Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

TC 11 Briefing Papers

Weighted distributed differential privacy ERM: Convex and non-convex☆

Yilin Kang[a,d], Yong Liu[a,b,c,*], Ben Niu[a], Weiping Wang[a]

[a] Institute of Information Engineering, Chinese Academy of Sciences, 89-A, Minzhuang Rd, Haidian District, Beijing, 100093, China

[b] Gaoling School of Artificial Intelligence, Renmin University of China, No. 59 Zhongguancun Street, Haidian District, Beijing, 100872, China

[c] Beijing Key Laboratory of Big Data Management and Analysis Methods, China

[d] School of Cyber Security, University of Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 11 December 2019
Revised 3 November 2020
Accepted 7 March 2021
Available online 18 April 2021

Keywords:
Distributed machine learning
Differential privacy
Weighted parties
Empirical risk minimization
Strongly convex
Polyak-Łojasiewicz condition

ABSTRACT

Distributed machine learning allows different parties to learn a single model over all data sets without disclosing their own data. In this paper, we propose a weighted distributed differentially private (WD-DP) empirical risk minimization (ERM) method to train a model in distributed setting, considering different weights of different clients. For the first time, we theoretically analyze the benefits brought by weighted paradigm in distributed differentially private machine learning. Our method advances the state-of-the-art differentially private ERM methods in distributed setting. By detailed theoretical analysis, we show that in distributed setting, the noise bound and the excess empirical risk bound can be improved by considering different weights held by multiple parties. Additionally, in some situations, the constraint: strongly convexity of the loss function in ERM is not easy to achieve, so we generalize our method to the condition that the loss function is not restricted to be strongly convex but satisfies the Polyak-Łojasiewicz condition. Experiments on real data sets show that our method is more reliable and we improve the performance of distributed differentially private ERM, especially in the case that data scales on different clients are uneven. Moreover, it is an attractive result that our distributed method achieves almost the same theoretical and experimental results as previous centralized methods.

1. Introduction

In recent years, machine learning has been widely used in many fields such as data mining and pattern recognition (He et al. (2015); Wang et al. (2018); Xu et al. (2018); Zhang et al. (2019)). Because of the need of data for training machine learning algorithms, tremendous data has been collected by individuals and companies. As a result, sensi-

☆ This work was supported by Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098; the National Natural Science Foundation of China [grant numbers 61703396, 62076234]; the Youth Innovation Promotion Association CAS; the Open Research Project of the State Key Laboratory of Media Convergence and Communication, Communication University of China, China (No. SKLMCC2020KF004).

* Corresponding author.
E-mail addresses: kangyilin@iie.ac.cn (Y. Kang), liuyonggsai@ruc.edu.cn (Y. Liu), niuben@iie.ac.cn (B. Niu), wangweiping@iie.ac.cn (W. Wang).

https://doi.org/10.1016/j.cose.2021.102275
0167-4048/

tive information disclosure is becoming a huge problem. In addition to data itself, model parameters trained by data can reveal sensitive information in an undirect way as well (Fredrikson et al. (2014); Shokri et al. (2017)).

To solve the problems mentioned above, differential privacy (Dwork (2011)) was proposed to preserve privacy in the field of machine learning and has been applied to principal component analysis (PCA) (Chaudhuri et al. (2013); Ge et al. (2018); Wang and Xu (2019b)), regression (Bernstein and Sheldon (2019); Chaudhuri and Monteleoni (2009); Smith et al. (2018); Zhang et al. (2012)), boosting (Dwork et al. (2010); Zhao et al. (2018)), generative adversarial networks (GAN) (Wu et al. (2019); Xu et al. (2019)), graph algorithms (Arora and Upadhyay (2019); Ullman and Sealfon (2019)), deep learning (Abadi et al. (2016); Farquhar and Gal (2019); Shokri and Shmatikov (2015)) and other fields.

There are mainly three methods to achieve differential privacy: output perturbation (Bassily et al. (2014); Dwork et al. (2006); Pathak et al. (2010); Zhang et al. (2017)), objective perturbation (Chaudhuri and Monteleoni (2009); Chaudhuri et al. (2011)) and gradient perturbation (Abadi et al. (2016); Bassily et al. (2014); Geyer et al. (2017)). Among them, gradient perturbation method is the most popular method because it can be applied to any gradient descent method, a commonly used optimization method in machine learning. Meanwhile, it not only protects the model parameters but also the gradients, which makes it more reliable.

Meanwhile, cooperations between organizations are becoming more common, multiple parties desire to train a machine learning model by combining data in the fields such as biomedicine and financial fraud detection. By combining training data, the performance of the model can be better because of the increasing of training data. However, sharing data between clients is unwise and may disclose personal sensitive information.

Distributed machine learning is an approach proposed to solve the multiple-party machine learning problem. Different parties own different training data sets (including sensitive information of individuals), and for privacy consideration, data is not shared between clients. Among many distributed learning strategies, *divide and conquer* is one of the simplest methods. It preserves privacy by minimizing information communications, which has caused widespread concerns of researchers. The first distributed differentially private machine learning method was proposed in Pathak et al. (2010), in which privacy was preserved by output perturbation. Jayaraman et al. (2018) introduced two differential privacy methods to distributed setting: output perturbation and gradient perturbation, achieving better theoretical and experimental results.

