Dynamic Privacy For Distributed Machine Learning Over Network

Tao Zhang
Tandon School of Engineering
New York University
NY
Email: t.z.1992.nyc@gmail.com

Quanyan Zhu
Tandon School of Engineering
New York University
NY
Email: quanyan.zhu@nyu.edu

Abstract

Privacy-preserving distributed machine learning becomes increasingly important due to the rapid growth of amount of data and the importance of distributed learning. This paper develops algorithms to provide privacy-preserving learning for classification problem using the regularized empirical risk minimization (ERM) objective function in a distributed fashion. We use the definition of differential privacy, developed by Dwork et al. privacy to capture the notion of privacy of our algorithm. We provide two methods. We first propose the dual variable perturbation, which perturbs the dual variable before next intermediate minimization of augmented Lagrange function over the classifier in every ADMM iteration. In the second method, we apply the output perturbation to the primal variable before releasing it to neighboring nodes. We call the second method primal variable perturbation. Under certain conditions on the convexity and differentiability of the loss function and regularizer, our algorithms is proved to provide differential privacy through the entire learning process. We also provide theoretical results for the accuracy of the algorithm, and prove that both algorithms converges in distribution. The theoretical results show that the dual variable perturbation outperforms the primal case. The tradeoff between privacy and accuracy is examined in the numerical experiment. Our experiment shows that both algorithms performs similar in managing the privacy-accuracy tradeoff, and primal variable perturbaiton is slightly better than the dual case.

1. Introduction

Distributed machine learning has become increasingly important due to the rapid growth of amount of data and the increasing of model complexity. In practice, the amount of training data can range from $1TB$ to $1PB$ [2]. With this training data, it is possible to develop complex models with 10^9 to 10^{12} parameters [2, 5]. In centralized learning, these training data are shared by all the nodes participating in the learning process via centralized collection, and the parameters are available to all these nodes. In many cases, especially for the *statistical learning*, all nodes must frequently use the shared data and parameters in order to improve the parameters during the learning process. The centralized learning is not encouraged due to several aspects such as high computational complexity, scalability, and communication overhead, to name a few. As a result, decentralization of the dataset as well as distributed algorithms become more and more important.

The main goal of distributed learning is to decentralize the problem to multiple local subproblems. There are many ways to establish the decentralization, and the *alternating direction method of multiplier* (ADMM) is a well suited algorithm to deal with large scale distributed optimization problems. ADMM algorithm trains the model purely based on the information exchange among the neighboring nodes, rather than the entire network, and it has been proved that ADMM for convex optimization problem is convergent to the centralized problem under some specific conditions [6].

Many benefits have raised in the field of distributed machine learning. Google, eBay, Linkedin and Apple were among the corporations to take advantage of the massive data collected from their customers or users. They use technology like machine learning to improve decision making, reducing cost, provide new products and services. The benefits of distributed machine learning are undeniable, but it also presents serious privacy issues; there are possible internal and external attacks to the training data, which are stored in digital databases, such as social network data, web search histories, financial information, and medical records.

The general ADMM-based distributed learning has a certain level of privacy by avoiding the centralized collection of training datasets. Indeed, the decentralization has avoided the direct sharing of local dataset that contains sensitive information. However, deleting or anonymizing the sensitive information from the training dataset may reduce the accuracy of the learning model; even if the accuracy is not affected, some sensitive information can be still re-identified from the remaining information. These kinds of attacks have been studied in many works; for example, the adversary can use some background knowledge and cross correlation with other databeses to extract the private information [32, 30]. Other examples such as when the dataset has certain structural features an attacker is able to learn from the private model. These attackers can be from the outside as well as inside of the learing network.

In this paper, we focus on the ADMM-based distributed machine learning on the problem of classification. We use the *empirical risk minimization* (ERM) to construct the objective function of the problem. The ERM method use the dataset to construct an approximation of the expected risk , which is usually referred to the *empirical risk*. The classifier is chosen by minimizing the empirical risk. In this paper, we regularize the ERM with an additional term, the regularizer, in the empricial loss function to avoid *overfitting*, which means that although the minimum of the empirical risk can be close to zero, the expected risk we are interested in can be very large.

Our goal is to develop an learning algorithm that can preserve the privacy of training data in every local node from both the internal and the external attackers during the entire learning process. Specifically, we develop randomized algorithms that can provide privacy in terms of α-*differential privacy* [4, 9] while keeping the learning procedure accurate. Our algorithms hold for loss functions and regularizers that satisfy specific conditions of convexity and differentiability. For training, we propose two privacy preserving estimates of the regularized ERM-beased optimization. The first is *primal variable perturbation*; this is based on the *output perturbation* developed by Dwork et al. [4], which adds noise to the output of the non-private regularized ERM algorithm. In our method, we add noise to the intermediate updated primal variable of each node of ADMM-based distributed algorithm before sharing this primal variable to neighboring nodes. We call the second case *dual variable perturbation*, in which we perturb the dual variable of every node at each ADMM iteration before next iteration.

Our results are applicable to general ERM optimization problems, and we use numerical experiments to the classification problem based on logistic regression. Differential privacy model aims to ensure that even if the adversary has knowledge of all the dataset except one data point, the adversary should not be able to distinguish whether an individual datapoint is present or absent, by adding randomness to the output of the algorithm; thus, the differential privacy not only aims to protect the specific data points present in the dataset but also all the possible datapoints for that dataset. Since there are no conditions ofr the dataset for the purpose of privacy preservation, the randomness incurs a cost in the performance while guaranteeing the differential privacy. Therefore, managing the tradeoff between privacy and accuracy is critical. Under the assumption that the data points in the dataset are drawn from an unknown but fixed distribution, we prove the accuracy of the distributed learning algorithm in terms of the privacy parameters. Another important issue is the convergence of ADMM. There are many convergences results for ADMM discussed in literature. Based on the accuracy analysis, we also discuss the convergence of our private ADMM method.

The contributions of this paper are shown as follows:

- We derive a method, *dual variable perturbation*, in which we add randomness to the dual variable before the next update of the primal variable. The differential privacy is guaranteed for every ADMM iteration as well as the final trained output.
- Based on the output perturbation developed by Dwork et al. [4], we develop a private ADMM-based distributed algorithm for regulatized ERM, which applies *primal variable perturbation*. In this technique, the randomness rises when every node transmits the primal parameter to the corresponding neighboring nodes. It is guaranteed to provide differential privacy for the every intermediate update. For the final update, we apply the dual variable perturbation in order to increase the accuracy.
- We provide the theoretical guarantees of accuracy of both algorithms with L_2 regularization. Based on the accuracy analysis, we also show that both algorithms are convergent in distribution with different probability densities.
- We implement our methods by experiments on a dateset of UCI Machine Learning Repositories [16]. We provide a method to select the optimal privacy parameter α by solving an optimization problem given a specific utility function of privacy. The test results show that both the algorithm performs similarly, but the primal variable perturbation slightly outperforms the dual variable perturbation. However, theoretical analysis shows that dual variable perturbation has higher probability of accuracy and better sample requirement than does the primal case. Both algorithms are suitable for the both types of attacks we are interested in.

1.1. Related Work

There has been a significant amount of literature on the distributed classification learning algorithms. These works mainly focus on either enhancing the efficiency of the learning model, or on producing a global classifier from multiple distributed local classifier trained at the corresponding individual node. In the first kind of these works, researchers focus on making the distributed algorithm suitable to datasets of very large size; some ([14]) use MapReduce to explore the performance improvements. The second kind of works includes methods such as ADMM methods ([1]) , parameter averaging ([10]) voting classifiction ([13]), mixing parameters ([7]). Our distributed algorithm is based on ADMM, in which the centralized problem acts as a group of coupled distributed convex optimization subproblems with the consensus constaints on the primal parameters.

Research on privacy has been studied in a significant number of works since at least [20]. Recent literature on privacy includes anonymization [22], privacy-preserving data mining [19,20,21], cryptographic approaches [33, 35]. Simple anonymization approaches are ineffective. Individual information can be re-identified by simply using a small amount of side information [32,16]. In privacy-preserving data mining research, the privacy can be pried through, for example, *composition attacks*, in which case the adversary have some prior knowledge. Other works on data perturbation for privacy (for instance [25, 26]) focus on additive or multiplicative perturbation of individual samples, which might affect certain relationships among different samples in the database.

The idea of increasing privacy by adding noise has been studied for decades (for example, [49]; and see [48] for more details). The main perturbation techniques can be summarized into two basic classes. One is *input perturbation*, in which the training datasets are randomly modified prior to learning. The other one is *output perturbation*, where the exact solution is obtained from the true datasets but the noisy randomized version of the solution is released. There exist some inherent limitations for these two methods. Since Agrawal and Srikant's work in [50], increasing number of work studies the limitations and applicability of noise perturbation, and the definitions of privacy started to expand. In Dwork et al.'s basic definition of privacy [4], ε-indistinguishability or differential privacy, a change in a single entry of the dataset incurs a small change in the distribution from the view of any adversary via a specific measure of distance in a worst-case scenario.

Differential privacy has been used in a large number of works in privacy research (for example, [4, 11, 27, 28, 29]) since it was first proposed by Dwork et al [4]. Differential privacy is immune to the composition attacks mentioned above [30]. Later works include differential-private contingency tables [10], and differential-private combinatorial optimization [8]. Moreover, Wasserman and Zhou study the differential privacy more statistically. A body of exist literature also studies the differential-private machine learning. For example, Kasiviswanathan et al. derives a general method for probabilistically approximately correct (PAC, [47]) in [46]. Other examples includes the work of Blum et al. in [9] that provides a method to deliver the dataset differentially privately. Many works studied the tradeoff privacy and accuracy while developing and exploring the theory of differential privacy (examples include [4, 9, 10, 27, 28, 29, 44]). There are two main approaches used in differential privacy: perturbation by Laplace noise, and other exponential mechanism. In this paper, we focus on the Laplace noise perturbation. Laplace noise perturbation, especially the Laplace noise addition, is the primary method for the differential privacy.

The rest of the paper is organized as follows. Section 2 outlines the centralized ERM objective function and then shows the equivalent distributed form; the corresponding privacy concerns are described. In Section 3, algorithms are produced, and the analysis of privacy guarantee is provided. Section 4 discusses the tradeoff between privacy and accuracy, and the convergence of the algorithms. Finally, Section 5 and 6 present numerical experiments and concluding remarks.

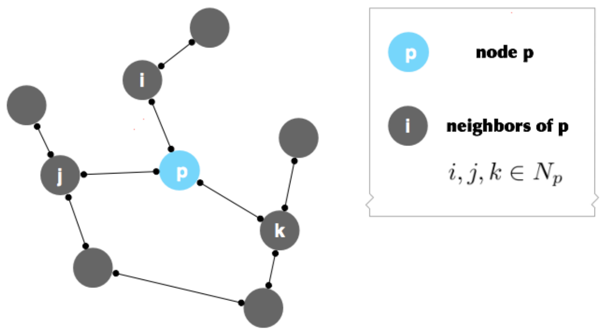


Figure 1. Network example where connectivity among nodes is denoted by a line joining them

2. Problem Statement

Consider a connected network shown in Figure 1, which contains P nodes described by one undirected graph $G(\mathscr{P}, \mathscr{E})$ with the set of nodes $\mathscr{P} = \{1, 2, 3, ..., P\}$, and edges $\mathscr{E}$ represened by lines denoting the linkes between connected nodes. A particular node $p \in \mathscr{P}$ only exchanges information between its neighboring node $j \in \mathscr{N}_p$, where $j \in \mathscr{N}_p$ is the set of all neighboring nodes of node p, and $N_p = |\mathscr{N}_p|$ is the number of neighboring nodes of node p. The network is connected but not necessary fully connected; there can be local cycles (e.g. local central node p and $i, j, k \in \mathscr{N}_p$). In a connected network, there must exist a path $i_1, i_2, i_3, ..., i_{m-1}, i_m$ of length at least 1 connecting node i_1 and i_m. Each node p contains a dataset D_p defined as follows:

$$D_p = \{(x_{ip}, y_{ip}) \subset X \times Y : i = 0, 1, ..., B_p\},$$

of size B_p with data vector $x_{ip} \in X \subseteq \mathbb{R}^d$, and the corresponding label $y_{ip} \in Y = \{-1, 1\}$. The entire network therefore has a set of data as:

$$\hat{D} = \bigcup_{p \in P} D_p.$$

The target of the centralized classification algorithm is to find a classifier $f : X \to Y$ using all available data $\hat{D}$ that enables each node in the network to classify any data x' input to a label $y' \in \{-1, 1\}$.

Suppose that $\hat{D}$ is available to the fusion center node, then we can choose the global classifier $f: X \rightarrow Y$ that minimizes the following centralized regularized emprical risk minimization problem (CR-ERM)

$$\min_{f} Z_C(f|\hat{D}) := \frac{C^R}{B_p} \sum_{p=1}^{P} \sum_{i=1}^{B_p} \hat{\mathscr{L}}(y_{ip},\ f^T x_{ip}) + \rho R(f), \tag{1}$$

where $C^R \leq B_p$ is a regularization parameter, and $\rho > 0$ is the parameter that controls the impact of the regularizer. The *loss function* $\hat{\mathscr{L}}(y_{ip},\ f^T x_{ip}) : \mathbb{R}^d \rightarrow \mathbb{R}$, is used to measure the quality of the classifier trained. In this paper, we focus on the specific loss function:

$$\hat{\mathscr{L}}(y_{ip},\ f^T x_{ip}) = \mathscr{L}(y_{ip} f^T x_{ip})$$

The function $R(f)$ is a regularizer that prevent overfitting. We aim to solve the centralized optimization problem (1) in a distributed fashion while achieving the same performance as in the centralized case. The decentralized equivalent enables node p to contribute by optimizing only the p-dependent terms of the objective function without exchanging any training data to other nodes $p' \neq p$. In this paper, we have the following assumptions

Assumption 1. - The loss function $\mathscr{L}$ is strictly convex and doubly differentiable of f with $|\mathscr{L}'| \leq 1$ and $|\mathscr{L}''| \leq c_1$, where c_1 is a constant. Both $\mathscr{L}$ and $\mathscr{L}'$ are continuous.

Assumption 2. - The regularizer function $R(\cdot)$ is continuous differentiable and 1-strongly convex. Both $R(\cdot)$ and $\nabla R(\cdot)$ are continuous.

Assumption 3. - We assume that $\|x_{ip}\| \leq 1$. Since $y_{ip} \in \{-1, 1\}$, then $|y_{ip}| = 1$.

2.1. Distributed ERM

To solve (1) in a distributed way, we first reform the objective function. The global variable f in CR-ERM is coupling the problem over the network. To decouple, we replace f by P copies of f; thus the global variable becomes auxiliary per-node variables $\{f_p\}_{p=1}^P$. Consensus constraints are required to force necessary global consistency condition $f_1 = f_2 = ... = f_P$ since the network is connected. Let Z_D denote $Z_D(\{f_p\}_{p \in \mathscr{P}} | \hat{D})$ be the decentralized objective function. An equivalent distributed form of the CR-ERM is

$$\min_{\{f_p\}_{p=1}^P} Z_D := \frac{C^R}{B_p} \sum_{p=1}^{P} \sum_{i=1}^{B_p} \mathscr{L}(y_{ip} f_p^T x_{ip}) + \sum_{p=1}^{P} \rho R(f_p).$$
$$s.t. f_p = f_j, p = 1, ..., P, j \in \mathscr{N}_p. \tag{2}$$

Now the problem (2) can be solved distributively by using the alternative direction method of multiplier (ADMM).

According to *Lemma* 1 in [1], if $\{f_p\}_{p=1}^P$ represnet a feasible solution of (2) and the network is connected, then problems (1) and (2) are equivalent, that is, $f = f_p, p = 1, ..., P$, where f is a feasible solution of (1).

In order to solve (2) by ADMM, we use the redundant variables $\{w_{jp}\}$ to assist to decouple f_p of node p from its neighors $j \in \mathscr{N}_p$. Thus the distrubted regularized emprical risk minimization problem (DR-ERM) becomes

$$\min_{\{f_p\}_{p=1}^P} Z_D.$$
$$s.t. f_p = w_{pj}, w_{pj} = f_j, p = 1, ..., P, j \in \mathscr{N}_p \tag{3}$$

Then the node-p-based individual objective function of (3) is

$$Z_p(f_p|D_p) := \frac{C^R}{B_p} \sum_{i=1}^{B_p} \mathscr{L}(y_{ip} f_p^T x_{ip}) + \rho R(f_p).$$

The augmented Lagrange funciton associated with the distributed optimization problem is:

$$\begin{aligned} L_D(\{f_p\}, \{w_{pj}\}, \{\lambda_{pj}^k\}) = & Z_D + \sum_{p=1}^{P} \sum_{i \in \mathscr{N}_p} \left(\lambda_{pi}^a\right)^T (f_p - w_{pi}) \\ & + \sum_{p=1}^{P} \sum_{i \in \mathscr{N}_p} \left(\lambda_{pi}^b\right)^T (w_{pi} - f_i) \\ & + \frac{\eta}{2} \sum_{p=1}^{P} \sum_{i \in \mathscr{N}_p} (\| f_p - w_{pi} \|^2 \\ & + \| w_{pi} - f_i \|^2). \end{aligned} \tag{4}$$

The distributed iterations solving (3) are:

$$\{f_p(t+1)\}_{p=1}^P = \arg \min_{\{f_p\}_{p=1}^P} L_D\big(\{f_p\}, \{w_{pj}(t)\}, \{\lambda_{pj}^k(t)\}\big), \tag{5}$$

$$\{w_{pj}(t+1)\}_{p=1}^{P} = \arg\min_{\{w_{pj}\}_{p=1}^{P}} L_D\big(\{f_p(t+1)\},\{w_{pj}\},\{\lambda_{pj}^{k}(t)\}\big), \tag{6}$$

$$\lambda_{pj}^{a}(t+1) = \lambda_{pj}^{a}(t) + \eta\big(f_p(t+1) - w_{pj}(t+1)\big),\quad p \in \mathscr{P},\ j \in \mathscr{N}_p, \tag{7}$$

$$\lambda_{pj}^{b}(t+1) = \lambda_{pj}^{b}(t) + \eta\big(w_{pj}(t+1) - f_p(t+1)\big)'\quad p \in \mathscr{P},\ j \in \mathscr{N}_p. \tag{8}$$

The general ADMM convergence is shown in Appendix A. Since the iterations (5)-(8) are proved to have the general form of ADMM iterations (see Appendix I), then the convergence of the decentralized regularized ERM is guaranteed.

From (4), the augmented Lagrange function is linear-quadratic in w_{pi}; thus, there is a closed form of $w_{pi}(t+1)$ at each iteration. Then we can replace w_{pi} terms in (5), (7), (8) by its closed expression. Moreover, by initializing the dual variables $\lambda_{pj}^{k} = \mathbf{0}_{d\times d}$, and let $\lambda_p(t) = \sum_{j\in\mathscr{N}_p} \lambda_{pj}^{k}$, $p \in \mathscr{P}$, $j \in \mathscr{N}_p$, $k = a, b$, we then can combine (7) and (8) into one update. As a result, the update procedures (5) to (8) can be further simplified through replacing w_{pi} by its corresponding closed form in (4). The simplified ADMM iteration is shown as follows, due to Lemma 3 of [1].

Let $L_N(t)$ denotes $L_N(\{f_p\},\{f_p(t)\},\{\lambda_p(t)\})$, and

$$L_N(t) = Z_D + 2\sum_{p=1}^{P} \lambda_p(t)^T f_p + \eta \sum_{p=1}^{P}\sum_{i\in\mathscr{N}_p} \parallel f_p - \frac{1}{2}(f_p(t) + f_i(t)) \parallel^2 .$$

The ADMM iterations (5)-(8) can be reduced to

$$\{f_p(t+1)\}_{p=1}^{P} = \arg\min_{\{f_p\}_{p=1}^{P}} L_N(\{f_p\},\{f_p(t)\},\{\lambda_p(t)\}), \tag{9}$$

$$\lambda_p(t+1) = \lambda_p(t) + \frac{\eta}{2}\sum_{j\in\mathscr{N}_p}[f_p(t+1) - f_j(t+1)]. \tag{10}$$

We denote the node-p-based non-private augmented Lagrange function $L_{Np}(\{f_p\},\{f_p(t)\},\{\lambda_p(t)\})$ as $L_{Np}(t)$:

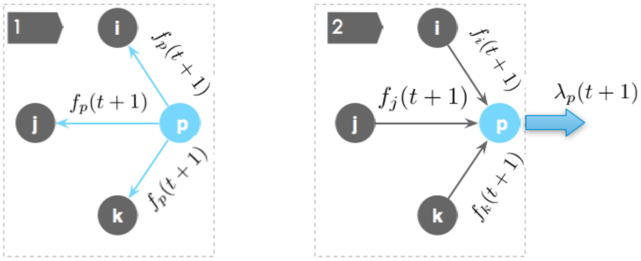


Figure 2. Visualization of iterations in Lemma 1.1: every node $p \in \mathscr{P}$ computes and broadcasts $f_p(t+1)$ to all neighbors *i, j, k* $\in \mathscr{N}_p$.2: once every node $p \in \mathscr{P}$ has received $f_x(t+1)$ from all $x \in \mathscr{N}_p$, it computes $\lambda_p(t+1)$.

$$L_{Np}(t) = \frac{C^R}{B_p}\sum_{i=1}^{B_p} \mathscr{L}(y_{ip} f_p^T x_{ip}) + \rho R(f_p) + 2\lambda_p(t)^T f_p + \eta \sum_{i\in\mathscr{N}_p} \parallel f_p - \frac{1}{2}(f_p(t) + f_i(t)) \parallel^2 . \tag{11}$$

Thus, every node p updates $f_p(t+1)$ at each iteration as follows

$$f_p(t+1) = \arg\min_{f_p} L_{Np}(t).$$

Algorithm 1 Distributed ERM

Required:Randomly initialize $f_p, \lambda_p = \mathbf{0}_{d\times 1}$ for every p

Inputs:$\hat{D}$

```
for $t = 0,1,2,3,...$ Do
    for $p = 1,2,3,...P$ Do
        Compute $f_p(t+1)$ via (9).
    end for
    for $p = 1,2,3,...P$ Do
        Broadcast $f_p(t+1)$ to all neighbors $j \in \mathscr{N}_p$
    end for
    for $p = 1,2,3,...P$ Do
        Compute $\lambda_p(t+1)$ via (10)
    end for
end for
```

Outputs: f^*

ADMM-based distributed ERM iterations (9) to (10) is illustrated in Figure 2 and summarized in Algorithm 1. Every node $p \in \mathscr{P}$ updates its local $d \times 1$ estimates $f_p(t)$ and $\lambda_p(t)$. At iteration $t+1$, node p updates the local $f_p(t+1)$ through

(9). Next, node p broadcasts the latest $f_p(t+1)$ to all its neighboring nodes $j \in \mathscr{N}_p$. Iteration $t+1$ finishes as each node updates the $\lambda_p(t+1)$ via (10).

