ only have at most one data point that is different, the second equality holds is because of the fact that this different data point could only be possessed by one node, say node i. Then there is $D_j=\hat{D}_j$ for $j\neq i$.

Given $\{f(r)\}_{r=0}^{2k-2}$, let $g_k(\cdot,D_i):\mathbb{R}^d\to\mathbb{R}^d$ denote the one-to-one mapping from $E_i(2k-1)$ to $F_i(2k-1)$ using dataset D_i. By Jacobian transformation, there is $\mathscr{F}_{F_i(2k-1)}(f_i(2k-1)|D_i)=\mathscr{F}_{E_i(2k-1)}(g_k^{-1}(f_i(2k-1),D_i))\cdot|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i)))|$, where $g_k^{-1}(f_i(2k-1),D_i)$ is the mapping from $F_i(2k-1)$ to $E_i(2k-1)$ using data D_i as shown in (49) and $\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i))$ is the Jacobian matrix of it. Then (48) yields:

$$\frac{\mathscr{F}_{F(0:2K)}(\{f(r)\}_{r=0}^{2K}|D_{all})}{\mathscr{F}_{F(0:2K)}(\{f(r)\}_{r=0}^{2K}|\hat{D}_{all})}=\prod_{k=1}^{K}\frac{\mathscr{F}_{E_i(2k-1)}(g_k^{-1}(f_i(2k-1),D_i))}{\mathscr{F}_{E_i(2k-1)}(g_k^{-1}(f_i(2k-1),\hat{D}_i))}\cdot\prod_{k=1}^{K}\frac{|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i)))|}{|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),\hat{D}_i)))|} \quad (51)$$

Consider the first part, $E_i(2k-1)\sim\exp\{-\alpha_i(k)||\epsilon||\}$, let $\hat{\epsilon}_i(2k-1)=g_k^{-1}(f_i(2k-1),\hat{D}_i)$ and $\epsilon_i(2k-1)=g_k^{-1}(f_i(2k-1),D_i)$

$$\prod_{k=1}^{K}\frac{\mathscr{F}_{E_i(2k-1)}(g_k^{-1}(f_i(2k-1),D_i))}{\mathscr{F}_{E_i(2k-1)}(g_k^{-1}(f_i(2k-1),\hat{D}_i))}=\prod_{k=1}^{K}\exp(\alpha_i(k)(||\hat{\epsilon}_i(2k-1)||-||\epsilon_i(2k-1)||))\leq\exp(\sum_{k=1}^{K}\alpha_i(k)||\hat{\epsilon}_i(2k-1)-\epsilon_i(2k-1)||) \quad (52)$$

Without loss of generality, let D_i and $\hat{D}_i$ be only different in the first data point, say (x_i^1,y_i^1) and $(\hat{x}_i^1,\hat{y}_i^1)$ respectively. By (49), Assumptions 4 and the facts that $||x_i^n||_2\leq 1$ (pre-normalization), $y_i^n\in\{+1,-1\}$.

$$||\hat{\epsilon}_i(2k-1)-\epsilon_i(2k-1)||=||\nabla O(f_i(2k-1),\hat{D}_i)-\nabla O(f_i(2k-1),D_i)||\leq\frac{2C}{B_i} \quad (53)$$

(52) can be bounded:

$$\prod_{k=1}^{K}\frac{\mathscr{F}_{E_i(2k-1)}(g_k^{-1}(f_i(2k-1),D_i))}{\mathscr{F}_{E_i(2k-1)}(g_k^{-1}(f_i(2k-1),\hat{D}_i))}\leq\exp(\sum_{k=1}^{K}\frac{2C\alpha_i(k)}{B_i}) \quad (54)$$

Consider the second part, the Jacobian matrix $\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i))$ is:

$$\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i))=-\frac{C}{B_i}\sum_{n=1}^{B_i}\mathscr{L}''(y_i^n f_i(2k-1)^T x_i^n)x_i^n(x_i^n)^T-\frac{\rho}{N}\nabla^2 R(f_i(2k-1))-2\eta V_i\mathbf{I}_d$$

Define

$$G(k)=\frac{C}{B_i}(\mathscr{L}''(\hat{y}_i^1 f_i(2k-1)^T\hat{x}_i^1)\hat{x}_i^1(\hat{x}_i^1)^T-\mathscr{L}''(y_i^1 f_i(2k-1)^T x_i^1)x_i^1(x_i^1)^T)\ ;$$
$$H(k)=-\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i))\ .$$

There is:

$$\frac{|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i)))|}{|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),\hat{D}_i)))|}=\frac{|\det(H(k))|}{|\det(H(k)+G(k))|}=\frac{1}{|\det(I+H(k)^{-1}G(k))|}=\frac{1}{|\prod_{j=1}^{r}(1+\lambda_j(H(k)^{-1}G(k)))|} \quad (55)$$

where $\lambda_j(H(k)^{-1}G(k))$ denotes the j-th largest eigenvalue of $H(k)^{-1}G(k)$. Since $G(k)$ has rank at most 2, $H(k)^{-1}G(k)$ also has rank at most 2. By Assumptions 4 and 5, the eigenvalue of $H(k)$ and $G(k)$ satisfy

$$\lambda_j(H(k))\geq\frac{\rho}{N}+2\eta V_i>0\ ;$$
$$-\frac{Cc_1}{B_i}\leq\lambda_j(G(k))\leq\frac{Cc_1}{B_i}\ .$$