However, distributed methods mentioned above do not consider about different weights of multiple parties when aggregating parameters. In real scenarios, data scales of different clients are always uneven[1]. Thus, the qualities of models trained by different clients are uneven, simply averaging without weights will lead worse performance on accuracy. For example, for the simplest case, supposing there are 2 clients, the first owns 100 data instances and the second owns 900 data instances. It is a common sense that the model trained by larger data set is better than which trained by smaller data set. When aggregating model parameters, if both clients are equally weighted by 0.5 (like previous methods Jayaraman et al. (2018); Pathak et al. (2010)), the local model trained by the first client is considered too much and the aggregated machine learning model will be 'dragged' by the first model (trained by 100 data instances) with a large probability. By applying different weights of multiple parties, the weights held by the first client and the second client are 0.1 and 0.9, respectively. Under weighted paradigm, all the local models are considered to reasonable extents, rather than 'over' or 'under' aggregated.

Besides, federated learning (McMahan et al. (2016)) and distributed machine learning are connected closely. In distributed machine learning, a 'server' absolutely controls all parties and different parties aim to train a large-scale machine learning model. And in federated learning, each party absolutely controls over its own data and can decide whether or not to disclose it. Despite of this, it can be observed that the frameworks of federated learning and distributed machine learning are similar: all parties train the models locally and a server aggregates the models. Thus, the 'boundary' between distributed machine learning and federated learning is blurred and it is hard to define the difference between them.

To get the global model, when aggregating parameters from different parties, apart from averaging, Alternating Direction Method of Multipliers (ADMM) (Boyd et al. (2011)) and Blockwise Model Update Filtering (BMUF) (Chen and Huo (2016)) are suitable methods and have been applied to lasso model (Zhang and Kwok (2014)), DNN (Chen and Huo (2016)), CNN (Choy (2015); Elgabli et al. (2019); Fu et al. (2019)), Matrix Factorization (Yu et al. (2014); Zhang and Kwok (2014)), LSTM (Chen et al. (2020); Chen and Huo (2016); Huang et al. (2020)) and other fields. Moreover, the combination of ADMM and differential privacy was studied in Ding et al. (2019); Huang et al. (2020); Wang et al. (2019). However, aggregating methods such as ADMM and BMUF require sending the gradients to the server at each iteration (multiple communications), which increases the communication complexity and privacy risk. In this paper, we focus on the more general method: averaging model parameters when getting the global model, in which only one communication is necessary.

To address the problems carried by uneven data scales over different clients when averaging model parameters, we propose Weighted Distributed Differential Privacy (WD-DP) method in this paper. This paper focuses on ERM, which is generally used in supervised learning and covers a variety of machine learning tasks. We consider weighted averaging paradigm, rather than simply averaging when aggregating models, in order to reduce the negative impact caused by uneven data scales, which leads better noise bound and excess empirical risk bound theoretically. This is the first time to analyze 'what we can get from the *weighted framework*' in distributed differentially private machine learning. Experiments on real data sets also show that the classification and regression performance of our method is much better than the

[1] In this paper, 'uneven data scales' means the sizes of data sets owned by different clients may differ a lot.

method proposed by Jayaraman et al. (2018), the best method in distributed differentially private ERM to the best of our knowledge.

Moreover, most previous theoretical analysis on differentially private ERM is based on strongly convex loss function and this constraint is not easy to guarantee in some situations. To solve the problem, first, we improve the proof process of the excess empirical risk bound proposed by Wang et al. (2017) and then generalize our method to non-convex loss functions which satisfy the Polyak-Łojasiewicz condition.

The contribution of this paper consists of three parts:

Weighted Distributed Differential Privacy Method. We propose a weighted distributed differential privacy paradigm: WD-DP in this paper, taking weights of different clients into account, in order to improve the performance of the model. For the first time, we theoretically analyze 'what can be benefited from the weighted paradigm' and show that weighted paradigm can reduce the negative impact caused by uneven data scales. By applying weights, different clients in the distributed system work as one, making the system more 'centralized'.

Superior Theoretical and Experimental Resutls. Theoretical and experimental results show that our proposed WD-DP method achieves better performance than simply averaging method (Jayaraman et al. (2018); Pathak et al. (2010)). By applying weighted paradigm, our proposed distributed method almost achieves the performance of the central method (Wang et al. (2017); Zhang et al. (2017)), which is an attractive result.

A More General Case. In previous distributed differential privacy methods, the loss function is always assumed to be strongly convex. In this paper, we extend this assumption to a more general case: the loss function satisfies the Polyak-Łojasiewicz condition, but does not need to be convex. Theoretical analysis and experiments show that under non-convex case, the results are similar.

The rest of the paper is organized as follows. We introduce some related work in Section 2, including distributed differentially private ERM, centralized differentially private ERM under non-convex condition, and federated learning. In Section 3, we propose our method: WD-DP in detail and then analyze the (ϵ, δ)-differential privacy of our method. We give the theoretical analysis of the excess empirical risk bounds of our method on both convex and non-convex conditions in Section 4. We present the experimental results in Section 5. Finally, we conclude the paper in Section 6.

2. Related work

In this section, we first introduce some related work over distributed differentially private machine learning. Then, we introduce some work on centralized differentially private ERM under non-convex condition. And finally, we claim the differences between our method and federated learning.

2.1. *Distributed setting*

The first distributed privacy preserving protocol was proposed by Pathak et al. (2010). There was a regularization term $\lambda N(\theta)$ in the objective function (which is not considered in this paper for the sake of simplicity), where θ is the model with p parameters (i.e. $\theta \in \mathbb{R}^p$). Different parties trained models locally and interacted with the curator to construct additive shares of a perturbed aggregated model. In this work, the delivery of parameters relied on homomorphic encryption (Paillier (1999)), which is expensive on computation. And differential privacy was guaranteed by output perturbation, adding Laplace noise.