Every iteration of our algorithm is still a minimization problem similar to the centralized problem (1). However, the number of variables participating in solving (9) per node per iteration, which is N_p, is much smaller than that in the centralized problem, which is $\sum_{p=1}^{P} N_p$. There are several methods to solve (9). For instance, projected gradient method, Newton method, and Broyden–Fletcher–Goldfarb–Shanno (BFGS) method that approximates the Newton method, to name a few.

ADMM based distributed machine learning has benefits due to high scalability, economic communication, and a certain level of privacy. The privacy arised is mainly due to the local parameter exchange among neighboring nodes instead of centralized communication. Dually, the parameter of each node is anonymous to the non-neighboring nodes. However, the neighboring nodes can access to the parameter without privacy protection; also, as shown in Section 1, simple anonymization is not good enough because it is still possible for adversary to extract the sensitive information with side information about the target.

2.2. Privacy Concerns

Although the data stored at each node is not exchanged during the entire ADMM algorithm, the potential privacy risk still exists. Suppose the dataset D_p stored at node p contains sensitive information in data point (x_i, y_i) that is not allowed to be released to other nodes in the network or anyone else outside.

In the distributed version of algorithm, node p optimizes only the p-dependent parts of the centralized problem. Let $K : \mathbb{R}^d \to \mathbb{R}$ be the randomized version of Algorithm 1, and let $\{f_p^*\}_{p\in\mathscr{P}}$ be the output of K at all the nodes. Then $\{f_p^*\}_{p\in\mathscr{P}}$ is random. Let K_p^t be the node-p-dependent sub-algorithm of K at iteration t, and let $f_p(t)$ be the output of $K_p^t(D_p)$ at iteration t inputing D_p. $f_p(t)$ is random at each t.

Consider an adversary, who knows all the data about node p except for the (x_{ip}, y_{ip}). The adversary is able to extract much additional information about (x_{ip}, y_{ip}) by observing the output of the algorithm. For the adversary inside the network, the sensitive information can even be leaked at any iteration of the training process. Therefore, it is necessary to develop a privacy preserved distributed ADMM algorithm for classification problem. We consider two types of attacks:

- **Type 1:** This attack is from adversaries outside the network, who do not have access to the intermediate ADMM iteration. The attack observes the output $\{f_p^*\}_{p\in\mathscr{P}}$ of algorithm K and aims to extract additional information of the private data point of the training dataset.
- **Type 2:** This attack is from the adversaries that can get access to the intermediate ADMM iterations. This attack aims to obtain additional information about the private data point of the the training dataset by observing the intermediate output $f_p(t)$ of K_p^t for all $p \in \mathscr{P}$.

We denote our privacy of distributed network based on the definition of *differential privacy* in [4]. Specifically, we require that a change of any single data point in the dataset might only change the distribution of the output of the algorithm slightly, which is visible to the adversary; this is done by adding randomness to the output of the algorithm. Let D_p and D'_p be two datasets differing in one data points; i.e., let $(x_{ip}, y_{ip}) \subset D_p$, and $(x'_{ip}, y'_{ip}) \subset D'_p$, then $(x_{ip}, y_{ip}) \neq (x'_{ip}, y'_{ip})$. In other words, their *Hamming Distance*

$$H_d(D_p, D'_p) = \sum_{i=0}^{B_p} \mathbf{1}\{i : x_i \neq x'_i\} \tag{12}$$

equals 1: i.e. $H_d(D_p, D'_p) = 1$.

Definition 1. *(Networked α_p-Differential Privacy) Consider a network consisits of P nodes $\mathscr{P} = 1, 2, ...P$, and each node p has a training dataset D_p, and $\hat{D} = \bigcup_{p\in P} D_p$. Let $K : \mathbb{R}^d \to \mathbb{R}$ be a randomized version of Algorithm 1. K outputs $\{f_p^*\}_{p\in\mathscr{P}}$, where $f_p^* = K(D_p)$ is the corresponding output at node p. Let D'_p be any dataset with*

$H_d(D'_p, D_p) = 1$, and let $g_p^* = K(D'_p)$. Then, K is networked α_p-differential private, if for any datasets D'_p for all $p \in \mathscr{P}$, known by the adversary of Type 1 attack, and for all possible sets of the outcomes $S \subseteq \mathbb{R}$, the following inequality holds:

$$\Pr[f_p^* \in S] \leq e_p^{\alpha} \cdot \Pr[g_p^* \in S]. \qquad (13)$$

The probability is taken over f_p^* the output of $K(\cdot)$ at each node $p \in \mathscr{P}$. The privacy raised is called networked α_p-differential Privacy.

Definition 1 specifies the privacy required against Type 1 attack. More specifically, networked α_p-differential private algorithms can prevent adversaries from obtaining much additional information by simply observing the output of the algorithm. This is because that no matter how the adversaries adjust the dataset D'_p ($H_d(D'_p, D_p) = 1$), the distribution of output can only change slightly.

For the privacy preserved against Type 2 attack, we have the following definition.

Definition 2. *(Dynamic α_p^t-Differential Privacy) Consider a network consisits of P nodes $\mathscr{P} = 1, 2, ...P$, and each node p has a training dataset D_p, and $\hat{D} = \bigcup_{p \in P} D_p$. Let $K : \mathbb{R}^d \to \mathbb{R}$ be a randomized version of Algorithm 1. Let K_p^t be the node-p-dependent sub-algorithm of K, optimizating ADMM iteration at t and outputing $f_p(t)$. Let D'_p be any dataset with $H_d(D'_p, D_p) = 1$, and let $g_p(t) = K_p^t(D'_p)$. We say that the algorithm K is dynamic α_p^t-differential private if for any dataset D'_p for all $p \in \mathscr{P}$ known by the adversary of Type 2 attack, and for all possible sets of the outcomes $S \subseteq \mathbb{R}$, the following inequality holds:*

$$\Pr[f_p(t) \in S] \leq e^{\alpha_p^t} \cdot \Pr[g_p(t) \in S], \qquad (14)$$

for all time t during a learning process. The probability is taken over $f_p(t)$, the output of K_p^t. The privacy raised for algorithm K is called dynamic α_p^t-differential Privacy.

Definition 2 provides the privacy against Type 2 attack. Dually, in dynamic α_p^t-differential private algorithms, adversaries of Type 2 attack cannot extract much additional information by observing the intermediate updates of $f_p(t)$. This is because that inputing any D'_p with $H_d(D'_p, D_p) = 1$ to the algorithm, the distribution of the output will not change much if any one data point is changed in D'_p.

Clearly, the algorithm with ADMM iterations shown in (9) to (11) is neither networked α_p-differential private nor dynamic α_p^t-differential private. This is because the intermedate and final optimal output f_p's are deterministic given dataset D_p. For D'_p with $H_d(D_p, D'_p) = 1$, the classifier will change completely, and the probability density $\Pr([f_p|D'_p]) = 0$, which leads to the ratio of probabilities $\frac{\Pr[f_p|D_p]}{\Pr[f_p|D'_p]} \to \infty$.

In order to provide the differential privacies defnined in Definition 1 and 2, we propose two algorithms, *dual variable perturbation* and *primal variable perturbation*, which are described in Section 3.1 and 3.2, respectively. Both algorithms can provide the two types of differential privacy defined in Section 2.2. However, we modify the primal variable perturbation by replacing the last ADMM iteration with dual variable perturbation in order to improve the accruacy of the final outputed classifier.

3. Dynamic Private Preserving

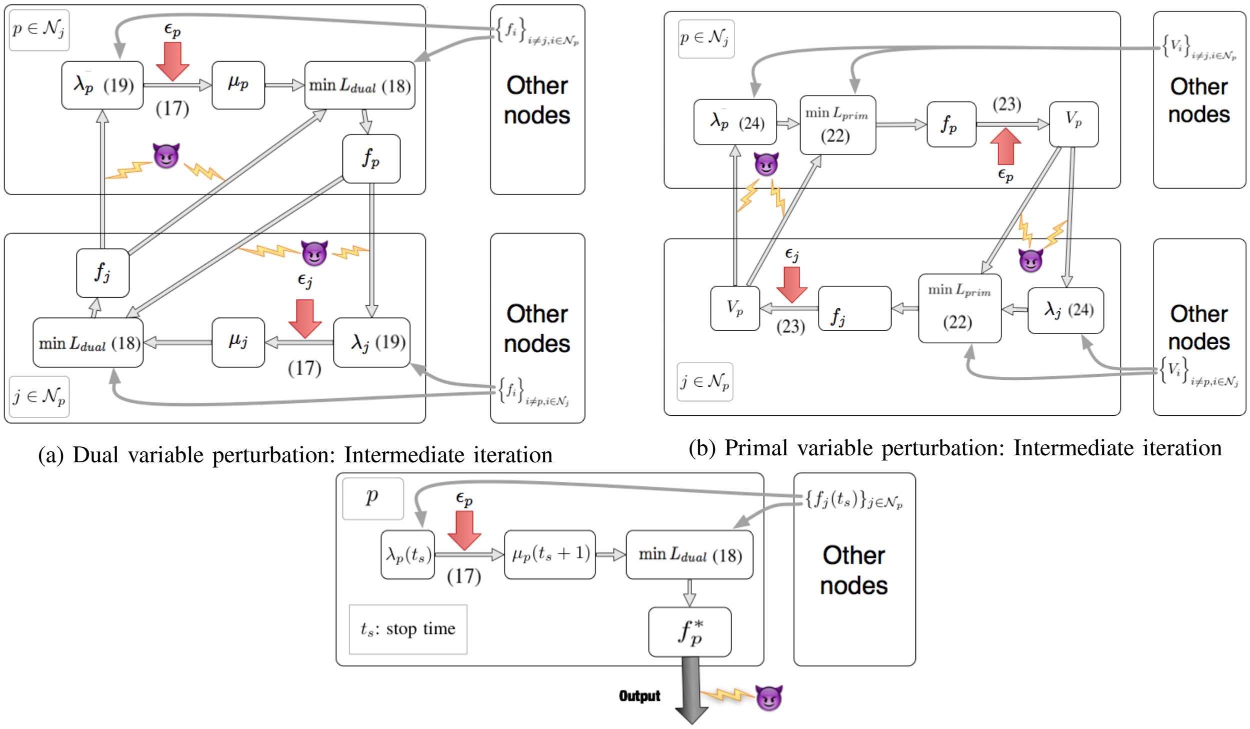
In this section, we describe two algorithms that provide networked and dynamic α-differential privacy defined in Section 2.2, respectively.

3.1. Dual Variable Perturbation

In order to provide differential privacy defined in Definition 1 and 2, we introduce our first private algorithm, *dual variable perturbation*, in which we perturb the dual variable $\{\lambda_p(t)\}_{p=1}^P$ with a random noise vector $\varepsilon_p(t)$, which has the probability density function:

$$\mathscr{K}(\varepsilon) \sim e^{-\zeta_p(t)\|\varepsilon\|}, \qquad (15)$$

where $\zeta_p(t)$ is a parameter related to the value of $\alpha_p(t)$. Let $\mu_p(t) = \lambda_p(t) + \varepsilon_p(t)$ be the perturbed dual variable. Now the corresponding node-p-based augmented Lagrange function $L_{Np}(t)$ becomes $L_{dual}\big(f_p, f_p(t), \mu_p(t+1), \{f_i(t)\}_{i \in \mathscr{N}_p}\big)$.



(a) Dual variable perturbation: Intermediate iteration

(b) Primal variable perturbation: Intermediate iteration

(c) Final iteration for both algorithms

Figure 3. Visualization of dual variable perturbation, primal variable perturbation. (a) shows the updates in the intermediate iterations of dual variable perturbation; (b) shows the primal variable perturbation during intermediate iterations; (c) describes the final updates for both algorithms. (a) and (b) prevent Type 2 attacks, and (c) prevents Type 1 attacks. The evail face and lighting symbols denote the attacks.

Use $L_{dual}(t)$ to denote $L_{dual}\big(f_p, f_p(t), \mu_p(t+1), \{f_i(t)\}_{i\in\mathcal{N}_p}\big)$, and we have

$$\begin{aligned} L_{dual}(t) =& \frac{C^R}{B_p}\sum_{i=1}^{B_p}\mathscr{L}(y_{ip}f_p^T x_{ip}) + \rho R(f_p) \\ &+ 2\mu_p(t+1)^T f_p + \frac{\Phi}{2} \parallel f_p \parallel^2 \\ &+ \eta\sum_{i\in\mathcal{N}_p} \parallel f_p - \frac{1}{2}(f_p(t) + f_i(t)) \parallel^2, \end{aligned} \tag{16}$$

where $\frac{\Phi}{2} \parallel f_p \parallel^2$ is an additional penalizer. As a result, the minimization of $L_{dual}(t)$ becomes random. We slightly change the iterations (9) to (10) as follows:

$$\mu_p(t+1) = \lambda_p(t) + \frac{C^R}{2B_p}\varepsilon_p(t+1), \tag{17}$$

$$f_p(t+1) = \arg\min_{f_p} L_{dual}(t), \tag{18}$$

$$\lambda_p(t+1) = \lambda_p(t) + \frac{\eta}{2}\sum_{j\in\mathcal{N}_p}[f_p(t+1) - f_j(t+1)]. \tag{19}$$

We perturb the dual variable $\lambda_p(t)$ via an additional variable μ_p in (19). This is because the dual variable is not exchanged during the training and is only used within the corresponding node; thus the direct perturbation to λ_p will affect the accuracy by the accumulated noise and is not necessary. We have the following theorem.

Theorem 1. *Let* $\hat{\alpha} = \alpha_p(t) - \ln\Big(1 + \frac{c_1}{\frac{B_p}{C^R}(\rho+2\eta N_p)}\Big)^2$. *If* $\hat{\alpha} > 0$, *then* $\Phi = 0$; *else, let* $\Phi = \frac{c_1}{\frac{B_p}{C^R}(e^{\alpha_p(t)/4}-1)} - \rho - 2\eta N_p$, *and as a result* $\hat{\alpha} = \alpha_p(t)/2$. *Under Assumption 1, 2 and 3, if the distributed classification optimization problem with objective function (2) can be solved by Algorithm 2, then the algorithm A_1 solving this distributed problem is dynamic α-differential private with $\alpha_p(t)$ for each node $p \in \mathscr{P}$ at time*

t. The ratio of conditional probabilities of $f_p(t)$ is bounded as:

$$\frac{Q(f_p(t)|D)}{Q(f_p(t)|D'_p)} \leq e^{\alpha_p(t)}, \tag{20}$$

where $Q(f_p(t)|D)$ and $Q(f_p(t)|D'_p)$ are the probability density functions of $f_p(t)$ given dataset D and D'_p, respectively, and $H_d(D,D'_p) = 1$.

Proof: See Appendix B

Algorithm 2 Dual Variable Perturbation

```
Required:Randomly initialize $f_p, \lambda_p = \mathbf{0}_{d\times 1}$ for every $p$
Inputs:$\hat{D}, \{[\alpha_p(1), \alpha_p(2), ...]\}_{p=1}^{P}$
for t = 0,1,2,3,... Do
    for p = 1,2,3,...P Do
        Let $\hat{\alpha} = \alpha_p(t) - \ln\left(1 + \frac{c_1}{\frac{B_p}{C^R}\left(\rho+2\eta N_p\right)}\right)^2$.
            If $\hat{\alpha} > 0$, then $\Phi = 0$, else, $\Phi = \frac{c_1}{\frac{B_p}{C^R}(e^{\alpha_p(t)/4}-1)} -$
            $\rho - 2\eta N_p$ and $\hat{\alpha} = \alpha_p(t)/2$.
        Draw noise $\varepsilon_p(t)$ according to (15) with
            $\zeta_p(t) = \hat{\alpha}$
        Compute $\mu_p(t+1)$ via (17)
        Compute $f_p(t+1)$ via (15)
        with augmented Lagrange function as (16).
    end for
    for p = 1,2,3,...P Do
        Broadcast $f_p(t+1)$ to all neighbors $j \in \mathscr{N}_p$
    end for
    for p = 1,2,3,...P Do
        Compute $\lambda_p(t+1)$ via (16)
    end for
end for
Outputs: $\{f_p^*\}_{p=1}^{P}$
```

The algorithm corresponding to Theorem 1 is illustrated in Figure 3 (a) and (c), and summarized in Algorithm 2. All nodes have its corresponding value of ρ. Every node $p \in \mathscr{P}$ updates its local estimates $\mu_p(t)$, $f_p(t)$ and $\lambda_p(t)$ at time t; at time $t+1$, node p first perturbs the dual variable $\lambda_p(t)$ obtained at time t to get $\mu_p(t+1)$ via (17), and then uses training dataset D_p to compute $f_p(t+1)$ via (18). Next, node p distributes $f_p(t+1)$ to all its neighboring nodes. The $(t+1)$-th update finishes when each node has updated its local $\lambda_p(t+1)$ via (19). The final iteration is exactly the same as the intermeidate iterations.

Theorem 1 links Algorithm 1 with Definition 2. We observe that an algorithm satisfies Definition 2 also satisfies Definition 1 for any individual node $p \in \mathscr{P}$. Therefore, any distributed algorithm that is dynamic α-differential private is also networked α-differential private. Thus, we have the following corollary.

Corollary 1.1. *If each node in the network chooses the same privacy parameter $\alpha_p(t) = \alpha^*(t)$ for all $p \in \mathscr{P}$ at each time, then the algorithm meets Theorem 1 also provide networked α-differential privacy with $\alpha = \alpha^*(t)$.*

Corollary 1.1 can be proved by directly substituting $\alpha_p(t) = \alpha^*(t)$ for all $p \in \mathscr{P}$.

The definitions of differential privacy in Section 2 (and also in, for example, [4, 11, 12]) only consider the change of output distribution corresponding to a change of a single entry of the dataset. However, in many cases, the sensitive information may be contained in more than one data point. Actually, Theorem 1 can be extended to deal with the dataset that has multiple sensitive data entries.

Corollary 1.2. *Let D and D'_p be two datasets with $H_d(D,D'_p) = c_2$, $c_2 \geq 1$. Then an algorithm meets Theorem 1 can compute a $f_p(t)$ that has the following bounded ratio of condtional densities:*

$$\frac{Q(f_p(t)|D)}{Q(f_p(t)|D'_p)} \leq e^{c_2\alpha'_p(t)}, \tag{21}$$

Proof: See Appendix C.

However, the output distribution must change corresponging to the a change of multiple data entries; as a result, the level of privacy has to decrease, especially for large c_2 and large B_P.

3.2. Primal Variable Perturbation

In this case, we perturb the primal variable $\{f_p(t+1)\}_{p=0}^{P}$ before releasing this variable to the neighboring nodes of each local node. This algorithm can also provide differential privacy defined in Definition 1 and 2. Let the node-p-based augmented Lagrange function $L_{prim}\big(f_p, f_p(t), \varepsilon_p(t), \lambda_p(t), \{V_i(t)\}_{i\in\mathscr{N}_p}\big)$ be represented as $L_{prim}(t)$:

$$\begin{aligned}L_{prim}(t) =& \frac{C^R}{B_p}\sum_{i=1}^{B_p} \mathscr{L}(y_{ip} f_p^T x_{ip}) + \rho R(f_p) + 2\lambda_p(t)^T f_p \\ &+ \eta \sum_{i\in\mathscr{N}_p} \parallel f_p - \frac{1}{2}(f_p(t) + V_i(t) - \varepsilon_p(t)) \parallel^2 .\end{aligned}$$

We use the un-perturbed primal $f_p(t)$ obtained at time t in the augmented Lagrange function and subtract the noise vector $\varepsilon_p(t)$ added at time t in order to reduce the noise in the minimization in (22); $\varepsilon_p(t)$ is static at time $t+1$. The privacy of releasing primal variable is not affected.

The following iterations specify the corresponding ADMM iterations.

The distributed iteration provideing dynamic $\alpha_p(t)$-differential privacy at time t is

$$f_p(t+1) = \arg\min_{f_p} L_{prim}(t), \tag{22}$$

$$V_p(t+1) = f_p(t+1) + \varepsilon_p(t+1), \tag{23}$$

$$\lambda_p(t+1) = \lambda_p(t) + \frac{\eta}{2}\sum_{j\in\mathscr{N}_p}[V_p(t+1) - V_j(t+1)], \tag{24}$$

where $\varepsilon_p(t+1)$ is the random noise vector with the density function (15). The aurgmented Lagrange function is (11). When the ADMM iteration meets the stop time, we input $\hat{D}$, and the latest $\{f_p(t)\}_p$ and $\{\lambda_p(t)\}_p$ obtained from (22) and (24), respectively, to Algorithm 1 to iterate (17) to (19) *one* time.