Implies

$$-\frac{c_1}{\frac{B_i}{C}(\frac{\rho}{N}+2\eta V_i)} \leq \lambda_j(H(k)^{-1}G(k)) \leq \frac{c_1}{\frac{B_i}{C}(\frac{\rho}{N}+2\eta V_i)} \ .$$

Since $2c_1 < \frac{B_i}{C}(\frac{\rho}{N}+2\eta V_i)$, there is

$$-\frac{1}{2} \leq \lambda_j(H(k)^{-1}G(k)) \leq \frac{1}{2}.$$

Since $\lambda_{\min}(H(k)^{-1}G(k)) > -1$, there is

$$\begin{aligned}\frac{1}{|1+\lambda_{\max}(H(k)^{-1}G(k))|^2} &\leq \frac{1}{|\det(I+H(k)^{-1}G(k))|} \\ &\leq \frac{1}{|1+\lambda_{\min}(H(k)^{-1}G(k))|^2} \ .\end{aligned}$$

Therefore,

$$\begin{aligned}&\prod_{k=1}^{K}\frac{|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i)))|}{|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),\hat{D}_i)))|} \\ &\leq \prod_{k=1}^{K}\frac{1}{|1-\frac{c_1}{\frac{B_i}{C}(\frac{\rho}{N}+2\eta V_i)}|^2} \\ &= \exp(-\sum_{k=1}^{K}2\ln(1-\frac{c_1}{\frac{B_i}{C}(\frac{\rho}{N}+2\eta V_i)})) \ . \end{aligned} \tag{56}$$

Since for any real number $x \in [0, 0.5]$, $-\ln(1-x) < 1.4x$. (56) can be bounded with a simper expression:

$$\begin{aligned}&\prod_{k=1}^{K}\frac{|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),D_i)))|}{|\det(\mathbf{J}(g_k^{-1}(f_i(2k-1),\hat{D}_i)))|} \\ &\leq \exp(\sum_{k=1}^{K}\frac{2.8c_1}{\frac{B_i}{C}(\frac{\rho}{N}+2\eta V_i)}) \ . \end{aligned} \tag{57}$$

Combine (54)(57), (51) can be bounded:

$$\begin{aligned}&\frac{\mathscr{F}_{F(0:2K)}(\{f(r)\}_{r=0}^{2K}|D_{all})}{\mathscr{F}_{F(0:2K)}(\{f(r)\}_{r=0}^{2K}|\hat{D}_{all})} \\ &\leq \exp(\sum_{k=1}^{K}\frac{2C}{B_i}(\frac{1.4c_1}{(\frac{\rho}{N}+2\eta V_i)}+\alpha_i(k))) \ . \end{aligned} \tag{58}$$

Therefore, the total privacy loss during T iterations can be bounded by any β:

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

REFERENCES

[1] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, ser. ICALP'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 1–12.

[2] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 4177–4184.

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

[4] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, 2011.

[5] Z. Xu, G. Taylor, H. Li, M. A. Figueiredo, X. Yuan, and T. Goldstein, "Adaptive consensus admm for distributed optimization," in *International Conference on Machine Learning*, 2017, pp. 3841–3850.

[6] Z. Xu, M. A. Figueiredo, and T. Goldstein, "Adaptive admm with spectral penalty parameter selection," *arXiv preprint arXiv:1605.07246*, 2016.

[7] C. Zhang and Y. Wang, "Privacy-preserving decentralized optimization based on admm," *arXiv preprint arXiv:1707.04338*, 2017.

[8] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 5445–5450.

[9] Q. Ling and A. Ribeiro, "Decentralized linearized alternating direction method of multipliers," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5447–5451.

[10] 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 Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[11] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *International Conference on Machine Learning*, 2014, pp. 1701–1709.

[12] Q. Ling, Y. Liu, W. Shi, and Z. Tian, "Weighted admm for fast decentralized network optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 22, pp. 5930–5942, 2016.

[13] M. Hale and M. Egerstedty, "Differentially private cloud-based multi-agent optimization with constraints," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 1235–1240.

[14] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the 2015 International Conference on Distributed Computing and Networking*. ACM, 2015, p. 4.

[15] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 50–64, 2017.

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

[17] T. Zhang and Q. Zhu, "Dynamic differential privacy for admm-based distributed classification learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 172–187, 2017.

[18] X. Zhang, M. Khalili, and M. Liu, "Improving the privacy and accuracy of admm-based distributed algorithms," *arXiv preprint arXiv:1806.02246*, 2018.

[19] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, no. May, pp. 1663–1707, 2010.

[20] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "Decentralized quadratically approximated alternating direction method of multipliers," in *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*. IEEE, 2015, pp. 795–799.

[21] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "Dlm: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.

[22] J. Kelner, "An algorithmist's toolkit," 2007. [Online]. Available: http://bit.ly/2C4yRCX

[23] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1069–1109, 2011.

[24] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[25] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.