By combining privacy with secure multi-party computation (SMC) (Tian et al. (2016)), Jayaraman et al. (2018) proposed a distributed method, guaranteeing differential privacy by output perturbation and gradient perturbation, adding noise within a SMC. In this work, the loss function $\ell(\cdot)$ was assumed G-Lipschitz and L-smooth, the noise bound was related to the training iterations T. The noise bound and excess empirical risk bound in this work are better than which proposed by Pathak et al. (2010). Particularly, in this method, parties aggregated parameters by simply averaging. As a result, if data scales on different parties are not even, the performance will decrease rapidly. Unfortunately, in real scenarios, data scales on clients are always uneven.

In the method WD-DP, proposed by this paper, we consider different weights held by different parties when aggregating models, and achieve better performance in distributed setting both theoretically and practically, no matter data scales are even or not. Additionally, previous method proposed by Jayaraman et al. (2018) is a special case of WD-DP, in the condition that the sizes of clients are the same. Thus, our method is more general and adapts to most scenarios. Moreover, for the first time, we analyze the theoretical improvements brought by the 'weighted framework', which is shown in Section 3 and Section 4. The comparison between our method and other methods mentioned above on noise bound and excess empirical risk bound is given in Table 1.

In Table 1, the noise bound and excess empirical risk bound of our method are better than the best previous distributed method we have known, proposed by Jayaraman et al. (2018), by a factor of $\frac{(mn_{(1)})^2}{n^2}$ and $\frac{(mn_{(1)})^2 \log(n)}{(\log(mn_{(1)})n)^2}$, respectively, where m is the number of parties, $n_{(1)}$ denotes the smallest size of data set owned by parties and n represents the total number of data instances over all data sets. Particularly, our method is much better when data scales on clients are uneven. Moreover, the method mentioned above is a special case of our method WD-DP under average setting. And obviously, the excess empirical risk bound of our method is far better than which in Pathak et al. (2010). It is worth emphasizing that although our method is proposed under distributed setting, it achieves almost the same theoretical performance as centralized methods.

2.2. *Non-convex ERM*

The first work on centralized non-convex differentially private ERM problem is Random Round Private SGD (Zhang et al. (2017)), guaranteeing (ϵ, δ)-differential pri-

Table 1 – Comparison between our method and other methods on noise bound and excess empirical risk bound.

	Gaussian Noise Bound	Excess Empirical Risk Bound	Distributed	Non-convex
Pathak et al. (2010)	None	$O\left(\frac{(m-1)^2(\lambda+1)}{n_{(1)}^2\lambda^2}+\frac{p^2(\lambda+1)\log^2(p/\delta)}{n_{(1)}^2\epsilon^2\lambda^2}+\frac{p(m-1)(\lambda+1)\log(p/\delta)}{n_{(1)}^2\epsilon\lambda^2}\right)$	Yes	No
Jayaraman et al. (2018)	$O\left(\frac{G^2T\log(1/\delta)}{m^2n_{(1)}^2\epsilon^2}\right)$	$O\left(\frac{pG^2L\log^2(mn_{(1)})\log(1/\delta)}{m^2n_{(1)}^2\lambda^2\epsilon^2}\right)$	Yes	No
Zhang et al. (2017)	$O\left(\frac{pG^2\log(n/\delta)\log(1/\delta)+G^2\epsilon^2}{\epsilon^2}\right)$	$O\left(\frac{G\sqrt{p\log(n/\delta)\log(1/\delta)}D}{n\epsilon}\right)$	No	Yes
Wang et al. (2017)	$O\left(\frac{G^2T\ln(1/\delta)}{n^2\epsilon^2}\right)$	$O\left(\frac{pG^2\log^2(n)\ln(1/\delta)}{n^2\epsilon^2}\right)$	No	Yes
Our Method WD-DP	$O\left(\frac{G^2T\ln(1/\delta)}{\boldsymbol{n^2}\epsilon^2}\right)$	$O\left(\frac{pG^2\boldsymbol{\log(n)}\ln(1/\delta)}{\boldsymbol{n^2}\epsilon^2}\right)$	Yes	Yes

vacy over non-convex loss function. In this method, the excess empirical risk bound was proportional to D, the upper bound of the ℓ_2 norm of the model's parameters (i.e. $\|\theta\| \leq D$ for all θ).

Theoretical analysis on noise bound and excess empirical risk bound of gradient perturbation method under non-convex condition in centralized setting was also studied in Wang and Xu (2019a); Wang et al. (2017). However, the proof process on the excess empirical risk bound can be improved, which will be detailed in Section 4 in this paper.

Bun and Steinke (2016) studied the upper bounds of the utility of centralized differentially private ERM problem with non-convex loss function. This work considered the problem in both low and high dimensional spaces and showed that for some special non-convex loss functions, the utility can be improved to a level similar to convex ones.

In this paper, first, we improve the proof process on excess empirical risk bound in Wang et al. (2017) (Section 4.1). Then, we extend our method to the condition that the loss function is not constrained strongly convex (or even convex). To the best of our knowledge, this is the first time to analyze non-convex differentially private ERM in distributed setting. The comparison between our method and previous centralized methods under non-convex condition is shown in Table 1.

In Table 1, by improving the proof process proposed by Wang et al. (2017), our excess empirical risk bound is tighter than before by a factor of $\log(n)$. Meanwhile, considering the parameter D is hard to control in general, our method is more reliable than the method proposed by Zhang et al. (2017), with a better noise bound.

2.3. *Federated learning*

McMahan et al. (2016) proposed a decentralized machine learning method: federated learning. In federated learning, training data is stored locally and a shared model is learned by aggregating local gradients, whose aim is to minimize communications between clients and the 'server'. However, extra privacy was not taken into account, except for decentralized data storage.