The following theorem states the result of primal variable perturbation.

Theorem 2. *Under Assumption 1, 2 and 3, if the distributed classification optimization problem with objective function (2) can be solved by Algorithm 3 with $\zeta_p(t) = \frac{\rho B_p \alpha_p(t)}{2}$, then the algorithm A_2 solving this distributed problem is dynamic α-differential private with $\alpha_p(t)$ for each node $p \in \mathscr{P}$ at time t. The ratio of conditional probabilities of $f_p(t)$ is bounded as in (20).*

Proof: See Appendix D.

Corollary 1.1 and 1.2 also hold for Theorem 2.

Corollary 2.1. *If all the nodes have the same privacy parameter $\alpha^*(t)$ at each time, then the algorithm meets Theorem 2 also provide networked α-differential privacy with $\alpha = \alpha^*(t)$.*

Similar to Corollary 1.1, Corollary 2.1 can be proved by substituting $\alpha = \alpha^*(t)$ for all $p \in \mathscr{P}$.

Corollary 2.2. *Let D and D'_p be two datasets with $H_d(D, D'_p) = c_2$, $c_2 \geq 1$. Any algorithm satisfies Theorem 2 can produce a private $f_p(t)$, which has the following bounded ratio of condtional densities at each iteration:*

$$\frac{Q(f_p(t)|D)}{Q(f_p(t)|D'_p)} \leq e^{c_2\alpha'_p(t)}. \tag{25}$$

The proof of Corollary 2.2 is the same as that of Corollary 1.2 in Appendix C.

Algorithm 3 Primal Variable Perturbation

```
Required:Randomly initialize $f_p, \lambda_p = \mathbf{0}_{d\times 1}$ for every $p$
Inputs:$\hat{D}, \{[\alpha_p(1), \alpha_p(2), ...]\}_{p=1}^P$
1: for t = 0,1,2,3,... Do
2:    for p = 1,2,3,...P Do
3:       Draw noise $\varepsilon_p(t)$ according to (15) with $\zeta_p(t) = \frac{\rho B_p \alpha_p(t)}{2C^R}$
4:       Compute $f_p(t+1)$ via (22)
5:       with augmented Lagrange function as (11).
6:       Compute $V_p(t+1)$ via (23)
7:    end for
8:    for p = 1,2,3,...P Do
9:       Broadcast $V_p(t+1)$ to all neighbors $j \in \mathscr{N}_p$
10:   end for
11:   for p = 1,2,3,...P Do
12:      Compute $\lambda_p(t+1)$ via (16)
13:   end for
14: if $t = stop\ time$
15 Input $\hat{D}$, and the latest $\{f_p(t)\}_p$ and $\{\lambda_p(t)\}_p$ obtained
      in above Step 4 and 12, respectively, to Algorithm 1 to
    iterate Step 1 once.
16: end for
Outputs: $\{f_p^*\}_{p=1}^P$
```

The algorithm associated with Theorem 2 is illustrated in Figure 3 (b)-(c), and is summarized in Algorithm 3. Each node $p \in \mathscr{P}$ updates $f_p(t)$,

$V_p(t)$ and $\lambda_p(t)$ at time t. Then, at time $t+1$, training dataset is used to compute $f_p(t+1)$ via (22), which is then perturbed to obtain $V_p(t+1)$ via (23). Next, $V_p(t+1)$ is distributed to all the neighboring nodes of node p. Finally, $\lambda_p(t+1)$ is updated via (24). The final iteration follows the dual variable perturbation.

4. Accuracy and Convergence Analysis

In this section, we discuss the accuracy of Algorithm 1 and 2. We establish performance bounds for regularization functions with L_2 norm. Our analysis is based on the following assumptions:

Assumption 4. - The data points $\{(x_{pi}, y_{pi})\}_{i=1}^{B_p}$ are drawn i.i.d. from a fixed but unknown probability distribution $\mathbb{P}^{xy}(x_{pi}, y_{pi})$.

Assumption 5. - $\varepsilon_p(t)$ is drawn from (15) with the same $\alpha_p(t) = \alpha(t)$ for all $p \in \mathscr{P}$.

We define the expected loss as

$$\hat{C}(f_p) := C^R \mathbb{E}_{(x,y)\sim\mathbb{P}^{xy}}(\mathscr{L}(yf^T x)).$$

Let $\hat{Z}$ be the expected objective as

$$\hat{Z}(f_p) := \hat{C}(f_p) + \rho R(f_p).$$

We also defined the constrained objectives for perturbed ADMM-based algorithms. Let $\varepsilon^{pi}(t) = \varepsilon_p(t) - \varepsilon_i(t)$, for $i \in \mathscr{N}_p$. Specifically, at each iteration t, we define:

$$Z_{dual}(f_p, t|D_p) := Z_p(f_p|D_p) + \frac{C^R}{B_p}\varepsilon_p(t)^T f_p,$$

$$\begin{aligned} Z_{prim}(f_p, t|D_p) := {} & Z_p(f_p|D_p) \\ & - \eta \sum_{i\in\mathscr{N}_p} \Big((f_p - \frac{1}{2}(f_p(t) + f_i(t))^T \\ & \cdot (\varepsilon^{pi}(t)) + \frac{1}{4}\big(\varepsilon^{pi}(t)\big)^2 \Big). \end{aligned}$$

Let $\varepsilon_p^t = \varepsilon_p(t)$, the noise vector generated at time t. The objective $Z_{dual}(f_p, t|D_p)$ (respectively, $Z_{prim}(f_p, t|D_p)$) is the corresponding node-p based objective function for the Algorithm 1 (respectively, Algorithm 2) if we fix the noise as ε_p^t generated at time t for $L_{dual}(f_p, t|D_p)$ throughout the entire ADMM process.

Let $\hat{f}_p(t+1)$, $f_p^{non}(t+1)$ and $f_p^*(t+1)$ be the population optimum, (non-private) empirical optimum, and private (empirical) optimum, respectively, defined at iteration $t+1$ as:

$$\hat{f}_p(t+1) = \arg\min_{f_p} \hat{Z}(f_p),$$

$$f_p^{non}(t+1) = \arg\min_{f_p} Z_p(f_p, t|D_p),$$

$$f_p^*(t+1) = \arg\min_{f_p} \overline{Z}(f_p, t|D_p),$$

where $\overline{Z}$ represents Z_{dual} or Z_{prim}, respectively.

Let $F_p(t+1) = \arg\min_{f_p} L_{nonP}(f_p, t|D_p)$be the updated non-private classifier at iteration $t+1$. From Theorem 9 (see Appendix A), the sequence $\{F_p(t+1)\}$ is bounded and converges to an optimal value $f_p^{non}(t+1)$ as time $t \to \infty$. Thus,there exists a constant $\Delta^{non}(t)$ such that:

$$\hat{C}(F_p(t)) - \hat{C}(f^{non}(t)) \leq \Delta^{non}(t).$$

Let $f_p(t+1)$ be the minimizer of the corresponding augmented Lagrange function of Z_{priv} at time t. Since both $Z_{dual}(f_p, t|D_p)$ and $Z_{prim}(f_p, t|D_p)$ are real and convex; similarly, the sequence $\{f_p(t)\}$ is bounded and $f_p(t)$ converges to $f_p^*(t)$, which is a limit point of $f_p(t)$, and there exists a constant $\Delta_p^{priv}(t) = \Delta_p^{dual}(t)$ or $\Delta_p^{prim}(t)$ given noise vector $\varepsilon_p(t)$ such that

$$\hat{C}(f_p(t)) - \hat{C}(f_p^*(t)) \leq \Delta_p^{priv}(t).$$

We will show that the performance of the algorithm can depend on the number of data points, B_p, of the dataset D_p, for all $p \in \mathscr{P}$. Let $f_p^0(t)$ be a reference classifier at time t with the expected loss as $\hat{C}^* = \hat{C}(f_p^0(t))$. Specifically, the performance of the algorithm is measured by the B_p, which is a function of $\parallel f_p^0(t) \parallel$ required to obtain a classifier $f_p(t)$ that minimizes the expected loss within some accuracy:

$$\hat{C}(f_p(t)) \leq \hat{C}^*(t) + \alpha_{acc} + \Delta_p^{priv}(t).$$

where α_{acc} is the *optimization accuracy*. We say that every learned $f_p(t)$ is α_{acc}-optimal if it satisfies the above inequality. First, we provide the theorem about the performance of the non-private ADMM-based algorithm.

Theorem 3. *Let $R(f_p(t)) = \frac{1}{2} \parallel f_p(t) \parallel^2$, and $f_p^0(t)$ such that $\hat{C}(f_p^0(t)) = C_E^*(t)$ for all $p \in \mathscr{P}$ at* *time t, and a real number $\delta > 0$. Let $F_p(t+1) = \arg\min_{f_p} L_{nonP}(f_p,t|D_p)$ be the output of Algorithm 1. If Assumption 1 and 4 are satisfied, then there exists a constant β_{non} such that if the number of data points, B_p in $D_p = \Big\{(x_{ip},y_{ip}) \subset \mathbb{R}^d \times \{-1,1\}\Big\}$ satisfy:*

$$B_p > \beta_{non} \max\left(\left\{\frac{C^R \parallel f_p^0(t+1) \parallel^2 \ln(\frac{1}{\delta})}{\alpha_{acc}^2}\right\}_{t=1}\right),$$

then $F_p(t+1)$ satisfies:

$$\mathbb{P}\left(\hat{C}(F_p(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc} + \Delta^{non}(t)\right) \geq 1-\delta.$$

Proof: See Appendix E.

We now establish the performance bounds for Algorithm 1, *dual variable perturbation*, which is summarized in the following theorem.

Theorem 4. *Let $R(f_p(t)) = \frac{1}{2} \parallel f_p(t) \parallel^2$, and $f_p^0(t)$ such that $\hat{C}(f_p^0(t)) = C_E^*(t)$ for all $p \in \mathscr{P}$, and a real number $\delta > 0$. If Assumption 1, 4 and 5 are satisfied, then there exists a constant β_{dual} such that if the number of data points, B_p in $D_p = \Big\{(x_{ip},y_{ip}) \subset \mathbb{R}^d \times \{-1,1\}\Big\}$ satisfy:*

$$B_p > \beta_{dual} \max\left(\left\{\frac{\parallel f_p^0(t+1) \parallel d \ln(\frac{d}{\delta})}{\alpha_{acc}\alpha_p(t)}\right\}_{t=1}, \left\{\frac{C^R c_1 \parallel f_p^0(t+1) \parallel^2}{\alpha_{acc}\alpha_p(t)}\right\}_{t=1}, \left\{\frac{C^R \parallel f_p^0(t+1) \parallel^2 \ln(\frac{1}{\delta})}{\alpha_{acc}^2}\right\}_{t=1}\right),$$

then $f_p^*(t+1)$ satisfies:

$$\mathbb{P}\left(\hat{C}(f_p^*(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc}\right) \geq 1-2\delta.$$

Proof: See Appendix F.

Corollary 4.1. *Let $f_p(t+1) = \arg\min L_{dual}(f_p,t|D_p)$ be the intermediate updated classifier of Algorithm 2 and let $f_p^0(t)$ be a reference classifier such that $\hat{C}(f_p^0(t) = \hat{C}^*(t)$. If all the conditions of Theorem 3 are satisfied, then, $f_p(t+1)$ satisfies*

$$\mathbb{P}\left(\hat{C}(f_p(t+1)) \leq \hat{C}^*(t) + \alpha_{acc} + \Delta_p^{dual}(t)\right) \geq 1-2\delta.$$

Proof: The following holds for $f_p(t)$ and $f_p^*(t)$

$$\hat{C}(f_p(t)) - \hat{C}(f_p^*(t)) \leq \Delta_p^{dual}(t).$$

From Theorem 3,

$$\mathbb{P}\left(\hat{C}(f_p^*(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc}\right) \geq 1-2\delta.$$

Therefore, we can have:

$$\mathbb{P}\left(\hat{C}(f_p(t+1)) \leq \hat{C}^*(t) + \alpha_{acc} + \Delta_p^{dual}\right) \geq 1-2\delta.$$

□

Theorem 4 and Corollary 4.1 can guarantee the privacy defined in both Definition 1 and 2. The following theorem is used to analyze the performance bound of un-perturbed classifier $f_p(t+1)$ in (22), which minimizes $L_{prim}(t)$ that involves noise vectors from $V_p(t)$ perturbed at the previous iteration.

Theorem 5. *Let $R(f_p(t)) = \frac{1}{2} \parallel f_p(t) \parallel^2$, and $f_p^0(t)$ such that $\hat{C}(f_p^0(t)) = C_E^*(t)$, and a real number $\delta > 0$. From Assumption 1, we have the loss function $\mathscr{L}(\cdot)$ is convex and differentiable with $\mathscr{L}'(\cdot) \leq 1$. If Assumption 4 and 5 are satisfied, then there exists a constant β_{prim}^A such that if the number of data points, B_p in $D_p = \Big\{(x_{ip},y_{ip}) \subset \mathbb{R}^d \times \{-1,1\}\Big\}$ satisfies:*

$$B_p > \beta_{prim}^A \max\left(\left\{\frac{C^R \parallel f_p^0(t+1) \parallel^3 \eta N_p d \ln(\frac{d}{\delta})}{\alpha_{acc}^2 \alpha_p(t)}\right\}_{t=1}, \left\{\frac{C^R \parallel f_p^0(t+1) \parallel^2 \ln(\frac{1}{\delta})}{\alpha_{acc}^2}\right\}_{t=1}\right),$$

then $f_p^*(t+1)$ satisfies:

$$\mathbb{P}\left(\hat{C}(f_p^*(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc}\right) \geq 1-2\delta.$$

Proof: See Appendix G.

Theorem 6. *Let $R(f_p(t)) = \frac{1}{2} \parallel f_p(t) \parallel^2$, and $f_p^0(t)$ such that $\hat{C}(f_p^0(t)) = C_E^*(t)$, and a real number $\delta > 0$. Let $f_p^*(t+1) = \arg\min Z_{prim}(t)$ be α_{acc}-accurate according to Theorem 4. From Assumption 1 we have that the loss function $\mathscr{L}(\cdot)$ is convex and differentiable with $\mathscr{L}'(\cdot) \leq 1$, and we also assume that $\mathscr{L}'$ satisfies:*

$$|\mathscr{L}'(a) - \mathscr{L}'(b)| \leq c_4|a-b|$$

for all pairs (a,b) with a constant c_4. If Assumption 4 and 5 are satisfied, then there exists a constant β_{prim}^B such that if the number of data points, B_p in $D_p = \Big\{(x_{ip}, y_{ip}) \subset \mathbb{R}^d \times \{-1,1\}\Big\}$ satisfies:

$$\begin{aligned}
B_p >& \beta_{prim}^B \max\Bigg(\Big\{\frac{C^R \parallel f_p^0(t+1) \parallel^3 \eta N_p d \ln(\frac{d}{\delta})}{\alpha_{acc}^2 \alpha_p(t)}\Big\}_{t=1},\\
&\Big\{\frac{C^R \parallel f_p^0(t+1) \parallel^2 \ln(\frac{1}{\delta})}{\alpha_{acc}^2}\Big\}_{t=1},\\
&\Big\{\frac{4C^B \parallel f^0(t+1) \parallel d\big(\ln(\frac{d}{\delta})\big)^2}{\alpha_{acc}\alpha_p(t)}\Big\}_{t=1},\\
&\Big\{\frac{4 \parallel f_p^0(t+1) \parallel^3 \eta N_p d \ln(\frac{d}{\delta})}{\alpha_{acc}^2 \alpha_p(t)}\Big\}_{t=1},\\
&\Big\{\frac{4\big(C^R\big)^{\frac{3}{2}} \parallel f_p^0(t+1) \parallel^2 d \ln(\frac{d}{\delta})}{\alpha_{acc}^{3/2} \alpha_p(t)}\Big\}_{t=1}\Bigg),
\end{aligned}$$

then $V_p^*(t+1) = f_p^*(t+1) + \varepsilon_p(t+1)$ satisfies:

$$\mathbb{P}\big(\hat{C}(V_p^*(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc}\big) \geq 1 - 3\delta.$$

Proof: See Appendix H.

Corollary 6.1. *Let $f_p(t+1) = \arg\min L_{prim}(f_p, t|D_p)$ be the intermediate updated classifier of Algorithm 3, and let $f_p^0(t)$ be a reference classifier such that $\hat{C}(f_p^0(t) = \hat{C}^*(t)$. If all the conditions of Theorem 5 are satisfied, then, $V_p(t+1) = f_p(t+1) + \varepsilon_p(t+1)$ satisfies*

$$\mathbb{P}\big(\hat{C}(V_p(t+1)) \leq \hat{C}^*(t) + \alpha_{acc} + \Delta_p^{prim}(t)\big) \geq 1 - 3\delta.$$

Proof: Since

$$\hat{C}(f_p(t)) - \hat{C}(f_p^*(t)) \leq \Delta_p^{prim}(t),$$

then

$$\hat{C}(V_p(t)) - \hat{C}(V_p^*(t)) \leq \Delta_p^{prim}(t).$$

From Theorem 5, $V_p^*(t+1)$ satisfies

$$\mathbb{P}\big(\hat{C}(V_p^*(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc}\big) \geq 1 - 3\delta.$$

Therefore, we have:

$$\mathbb{P}\big(\hat{C}(V_p(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc} + \Delta_p^{prim}(t)\big) \geq 1 - 3\delta.$$

□

Since at the last iteration of primal variable perturbation we use the same iteration as that of the dual variable perturbation, Theorem 6 and Corollary 6.1 only guarantee the dynamic α_p^t-differential privacy for primal variable perturbation. As a result, we combine the conditions of Theorem 4 and 6 to guarantee the networked α_p-differential privacy. Thus, we have the following corollary.

Corollary 6.2. *Let f_p^* be the final output classifier of Algorithm 3 of node p, and let $f_p^0(t)$ be a reference classifier such that $\hat{C}(f_p^0(t) = \hat{C}^*(t)$. If all the conditions of Theorem 4 and 6 are satisfied, then, f_p^* satisfies*

$$\mathbb{P}\big(\hat{C}(f_p^*) \leq \hat{C}^*(t) + \alpha_{acc} + \Delta_p^{dual}(t)\big) \geq 1 - 5\delta.$$

Proof: We need all the conditions of Theorem 6 to be satisfied in order to guarantee the privacy during the intermediate iterations. All the conditions of Theorem 4 are satisfied so that the networked α_p-differential privacy is provided. Combining Theorem 4 and 6 gives the probability no less than $1-5\delta$. □

Clearly, the privacy rises by trading the accuracy. It is essential to manage the tradeoff between the privacy and accuracy in order to establish both the privacy and accuracy with at least satisfied level.

Another important issue we care about is the convergence of the Algorithm 1 and 2. Our analysis based on the assumption that all the conditions of Theorem 3 to 5 are satisfed. As shown in Appendix A, the non-private ADMM algorithm is convergent. In our private algorithms, the augmented Lagrange function (11) and (16) are solvable since both of them are convex. Also, the matrix $A = I_d$ is an identity matrix in our case, thus $A^T A$ is nonsingular. Theorem 9 shows that the non-private ADMM-based optimizaiton is convergent. However, our algorithms do not necessarily converge to one optimum classifier for all the nodes; different node can have different value of convergent classifier, but all of them have similar performance.

We first analysis the convergence of dual variable perturbation. We summarize the convergence analysis in the following theorem.

Proposition 7. *Let $f_p^0(t)$ be a reference classifier such that $\hat{C}(f_p^0) = C_E^*(t)$ for all node $p \in \mathscr{P}$ at*

time t. If all the conditions of Theorem 4 are satisfied, then $f_p(t) = \arg\min L_{dual}(f_p, t-1|D_p)$ is convergent in distribution with probability $\leq 1-2\delta$.
Proof: See Appendix J.

The convergence of primal variable perturbation only consider the primal variable $f_p(t+1)$ at each time before perturbation. It is summarized in the folowing theorem.

Proposition 8. *Let $R(f_p(t)) = \frac{1}{2} \parallel f_p(t) \parallel^2$, and $f_p^0(t)$ such that $\hat{C}(f_p^0(t)) = C_E^*(t)$, and a real number $\delta > 0$. If all the conditions of Theorem 6 is satisfied, then $f_p(t) = \arg\min L_{prim}(f_p, t|D_p)$ is convergent in distribution with probability $1-3\delta$.*
Proof: See Appendix K.

5. Numerical Experiments

In this section, we test Algorithm 2 and 3 with real world training dataset. Consider the following examples. The classification method used is *logistic regression*. Potential application scenarios include but nor limited to the following two.

Example 5.1. *(Potential Customer Classification)* Consider a network of P companies agreed to collaborate to develop an algorithm that can classify the target customers by predicting their annual incomes based on thier information such as age, sex, occupation, and education. Suppose D_p is the customer data records stored at company p. The learning process of the algorithm is based on all available datasets $\{D_p\}_{p=1}^P$, rather than company p alone. The company p learns the model only by its own training dataset D_p, and there is no data exchange between different companies. The intermediate updated classifier $f_p(t)$ is the only shared information. Moreover, company p only communicate with its neighboring companies. The companies want to increase the privacy level of the algorithm, and make sure the final algorithm and also the learning process preserves the privacy of the sensitive information against other companies in this network as well as other parties from outside.