Based on McMahan et al. (2016), a user-level differentially private LSTM (Long Short-Term Memory) was proposed in McMahan et al. (2017), and was applied to language models. Geyer et al. (2017) proposed a method to guarantee whether a client participants in federated learning cannot be inferred by malicious adversaries, preserving client-level privacy. However, all previous differentially private federated learning methods concentrate on a certain problem or a certain machine learning model (such as LSTM), rather than a general paradigm.

From our perspective, federated learning is a kind of distributed machine learning method, focusing more on communication complexity (communication cost). Thus, some differences between our proposed WD-DP method and federated learning should be clarified in detail. Although federated learning considers about different weights held by different clients, it ignores the effects caused by weighted framework. However, in this paper, we focus more on the theoretical analysis and answer the question: "how can we benefit from weighted distributed framework". This paper theoretically prove that by considering weights, the noise bound and the excess empirical risk bound in distributed machine learning can be improved, for the first time. Meanwhile, we focus on a general paradigm: ERM, covering a variety of machine learning tasks. Considering that our method is a general paradigm and the averaging framework is easy to apply, we do not place too much emphasis on application scenarios. And of course, our method can be easily applied to federated learning. For the same reason, the comparisons in this paper are mainly applied between previous differentially private ERM methods and our method.

3. WD-DP: Weighted distributed differential privacy empirical risk minimization

In this section, we first introduce some basic definitions in distributed machine learning. Then, we propose WD-DP method in detail and give theoretical analysis of (ϵ, δ)-DP of our algorithm.

Given a d-dimensional vector $\mathbf{x} = [x_1, x_2, \ldots, x_d]^\top$, denote its ℓ_2-norm: $\left(\sum_{i=1}^{d}|x_i|^2\right)^{\frac{1}{2}}$ by $\|\mathbf{x}\|$. $\tilde{O}(\cdot)$ is similar to $O(\cdot)$, but hiding factors polynomial in $\log n$ and $\log(1/\delta)$. Denote the probability distribution of data as $\mathcal{D}^n$, for two databases $D, D' \in \mathcal{D}^n$ differing by one single element, they are denoted as $D \sim D'$, called *adjacent databases*.

Definition 1 (Dwork et al.(2014)). Randomized function $\mathcal{A}: \mathcal{D}^n \to \mathbb{R}^p$ is (ϵ, δ)-differential privacy ((ϵ, δ)-DP) if

$$\mathbb{P}[\mathcal{A}(D) \in S] \leq e^{\epsilon}\mathbb{P}[\mathcal{A}(D') \in S] + \delta,$$

where $S \in \text{range}(\mathcal{A})$ and p is the number of parameters.

According to the definition, differential privacy guarantees that data sets D, D' lead to similar distributions on the output of a randomized algorithm $\mathcal{A}$. This implies that an adversary will draw essentially the same conclusions about an individual whether or not that individual's data was used even if many records are known a priori to the adversary.

In differentially private machine learning, the adversary can get the machine learning model θ (at each iteration) and attempts to infer the sensitive information of individuals included in data set. Some kind of attacks, such as membership inference attack, attribute inference attack, memorization attack (Backes et al. (2016); Carlini et al. (2019); Jayaraman and Evans (2019)), have been found to be thwarted by differential privacy mechanisms.

The objective function in centralized ERM is defined as:

$$L_D(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, x_i, y_i),$$

where (x_i, y_i) denotes data instance, $\ell(\cdot)$ is the loss function.

Definition 2 (Dwork et al.(2014)). For adjacent databases $D \sim D'$, the ℓ_2-sensitivity of a function $f(\cdot)$ is defined as:

$$S(f) = \max \|f(D) - f(D')\|, \tag{1}$$

ℓ_2-sensitivity captures the magnitude by which a single individual's data can change function $f(\cdot)$ in the worst case.

3.1. *Distributed differential privacy*

Suppose there are m parties $P_1, P_2, \ldots, P_m$, owning data sets $D_1, D_2, \ldots, D_m$ with size $n_1, n_2, \ldots, n_m$, respectively. Different parties train their own model $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(m)}$ locally to prevent data disclosing (in this paper, we denote model by parameters), and then their models are aggregated by a trusted third party (called server).

So, in distributed setting, the objective function is:

$$L_D(\theta) = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{n_j}\sum_{i=1}^{n_j} \ell(\theta, x_i^{(j)}, y_i^{(j)}), \tag{2}$$

where $(x_i^{(j)}, y_i^{(j)})$ denotes data owned by party j.

By Eq. (2), when it comes to gradient perturbation, at round t, with learning rate η, we have the updating criteria on party j:

$$\theta_{t+1}^{(j)} = \theta_t^{(j)} - \eta(\nabla L_{D_j}(\theta_t^{(j)}) + z_t),$$

and by simply averaging (Jayaraman et al. (2018); Pathak et al. (2010)), the updating criteria on server after T local iterations is:

$$\theta^{(c)} = \frac{1}{m}\sum_{j=1}^{m}\theta_T^{(j)}, \tag{3}$$

where $L_{D_j}(\theta)$ represents the objective function over party j, $z_t \sim \mathcal{N}(0, \sigma^2 I_p)$ is Gaussian noise guaranteeing differential privacy and $\theta^{(c)}$ denotes the aggregated model on server.

It can be observed that each party P_i processes its own data set D_i, without sharing data with other parties. After all m models trained, they are aggregated by the server. The aim is to learn a classifier from the union of all the data $D_1 \cup D_2 \ldots \cup D_m$ without data exchange. And $\theta^{(c)}$ is the final model we want.