Example 5.2. *(International Collaborative Anti-Terrorist)* Consider a group of countries with corresponding datasets D containing intelligence about the terrorism. All the countries are willing to collaborate in order to classify jointly possible terrorist entering their countries. However, the confidential information involved in the intelligence prevents each countries to open access to the dataset of other countries. In this case, differential privacy model can preserve the confidential intelligence while producing an accurate classifier of terrorist.

5.1. Privacy Preserved Logistic Regression

In the experiments, we use our algorithm to develop a dynamic differential private logisitic regression. The logistic regression has the loss function:

$$\mathscr{L}_{LR}(y_{ip} f^T x_{ip}) = log(1 + exp(-y_{ip} f_p^T x_{ip})). \tag{26}$$

The first derivative and teh second derivative are:

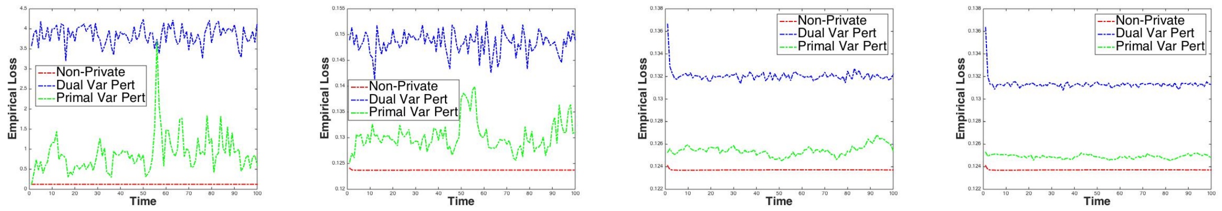
$$\mathscr{L}'_{LR} = \frac{-y_{ip} x_{ip}}{1 + exp(y_{ip} f_p^T x_{ip})}$$

$$\mathscr{L}''_{LR} = \frac{y_{ip}^2 x_{ip} x_{ip}^T}{(1 + exp(y_{ip} f_p^T x_{ip})(1 + exp(-y_{ip} f_p^T x_{ip})},$$

which can be bounded as $|\mathscr{L}'_{LR}| \leq 1$ and $\mathscr{L}''_{LR} \leq \frac{1}{4}$, respectively, according to Assumption 3. Therefore, the loss function of logistic regression satisfies the conditions shown in Assumption 1. In this case, $R(F_p) = \frac{1}{2} \parallel f_p \parallel^2$, and $c_1 = \frac{1}{4}$. And we can directly apply the loss function $\mathscr{L}_{LR}$ to Theorem 1 and 2 with $R(f) = \frac{1}{2} \parallel f_p \parallel^2$, and $c_1 = \frac{1}{4}$, and then it can provide $\alpha_p(t)$-differential privacy for any $p \in \mathscr{P}$ at time $t = 1, 2, ...$ of a distributed logistic regression problem.

5.2. Pre-Processing

We test our algorithms to this example. The classification method used is *logistic regression*. We simulate the customer information by *Adult* dataset from UCI Machine Learning Repository [11], which contains demographic information such as age, sex, education, occupation, marital status, and native country. There are 48842 data samples. The prediction task is to determine



(a) $\alpha = 0.01$

(b) $\alpha = 0.1$

(c) $\alpha = 0.5$

(d) $\alpha = 1$

Figure 4. Convergence of algorithms, at iteration 100 (before the stop time)

whether a person's annul income is greater than $\$50k$.

In order to process the Adult dataset to our algorithm, we remove all the missing data points, and follow the data cleaning process of [12]. Also, we convert the categorial attributes to a binary vector. For other non-numerical descriptive attributes such as different countries in the category of native country, we replace them by their own frequency of occurance in the corresponding category. Moreover, each column is first normalized to make sure the maximum value is 1; then each row is normalized so that the L_2 norm of each data sample is at most 1.

5.3. Privacy-Accuracy Tradeoff

In this experiment, we study the privacy-accuracy tradeoff of Algorithm 2 and 3. The privacy is quantified by the value of $\alpha_p(t)$.

When $\alpha_p(t)$ becomes larger, the ratio of densities of the classifier $f_p(t)$ on two different data sets is larger, which implies a higher belief of the adversary when one data point in data set D is changed; thus, it provides lower privacy. However, the accuracy of the algorithm increases as $\alpha_p(t)$ becomes larger. As shown in Figure 4, larger $\alpha_p(t)$ gives better convergence of the algorithms; moreover, from Figure 4, we can see that the dual variable perturbation is slightly more robust to noise than is the primal case given the same value of $\alpha_p(t)$. When $\alpha_p(t)$ is small, the model is more private but less accurate. Therefore, the utilities of privacy and accuracy shoud satisfy the following assumption:

Assumption 6. - the utilities of privacy and accuracy should be monotonic with respect to $\alpha_p(t)$ but in different directions, say decreasingly and increasingly, respectively.

As a result, The quality of classifier is measured by the empirical loss $\overline{C}(t) = \frac{C^R}{B_p}\sum_{i=1}^{B_p}\mathscr{L}(y_{ip}f_p(t)^T x_{ip})$. Given the dataset D_p and a $\alpha_p(t_x)$ at a specific time t_x, there exists a corresponding $f_p(t_x)$ minimizing (16). Thus, there must be a function $L_{acc}()$ to capture relationship between $\alpha_p(t)$ and $\overline{C}(t)$: $L_{acc}(\alpha_p(t)) = \overline{C}(t)$. The function L_{acc} is obtained by curve fitting given the experimental data points $(\alpha_p(t), \overline{C}(t))$. Let $U_{priv}(\alpha_p(t))$ be the utility of privacy. Besides the decreasing monotonicity, $U_{priv}(\alpha_p(t))$ should be convex and doubly differentiable function of $\alpha_p(t)$.

Given the privacy utility function $U_{priv}(\alpha_p(t))$, there exists an optimal value of $\alpha_p^*(t)$ that minimizes the following problem:

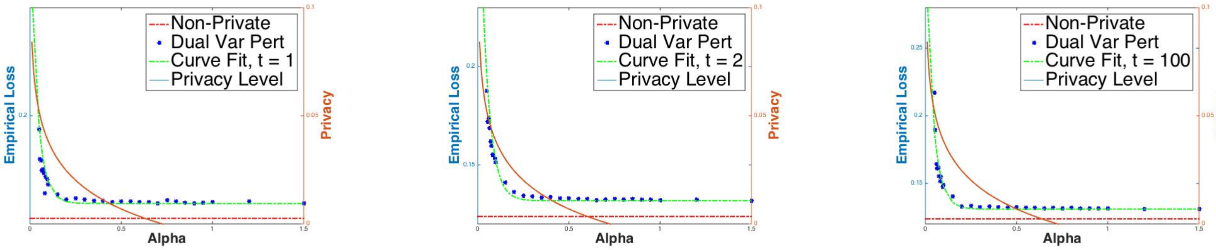
$$\min \mathscr{J}(t) = L_{acc}(\alpha_p(t)) - U_{priv}(\alpha_p(t))$$
$$s.t.\ 0 < \alpha_p(t) \le \alpha_U,\ 0 \le L_{acc}(\alpha_p(t)) \le c_3 \tag{27}$$

where α_U and c_3 are the threshold values for $\alpha_p(t)$ and L_{acc}, respectively, beyond which is considered as non-private and non-accurate, respectively. The above discussion is summarized in the following definition.

Definition 3. *(Optimal Private) If there exists a value of privacy parameter $\alpha_p^*(t)$ that minimizes (35):*

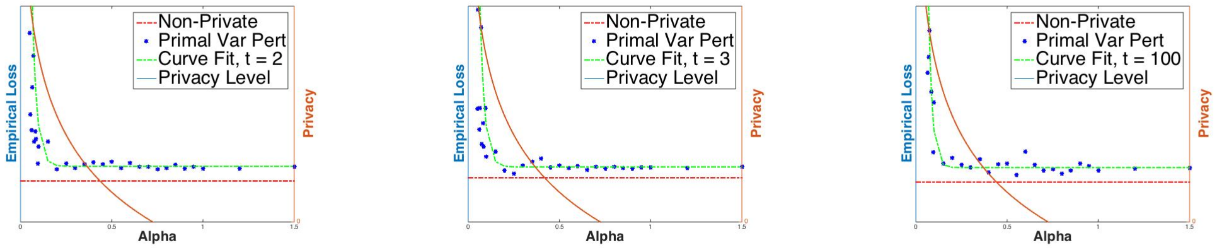
$$\alpha_p^*(t) = \arg\min_{\alpha_p(t)} \mathscr{J}(t)$$
$$s.t.\ \alpha_L \le \alpha_p(t) \le \alpha_U,\ 0 \le L_{acc}(\alpha_p(t)) \le c_3 \tag{28}$$

then by choosing thie value as the privacy prarmeter, every iteration of Algorithm 1 and 2 for each node $p \in \mathscr{P}$ is optimal private.



(a) $t = 1$ (b) $t = 2$ (c) $t = 100$

Figure 5. Privacy-Accuracy Tradeoff: Dual Variable Perturbation, with $\omega_{p1} = 0.02$, $\omega_{p2} = 6$, $\omega_{p3} = 9$, $\omega_{p4} = 1$(before the stop time)



(a) $t = 2$ (b) $t = 3$ (c) $t = 100$

Figure 6. Privacy-Accuracy Tradeoff: Primal Variable Perturbation, with $\omega_{p1} = 0.02$, $\omega_{p2} = 6$, $\omega_{p3} = 9$, $\omega_{p4} = 1$ (before the stop time)

For training the classifier, we tried a few fixed values of ρ and test the empirical loss $L_{ep}(t)$ of the classifier. Then, we selected the value of ρ that minimizes the empirical loss for a fixed α_p (0.3 in this experiment). We also test the non-private version of Algorithm, and the corresponding minimum value of ρ is obtained as the control. We chose the corresponding optimal value of the regularization prarmeter ρ for each algorithm as shown in Table 1.

Table 1. Optimal value of Λ for different algorithms

Algorithm	Non-Private	Dual Var Pert	Primal Pert
ρ	10^{-10}	$10^{-2.5}$	10^{-1}

Table 2. Value of C^R for different algorithms

Algorithm	Non-Private	Dual Var Pert	Primal Pert
C^R	1750	1750	146

Table 2 shows the values of C^R chosen for each algorithm and the non-private case. Figure (4) shows the convergence of dual and primal variable perturbations at different value of $\alpha_p(t)$. Larger values of α_p yields better convergence for both perturbations. Moreover, the dual variable perturbation has smaller variance of empirical loss than does the primal perturbation. However, larger α_p incurs poorer privacy. This tradeoff is discussed below.

The utility function of privacy is chosen according to the specification in Section 4 as:

$$U_{priv}(\alpha_p(t)) = \omega_{p1} \cdot \ln \frac{\omega_{p2}}{\omega_{p3}\alpha_p(t) + \omega_{p4}\alpha_p^2(t)}, \tag{29}$$

where ω_{p1} and ω_{p2} are two positive constants. Taking the derivatives and double derivatives with respective to $\alpha_p(t)$,

$$U'_{priv}(\alpha_p(t)) = -\frac{\omega_{p1}\left(\omega_{p3} + 2\omega_{p4}\alpha_p(t)\right)}{\omega_{p3}\alpha_p(t) + \omega_{p4}\alpha_p^2(t)},$$

$$U''_{priv}(\alpha_p(t)) = \frac{-2\omega_{p1}\omega_{p4} + \omega_{p1}\left(\omega_{p3} + 2\omega_{p4}\alpha_p(t)\right)^2}{\alpha_p^2(t) + \omega_{p3}}.$$

For $\alpha_p(t) > 0$, $U'_{priv}(\alpha_p(t)) < 0$ and $U''_{priv}(\alpha_p(t)) > 0$, which imply decreasingly monotonicity and convexity, respectively. The function $L_{acc}(\alpha_p(t))$ is determined by data fitting from $\{(\alpha_p(t), L_{ep}(t)\}_{t=0}$. In our experiment, we choose $\omega_{p1} = 0.02$, $\omega_{p2} = 6$, $\omega_{p3} = 9$, $\omega_{p4} = 1$.

Figure 5 shows the privacy-accuracy tradeoff of dual variable perturbation at different iterations. From curve fitting, we model the function

$$L_{acc}(\alpha_p(t)) = c_4 \cdot e^{-c_5 \alpha_p(t)} + c_6 \qquad (30)$$

where c_4, c_5, c_6 and are three non-negative constant. From the experimental results, we determine $c_4 = 0.2$, $c_5 = 25$, $c_6 = \min\{L_{ep}(t_1)\}_{t_1=0}$; these values are applicable for all iteraions.

Figure 6 presents the privacy-accuracy tradeoff of primal perturbation at different iterations. We model the function L_{acc} the same as (38). From the plots in Figure 5, we can see that the experimental results of $L_{acc}(\alpha_p(t))$ given $\{\alpha_p(t)\}$ for primal variable perturbation experiences more oscillation than the dual variable perturbation does. For iteration $t > 1$, $c_4 = 20$, $c_5 = 20$, $c_6 = \frac{1}{81}\sum_{t=20}^{100} L_{ep}(t)$. Figure 7 and 8 compare the privacy-accuracy tradeoff of dual and primal variable perturbations in terms of empirical loss and misclassification error rate, respectively. As shown, the reaction of empirical loss of dual variable perturbation is more stable than the primal variable perturbation for most values of $\alpha_p(t)$. Moreover, the dual perturbation gives better error rate for most of α_p, which implies better management of tradeoff between privacy and accuracy.

We determine $\alpha_U = 1$, $c_3 = 0.135$. Let α_p^* be the value such that the corresponding is optimal private. Substitute (37) and (38) to (35), and we then take derivative of $\mathscr{T}$ in (35) with respect to $\alpha_p(t)$, and set it to 0 at α_p^*:

$$\begin{aligned}\omega_{p1}(\omega_{p3} - 2\omega_{p4}\alpha_p(t)) =& c_4 c_5 (\omega_{p3}\alpha_p(t) \\ &+ \omega_{p4}\alpha_p^2(t)) \cdot e^{-c_5\alpha_p(t)}.\end{aligned}$$

The optimum value of $\alpha_p(t)$ at each time t is obtained by soving the above equation.

Figure 7 and 8 shows the privacy-accuracy tradeoff of the final optimum classifier in terms of empirical loss and misclassification error rate (MER). The MER is determined by the fraction of times the trained classifier predict a wrong label. We can see that primal variable peroforms slightly better than dual variable perturbation with respect to the empirical loss.

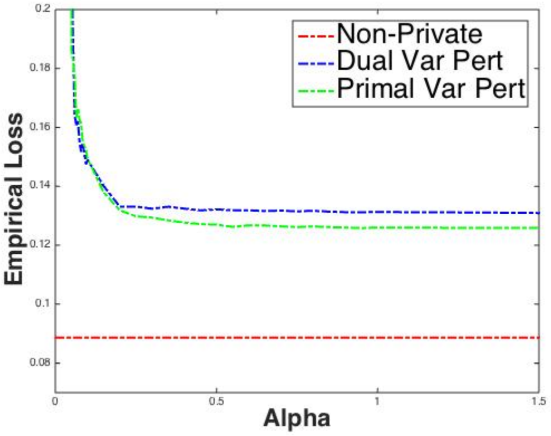


Figure 7. Privacy-Accuracy Tradeoff: Empirical Risk vs. α_p of final optimum output

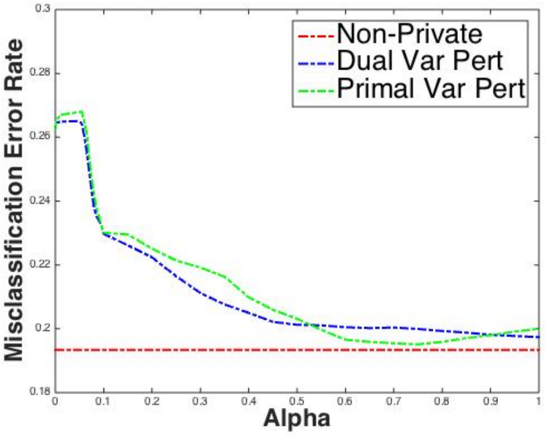


Figure 8. Privacy-Accuracy Tradeoff: Misclassification Error Rate vs. α_p at iteration 100

6. Conclusion

This work developed two ADMM-based algorithms to solve a centralized regularized ERM in a distributed fashion while providing α-differential privacy for the ADMM iterations as well as the final trained output. Thus, the sensitive information stored in the training dataset at each node is protected against both the internal and the external adversares.

Based on distributed training datasets, Algorithm 1 perturbs the dual variable $\lambda_p(t)$ for every node $p \in \mathscr{P}$ at iteration t; For the next iteration, $t+1$, the perturbed version of $\lambda_p(t)$ is involved in the update of primal variable $f_p(t+1)$. Thus the perturbation created at time t provides privacy at time $t+1$. In Algorithm 2, we perturb the primal variable $f_p(t)$, whose noisy version is then released to the neighboring nodes. In this case, the perturbation added at time t make the training process private at time t. Moreover, since the primal variables are shared among all the nieghboring nodes, at time t, the noise directly involved in the optimization of parameter update comes from multiple nodes; as a result, the updated variable has more randomness than the dual perturbation case.

In general, the accuracy decreases as privacy requirements are more stringent. The tradeoff between the privacy and accuracy is studied. Our experiments on real data from UCI Machine Learning Repository show that dual variable perturbation is more robust to the noise than the primal variable perturbation. The dual variable perturbation outperforms the primal case at balancing the privacy-accuracy tradeoff as well as learning quality.

However, there are several conditions for the loss function and the regularizer function, which are summarized in Assumption 1 to 3. The conditions for dual variable perturbation and primal variable perturbation are similar except that the loss funciton is required to be bounded doubly differentiable for dual variable perturbation. Thus, for the loss functions and regularizer functions satisfing Assumption 1 to 3, we recommand the dual variable perturbation algorithm, which can obtain more accurate results while keep the α-differential privacy to a good level.

Appendix A. Alternating Direction Method of Multipliers

Consider a convex optimization problem:

$$\begin{aligned} &\min_x \; g_1(x) + g_2(Ax) \\ &s.t. \; x \in S_1, \; Ax \in S_2, \end{aligned} \tag{31}$$

where $g_1 : R^{s_1} \to R$ and $g_2 : R^{s_2} \to R^1$ are both convex functions, $A \in R^{s_2 \times s_1}$ is a matrix, $S_1 \in R^{s_1}$ and $S_2 \in R^{s_2}$ are two non-empty polyhedral sets. Using an additional auxiliary variable $v \in R^{s_2}$ yields an equivalent form of (33) as:

$$\begin{aligned} &\min_x \; g_1(x) + g_2(v). \\ &s.t. \; Ax = v \\ &\quad x \in S_1, \; v \in S_2 \end{aligned} \tag{32}$$

The corresponding augmented Lagrange function of (34) is:

$$\begin{aligned} L(x,v,\lambda) =& g_1(x) + g_2(v) + \lambda^T(Ax - v) \\ &+ \frac{\eta}{2} \parallel Ax - v \parallel^2, \end{aligned} \tag{33}$$

where $\lambda \in R^{s_2}$ is the Lagrange multiplier corresponding to the constraints $Ax = v$, and $\eta > 0$ is a penalty parameter that controls the effect of constraints violation in (34). The ADMM first minimizes $L(x,v,\lambda)$ with respect to primal variable x, and then keeping the value of x fixed at the just computed value, with respect to the auxiliary variable v. After that, the dual variable λ is updated in a gradient ascending manner. Specifically, ADMM iterates at time $t+1$ is:

$$x(t+1) = \min_x L(x, v(t), \lambda(t)), \tag{34}$$

$$v(t+1) = \min_x L(x(t+1), v, \lambda(t)), \tag{35}$$

$$\lambda(t+1) = \lambda(t) + \eta(Ax(t+1) - v(t+1)), \tag{36}$$

The following theorem states the convergence of ADMM.

Theorem 9. *([51]) Assume (33) is solvable, and either $A^T A$ is nonsingular or S_1 is bounded. Then*

- *a sequence $\{x(t), v(t), \lambda(t)\}$ generated by ADMM iterations (36) to (38) is bounded, and every limit points of $\{x(t)\}$ is an optimal solution of (33): $\{x(t), v(t)\}$ converges to a solution of (33).*
- *$\{\lambda\}$ converges to a solution of the dual problem:*

$$\min_{\lambda \in S_2} G_1(\lambda) + G_2(\lambda), \tag{37}$$

where

$$G_1(\lambda) = \inf_x g_1(x) + \lambda^T Ax,$$

$$G_2(\lambda) = \inf_v g_2(v) - \lambda^T v.$$

Appendix B. Proof of Theorem 1

Proof: (**Theorem 1**)

Let $f_p(t+1)$ be the optimal primal variable with zero duality gap. From the Assumption 1 and 2, we know that both the loss funciton $\mathscr{L}$ and the regularizer $R(\cdot)$ are differentiable and convex, and by using the Karush-Kuhn-Tucker (KKT) optimality condition (stationarity), we have

$$\begin{aligned}0 =& \frac{C^R}{B_p}\sum_{i=1}^{B_p} y_{ip}\mathscr{L}'(y_{ip}f_p(t+1)^T x_{ip})x_{ip} + \rho\nabla R(f_p)\\ &+ 2\big(\frac{C^R}{2B_p}\varepsilon_p(t) + \lambda_p(t)\big) + (\Phi + 2\eta N_p)f_p(t+1)\\ &- \eta\sum_{i\in\mathscr{N}_p}(f_p(t) + f_i(t)),\end{aligned}$$

from which we can establish the relationship between the noise $\varepsilon_p(t)$ and the optimal primal variable $f_p(t+1)$ as:

$$\begin{aligned}\varepsilon_p(t) =& -\sum_{i=1}^{B_p} y_{ip}\mathscr{L}'(y_{ip}f_p(t+1)^T x_{ip})x_{ip} - \frac{B_p}{C^R}\rho\nabla R(f_p)\\ &- \frac{2B_p}{C^R}\lambda_p(t) - \frac{B_p}{C^R}(\Phi + 2\eta N_p)f_p(t+1)\\ &+ \frac{B_p\eta}{C^R}\sum_{i\in\mathscr{N}_p}(f_p(t) + f_i(t)).\end{aligned} \tag{38}$$

Under Assumption 1, the augmented Lagrange function $L_{dual}(t)$ is strictly convex, thus there is a unique value of $f_p(t+1)$ for fixed $\varepsilon_p(t)$ and dataset D_p. The equation (38) shows that for any value of $f_p(t+1)$, we can find a unique value of $\varepsilon_p(t)$ such that $f_p(t+1)$ is the minimizer of L_{dual}. Therefore, given a dataset D_p, the relation between $\varepsilon_p(t)$ and $f_p(t+1)$ is bijective.