3.2. *Weighted distributed differential privacy*

Traditional methods (Jayaraman et al. (2018); Pathak et al. (2010)) use Eq. (3) to aggregate parameters by simply averaging. However, this method pays more attention on data instances in small data sets, which leads worse noise bound and excess empirical risk bound. In real scenarios, data scales on clients are always uneven, so simply averaging leads worse performance (an example is given in Section 1).

To solve the problem mentioned above, instead of simply averaging the parameters when aggregating models, we take weights of different parties into account. The weight is related to the data set's size owned by corresponding client, leading the updating criteria on the server to:

$$\theta^{(c)} = \sum_{j=1}^{m}\left(\frac{n_j}{n} + z_w\right)\theta_T^{(j)},$$

where z_w is the random noise to preserve the privacy when applying weights.

This formulation is similar to Eq. (3). It seems like a simple trick, only adding weights on the models of clients. However, in this paper, for the first time, we analyze "what can be given by adding weights'. Moreover, we theoretically prove that our *weighted distributed framework* is a general paradigm, achieving almost the same performance as previous centralized methods, which is an attractive result benefitting from "weighted framework'.

When considering about weights of different parties, data instances in different parties are paid same attentions, which reduces the negative impacts caused by a single bad data instance, rare but special high noise generated for guaranteeing differential privacy or uneven data scales.

Although in the setting of federated learning (such as McMahan et al. (2016)), weights of different parties are also considered, they are seemed as public knowledge, which may also reveal the client's side information and the adversary can use the weights to infer the identity of each client. However, in distributed setting, the server controls all the parties and weights of parties are only owned by the server, which decreases the privacy risk caused by the weights. Moreover, our method considers this kind of risk and preserves the privacy which may disclose by weights (detailed in Theorem 1).

Our method is detailed in Algorithm 1.

In this paper, we assume that the privacy budgets (ϵ_1, ϵ_2, δ_1 and δ_2) are assigned by the trusted server. As a result, the random variables z_t added to each party share the same variance (line 3–5 in Algorithm 1) and (ϵ, δ)-differential privacy is achieved on the final model $\theta^{(c)}$. However, different clients may

Algorithm 1 Weighted Distributed Differential Privacy ERM Method: WD-DP.

Require: m parties indexed by j, number of local iteration rounds T, learning rate η

function DISTRIBUTEDLEARNING(m, T, η)
  First, m parties download random $\theta^{(c)}$ from the server as initialization.
  Party j **(**$j \in \{1, 2, \ldots, m\}$**) executes at round** t:
    $\theta_{t+1}^{(j)} = \theta_t^{(j)} - \eta(\nabla L_{D_j}(\theta_t^{(j)}) + z_t)$,
    where $z_t \sim \mathcal{N}(0, \sigma^2 I_p)$.
  Server executes after T **local rounds**:
    $\theta^{(c)} = \sum_{j=1}^{m} \left(\frac{n_j}{n} + z_w\right)\theta_T^{(j)}$,
    where $z_w \sim \mathcal{N}(0, \sigma_w^2)$.
  return $\theta^{(c)}$.
end function

have different privacy requirements (different privacy budgets). It is a different scenario, but can be easily analyzed by following our thoughts, which is detailed in Theorem 1.

3.2.1. *Differential privacy on final model*

In this paper, we guarantee (ϵ, δ)-DP using Gaussian Mechanism (Dwork et al. (2006)) and moments accountant method (Abadi et al. (2016)).

Theorem 1. *In Algorithm 1, for $\epsilon_1, \delta_1, \epsilon_2, \delta_2 > 0$, if $\ell(\theta, x, y)$ is G-Lipschitz over θ, with*

$$\sigma^2 = c\frac{G^2 T \ln(1/\delta_1)}{n^2\epsilon_1^2}, \tag{4}$$

and

$$\sigma_w^2 = c\frac{\ln(1/\delta_2)}{n^2\epsilon_2^2}, \tag{5}$$

$\theta^{(c)}$ is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP for some constant c.

Proof of Theorem 1.. The output of Algorithm 1: $\theta^{(c)}$ satisfies $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP, in which ϵ_1, δ_1 are for DP when training and ϵ_2, δ_2 are for DP when applying weights.

First, we analyze the privacy when training (guaranteed by Eq. (4)).

The t^{th} query which may disclose privacy can be seemed as a randomized mechanism:

$$\begin{aligned} M_t &= \sum_{j=1}^{m}\left(\frac{n_j}{n} + z_w\right)\left[\frac{1}{n_j}\sum_{i=1}^{n_j}\nabla\ell(\theta_t^{(c)}, x_i^{(j)}, y_i^{(j)}) + z_t\right] \\ &= \underbrace{\frac{1}{n}\sum_{i=1}^{n}\nabla\ell(\theta_t^{(c)}, x_i, y_i) + z_t}_{A_t} \\ &\quad + \underbrace{\sum_{j=1}^{m}\frac{z_w}{n_j}\left[\sum_{i=1}^{n_j}\nabla\ell(\theta_t^{(c)}, x_i^{(j)}, y_i^{(j)}) + z_t\right]}_{B_t}, \end{aligned} \tag{6}$$

where $\theta_t^{(c)}$ represents $\theta^{(c)}$ after t local rounds.

First, we consider $A_t = \frac{1}{n}\sum_{i=1}^{n}\nabla\ell(\theta_t^{(c)}, x_i, y_i) + z_t$.