Let D_p and D'_p be two datasets with $H_d(D_p, D'_p) = 1$, $(x_i, y_i) \in D_p$ and $(x'_i, y'_i) \in D'_p$ are the corresponding two different data points. Let two matrices $\mathbf{J}_f(\varepsilon_p(t)|D_p)$ and $\mathbf{J}_f(\varepsilon'_p(t)|D'_p)$ denote the Jacobian matrices of mapping from $f_p(t+1)$ to $\varepsilon_p(t)$ and $\varepsilon'_p(t)$, respectively. Then, transformation from noise $f_p(t+1)$ to $\varepsilon_p(t)$ by Jacobian yields:

$$\begin{aligned}&\frac{Q(f_p(t+1)|D_p)}{Q(f_p(t+1)|D'_p)}\\ &= \frac{q(\varepsilon_p(t)|D_p)}{q(\varepsilon'_p(t)|D'_p)}\frac{|det(\mathbf{J}_f(\varepsilon_p(t)|D_p))|^{-1}}{|det(\mathbf{J}_f(\varepsilon'_p(t)|D'_p))|^{-1}},\end{aligned} \tag{39}$$

where $q(\varepsilon_p(t)|D_p)$ and $q(\varepsilon'_p(t)|D'_p)$ are the densities of $\varepsilon_p(t)$ and $\varepsilon'_p(t)$, respectively, given $f_p(t+1)$ when the datasets are D_p and D'_p, respectively.

Therefore, in order to prove the ratio of conditional densities of optimal primal variable is bounded as:

$$\frac{Q(f_p(t)|D)}{Q(f_p(t)|D'_p)} \le e^{\alpha_p(t)},$$

we have to show:

$$\begin{aligned}\frac{q(\varepsilon_p(t)|D_p)}{q(\varepsilon'_p(t)|D'_p)}\cdot\frac{|det(\mathbf{J}_f(\varepsilon_p(t)|D_p))|^{-1}}{|det(\mathbf{J}_f(\varepsilon'_p(t)|D'_p))|^{-1}}\\ \le e^{\alpha_p(t)}.\end{aligned}$$

We first bound the ratio of the determinant of Jacobian matrices, and then the ratio of conditional densities of the noise vectors.

Let x^a be the a-th element of the vector x, and (a,b). Let $\mathbf{E} \in \mathbb{R}^{d\times d}$ be a matrix, then let $\mathbf{E}^{(a,b)}$ denote the (a,b)-th entry of the matrix $\mathbf{E}$. Thus, the (m,n)-th entry of $\mathbf{J}_f(\varepsilon_p(t))$ is:

$$\begin{aligned}\mathbf{J}_f(\varepsilon_p(t))^{(m,n)} =& -\sum_{i=1}^{B_p}(y_i^2\mathscr{L}''(y_i f_p(t+1)^T x_i)x_i^{(m)}x_i^{(n)}\\ &- \frac{B_p}{C^R}\rho\nabla^2 R(f_p(t+1))^{(m,n)}\\ &- \frac{B_p}{C^R}(\Phi + 2\eta N_p)\mathbb{1}(j=k).\end{aligned}$$

Let $\mathbf{J}_f^0(x_i, y_i) = (y_i^2\mathscr{L}''(y_i f_p(t+1)^T x_i)x_i x_i^T$, thn the Jacobian matrix can be expressed as:

$$\begin{aligned}\mathbf{J}_f(\varepsilon_p(t)|D_p) =& -\sum_{i=1}^{B_p}\mathbf{J}_f^0(x_i, y_i) - \frac{B_p}{C^R}\rho\nabla^2 R(f_p(t+1))\\ &- \frac{B_p}{C^R}(\Phi + 2\eta N_p)\mathbf{I}_d.\end{aligned}$$

Let $\mathbf{M} = \mathbf{J}_f^0(x'_i, y'_i) - \mathbf{J}_f^0(x_i, y_i)$, and $\mathbf{H} = -\mathbf{J}_f(\varepsilon_p(t)|D_p)$, and thus $\mathbf{J}_f(\varepsilon_p(t)|D'_p) = -(\mathbf{M} + \mathbf{H})$. Let $h_j(\mathbf{W})$ be the j-th largest eigenvalue of a symmetric matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$ with rank θ Then we have the following fact:

$$det(\mathbf{I}+\mathbf{W}) = \prod_{j}^{\theta}(1+h_j(\mathbf{W})).$$

Since the matrix $x_i x_i^T$ has rank 1, then matrix $\mathbf{M}$ has rank at most 2; thus matrix $\mathbf{H}^{-1}\mathbf{M}$ has rank at most 2; therefore, we have:

$$\begin{aligned} det(\mathbf{H}+\mathbf{M}) &= det(\mathbf{H}) \cdot det(\mathbf{I}+\mathbf{H}^{-1}\mathbf{M}) \\ &= det(\mathbf{H}) \cdot (1+h_1(\mathbf{H}^{-1}\mathbf{M}))(1+h_2(\mathbf{H}^{-1}\mathbf{M}). \end{aligned}$$

Thus, the ratio of determinants of the Jacobian matrices can be expressed as:

$$\begin{aligned} \frac{|det(\mathbf{J}_f(\varepsilon_p(t)|D_p))|^{-1}}{|det(\mathbf{J}_f(\varepsilon'_p(t)|D'_p))|^{-1}} =& \frac{|det(\mathbf{H}+\mathbf{M})|}{|det(\mathbf{H})|} \\ =& |det(\mathbf{I}+\mathbf{H}^{-1}\mathbf{M})| \\ =& (1+h_1(\mathbf{H}^{-1}\mathbf{M}))(1+h_2(\mathbf{H}^{-1}\mathbf{M}) \\ =& |1+h_1(\mathbf{H}^{-1}\mathbf{M})+h_2(\mathbf{H}^{-1}\mathbf{M}) \\ & +h_1(\mathbf{H}^{-1}\mathbf{M})h_2(\mathbf{H}^{-1}\mathbf{M})|. \end{aligned}$$

Based on Assumption 2, all the eigenvalues of $\nabla^2 R(f_p(t+1))$ is greater than 1 [32]. Thus, from Assumption 1, matrix $\mathbf{H}$ has all eigenvalues at least $\frac{B_p}{C^R}\left(\rho+\Phi+2\eta N_p\right)$. Therefore, $|h_1(\mathbf{H}^{-1}\mathbf{M})| \leq \frac{|h_i(\mathbf{M})|}{\frac{B_p}{C^R}\left(\rho+\Phi+2\eta N_p\right)}$.

Let $\sigma_i(\mathbf{M})$ be the non-negative singular value of the symmetric matrix $\mathbf{M}$. According to [3], we have the inequality

$$\sum_i |h_i(\mathbf{M})| \leq \sum_i \sigma_i(\mathbf{M}). \tag{40}$$

Thus, we have

$$|h_1(\mathbf{M})|+|h_2(\mathbf{M})| \leq \sigma_1(\mathbf{M})+\sigma_2(\mathbf{M}).$$

Let $\parallel X \parallel_\Sigma = \sum_i \sigma_i$ be the trace norm of X. Then according to the *trace norm inequality*, we have:

$$\parallel \mathbf{M} \parallel_\Sigma \leq \parallel \mathbf{J}^0(x'_i,y'_i) \parallel_\Sigma + \parallel -\mathbf{J}^0(x_i,y_i) \parallel_\Sigma .$$

As a result, based on the upper bounds from Assumption 1 and 3, we have:

$$\begin{aligned} |h_1(\mathbf{M})|+|h_2(\mathbf{M})| &\leq \parallel \mathbf{J}^0(x'_i,y'_i) \parallel_\Sigma + \parallel -\mathbf{J}^0(x_i,y_i) \parallel_\Sigma \\ &\leq |(y_i^2 \mathscr{L}''(y_i f_p(t+1)^T x_i)| \cdot \parallel x_i \parallel \\ &+ |(y_i'^2 \mathscr{L}''(y'_i f_p(t+1)^T x'_i)| \cdot \parallel x'_i \parallel \\ &\leq 2c_1, \end{aligned}$$

which follows $h_1(\mathbf{M})h_2(\mathbf{M}) \leq c_1^2$. Finally, the ratio of determinants of Jacobian matrices is bounded as:

$$\begin{aligned} \frac{|det(\mathbf{J}_f(\varepsilon_p(t)|D_p))|^{-1}}{|det(\mathbf{J}_f(\varepsilon'_p(t)|D'_p))|^{-1}} &\leq (1+\frac{c_1}{\frac{B_p}{C^R}\left(\rho+\Phi+2\eta N_p\right))})^2 \\ &= e^{\overline{\alpha}}, \end{aligned} \tag{41}$$

where $\overline{\alpha} = \ln\left(1+\frac{c_1}{\frac{B_p}{C^R}\left(\rho+\Phi+2\eta N_p\right)}\right)^2$.

Now, we bound the ratio of densities of $\varepsilon_p(t)$. Let $sur(E)$ be the surface area of the sphere in d dimension with radius E, and $sur(E) = sur(1)\cdot E^{d-1}$. We can write:

$$\begin{aligned} \frac{q(\varepsilon_p(t)|D_p)}{q(\varepsilon'_p|D'_p)} &= \frac{\mathscr{K}(\varepsilon_p(t))\frac{\|\varepsilon_p(t)\|^{d-1}}{sur(\|\varepsilon_1(t)\|)}}{\mathscr{K}(\varepsilon'_p(t))\frac{\|\varepsilon'_p(t)\|^{d-1}}{sur(\|\varepsilon'_p(t)\|)}} \\ &\leq e^{\zeta_p(t)(\|\varepsilon'_p(t)\|-\|\varepsilon_p(t)\|)} \\ &\leq e^{\hat{\alpha}}, \end{aligned} \tag{42}$$

where $\hat{\alpha}$ is a constant satisfying the above inequality. Since we want to bound the ratio of densities of $f_p(t+1)$

$$\frac{Q(f_p(t+1)|D_p)}{Q(f_p(t+1)|D'_p)} \leq e^{\alpha_p(t)},$$

we need $\hat{\alpha} \leq \alpha_p(t)-\overline{\alpha}$. For non-negative Φ, let

$$\hat{\alpha} = \alpha_p(t) - \ln\left(1+\frac{c_1}{\frac{B_p}{C^R}\left(\rho+2\eta N_p\right)}\right)^2.$$

If $\hat{\alpha} > 0$, then we fix $\Phi = 0$, and thus $\hat{\alpha} = \alpha_p(t) - \overline{\alpha}$. Otherwise, let $\Phi = \frac{c_1}{\frac{B_p}{C^R}(e^{\alpha_p(t)/4}-1)} - \rho - 2\eta N_p$, and $\hat{\alpha} = \frac{\alpha_p(t)}{2}$ which makes $\hat{\alpha} = \alpha_p(t)-\overline{\alpha}$. Therefore, we can have

$$\frac{|det(J_f(b_1|D_p))|^{-1}}{|det(J_f(b_2|D'_p))|^{-1}} \leq e^{\alpha_p(t)-\hat{\alpha}}.$$

From the upper bounds stated in Assumption 1 and 3, the l_2 norm of the difference of ε_1 and ε_2 can be bounded as:

$$\begin{aligned} \parallel \varepsilon'_p(t)-\varepsilon_p(t) \parallel =& \sum_{i=1}^{B_p} \parallel y_{ip}\mathscr{L}'(y'_{ip} f_p(t+1)^T x'_{ip})x'_{ip} \\ & -(y_{ip}\mathscr{L}'(y_{ip} f_p(t+1)^T x_{ip})x_{ip} \parallel \\ \leq& 2. \end{aligned}$$

Thus,$\| \varepsilon'_p(t) \| - \| \varepsilon_p(t) \| \leq \| \varepsilon'_p(t) - \varepsilon_p(t) \| \leq 2$. Therefore, by selecting $\zeta_p(t) = \frac{\hat{\alpha}}{2}$, we can bound the ratio of conditional densities of $f_p(t+1)$ as:

$$\frac{Q(f_p(t+1)|D_p)}{Q(f_p(t+1)|D'_p)} \leq e^{\alpha_p(t)},$$

and prove that the dual variable perturbation can provide $\alpha_p(t)$-differential privacy. □

Appendix C. Proof of Corollary 1.2

Proof: (**Corollary 1.2**) We prove this corollary by induction. For $c_2 = 1$, it is true since this is exactly the case of Theorem 1. Suppose Corollary 1.2 is held for $H_d(D_p, D'_p) = c_2$. Let $H_d(D_p, D'_p) = c_2 + 1$. Clearly, there must exist a dataset D''_p such that $H_d(D_p, D''_p) = 1$, and $H_d(D'_p, D''_p) = c_2$. Thus, from (13), we have:

$$\begin{aligned}\frac{Q(f_p(t)|D_p)}{Q(f_p(t)|D'_p)} &= \frac{Q(f_p(t)|D_p)}{Q(f_p(t)|D''_p)} \cdot \frac{Q(f_p(t)|D''_p)}{Q(f_p(t)|D'_p)} \\ &\leq e^{\alpha_p(t)} e^{c_2 \alpha_p(t)} = e^{(c_2+1)\alpha_p(t)}.\end{aligned} \tag{43}$$

Therefore, the induction hypothesis is true and Corollary 1.2 is proven. □

Appendix D. Proof of Theorem 2

Proof: (**Theorem 2**)

Let D_p and D'_p be two datasets with $H_d(D_p, D'_p) = 1$. Since only $V_p(t)$ is released, then our target is to prove the following:

$$\frac{Q(V_p(t+1)|D_p)}{Q(V_p(t+1)|D'_p)} \leq e^{\alpha_p(t)}. \tag{44}$$

From (23), we have:

$$\frac{Q(V_p(t+1)|D_p)}{Q(V_p(t+1)|D'_p)} = \frac{\mathscr{K}(\varepsilon_p(t))}{\mathscr{K}(\varepsilon'_p(t))} = \frac{e^{-\zeta_p(t)\|\varepsilon_p(t)\|}}{e^{-\zeta_p(t)\|\varepsilon'_p(t)\|}}. \tag{45}$$

Therefore, in order to make the model to provide $\alpha_p(t)$-differential privacy, we need to find a $\zeta_p(t)$ that satisfies

$$\zeta_p(t)(\| \varepsilon_p(t) \| - \| \varepsilon'_p(t) \|) \leq \alpha_p(t). \tag{46}$$

Let $V^A = \arg\min_{V_p} L_{prim}(t|D_p)$, and $V^B = \arg\min_{V_p} L_{prim}(t|D'_p)$, where $L_{prim}(t|D)$ is the augmented Lagrange function for primal variable perturbation given dataset D.

Let F, G be defined at each node $p \in \mathscr{P}$ as:

$$F(V_p(t)) = L_{prim}(t|D_p),$$

$$G(V_p(t)) = L_{prim}(t|D'_p) - L_{prim}(t|D_p).$$

Thus, $G(V_p) = \frac{C^R}{B_p} \sum_{i=1}^{B_p} (\mathscr{L}(y'_{ip} V_p^T x'_{ip}) - \mathscr{L}(y_{ip} V_p^T x_{ip}))$. According to Assumption 2, we can imply that $L_{prim}(t|D_p) = F(V_p(t))$ and $L_{prim}(t|D'_p) = F(V_p(t)) + G(V_p(t))$ are both ρ-strong convex. Differentating $G(V_p(t))$ with respect to $V_p(t)$ gives:

$$\begin{aligned}\nabla G(V_p) =& \frac{C^R}{B_p} (y'_{ip} \mathscr{L}'(y'_{ip} V_p^T x'_{ip}) x'_{ip} \\ & - (y_{ip} \mathscr{L}(y_{ip} V_p^T x_{ip}) x_{ip}.\end{aligned}$$

From Assumption 1 and 3, $\| \nabla G(V_p) \| \leq \frac{2C^R}{B_p}$. From definitions of V^A and V^B, we have:

$$\nabla F(V^A) = \nabla F(V^B) + \nabla F(V^B) = 0$$

From Lemma 14 in [52] and the fact that $F(\cdot)$ is ρ-strongly convex, weh have the following inequality:

$$\langle \nabla F(V^A) - F(V^B), V^A - V^B \rangle \geq \rho \| V^A - V^B \|^2;$$

therefore, Cauchy-Schwarz inequality yields:

$$\begin{aligned}&\| V^A - V^B \| \cdot \| \nabla G(V^B) \| \\ &\geq (V^A - V^B)^T \nabla G(V^B) \\ &= \langle \nabla F(V^A) - F(V^B), V^A - V^B \rangle \\ &\geq \rho \| V^A - V^B \|^2 .\end{aligned}$$

Dividing both sides by $\rho \| V^A - V^B \|$ gives:

$$\| V^A - V^B \| \leq \frac{1}{\rho} \| \nabla G(V^B) \| \leq \frac{2C^R}{\rho B_p}. \tag{47}$$

From (23), we have

$$\| V^A - V^B \| \leq \frac{1}{\rho} \| \nabla G(V^B) \| = \| \varepsilon_p(t) - \varepsilon'_p(t) \| .$$

Thus, we can bound

$$\begin{aligned}\zeta_p(t)(\| \varepsilon_p(t) \| - \| \varepsilon'_p(t) \|) &\leq \zeta_p(t)(\| \varepsilon_p(t) - \varepsilon'_p(t) \|) \\ &\leq \frac{2C^R}{B_p \rho} \zeta_p(t)\end{aligned}$$

Therefore, by choosing $\zeta_p(t) = \frac{\rho B_p \alpha_p(t)}{2C^R}$, the inequality (43) holds; thus primal variable perturbation is dynamic α_p-differential private at each node p.

□

Appendix E. Proof of Theorem 3

Proof: **(Theorem 3)** Let

$$\hat{f}_p(t+1) = \arg\min \hat{Z}(f_p,t),$$
$$\tilde{f}_p(t+1) = \arg\min Z_p(f_p,t|D_p),$$

and let $f_p^{non}(t+1)$ be the estimated optimum that is practical result of the algorithm. We assume that $f_p^{non}(t+1)$ is very close to the actually so that $Z_p(f_p^{non}(t+1),t|D_p) - Z_p(\tilde{f}_p(t+1),t|D_p) \approx 0$. For the non-private ERM, Shalev-Shwartz and Srebro in [53] show that for a specific reference classifier $f_0(t+1)$ at time $t+1$ such that $\hat{C}(f^0(t+1)) = C^*E$, we have:

$$\begin{aligned}\hat{C}(f_p^{non}(t+1)) =& \hat{C}^* \\ &+ \big(\hat{Z}(f_p^{non}(t+1),t) - \hat{Z}(\hat{f}_p(t+1),t)\big) \\ &+ \big(\hat{Z}(\hat{f}_p(t+1),t) - \hat{Z}(f_p^0(t+1),t)\big) \\ &+ \frac{\rho}{2} \parallel f_p^0(t+1) \parallel^2 - \frac{\rho}{2} \parallel f_p^{non}(t+1) \parallel^2 .\end{aligned}$$

From Sridharan et al. [54], we have, with probability at least $1-\delta$

$$\begin{aligned}&\hat{Z}(f_p^{non}(t+1),t) - \hat{Z}(\hat{f}_p(t+1),t) \\ &\leq 2\big(Z_p(f_p^{non}(t+1),t|D_p) - Z_p(\tilde{f}_p(t+1),t|D_p)\big) \\ &+ \mathscr{O}\Big(C^R \frac{\ln(\frac{1}{\delta})}{B_p \rho}\Big).\end{aligned}$$

Since $Z_p(f_p^{non}(t+1),t|D_p) - Z_p(\tilde{f}_p(t+1),t|D_p) \approx 0$, then

$$\begin{aligned}&\hat{Z}(f_p^{non}(t+1),t) - \hat{Z}(\hat{f}_p(t+1),t) \\ &\leq \mathscr{O}\Big(C^R \frac{\ln(\frac{1}{\delta})}{B_p \rho}\Big).\end{aligned}$$

If we choose $\rho \leq \frac{\alpha_{acc}}{\| f_p^0(t+1) \|^2}$, then

$$\frac{\rho}{2} \parallel f_p^0(t+1) \parallel^2 - \frac{\rho}{2} \parallel f_p^{non}(t+1) \parallel^2 \leq \frac{\alpha_{acc}}{2}.$$

Thus

$$\hat{C}(f_p^{non}(t+1)) \leq \hat{C}^* + \mathscr{O}\Big(C^R \frac{\ln(\frac{1}{\delta})}{B_p \rho}\Big) + \frac{\alpha_{acc}}{2}.$$

Therefore, we can find the value of B_p by solving

$$\mathscr{O}\Big(C^R \frac{\ln(\frac{1}{\delta})}{B_p \rho}\Big) + \frac{\alpha_{acc}}{2} \leq \alpha_{acc}$$

We get:

$$B_p > \beta_{non} \max\left(\frac{C^R \parallel f_p^0(t+1) \parallel^2 \ln(\frac{1}{\delta})}{\alpha_{acc}^2}\right).$$

If we determine different reference classifier $f_p^0(t+1)$ at different time, then we need to find the maximum value across the time and among different value of $\parallel f_p^0(t+1) \parallel$:

$$B_p > \beta_{non} \max\left(\Big\{\frac{C^R \parallel f_p^0(t+1) \parallel^2 \ln(\frac{1}{\delta})}{\alpha_{acc}^2}\Big\}_{t=1}\right).$$

Let $F_p(t+1) = \arg\min_{f_p} L_{non}(f_p,t|D_p)$. Since

$$\hat{C}(F_p(t+1)) = \hat{C}(f_p^{non}(t+1)) + \Delta^{non}(t),$$

then

$$\hat{C}(F_p(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc} + \Delta^{non}(t),$$

with probability no less than $1-\delta$. □

Appendix F. Proof of Theorem 4

Proof: **(Theorem 4)** First we define the following optimal variables:

$$\hat{f}_p(t+1) = \arg\min \hat{Z}(f_p,t),$$
$$f_p^{non}(t+1) = \arg\min Z_p(f_p,t|D_p),$$
$$f_p^*(t+1) = \arg\min Z_{dual}(f_p,t|D_p),$$

and as defined in Theorem 3, $\hat{C}(f_p^0(t+1)) = \hat{C}^*$ at time $t+1$. We use the analysis of Shalev-Shwartz and Srebro in [53] (also see the work of Chaudhuri et al. in [11]), and have the follows:

$$\begin{aligned}\hat{C}(f_p^*(t+1)) =& \hat{C}(f_p^0(t+1)) \\ &+ \big(\hat{Z}(f_p^*(t+1),t) - \hat{Z}(\hat{f}_p(t+1),t)\big) \\ &+ \big(\hat{Z}(\hat{f}_p(t+1),t) - \hat{Z}(f_p^0(t+1),t)\big) \\ &+ \frac{\rho}{2} \parallel f_p^0(t+1) \parallel^2 - \frac{\rho}{2} \parallel f_p^*(t+1) \parallel^2 .\end{aligned} \tag{48}$$

Now we bound each terms in the right hand side of (47) as follows. From Assumption 1, we have

$\mathscr{L}' \leq c_1$. By choosing $B_p > \frac{5c_1C^R\|f_p^0(t+1)\|^2}{\alpha_{acc}\alpha_p(t)}$, and $\rho > \frac{\alpha_{acc}}{2\|f_p^0(t+1)\|^2}$, and since $\alpha_p(t) \leq 1$, we have:

$$\begin{aligned}\hat{\alpha} =&\alpha_p(t) - \ln\Big(1 + \frac{c_1}{\frac{B_p}{C^R}\big(\rho + 2\eta N_p\big)}\Big)^2\\ >&\alpha_p(t) - \ln(1 + \frac{c_1C^R}{B_p\rho})^2\\ >&\alpha_p(t) - \ln(1 + \frac{2\alpha_p(t)}{5})^2\\ >&\alpha_p(t) - \frac{4\alpha_p(t)}{5} = \frac{\alpha_p(t)}{5}.\end{aligned}$$

Then, according to Algorithm 1, we choose the corresponding $\zeta_p(t) = \frac{\alpha_p(t)}{4}$ because $\hat{\alpha} > 0$. Let Λ be the event

$$\begin{aligned}\Lambda :=\Big\{&Z_p(f_p^*(t+1),t|D_p) \leq Z_p(f_p^{non}(t+1),t|D_p)\\ &+\frac{16d^2\Big(\ln(\frac{d}{\delta})\Big)^2}{\rho B_p^2\alpha_p(t)^2}\Big\}.\end{aligned}$$

Since $\hat{\alpha} > \frac{\alpha_p(t)}{2} > 0$, and applying Lemma 11 yields:

$$\mathbf{P}_{\varepsilon_p(t)}\Big(\Lambda\Big) \geq 1 - \delta.$$

From the work of Sridharan et al. in [54], the following inequality holds with probability $1-\delta$

$$\begin{aligned}\hat{Z}(f_p^*(t+1)) - \hat{Z}(\hat{f}_p(t+1)) \leq 2\Big(&Z_p(f_p^*(t+1),t|D_p)\\ &-Z_p(f_p^{non}(t+1),t|D_p)\Big)\\ &+\mathscr{O}\Big(\frac{\ln(\frac{1}{\delta})}{B_p\rho}\Big)\\ \leq&\frac{32d^2\Big(\ln(\frac{d}{\delta})\Big)^2}{\rho B_p^2\alpha_p(t)^2}\\ &+\mathscr{O}\Big(\frac{\ln(\frac{1}{\delta})}{B_p\rho}\Big).\end{aligned}$$

The big-$\mathscr{O}$ notation hides only fixed numerical constants, which depend on the derivative of the loss function and the upper bounds of the data points shown in Assumption 3. Combining the above two processes, $\hat{Z}(f_p^*(t+1)) - \hat{Z}(\hat{f}_p(t+1))$ is bounded as shown above with probability $1-2\delta$.

From the definitions of $f_p^0(t+1)$ and $\hat{f}_p(t+1)$, we can get $\hat{Z}(\hat{f}_p(t+1),t) - \hat{Z}(f_p^0(t+1),t) < 0$. Since $P \geq 1$, then by selecting $\rho = \frac{\alpha_{acc}}{\|f_p^0(t+1)\|^2}$, we can bound

$$\frac{\rho}{2} \parallel f_p^0(t+1) \parallel^2 - \frac{\rho}{2} \parallel f_p^*(t+1) \parallel^2 \leq \frac{\alpha_{acc}}{2}.$$

Therefore, from (47), we have:

$$\begin{aligned}\hat{C}(f_p^*(t+1)) \leq& C_E^* + \frac{32d^2\Big(\ln(\frac{d}{\delta})\Big)^2}{\rho B_p^2\alpha_p(t)^2}\\ &+\mathscr{O}\Big(C^R\frac{\ln(\frac{1}{\delta})}{B_p\rho}\Big) + \frac{\alpha_{acc}}{2},\end{aligned}$$

with $\rho = \frac{6\alpha_{acc}}{\|f_p^0(t+1)\|^2}$. The lower bounds of B_p is determined by solving the following:

$$\frac{32d^2\Big(\ln(\frac{d}{\delta})\Big)^2}{\rho B_p^2\alpha_p(t)^2} + \mathscr{O}\Big(C^R\frac{\ln(\frac{1}{\delta})}{B_p\rho}\Big) + \frac{\alpha_{acc}}{2} \leq \alpha_{acc}.$$

□

Lemma 10. *Let Z be a gamma random variable with density function $\Gamma(k,\theta)$, where k is an integer, and let $\delta > 0$. Then we have:*

$$\mathbb{P}(Z < k\theta\ln(\frac{k}{\delta})) \geq 1 - \delta.$$

Proof: **(Lemma 10)** Since Z is a gamma random variable $\Gamma(k,\theta)$, then we can express Z as follows:

$$Z = \sum_{i=1}^{k} Z_i,$$

where $\{Z_i\}_{i=1}^k$ are independent exponential random variable with density function $Exp(\frac{1}{\theta})$; thus, for each Z_i we have:

$$\mathbb{P}(Z_i \leq \theta ln(\frac{k}{\delta})) = 1 - \frac{\delta}{k}.$$

Since $Z_{ii=1}$ are independent, we have:

$$\begin{aligned}\mathbb{P}(Z < k\theta\ln(\frac{k}{\delta})) =& \prod_{i=1}^{k}\mathbb{P}(Z_i \leq \theta ln(\frac{k}{\delta}))\\ =& (1-\frac{\delta}{k})^k \geq 1 - \delta.\end{aligned}$$

□

Lemma 11. *Let $\hat{\alpha} > 0$, and $f_p^*(t+1) = \arg\min Z_{dual}(f_p,t|D_p)$, and $f_p^{non}(t+1) =$*

$\arg\min Z_p(f_p,t|D_p)$. *Let* Λ *be the event*

$$\Lambda := \Big\{ Z_p(f_p^*(t+1),t|D_p) \leq Z_p(f_p^{non}(t+1),t|D_p) + \frac{16d^2\Big(\ln(\frac{d}{\delta})\Big)^2}{\rho B_p^2 \alpha_p(t)^2} \Big\}.$$

Under Assumption 1 and 2, we have:

$$\boldsymbol{P}_{\varepsilon_p(t)}\Big(\Lambda\Big) \geq 1-\delta.$$

The probability $\boldsymbol{P}_{\varepsilon_p(t)}$ *is taken over the noise vector* $\varepsilon_p(t)$.

Proof: **(Lemma 11)** Since $\hat{\alpha} > 0$, $\Phi = 0$; then $f_p^*(t+1) = \arg\min Z_{dual}(f_p,t|D_p)$ can be expressed as:

$$f_p^*(t+1) = \arg\min \Big(Z_p(f_p,t|D_p) + 2\varepsilon_p(t)^T f_p \Big).$$

Thus, we have:

$$\begin{aligned} Z_p(f_p^*(t+1),t|D_p) \leq{}& Z_p(f_p^{non}(t+1),t|D_p) \\ &+ \frac{C^R}{B_p}\varepsilon_p(t)^T(f_p^{non}(t+1) - f_p^*(t+1)). \end{aligned}$$

Firstly, we bound the l_2-norm $\parallel f_p^{non}(t+1) - f_p^*(t+1) \parallel$. We use the similar procedure to establish (46) in Appendix D by setting $F(Y) = Z_p(Y,t|D_p)$ and $G(Y) = \frac{C^R}{B_p}\varepsilon_p(t)$; thus, based on Assumption 1 and 2, we have:

$$\begin{aligned} \parallel f_p^{non}(t+1) - f_p^*(t+1) \parallel \leq & \frac{1}{\rho} \parallel \nabla\big(2\varepsilon_p(t)^T f_p\big) \parallel \\ \leq & \frac{C^R \parallel \varepsilon_p(t) \parallel}{B_p \rho}. \end{aligned}$$

Cauchy-Schwarz inequality yields:

$$\begin{aligned} & Z_p(f_p^*(t+1),t|D_p) - Z_p(f_p^{non}(t+1),t|D_p) \\ \leq & \parallel Z_p(f_p^*(t+1),t|D_p) - Z_p(f_p^{non}(t+1),t|D_p) \parallel \\ \leq & \frac{2}{B_p} \parallel \varepsilon_p(t)^T(f_p^{non}(t+1) - f_p^*(t+1) \parallel \\ \leq & \frac{\big(C^R\big)^2 \parallel \varepsilon_p(t) \parallel^2}{B_p^2 \rho}. \end{aligned}$$

Since the noise vector $\varepsilon_p(t)$ is drawn from

$$\mathscr{K}(\varepsilon) \sim e^{-\zeta_p(t)\|\varepsilon\|},$$

then $\parallel \varepsilon_p(t) \parallel$ is drawn from $\Gamma(d, \frac{1}{\zeta_p(t)}) = \Gamma(d, \frac{2}{\hat{\alpha}})$. Then by using Lemma 10 with $\parallel \varepsilon_p(t) \parallel \leq \frac{2d\ln(\frac{d}{\delta})}{\hat{\alpha}}$, we have:

$$\begin{aligned} & L_{nonP}(f_p^*(t+1),t|D_p) - L_{nonP}(f_p^{non}(t+1),t|D_p) \\ & \leq \frac{4d^2\Big(\ln(\frac{d}{\delta})\Big)^2}{\rho B_p^2 \alpha_p(t)^2}. \end{aligned}$$

with probability no less than $1-\delta$. □

Appendix G. Proof of Theorem 5

Proof: **(Theorem 5)** Similar to the proof of Theorem 4 in Appendix F, we define the following optimal variables:

$$\hat{f}_p(t+1) = \arg\min Z_E(f_p,t),$$

$$f_p^{non}(t+1) = \arg\min Z_p(f_p,t|D_p),$$

$$f_p^*(t+1) = \arg\min Z_{prim}(f_p,t|D_p).$$

Let $\hat{C}(f_p^0(t+1)) = \hat{C}^*$ at time $t+1$. We use the analysis of Shalev-Shwartz and Srebro in [53] (also see the work of Chaudhuri et al. in [11]), and have the follows,

$$\begin{aligned} \hat{C}(f_p^*(t+1)) = & \hat{C}(f_p^0(t+1)) \\ & + \big(\hat{Z}(f_p^*(t+1),t) - \hat{Z}(\hat{f}_p(t+1),t)\big) \\ & + \big(\hat{Z}(\hat{f}_p(t+1),t) - \hat{Z}(f_p^0(t+1),t)\big) \\ & + \frac{\rho}{2} \parallel f_p^0(t+1) \parallel^2 - \frac{\rho}{2} \parallel f_p^*(t+1) \parallel^2 . \end{aligned} \tag{49}$$

According to Theorem 2, we choose $\zeta_p(t) = \frac{\rho B_p \alpha_p(t)}{2C^R} > 0$. Thus, applying Lemma 14, we have:

$$\begin{aligned} & Z_p(f_p^*(t+1),t|D_p) - Z_p(f_p^{non}(t+1),t|D_p) \\ & \leq \frac{16\big(C^R\big)^2 \eta^2 N_p^2 d^2 \big(\ln(\frac{d}{\delta})\big)^2}{\rho^3 B_p^2 \alpha_p(t)^2}, \end{aligned}$$

with probability no smaller than $1-\delta$. Then we use the result of Sridharan et al. in [54], with probability no smaller than $1-\delta$:

$$\begin{aligned}\hat{Z}(f_p^*(t+1))-\hat{Z}(\hat{f}_p(t+1)) \leq 2\Big(&Z_p(f_p^*(t+1),t|D_p)\\ &-Z_p(f_p^*(t+1),t|D_p)\Big)\\ &+\mathscr{O}\Big(\frac{\ln(\frac{d}{\delta})}{B_p\rho}\Big)\\ \leq&\frac{32\big(C^R\big)^2\eta^2N_p^2d^2\big(\ln(\frac{d}{\delta})\big)^2}{\rho^3B_p^2\alpha_p(t)^2}\\ &+\mathscr{O}\Big(\frac{\ln(\frac{1}{\delta})}{B_p\rho}\Big).\end{aligned}$$

Combining the above two processes, we have the probability no smaller than $1-2\delta$.

In order to bound the last two terms in (48), we select $\rho=\frac{\alpha_{acc}}{\|f_p^0(t+1)\|^2}$; as a result,

$$\frac{\rho}{2}\parallel f_p^0(t+1)\parallel^2-\frac{\rho}{2}\parallel f_p^*(t+1)\parallel^2\leq\frac{\alpha_{acc}}{2}.$$

From the definitions of $\hat{f}_p(t+1)$ and $f_p^0(t+1)$, we have:

$$\hat{Z}(\hat{f}_p(t+1),t)-\hat{Z}(f_p^0(t+1),t)\leq 0.$$

The value of B_p is determined such that

$$\hat{C}(f_p^*(t+1))\leq\hat{C}^*+\alpha_{acc}.$$

Therefore, we find the bounds of B_p by solving

$$\frac{32\big(C^R\big)^2\eta^2N_p^2d^2\big(\ln(\frac{d}{\delta})\big)^2}{\rho^3B_p^2\alpha_p(t)^2}+\mathscr{O}\Big(\frac{C^R\ln(\frac{1}{\delta})}{B_p\rho}\Big)+\frac{\alpha_{acc}}{2}\leq\alpha_{acc},$$

with $\rho=\frac{\alpha_{acc}}{\|f_p^0(t+1)\|^2}$. □

Lemma 12. Let f and g be two probability density functions. If there exists a constant c_6 such that $f(x)=c_6g(x)$ for all $x\in\mathbb{R}^d$, then:

$$f(x)=g(x).$$

Proof: **(Lemma 12)** From the property of probability density function, we have:

$$\begin{aligned}1=&\int_{-\infty}^{\infty}f(x)dx\\ =&c_6\cdot\int_{-\infty}^{\infty}g(x)dx\\ =&c_6.\end{aligned}$$

Therefore, $c_6=1$, and $f(x)=g(x)$. □

Lemma 13. Let $\{Z_j\}_{j=1}^K$ be independent gamma random variables with density $\Gamma(\beta_j,h)$. Then $Z=\sum_{j=1}^K$ is a gamma random variable with $\Gamma(\sum_j^K\beta_j,h)$

Proof: **(Lemma 13)** We prove Lemma 11 by induction. First, we show it is true for K = 2. Let $g(\cdot)=\Gamma(\beta_1+\beta_2,h)$, and $f_{Z_1+Z_2}(z)$ be the joint probability density of Z_1 and Z_2. Then, we have $f_{Z_1+Z_2}(z)=0=g(z)$ for all $z<0$. Let $r>0$, then

$$\begin{aligned}f_{Z_1+Z_2}(r)=&(f_{Z_1}*f_{z_2})(r)\\ =&\int_0^r f_{Z_1}(x)*f_{Z_2}(r-x)dx\\ =&\frac{1}{\Gamma(\beta_1)\Gamma(\beta_2)}\int_0^r he^{-hx}(hx)^{\beta_1-1}he^{h(r-x)}\\ &\cdot\big(h(r-x))\big)^{\beta_2-1}dx\\ =&\frac{h^{\beta_1+\beta_2}e^{-hr}}{\Gamma(\beta_1)\Gamma(\beta_2)}\int_0^r x^{\beta_1-1}(r-x)^{\beta_2-1}dx\\ =&\frac{he^{-hr}h^{\beta_1+\beta_2-1}}{\Gamma(\beta_1)\Gamma(\beta_2)}\int_0^1(ry)^{\beta_1-1}\big(r(1-y)\big)^{\beta_2-1}rdy\\ =&\frac{he^{-hr}(rh)^{\beta_1+\beta_2-1}}{\Gamma(\beta_1+\beta_2)}\cdot\frac{\Gamma(\beta_1+\beta_2)}{\Gamma(\beta_1)\Gamma(\beta_2)}\\ &\cdot\int_0^1 y^{\beta_1-1}(1-y)^{\beta_2-1}dy\\ =&g(r)\cdot c_6,\end{aligned}\tag{50}$$

where $c_6=\frac{\Gamma(\beta_1+\beta_2)}{\Gamma(\beta_1)\Gamma(\beta_2)}\int_0^1 y^{\beta_1-1}(1-y)^{\beta_2-1}dy$ is a constant. From Lemma 11, we prove that $f_{Z_1+Z_2}(z)=g(z)$.

Now we assume it is also true for $K=K$. We next prove it is also true for $K'=K+1$. Let $f^K(z)=f_{\sum_{j=1}^K}(z)$, and $g^K(z)=\Gamma(\sum_{j=1}^K\beta_j,h)$. Then we have:

$$f^{K+1}(z)=(f^K(z)*f_{Z_{K+1}})(z).$$

By replacing $f_{Z_1}(z)$ by $f^K(z)$, $f_{Z_2}(z)$ by $f_{Z_{K+1}}(z)$, β_1 by $\sum_{j=1}^K\beta_j$, and β_2 by β_{K+1} in (47), we can prove

$$f^{K+1}(z)=g^{K+1}(z)\cdot c_7,$$

where $c_7=\frac{\Gamma(\sum_{j=1}^K\beta_j+\beta_{K+1})}{\Gamma(\sum_{j=1}^K\beta_j)\Gamma(\beta_{K+1})}\int_0^1 y^{\sum_{j=1}^K\beta_j-1}(1-y)^{\beta_{K+1}-1}dy$ is a constant. Thus form Lemma 12, $f^{K+1}(z)=g^{K+1}(z)$ Therefore, by induction, Lemma 13 is proved.

□

The following Lemma is analogous to Lemma 11

Lemma 14. *Let $\zeta_p(t) > 0$, and $f_p^*(t+1) = \arg\min Z_{prim}(f_p,t|D_p)$, and $f_p^{non}(t+1) = \arg\min Z_p(f_p,t|D_p)$. Suppose that the noise vector $\varepsilon_t(t)$ generated at time t has the same value of $\alpha_p(t)$ for all $p \in \mathscr{P}$. Let Λ be the event*

$$\Lambda := \Big\{ Z_p(f_p^*(t+1),t|D_p) \leq Z_p(f_p^{non}(t+1),t|D_p) + \frac{16\big(C^R\big)^2\eta^2 N_p^2 d^2\big(\ln(\frac{d}{\delta})\big)^2}{\rho^3 B_p^2 \alpha_p(t)^2} \Big\}.$$

If the loss function $\mathscr{L}$ is convex and differentiable with $|\mathscr{L}| \leq 1$, then we have:

$$\boldsymbol{P}_{\varepsilon_p(t)}\Big(\Lambda\Big) \geq 1-\delta.$$

The probability $\boldsymbol{P}_{\varepsilon_p(t)}$ is taken over the noise vector $\varepsilon_p(t)$.