In the moments accountant method (Abadi et al. (2016)), the λ^{th} moment $\alpha_M(\lambda; D, D')$ on randomized mechanism M is defined as:

$$\alpha_M(\lambda; D, D') = \log \mathbb{E}_{o\sim M(D)}[\exp(\lambda c(o; M, D, D'))], \tag{7}$$

where $c(o; M, D, D')$ is privacy loss at output o, defined as:

$$c(o; M, D, D') = \log\frac{\mathbb{P}[M(D) = o]}{\mathbb{P}[M(D') = o]}. \tag{8}$$

In order to preserve privacy, it is necessary to bound all possible $\alpha_M(\lambda; D, D')$.

So, $\alpha_M(\lambda)$ is defined as:

$$\alpha_M(\lambda) = \max_{D,D'}\alpha_M(\lambda; D, D').$$

Considering that:

$$\begin{aligned} A_t &= \frac{1}{n}\sum_{i=1}^{n}\nabla\ell(\theta_t^{(c)}, x_i, y_i) + z_t \\ &= \frac{1}{n}\sum_{i=1}^{n}\nabla\ell(\theta_t^{(c)}, x_i, y_i) + \mathcal{N}(0, \sigma^2 I_p). \end{aligned}$$

We denote probability distributions on adjacent databases D and D' over mechanism A_t as P and Q:

$$P = \nabla L_D(\theta_t^{(c)}) + \mathcal{N}(0, \sigma^2 I_p) = \mathcal{N}(\nabla L_D(\theta_t^{(c)}), \sigma^2 I_p),$$
$$Q = \nabla L_{D'}(\theta_t^{(c)}) + \mathcal{N}(0, \sigma^2 I_p) = \mathcal{N}(\nabla L_{D'}(\theta_t^{(c)}), \sigma^2 I_p).$$

The Rényi divergence D_α over distributions $P(x)$ and $Q(x)$ is defined as (Bun and Steinke (2016)):

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1}\log\left(\mathbb{E}_{x\sim P}\left[\left(\frac{P(x)}{Q(x)}\right)^{\alpha-1}\right]\right). \tag{9}$$

By Eq. (7), (8), (9) and definitions P, Q, we have equations below over mechanism A_t:

$$\begin{aligned} \alpha_{A_t}(\lambda) &= \log \mathbb{E}_{o\sim P}\left[\exp\left(\lambda\log\left(\frac{P}{Q}\right)\right)\right] \\ &= \log \mathbb{E}_{o\sim P}\left[\left(\frac{P}{Q}\right)^{\lambda}\right] \\ &= \lambda D_{\lambda+1}(P\|Q). \end{aligned}$$

By Lemma 2.5 in Bun and Steinke (2016), we have:

$$\lambda D_{\lambda+1}(P\|Q) = \frac{\lambda(\lambda+1)\left\|\nabla L_D(\theta_t^{(c)}) - \nabla L_{D'}(\theta_t^{(c)})\right\|^2}{2\sigma^2}.$$

Note that $\ell(\cdot)$ is G-Lipschitz (denoted as G below), and suppose the only different element between D and D' is the n^{th} one, we have:

$$\begin{aligned} &\nabla L_D(\theta_t^{(c)}) - \nabla L_{D'}(\theta_t^{(c)}) \\ &= \frac{1}{n}\left(\sum_{i=1}^{n-1}\nabla\ell(\theta_t^{(c)}, x_i, y_i) + \nabla\ell(\theta_t^{(c)}, x_n, y_n)\right) \\ &\quad - \frac{1}{n}\left(\sum_{i=1}^{n-1}\nabla\ell(\theta_t^{(c)}, x_i, y_i) + \nabla\ell(\theta_t^{(c)}, x_n', y_n')\right) \\ &= \frac{1}{n}\left(\nabla\ell(\theta_t^{(c)}, x_n, y_n) - \nabla\ell(\theta_t^{(c)}, x_n', y_n')\right) \\ &\overset{(G)}{\leq} \frac{2G}{n}. \end{aligned}$$

Thus, we have:

$$\alpha_{A_t}(\lambda) = \lambda D_{\lambda+1}(P\|Q) \leq \frac{2G^2\lambda(\lambda+1)}{\sigma^2 n^2}.$$

By Theorem 2.1 in Abadi et al. (2016), for some constant c_1, we have:

$$\alpha_A(\lambda) \le \sum_{t=1}^{T} \alpha_{A_t}(\lambda) \le c_1\lambda^2 \frac{G^2T}{\sigma^2n^2}.$$

Taking $\sigma^2 = c\frac{G^2T\ln(1/\delta_1)}{n^2\epsilon_1^2}$ for some constant c:

$$\alpha_A(\lambda) \le c_1\lambda^2 \frac{G^2T}{\sigma^2n^2} \le \frac{\lambda\epsilon_1}{2},$$

and as a result, we have:

$$\frac{c_1\lambda^2G^2T\epsilon_1^2}{cG^2T\ln(1/\delta_1)} \le \frac{\lambda\epsilon_1}{2},$$

which means:

$$\delta_1 \le \exp(\frac{-\lambda\epsilon_1}{2}).$$

Due to Theorem 2.2 in Abadi et al. (2016), A_t is (ϵ_1, δ_1)-DP.

In Eq. (6), differential privacy can be guaranteed by summing part A_t over T iterations. Part B_t gives more randomness to M_t and as a result, M_t strictly satisfies (ϵ_1, δ_1)-differential privacy.

Then, we analyze the privacy when applying weights (guaranteed by Eq. (5)).