Proof: **(Lemma 14)**

Let $\varepsilon^{pi}(t) = \varepsilon_p(t) - \varepsilon_i(t)$ with probability density $P_{\varepsilon^{pi}}$. Let $f_p^*(t+1) = \arg\min Z_{prim}(f_p,t|D_p)$, and it can be expressed as:

$$\begin{aligned} f_p^*(t+1) = \arg\min \Big(& Z_p(f_p,t|D_p) \\ & - \eta \sum_{i\in\mathscr{N}_p} \Big((f_p - \frac{1}{2}(f_p(t) + f_i(t))^T \\ & \cdot (\varepsilon^{pi}(t)) + \frac{1}{4}\big(\varepsilon^{pi}(t)\big)^2 \Big). \end{aligned}$$

Thus, we have:

$$\begin{aligned} & Z_p(f_p^*(t+1),t|D_p) \\ & \qquad \leq Z_p(f_p^{non}(t+1),t|D_p) \\ & \qquad -\eta \sum_{i\in\mathscr{N}_p} (f_p^{non}(t+1) - f_p^*(t+1))^T \cdot \varepsilon^{pi}. \end{aligned}$$

Firstly, we bound the l_2-norm $\| f_p^{non}(t+1) - f_p^*(t+1) \|$. We use the similar procedure to establish (46) in Appendix D by setting $F(\cdot) = Z_p(Z,t|D_p)$ and $G(Z) = \eta \sum_{i\in\mathscr{N}_p} \big(\varepsilon^{pi}\big)^T(\cdot)$; thus,based on Assumption 1 and 2, we have:

$$\begin{aligned} & \| f_p^{non}(t+1) - f_p^*(t+1) \| \\ & \leq \frac{1}{\rho} \| \sum_{i\in\mathscr{N}_p} \nabla(\eta N_p(f_p^*(t+1))^T \varepsilon^{pi}) \| \\ & \leq \sum_{i\in\mathscr{N}_p} \frac{\eta \| \varepsilon^{pi}(t) \|}{\rho} = \sum_{i\in\mathscr{N}_p} \frac{\eta\Big(\| \varepsilon_p(t) - \varepsilon_j(t) \| \Big)}{\rho} \\ & \leq \sum_{i\in\mathscr{N}_p} \frac{\eta\Big(\| \varepsilon_p(t) \| + \| \varepsilon_j(t) \| \Big)}{\rho}. \end{aligned}$$

Since $\alpha_p(t)$ is the same for all $p \in \mathscr{P}$ at time t; thus $\zeta_j(t) = \frac{\rho B_p \alpha_p(t)}{2C^R}$ for all $j \in \mathscr{P}$. Since $\varepsilon_j(t)$ is drawn from (15), then, $\| \varepsilon_p(t) \|$ is gamma with $\Gamma(d, \frac{1}{\zeta_p(t)})$ for all $p \in \mathscr{P}$. Let

$$\| \varepsilon_{pi} \|^{\oplus} = \| \varepsilon_p(t) \| + \| \varepsilon_i(t) \| .$$

Thus

$$\begin{aligned} \| f_p^{non}(t+1) - f_p^*(t+1) \| \leq & \sum_{i\in\mathscr{N}_p} \frac{\eta\Big(\| \varepsilon_{pi} \|^{\oplus} \Big)}{\rho} \\ & = \frac{\eta N_p\Big(\| \varepsilon_{pi} \|^{\oplus} \Big)}{\rho}. \end{aligned}$$

Cauchy-Schwarz inequality yields:

$$\begin{aligned} & Z_p(f_p^*(t+1),t|D_p) - Z_p(f_p^*(t+1),t|D_p) \\ & \leq \| Z_p(f_p^*(t+1),t|D_p) - Z_p(f_p^*(t+1),t|D_p) \| \\ & \leq \frac{\eta^2 N_p^2 \Big(\| \varepsilon_{pi} \|^{\oplus} \Big)^2}{\rho}, \end{aligned}$$

and from Lemma 12 we have the $P_{\|\varepsilon^{pj}\|} = \Gamma(2d, \frac{2C^R}{\rho B_p \alpha_p(t)})$. Applying Lemma 10 with $\| \varepsilon^{pj}(t) \|^{\oplus} \leq \frac{4C^R d \ln(\frac{d}{\delta})}{\rho B_p \alpha_p(t)}$ yields:

$$\begin{aligned} & Z_p(f_p^*(t+1),t|D_p) - Z_p(f_p^{non}(t+1),t|D_p) \\ & \leq \frac{16\big(C^R\big)^2 \eta^2 N_p^2 d^2 \big(\ln(\frac{d}{\delta})\big)^2}{\rho^3 B_p^2 \alpha_p(t)^2} \end{aligned}$$

with probability no smaller than $1-\delta$ □

Appendix H.
Proof of Theorem 6

Proof: **(Theorem 6)** Again, we define the following optimal variables:

$$\hat{f}_p(t+1) = \arg\min Z_E(f_p, t),$$

$$f_p^{non}(t+1) = \arg\min Z_p(f_p, t|D_p),$$

$$f_p^*(t+1) = \arg\min Z_{prim}(f_p, t|D_p),$$

$$V_p^*(t+1) = f_p^*(t+1) + \varepsilon_p(t).$$

Now we make $f_p^0(t+1)$ such that $\hat{C}(f_p^*(t+1)) = \hat{C}^*(t+1)$ be the reference at time $t+1$. We use the analysis of Shalev-Shwartz and Srebro in [53] (also see the work of Chaudhuri et al. in [11]), and have the follows,

$$\begin{aligned}\hat{C}(V_p^*(t+1)) =& \hat{C}(f_p^0(t+1)) \\ &+ \big(\hat{Z}(V_p^*(t+1), t) - \hat{Z}(\hat{f}_p(t+1), t)\big) \\ &+ \big(\hat{Z}(\hat{f}_p(t+1), t) - \hat{Z}(f_p^0(t+1), t)\big) \\ &+ \frac{\rho}{2} \parallel f_p^0(t+1) \parallel^2 - \frac{\rho}{2} \parallel V_p^*(t+1) \parallel^2 .\end{aligned} \tag{51}$$

If $R(f_p(t)) = \frac{1}{2} \parallel f_p(t) \parallel^2$, then $\parallel \nabla^2 R(f_p(t)) \parallel \leq 1$. Thus, we can apply Lemma 15 with $\tau = 1$:

$$\begin{aligned}&Z_{prim}(V_p^*(t+1), t|D_p) - Z_{prim}(f_p^*(t+1), t|D_p) \\ &\leq \frac{4\big(C^R\big)^2 d^2 \Big(\rho + c_4 C^R\Big) \Big(\ln(\frac{d}{\delta})\Big)^2}{\rho^2 B_p^2 \alpha_p(t)^2},\end{aligned}$$

with probability $\geq 1-\delta$ over the noise. In the proof of Theorem 5, we have, with probability $1-\delta$:

$$\begin{aligned}&Z_p(f_p^*(t+1), t|D_p) - Z_p(f_p^{non}(t+1), t|D_p) \\ &\leq \frac{4\eta^2 N_p^2 d^2 \big(\ln(\frac{d}{\delta})\big)^2}{\rho^3 B_p^2 \alpha_p(t)^2}.\end{aligned}$$

Therefore, with probability $1-2\delta$, we have

$$\begin{aligned}&Z_p(V_p^*(t+1), t|D_p) - Z_p(f_p^{non}(t+1), t|D_p) \\ &\leq \frac{4\eta^2 N_p^2 d^2 \big(\ln(\frac{d}{\delta})\big)^2}{\rho^3 B_p^2 \alpha_p(t)^2} + \frac{4d^2 \Big(\rho + c_4\Big) \Big(\ln(\frac{d}{\delta})\Big)^2}{\rho^2 B_p^2 \alpha_p(t)^2}.\end{aligned}$$

Sridharan et al. in [54] shows, with probability $1-\delta$:

$$\begin{aligned}&\hat{Z}(V_p^*(t+1)) - \hat{Z}(\hat{f}_p(t+1)) \\ &\leq 2\Big(Z_{prim}(V_p(t+1), t|D_p) - Z_{prim}(f_p^*(t+1), t|D_p)\Big) \\ &+ \mathscr{O}\Big(C^R \frac{\ln(\frac{d}{\delta})}{B_p \rho}\Big) \\ \leq& \frac{8\big(C^R\big)^2 d^2 \Big(\rho + c_4 C^R\Big) \Big(\ln(\frac{d}{\delta})\Big)^2}{\rho^2 B_p^2 \alpha_p(t)^2} + \frac{8\eta^2 N_p^2 d^2 \big(\ln(\frac{d}{\delta})\big)^2}{\rho^3 B_p^2 \alpha_p(t)^2} \\ &+ \mathscr{O}\Big(C^R \frac{\ln(\frac{1}{\delta})}{B_p \rho}\Big).\end{aligned}$$

Combining the above two processes, we have the probability no smaller than $1-3\delta$.

Since $\hat{f}_p(t+1) = \arg\min \hat{Z}(f_p, t)$, then $\big(\hat{Z}(\hat{f}_p(t+1), t) - \hat{Z}(f_p^0(t+1), t) \leq 0$. For the last two terms, we select $\rho = \frac{\alpha_{acc}}{\|f_p^0(t+1)\|^2}$ in order to make them bounded by $\frac{\alpha_{acc}}{2}$.

The value of B_p is determined by solving

$$\begin{aligned}&\frac{8\big(C^R\big)^2 d^2 \Big(\rho + c_4 C^R\Big) \Big(\ln(\frac{d}{\delta})\Big)^2}{\rho^2 B_p^2 \alpha_p(t)^2} + \frac{8\eta^2 N_p^2 d^2 \big(\ln(\frac{d}{\delta})\big)^2}{\rho^3 B_p^2 \alpha_p(t)^2} \\ &+ \mathscr{O}\Big(C^R \frac{\ln(\frac{1}{\delta})}{B_p \rho}\Big) + \frac{\alpha_{acc}}{2} = \alpha_{acc},\end{aligned}$$

with $\rho = \frac{\alpha_{acc}}{\|f_p^0(t+1)\|^2}$, such that

$$\mathbb{P}\big(\hat{C}(V_p^*(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc}\big) \geq 1-3\delta.$$

We get:

$$\begin{aligned}B_p =& \max\Bigg(\Big\{\frac{4C^B \parallel f^0(t+1) \parallel d\big(\ln(\frac{d}{\delta})\big)^2}{\alpha_{acc}\alpha_p(t)}\Big\}_{t=1}, \\ &\Big\{\frac{4 \parallel f_p^0(t+1) \parallel^3 \eta N_p d \ln(\frac{d}{\delta})}{\alpha_{acc}^2 \alpha_p(t)}\Big\}_{t=1}, \\ &\Big\{\frac{4\big(C^R\big)^{\frac{3}{2}} \parallel f_p^0(t+1) \parallel^2 d \ln(\frac{d}{\delta})}{\alpha_{acc}^{3/2} \alpha_p(t)}\Big\}_{t=1}\Bigg).\end{aligned}$$

However, the accuracy of $V_p^*(t+1)$ depends on $f_p^*(t+1)$, thus we also have to make

$$\mathbb{P}\big(\hat{C}(f_p^*(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc}\big) \geq 1-2\delta.$$

Combining the result of Theorem 5, we have

$$\begin{aligned}B_p >&\beta^B_{prim}\max\Big(\Big\{\frac{C^R \parallel f_p^0(t+1) \parallel^3 \eta N_p d\ln(\frac{d}{\delta})}{\alpha_{acc}^2\alpha_p(t)}\Big\}_{t=1},\\ &\Big\{\frac{C^R \parallel f_p^0(t+1) \parallel^2 \ln(\frac{1}{\delta})}{\alpha_{acc}^2}\Big\}_{t=1},\\ &\Big\{\frac{4C^B \parallel f^0(t+1) \parallel d\big(\ln(\frac{d}{\delta})\big)^2}{\alpha_{acc}\alpha_p(t)}\Big\}_{t=1},\\ &\Big\{\frac{4 \parallel f_p^0(t+1) \parallel^3 \eta N_p d\ln(\frac{d}{\delta})}{\alpha_{acc}^2\alpha_p(t)}\Big\}_{t=1},\\ &\Big\{\frac{4\big(C^R\big)^{\frac{3}{2}} \parallel f_p^0(t+1) \parallel^2 d\ln(\frac{d}{\delta})}{\alpha_{acc}^{3/2}\alpha_p(t)}\Big\}_{t=1}\Big).\end{aligned}$$

As a result, the value of B_p is determined by taking the intersection of

□

Lemma 15. Assume $R(f_p(t))$ is doubly differentiable w.r.t. $f_p(t)$ with $\parallel \nabla^2 R(f_p(t)) \parallel\leq \tau$ for all $f_p(t)$. Suppose the loss function $\mathscr{L}$ is differentiable, $\mathscr{L}'$ is continuous, and satisfies

$$|\mathscr{L}'(a)-\mathscr{L}'(b)|\leq c_4|a-b|$$

for all pairs (a,b) with a constant c_4. Let $f_p^*(t+1) = \arg\min Z_{prim}(f_p,t|D_p)$, and $V_p^*(t+1) = f_p^*(t+1)+\varepsilon_p(t)$, where the noise vector $\varepsilon_p(t)$ is drawn from (15) with the same $\alpha_p(t)$ for all $p\in\mathscr{P}$ at time t. Let Λ be the event

$$\begin{aligned}\Lambda :=\Big\{&Z_{prim}(V_p^*(t+1),t|D_p)\leq Z_{prim}(f_p^*(t+1),t|D_p)\\ &+\frac{4\big(C^R\big)^2 d^2\Big(\rho\tau+c_4C^R\Big)\Big(\ln(\frac{d}{\delta})\Big)^2}{\rho^2B_p^2\alpha_p(t)^2}.\end{aligned}$$

Under Assumption 1 and 2, we have:

$$\mathbf{P}_{\varepsilon_p(t)}\Big(\Lambda\Big)\geq 1-\delta.$$

The probability $\mathbf{P}_{\varepsilon_p(t)}$ is taken over the noise vector $\varepsilon_p(t)$.

Proof: **(Lemma 15)** From Assumption 3, we know that the data points in dataset D_p satisfy: $\parallel x_{ip} \parallel\leq 1$, and $|y_{ip}|=1$. From Assumption 1 and 2, $R(\cdot)$ and $\mathscr{L}$ are differentiable. Suporse $R(\cdot)$ is doubly differentiable and $\nabla^2 R(\cdot)\leq\tau$. Let $0\leq\varphi\leq 1$, then the *Mean Value Theorem* and Cauchy-Schwarz inequality give:

$$\begin{aligned}&Z_{prim}(V_p^*(t+1),t|D_p)-Z_{prim}(f_p^*(t+1),t|D_p)\\ &=(V_p^*(t+1)-f_p^*(t+1))^T\nabla Z_{prim}\Big(\varphi f_p^*(t+1)\\ &+(1-\varphi)V_p^*(t+1)\Big)\\ &\leq\parallel V_p^*(t+1)-f_p^*(t+1)\parallel\\ &\cdot\parallel\nabla Z_{prim}\Big(\varphi f_p^*(t+1)+(1-\varphi)V_p^*(t+1)\Big)\parallel.\end{aligned}$$

Let $\varepsilon^{pi}(t)=\varepsilon_p(t)-\varepsilon_i(t)$. From the definition of $Z_{prim}(f_p,t|D_p)$, we have:

$$\begin{aligned}Z_{prim}(f_p,t|D_p)=&Z_p(f_p,t|D_p)\\ &-\eta\sum_{i\in\mathscr{N}_p}\Big((f_p-\frac{1}{2}(f_p(t)+f_i(t))^T\cdot(\varepsilon^{pi}(t))\\ &+\frac{1}{4}(\varepsilon^{pi}(t))^2\Big).\end{aligned}$$

Taking the derivative of Z_{prim} w.r.t. f_p gives

$$\begin{aligned}\nabla Z_{prim}(f_p,t|D_p)=&\frac{C^R}{B_p}\sum_{i=1}^{B_p}y_{ip}\mathscr{L}'(y_{ip}f_p^Tx_{ip})x_{ip}\\ &+\rho\nabla R(f_p)-\eta\sum_{j\in\mathscr{N}_p}\varepsilon^{pi}(t).\end{aligned}$$

Since $\nabla Z_{prim}(f_p^*(t+1),t|D_p)=0$, then we have:

$$\begin{aligned}&\nabla Z_{prim}\Big(\varphi f_p^*(t+1)+(1-\varphi)V_p^*(t+1)|D_p\Big)\\ &=\nabla Z_{prim}(f_p^*(t+1),t|D_p)\\ &-\rho\Big(\nabla R(f_p^*(t+1))-\nabla R\big(\varphi f_p^*(t+1)\\ &+(1-\varphi)V_p^*(t+1)\big)\Big)\\ &-\frac{C^R}{B_p}\sum_{i=1}^{B_p}\Bigg(y_{ip}\Big(\mathscr{L}'(y_{ip}f_p^*(t+1)^Tx_{ip})\\ &-\mathscr{L}'(y_{ip}\big(\varphi f_p^*(t+1)+(1-\varphi)V_p^*(t+1)\big)^Tx_{ip})\Big)x_{ip}\Bigg).\end{aligned}$$

Let

$$\begin{aligned}T=&y_{ip}\Big(\mathscr{L}'(y_{ip}f_p^*(t+1)^Tx_{ip})\\ &-\mathscr{L}'(y_{ip}\big(\varphi f_p^*(t+1)+(1-\varphi)V_p^*(t+1)\big)^Tx_{ip})\Big)x_{ip}.\end{aligned}$$

Based on the condition on the loss function:

$$|\mathscr{L}'(a)-\mathscr{L}'(b)|\leq c_4|a-b|,$$

we can bound T as:

$$\begin{aligned}T \leq & |y_{ip}| \parallel x_{ip} \parallel \\ & \cdot |\mathscr{L}'(y_{ip} f_p^*(t+1)^T x_{ip}) \\ & - \mathscr{L}'(y_{ip}\big(\varphi f_p^*(t+1) + (1-\varphi) V_p^*(t+1)\big)^T x_{ip})| \\ \leq & |y_{ip}| \parallel x_{ip} \parallel \cdot c_4 \cdot |y_{ip}(1-\varphi)(f_p^*(t+1) - V_p^*(t+1))^T x_{ip}| \\ \leq & c_4 \cdot (1-\varphi)|y_{ip}|^2 \parallel x_{ip} \parallel^2 \parallel f_p^*(t+1) - V_p^*(t+1) \parallel \\ \leq & c_4 \cdot (1-\varphi) \parallel f_p^*(t+1) - V_p^*(t+1) \parallel .\end{aligned}$$

Since we assume $R(\cdot)$ is doubly differentiable, we then apply the *Mean Value Theorem*:

$$\begin{aligned}& \parallel \nabla R(f_p^*(t+1)) - \nabla R\big(\varphi f_p^*(t+1) + (1-\varphi) V_p^*(t+1)\big) \parallel \\ & \leq (1-\varphi) \parallel f_p^*(t+1) - V_p^*(t+1) \parallel \cdot \parallel \nabla^2 R(\xi) \parallel,\end{aligned}$$

where $\xi \in \mathbb{R}^d$. Therefore, we have:

$$\begin{aligned}& \nabla Z_{prim}\Big(\varphi f_p^*(t+1) + (1-\varphi) V_p^*(t+1)|D_p\Big) \\ & \leq (1-\varphi) \parallel f_p^*(t+1) - V_p^*(t+1) \parallel \cdot \rho \cdot \parallel \nabla^2 R(\xi) \parallel \\ & + C^R c_4 \cdot (1-\varphi) \parallel f_p^*(t+1) - V_p^*(t+1) \parallel \\ & \leq (1-\varphi) \cdot \parallel f_p^*(t+1) - V_p^*(t+1) \parallel \Big(\rho\tau + C^R c_4\Big) \\ & \leq \parallel f_p^*(t+1) - V_p^*(t+1) \parallel \Big(\rho\tau + C^R c_4\Big).\end{aligned}$$

Since $f_p^*(t+1) - V_p^*(t+1) = \varepsilon_p(t)$, with density $\Gamma(d, \frac{2C^R}{\rho B_p \alpha_p(t)})$ then we can apply Lemma 10 to $\parallel f_p^*(t+1) - V_p^*(t+1) \parallel$. Thus, with $\parallel f_p^*(t+1) - V_p^*(t+1) \parallel \leq \frac{2C^R d \ln(\frac{d}{\delta})}{\rho B_p \alpha_p(t)}$, we have:

$$\begin{aligned}& Z_{prim}(V_p^*(t+1), t|D_p) - Z_{prim}(f_p^*(t+1), t|D_p) \\ & \leq \frac{4\big(C^R\big)^2 d^2 \Big(\rho\tau + c_4 C^R\Big)\Big(\ln(\frac{d}{\delta})\Big)^2}{\rho^2 B_p^2 \alpha_p(t)^2},\end{aligned}$$

with probability no less than $1-\delta$.

□

Appendix I.
Proof that iterations (5) to (8) are convergent ADMM algorithm shown in Appendix A

The goal here is to cast (3) in the form of (31) and show that updates (5)-(8) correspond to (33)-(35) in Appendix A. We first reform the constraints $\{f_p = w_{pj}, w_{pj} = f_j\}_{p \in \mathscr{P}, j \in \mathscr{N}_p}$ to $\mathbf{A}f = w$, where $f = [f_1, f_2, ..., f_P]^T$. For all the nodes in the network, the constraint $f_p = w_{pj}$ can be written as:

$$\begin{aligned}& \{f_1 = w_{1j}\}_{j \in \mathscr{N}_1} \\ & \quad \vdots \\ & \{f_P = w_{Pj}\}_{j \in \mathscr{N}_P}.\end{aligned} \tag{52}$$

Let

$$w = [\{w_{1j}^T\}_{j \in \mathscr{N}_1}, ..., \{w_{Pj}^T\}_{j \in \mathscr{N}_P}]^T$$

and let $\mathbf{A}$ be a block-diagonal matrix with diagonal

$$\mathbf{A}_i = [\mathbf{I}_d, ..., \mathbf{I}_d]^T.$$

Thus,

$$\mathbf{A} = \begin{vmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_P \end{vmatrix}.$$

Therefore, we can write (51) in the form of matrix and vector as:

$$\mathbf{A}f = w. \tag{53}$$

Let $|\mathscr{E}|$ be the number of links in the network. Then there are $\sum_{i=1}^P N_p = 2|\mathscr{E}|$, where the factor 2 of $2|\mathscr{E}|$ is from the fact that there are two constraints between two nodes: $f_p = w_{pj}$ and $f_j = w_{jp}$. Then matrix $A \in \mathbb{R}^{2d|\mathscr{E}| \times dP}$, and $A_i \in \mathbb{R}^{dN_p \times dN_p}$.