It is easy to follow that the ℓ_2-sensitivity when applying weights is $\frac{1}{n}$ (i.e. differing by a single element, the maximum change on weights is $\frac{1}{n}$).

By Gaussian mechanism (Dwork et al. (2014)), (ϵ, δ)-DP is guaranteed if random noise $z \sim \mathcal{N}(0, \sigma^2)$ is added to a query, with $\sigma \ge \frac{\sqrt{2\ln(1.25/\delta)}S(f)}{\epsilon}$, where $S(f)$ is the ℓ_2-sensitivity of the query.

Therefore, with σ_w is Eq. (5), (ϵ_2, δ_2)-differential privacy is guaranteed by Gaussian mechanism.

Then, by applying composition theorems (Dwork et al. (2014)) over processes **delivering weights** and **training models**, Algorithm 1 is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-differential privacy overall. □

In Theorem 1, $\ell(\cdot)$ is G-Lipschitz, but not constrained *convex*. Thus, Theorem 1 is general in both convex and non-convex conditions.

Although our proposed WD-DP considers distributed setting, Theorem 1 is not related to the number of parties m or the data scale n_j on client j. As a result, Gaussian noise guaranteeing differential privacy is not related to multiple parties, but has the same form as in centralized setting.

Moreover, we consider different weights held by parties when aggregating models, so the bound is tighter than which introduced by Jayaraman et al. (2018) by a factor of $\frac{(mn_{(1)})^2}{n^2}$, where $n_{(1)}$ is the smallest size of data sets owned by all the parties. When data scales on clients are not even, our method is much better.

3.2.2. *Differential privacy on clients*

After analyzing the privacy on the final model, models of clients should be paid attentions. Discussions of differential privacy in this part are based on adjacent datasets D_j and D'_j over each client j.

Theorem 2. *In Algorithm 1, with σ the same as in Eq. (4), the model $\theta^{(j)}$ on client j is $(\frac{n}{n_j}\epsilon_1, \delta_1)$-differential privacy.*

Here, ϵ_1 and δ_1 are the same as in Theorem 1 (assigned by the trusted server), rather than locally determined. Additionally, like claimed in Section 3.2, our analysis can be easily generalized to the cases that privacy budgets are determined locally. For models on clients, there is no need to consider the privacy loss when delivering weights, so ϵ_2 and δ_2 are not considered.

Proof of Theorem 2.. Denote M_t on client j as $M_t^{(j)}$, like in Theorem 1, we have:

$$M_t^{(j)} = \frac{1}{n_j}\sum_{i=1}^{n_j} \nabla\ell(\theta_t^{(j)}, x_i^{(j)}, y_i^{(j)}) + \mathcal{N}(0, \sigma^2 I_p).$$

Then, following the proof steps in Theorem 1, for some constant c_1, we have:

$$\alpha_M^{(j)}(\lambda) \le c_1\lambda^2 \frac{G^2T}{\sigma^2n_j^2},$$

where $\alpha_M^{(j)}(\lambda)$ represents *moment* (Abadi et al. (2016)) on client j.

Taking σ the same as in Eq. (4), for some constant c, we have:

$$\alpha_M^{(j)}(\lambda) \le c\frac{\lambda^2\epsilon_1^2n^2}{\log(1/\delta_1)n_j^2}.$$

We can guarantee that:

$$\alpha_M^{(j)}(\lambda) \le c\frac{\lambda^2\epsilon_1^2n^2}{\log(1/\delta_1)n_j^2} \le \lambda\epsilon_1\frac{n}{2n_j},$$

and as a result:

$$\delta_1 \le \exp(-\frac{2n}{n_j}\lambda\epsilon_1) \le \exp(-\frac{n}{2n_j}\lambda\epsilon_1),$$

which means $(\frac{n}{n_j}\epsilon_1, \delta_1)$-DP due to Theorem 2.2 in Abadi et al. (2016). □

Remark 1. In Theorem 2, for clients with larger data scales, we preserve more privacy (denoted by smaller $\frac{n}{n_j}\epsilon_1$) and vice versa. It is in line with the 'common sense': larger data sets are more important and need more protections.

3.3. *The relationship between ϵ and δ*

In this part, following Triastcyn and Faltings (2019), we discuss the relationship between ϵ and δ. In our algorithm, ϵ_1 is connected with δ_1 and ϵ_2 is connected with δ_2, the discussion holds for both of the pairs.

In the field of differential privacy, the probability of the failure of ϵ-DP is represented by δ, i.e.

$$\mathbb{P}[c(o; M, D, D') \ge \epsilon] \le \delta, \tag{10}$$

where c is the privacy loss, defined in (8).

By extended Markov's inequality, for monotonically increasing non-negative function $\varphi(\cdot)$:

$$\mathbb{P}[|c(o; M, D, D')| \geq \epsilon] \leq \frac{\mathbb{E}[\varphi(|c(o; M, D, D')|)]}{\varphi(\epsilon)}.$$

By applying $\varphi(x) = e^{\lambda x}$, we have[2]:

$$\mathbb{P}[c(o; M, D, D') \geq \epsilon] \leq \frac{\mathbb{E}[e^{\lambda c(o;M,D,D')}]}{e^{\lambda\epsilon}} \overset{(10)}{=} \delta,$$

where $e^{\lambda c(o;M,D,D')} = \left(\frac{\mathbb{P}(M(D)=o)}{\mathbb{P}(M(D')=o)}\right)^{\lambda}$, a Rényi divergence.

Thus, ϵ and δ are connected by the expectation of the privacy loss.