Now we consider the constraint $f_p = w_{jp}$. We can also list it acorss the nodes as:

$$\begin{aligned}& \{f_1 = w_{j1}\}_{j \in \mathscr{N}_1} \\ & \quad \vdots \\ & \{f_P = w_{jP}\}_{j \in \mathscr{N}_P}.\end{aligned} \tag{54}$$

Similarly, let

$$w^1 = [\{w_{j1}^T\}_{j \in \mathscr{N}_1}, ..., \{w_{jP}^T\}_{j \in \mathscr{N}_P}]^T.$$

and then we can write (53) in the form as:

$$\mathbf{A}f = w^1. \tag{55}$$

It can be observed that replacing each w_{ij} in w by w_{ji} gives w^1. Now we express w^1 in terms of w. Let $\mathbf{S}^w$ be a $2|\mathscr{E}| \times 2|\mathscr{E}|$ matrix defined as:

$$\mathbf{S}^w = [\{s_{1j}\}_{j \in \mathscr{N}_1}, ..., \{s_{Pj}\}_{j \in \mathscr{N}_P}],$$

where

$$s_{pj} = [\big(s_{pj}^1\big)^T, ..., \big(s_{pj}^P\big)^T]^T$$

is a $2|\mathscr{E}| \times 1$ indictor vector. Let $\delta(\cdot,\cdot)$ be the Kronecker's delta. Then

$$s^a_{pj} = [\{\delta(p-b, j-b)\}_{b\in\mathscr{N}_a}]^T.$$

Thus, we can write w^1 in terms of w as:

$$w^1 = \big(\mathbf{S}^w \otimes \mathbf{I}_d\big)w. \tag{56}$$

Therefore, (54) can be written as:

$$\mathbf{A}f = \big(\mathbf{S}^w \otimes \mathbf{I}_d\big)w, \tag{57}$$

where $\otimes$ denotes Kronecker product.

Let $\mathbf{A}^1 = \big[A^T A^T\big]^T$, and $\mathbf{S}^1 = \big[\mathbf{I}^T_{2d|\mathscr{E}|}\big(\mathbf{S}^w \otimes \mathbf{I}_d\big)^T\big]^T$. Then, we can combine (54) and (56) as:

$$\mathbf{A}^1 f = \mathbf{S}^1 w. \tag{58}$$

Thus, we can re-write (3) as:

$$\begin{aligned} \min\ Z_{dec} = &\frac{C^R}{B_p}\sum_{p=1}^{P}\sum_{i=1}^{B_p}\mathscr{L}(y_{ip}f_p^T x_{ip}) + \sum_{p=1}^{P}\rho R(f_p) \\ &s.t.\ \ \mathbf{A}^1 f = \mathbf{S}^1 w. \end{aligned} \tag{59}$$

Now, let

$$\begin{aligned} &g_1(f) = \frac{C^R}{B_p}\sum_{p=1}^{P}\sum_{i=1}^{B_p}\mathscr{L}(y_{ip}f_p^T x_{ip}) + \sum_{p=1}^{P}\rho R(f_p) \\ &g_2(w) = 0 \\ &S_1 = \mathbb{R}^{dP} \\ &S_2 = \big\{w \in \mathbb{R}^{4d|\mathscr{E}|} | w = \mathbf{S}^1 w' for\ some w' \in \mathbb{R}^{2d|\mathscr{E}|}\big\}. \end{aligned}$$

Thus, problem (3) has the type of (31). Therefore, the ADMM-based algorithm with updates (5)-(8) is convergent according to Theorem 9 in Appendix A.

Appendix J. Proof of Proposition 7

Proof: **(Proposition 7)** According to Corollary 4.1, $f_p(t)$ is α_{acc}-optimal at each time t, and

$$\mathbb{P}\big(\hat{C}(f_p(t)) \leq \hat{C}^*(t) + \alpha_{acc} + \Delta_p^{dual}(t)\big) \geq 1 - 2\delta,$$

and from Theorem 3, we have

$$\mathbb{P}\big(\hat{C}(F_p(t)) \leq \hat{C}^*(t) + \alpha_{acc} + \Delta^{non}(t)\big) \geq 1 - \delta.$$

Then, we have:

$$\mathbb{P}\big(\hat{C}(f_p(t)) \leq \hat{C}(F_p(t)) + \Delta_p^{dual}(t) - \Delta^{non}(t)\big) \geq 1 - 2\delta.$$

It also holds for $t \to \infty$ when the $F_p(t)$ converges to $f_p^{non}(t+1)$. Therefore, $f_p(t)$ performs similar to $f_p^{non}(t)$, and the error between them is caused by the noise $\{\varepsilon_p(t)\}$.

Taking the gradient of L_{dual} (16) and setting it to 0 at $f_p(t)$ give (37) in Appendix:

$$\begin{aligned} \varepsilon_p(t) = &- \sum_{i=1}^{B_p} y_{ip}\mathscr{L}'(y_{ip}f_p(t+1)^T x_{ip})x_{ip} - \frac{B_p}{C^R}\rho\nabla R(f_p) \\ &- \frac{2B_p}{C^R}\lambda_p(t) - \frac{B_p}{C^R}(\Phi + 2\eta N_p)f_p(t+1) \\ &+ \frac{B_p\eta}{C^R}\sum_{i\in\mathscr{N}_p}(f_p(t) + f_i(t)). \end{aligned}$$

Following the similar argument in the proof of Theorem 1 in Appendix B, we claim that the relation between $\varepsilon_p(t)$ and $f_p(t+1)$ is bijective.

Let $\mathbf{J}_f(\varepsilon_p(t)|D_p)$ be the Jacobian matrix transforming from $f_p(t+1) \to \varepsilon_p(t)$ as (See Appendix B for more details):

$$\begin{aligned} \mathbf{J}_f(\varepsilon_p(t)|D_p) = &- \sum_{i=1}^{B_p}\mathbf{J}_f^0(x_i, y_i) - \frac{B_p}{C^R}\rho\nabla^2 R(f_p(t+1)) \\ &- \frac{B_p}{C^R}(\Phi + 2\eta N_p)\mathbf{I}_d. \end{aligned}$$

By transformation through Jacobian, we have:

$$\begin{aligned} &Q(f_p(t)|D_p) \\ &\quad = \mathscr{K}(\varepsilon_p(t))\frac{\parallel \varepsilon_p(t) \parallel^{d-1}}{sur(\parallel \varepsilon_p(t) \parallel)}|det(\mathbf{J}_f(\varepsilon_p(t)|D_p))|^{-1} \\ &\quad = \mathscr{K}(\varepsilon_p(t))\frac{1}{sur(\parallel 1 \parallel)}|det(\mathbf{J}_f(\varepsilon_p(t)|D_p))|^{-1}, \end{aligned}$$

where $sur(E)$ is the surface area of the sphere in d dimension with radius E, and $sur(E) = sur(1)\cdot E^{d-1}$. Since $\varepsilon_p(t)$ is Laplace random variable with density $\mathscr{K}$, and $|det(\mathbf{J}_f(\varepsilon_p(t)|D_p))|$ is a bounded function of $f_p(t+1)$. Thus, $Q(f_P(t)|D_p)$ is a bounded density function. Therefore, with probability greater than $1-2\delta$, Algorithm 2 converges in distribution. □

Appendix K. Proof of Proposition 8

Proof: **(Proposition 8)** From Corollary 6.1, we have:

$$\begin{aligned} \mathbb{P}\big(\hat{C}(V_p(t+1)) \leq \hat{C}^*(t+1) + \alpha_{acc} + &\Delta_p^{prim}(t)\big) \\ &\geq 1 - 3\delta, \end{aligned}$$

and from Theorem 3,

$$\mathbb{P}\left(\hat{C}(F_p(t)) \leq \hat{C}^*(t) + \alpha_{acc} + \Delta^{non}(t)\right) \geq 1-\delta,$$

then,

$$\mathbb{P}\left(\hat{C}(V_p(t+1)) \leq \hat{C}(F_p(t)) + \Delta_p^{prim}(t) - \Delta^{non}(t)\right) \geq 1-3\delta.$$

It also holds for $t \to \infty$ when the $F_p(t)$ converges to $f_p^{non}(t+1)$. Therefore, $V_p(t)$ performs similar to $f_p^{non}(t)$, and the error between them is caused by the noise $\{\varepsilon_p(t)\}$.

Let $f_p(t+1) = \arg\min L_{prim}(t)$ with zero duality gap, and let $\varepsilon^{pi}(t) = \varepsilon_p(t) - \varepsilon_i(t)$. Under the Assumption 1 and 2, using the Karush-Kuhn-Tucker (KKT) optimality condition (stationarity), we have

$$\begin{aligned} 0 =& \frac{C^R}{B_p}\sum_{i=1}^{B_p} y_{ip}\mathscr{L}'(y_{ip}f_p(t+1)^T x_{ip})x_{ip} + \rho\nabla R(f_p) \\ & -\eta\sum_{i\in\mathcal{N}_p}\varepsilon^{pi}(t). \end{aligned}$$

Let $\varepsilon^p(t) = \sum_{i\in\mathcal{N}_p}\varepsilon^{pi}(t)$. Then we can establish the relationship between the noise $\varepsilon^{pi}(t)$ and the optimal primal variable $f_p(t+1)$ as:

$$\begin{aligned} \varepsilon^p(t) =& \frac{C^R}{B_p\eta}\sum_{i=1}^{B_p} y_{ip}\mathscr{L}'(y_{ip}f_p(t+1)^T x_{ip})x_{ip} \\ & + \frac{\rho}{\eta}\nabla R(f_p). \end{aligned}$$

Again, following the similar argument in the proof of Theorem 1 in Appendix B, we claim that there is bijection between $\varepsilon_p(t)$ and $f_p(t+1)$.

Let $\mathbf{J}_f^1(\varepsilon^p(t)|D_p)$ be the Jacobian matrix transforming from $f_p(t+1)$ to $\varepsilon^p(t)$ based on the above equation. Let $\mathbf{J}_f^1(\varepsilon^p(t)|D_p)^{(a,b)}$ be the (a,b) entry of matrix $\mathbf{J}_f^1(\varepsilon^p(t)|D_p)$, then

$$\begin{aligned} &\mathbf{J}_f^1(\varepsilon^p(t)|D_p)^{(a,b)} \\ &\quad = \frac{C^R}{B_p\eta}\sum_{i=1}^{B_p} y_{ip}^2\mathscr{L}''(y_{ip}f_p(t+1)^T x_{ip})x_{ip}^{(a)}x_{ip}^{(b)} \\ &\quad + \frac{\rho}{\eta}\nabla^2 R(f_p)^{(a,b)}. \end{aligned}$$

Thus,

$$\begin{aligned} \mathbf{J}_f^1(\varepsilon^p(t)|D_p) =& \frac{C^R}{B_p\eta}\sum_{i=1}^{B_p} y_{ip}^2\mathscr{L}''(y_{ip}f_p(t+1)^T x_{ip})x_{ip}x_{ip}^T \\ & + \frac{\rho}{\eta}\nabla^2 R(f_p). \end{aligned}$$

We now find the probability density function of $\varepsilon^{pi}(t)$. Since $\varepsilon_p(t)$ and $\varepsilon_i(t)$ are independent, then their joint density function $P_{pi}(z)$ is:

$$P_{pi}(z) = \frac{1}{\kappa}e^{-\left(\zeta_p(t)+\zeta_j(t)\right)\|z\|},$$

where κ is a normalizing constant. Since $\alpha_p(t)$ is fixed for all nodes at time t, then all the nodes have the same value of $\zeta_p(t) = \zeta(t)$. Then

$$P_{pi}(\varepsilon_p(t),\varepsilon_i(t)) = \frac{1}{\kappa}e^{-2\zeta(t)\left(\|\varepsilon_p(t)\|-\|\varepsilon_i(t)\|\right)}.$$

Then the cumulative distribution function of $\varepsilon^{pi}(t)$ is

$$\begin{aligned} F^{\varepsilon^{pi}(t)}(z) =& P(\varepsilon^{pi} \leq z) \\ =& \int_{\infty}^{\infty}\int_{\varepsilon^p - z}^{\infty} P_{pi}(\varepsilon_p(t),\varepsilon_i(t))P_{pi}(z)d\varepsilon_p(t)d\varepsilon_j(t). \end{aligned}$$

Thus, the density function of $\varepsilon^{pi}(t)$ is

$$P^{pi}(z) = \frac{dF^{\varepsilon^{pi}(t)}(z)}{dz}.$$

Therefore, the density function of $\varepsilon^p(t) = \sum_{i\in\mathcal{N}_p}\varepsilon^{pi}(t)$ can be expressed as:

$$P_{\varepsilon^p(t)}(z) = \prod_{i\in\mathcal{N}_p}^{N_p} * P^{pi}(z),$$

where $\prod_{i\in\mathcal{N}_p}^{N_p} *$ is the N_p-fold convolution.

By transformation through Jacobian, we have:

$$\begin{aligned} Q^A(f_p(t+1)|D_p) =& P_{\varepsilon^p(t)}(\varepsilon_p(t))\frac{\|\varepsilon_p(t)\|^{d-1}}{sur(\|\varepsilon_1(t)\|)} \\ & \cdot|det(\mathbf{J}_f^1(\varepsilon^p(t)|D_p))|^{-1} \\ =& P_{\varepsilon^p(t)}(\varepsilon_p(t))\frac{1}{sur(\|1\|)} \\ & \cdot|det(\mathbf{J}_f^1(\varepsilon^p(t)|D_p))|^{-1}, \end{aligned}$$

where $sur(E)$ is the surface area of the sphere in d dimension with radius E, and $sur(E) = sur(1)\cdot E^{d-1}$. $|det(\mathbf{J}_f^1(\varepsilon^p(t)|D_p))|$ is a bounded function of $f_p(t+1)$.

Since $V_p(t+1) = f_p(t+1) + \varepsilon_p(t+1)$ and $f_p(t+1)$ and $\varepsilon_p(t+1)$ are independent, then we can find the probability density function, $P_{V_p}^{t+1}$, of $V_p(t+1)$ as:

$$P_{V_p}^{t+1}(z) = (Q^A(f_P(t+1)|D_p) * \mathscr{K})(z).$$

Therefore, with probability greater than $1-3\delta$, Algorithm 2 converges in distribution.

□

References

[1] P.A. Forero, A. Cano, and G. Giannakis. Consensus-based distributed support vector machines. JMLR, 11:1663–1707, 2010.

[2] Mu Li, Dave Andersen, Alex Smola, Junwoo Park, Amr Ahmed, Vanja Josifovski, James Long, Eugene Shekita, Bor-Yiing Su. Scaling Distributed Machine Learning with the Parameter Server . Operating Systems Design and Implementation (OSDI), 2014

[3] Peter Chilstrom. Singular Value Inequalities: New Approaches to Conjectures. 2013.

[4] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, Proc. of the Third Theory of Cryptography Conference – TCC 2006, volume 3876 of Lecture Notes in Computer Science, pages 265–284. Springer-Verlag, 2006.

[5] K. Canini. Sibyl: A system for large scale supervised machine learning. Google, 2012.

[6] S. Boyd1, N. Parikh, E. Chu, B. Peleato and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Foundations and Trends in Machine Learning Vol. 3, No. 1 (2010) 1–122.

[7] McDonald, Ryan, Hall, Keith, and Mann, Gideon. Distributed training strategies for the structured perceptron. In NAACL HLT, 2010.

[8] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private approximation algorithms. In Proceedings of the 2010 ACM-SIAM Symposium on Discrete Algorithms (SODA), 2010.

[9] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. Proceedings of the 39th STOC, 2007.

[10] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In Proceedings of the 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), pages 273–282, 2007.

[11] K Chaudhuri, C Monteleoni, AD Sarwate. Differentially private empirical risk minimization. Journal of machine learning research: JMLR 12, 1069, 2011.

[12] C. Dwork and J. Lei. Differential privacy and robust statistics. In Proceedings of the 41st ACM Symposium on Theory of Computing (STOC), 2009.

[13] Collins, Michael. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In EMNLP, 2002.

[14] P. Chilstrom, Singular Value Inequalities: New Approaches to Conjectures. UNF Theses and Dissertations. 2013

[15] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, Cambridge, UK, 2004

[16] Bauer, Eric and Kohavi, Ron. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36(1-2), 1999.

[17] Teo, Choon Hui, Vishwanthan, S.V.N., Smola, Alex J., and Le, Quoc V. Bundle methods for regularized risk minimization. J. Mach. Learn. Res., 11:311–365, March 2010.

[18] L. Wasserman and S. Zhou. A statistical framework for differential privacy. Journal of the American Statistical Association, 105(489):375–389, 2010.

[19] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, EUROCRYPT, volume 4004 of Lecture Notes in Computer Science, pages 486–503. Springer, 2006

[20] T. Dalenius. Towards a methodology for statistical disclosure control. Statistik Tidskrift, 15:429–444, 1977.

[21] R. Agrawal and R. Srikant. Privacy-preserving data mining. SIGMOD Record, 29(2):439–450, 2000.

[22] L. Sweeney. k-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and KnowledgeBased Systems, 10(5), 2002.

[23] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), pages 211–222, 2003.

[24] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In Proceedings of the 22nd International Conference on Data Engineering (ICDE), 2006.

[25] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. Information Systems, 29(4), 2004.

[26] J. Kim and W. Winkler. Multiplicative noise for masking continuous data. Statistics, 2003.

[27] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of LP decoding. Proceedings of the 39th STOC, 2007.

[28] F. McSherry and K. Talwar. Mechanism design via differential privacy. Proceedings of the 48th FOCS, 2007.

[29] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In In Proceedings of the 24th PODS, 2005.

[30] S.R. Ganta, S.P. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge

[31] S. Zhou, K. Ligett, and L. Wasserman. Differential privacy with compression. In Proceedings of the 2009 International Symposium on Information Theory, Seoul, South Korea, 2009.

[32] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets (how to break anonymity of the netflix prize dataset). In Proceedings of 29th IEEE Symposium on Security and Privacy, pages 111–125, 2008.

[33] B. Pinkas. Cryptographic techniques for privacy-preserving data mining. ACM SIGKDD Explorations Newsletter, 4(2), 2002.

[34] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. J. Strauss, and R. N. Wright. Secure multiparty computation of approximations. ACM Trans. Algorithms, 2(3):435–472, 2006.

[35] Lecture 8. Convergence in Ditribution. http://www.math.uah.edu/stat/dist/Convergence.html

[36] Wei Shi, Qing Ling, etc.On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 62, NO. 7, APRIL 1, 2014

[37] Asuncion and Newman. *Adult*. UCI Machine Learning Repository, 2007

[38] Statistical Consulting Group, San Diego State University. http://www.scg.sdsu.edu/dataset-adult_r.html

[39] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou R3579X? Anonymized social networks, hidden patterns, and structural steganography. In Proceedings of the 16th International World Wide Web Conference, 2007.

[40] R. Wang, Y. F. Li, X. Wang, H. Tang, and X.-Y. Zhou. Learning your identity and disease from research papers: information leaks in genome wide association study. In ACM Conference on Computer and Communications Security, pages 534–544, 2009.

[41] O. L. Mangasarian, E. W. Wild, and G. Fung. Privacy-preserving classification of vertically partitioned data via random kernels. ACM Transactions on Knowledge Discovery from Data, 2(3), 2008.

[42] J. Kim and W. Winkler. Multiplicative noise for masking continuous data. Statistics, 2003.

[43] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, Cambridge, UK, 2004.

[44] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. Proc. CRYPTO, 2004.

[45] R. Nishihara, L. Lessard, B. Recht, A. Packard and M. Jordan. A General Analysis of the Convergence of ADMM. International Conference on Machine Learning 32, 2015

[46] S. A. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In Proc. of FOCS, 2008.

[47] VALIANT, L. G. A theory of the learnable. Communications of the ACM 27 (1984), 1134–1142.

[48] [1] N. R. Adam and J. C. Wortmann. Security-control methods for statistical databases: a comparative study. ACM Computing Surveys, 25(4), December 1989.

[49] Dorothy E. Denning. Secure statistical databases with random sample queries. ACM Transactions on Database Systems, 5(3):291–315, September 1980.

[50] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein, editors, SIGMOD Conference, pages 439–450. ACM, 2000.

[51] D. Bertsekas and J. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Athena Scientific, 1997.

[52] S. Shalev-Shwartz. Online Learning: Theory, Algorithms, and Applications. PhD thesis, The Hebrew University of Jerusalem, July 2007.

[53] S. Shalev-Shwartz and N. Srebro. SVM optimization : Inverse dependence on training set size. In The 25th International Conference on Machine Learning (ICML), 2008.

[54] K. Sridharan, N. Srebro, and S. Shalev-Shwartz. Fast rates for regularized objectives. In Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS), 2008.