4. Theoretical analysis over convex and non-convex conditions

In this section, first we give the analysis of excess empirical risk of our method WD-DP under strongly convex condition. Then, we generalize it to non-convex loss functions which satisfy the Polyak-Łojasiewicz condition. To the best of our knowledge, this is the first theoretical analysis of excess empirical risk bound on non-convex distributed differentially private ERM.

4.1. *Convex*

In this part, we give the theoretical analysis on the excess empirical risk under the condition that the objective function $L(\cdot)$ is λ-strongly convex.

Theorem 3. *Suppose that the loss function $\ell(\theta, x, y)$ is G-Lipschitz and L-smooth over θ, $L_D(\theta)$ is λ-strongly convex and differentiable, with σ in Eq. (4) and learning rate $\eta = \frac{1}{L}$. We have:*

$$\mathbb{E}[L_D(\theta_T)] - L_D^* \leq O\left(\frac{pG^2 \ln(1/\delta_1)\log(n)}{n^2\epsilon_1^2}\right),$$

where $T = \tilde{O}\left(\log(\frac{n^2\epsilon_1^2}{pG^2\ln(1/\delta_1)})\right)$, $L_D^* = \min_{\theta\in\mathbb{R}^p} L_D(\theta)$ and p is the number of parameters.

Proof of Theorem 3.. According to the updating criteria of gradient descent, we have[3]:

$$\theta_{t+1} - \theta_t = -\eta(\nabla L_D(\theta_t) + z_t) = -\frac{1}{L}(\nabla L_D(\theta_t) + z_t). \tag{11}$$

First, we build a connection between $L_D(\theta_{t+1})$ and $L_D(\theta_t)$.

Note that the loss function $\ell(\cdot)$ is L-smooth (denoted as L below) and the objective function $L_D(\theta)$ is differentiable (denoted as d below), we have:

$$\begin{aligned}
&\mathbb{E}_{z_t}[L_D(\theta_{t+1}) - L_D(\theta_t)] \\
&\overset{(L,d)}{\leq} \mathbb{E}_{z_t}\left[< \nabla L_D(\theta_t), \theta_{t+1} - \theta_t > + \tfrac{L}{2}\left\|\theta_{t+1} - \theta_t\right\|^2\right] \\
&\leq -\tfrac{1}{L}\left\|\nabla L_D(\theta_t)\right\|^2 - \tfrac{1}{L} < \nabla L_D(\theta_t), z_t > \\
&\quad + \tfrac{1}{2L}\left\|\nabla L_D(\theta_t)\right\|^2 + \tfrac{1}{2L}\mathbb{E}_{z_t}\|z_t\|^2 \\
&\quad + \tfrac{1}{L} < \nabla L_D(\theta_t), z_t > \\
&-\tfrac{1}{2L}\left\|\nabla L_D(\theta_t)\right\|^2 + \tfrac{1}{2L}\mathbb{E}_{z_t}\|z_t\|^2.
\end{aligned} \tag{12}$$

Then, we can connect $L_D(\theta_t)$ with L_D^*. Note that $L_D(\theta)$ is λ-strongly convex and differentiable, from Csiba and Richtárik (2017), we have:

$$\left\|\nabla L_D(\theta_t)\right\|^2 \geq 2\lambda(L_D(\theta_t) - L_D^*). \tag{13}$$

For random variable X, we have:

$$\mathbb{E}(X^2) = \mathbb{E}^2(X) + v(X), \tag{14}$$

where $v(X)$ denotes variance of X.

By Eq. (13) and Eq. (14), Eq. (12) can be transferred to:

$$\begin{aligned}
&\mathbb{E}_{z_t}[L_D(\theta_{t+1}) - L_D(\theta_t)] \\
&\leq -\tfrac{\lambda}{L}(L_D(\theta_t) - L_D^*) + \tfrac{1}{2L}\left(\mathbb{E}_{z_t}^2\|z_t\| + v(\|z_t\|)\right) \\
&= -\tfrac{\lambda}{L}(L_D(\theta_t) - L_D^*) + \tfrac{p\sigma^2}{2L}.
\end{aligned}$$

In this way, we can fill the gap between $L_D(\theta_T)$ and L_D^*.

By summing over T iterations, we have:

$$\begin{aligned}
&\mathbb{E}[L_D(\theta_T)] - L_D^* \\
&\leq (1 - \tfrac{\lambda}{L})^T(L_D(\theta_0) - L_D^*) \\
&\quad + \tfrac{p\sigma^2}{2L}((1 - \tfrac{\lambda}{L})^0 + (1 - \tfrac{\lambda}{L})^1 + \ldots + (1 - \tfrac{\lambda}{L})^{T-1}) \\
&= (1 - \tfrac{\lambda}{L})^T(L_D(\theta_0) - L_D^*) + \tfrac{p\sigma^2}{2L}\tfrac{L}{\lambda}(1 - (1 - \tfrac{\lambda}{L})^T) \\
&\leq (1 - \tfrac{\lambda}{L})^T(L_D(\theta_0) - L_D^*) + \tfrac{p\sigma^2}{2\lambda}.
\end{aligned} \tag{15}$$

Taking $T = \tilde{O}\left(\log(\frac{n^2\epsilon_1^2}{pG^2\ln(1/\delta_1)})\right)$, we have:

$$\mathbb{E}[L_D(\theta_T)] - L_D^* \leq O\left(\frac{pG^2\ln(1/\delta_1)\log(n)}{n^2\epsilon_1^2}\right).$$

□

Remark 2. In Wang et al. (2017), Eq. (15) is scaling to:

$$\mathbb{E}[L_D(\theta_T)] - L_D^* \leq (1 - \frac{\lambda}{L})^T(L_D(\theta_0) - L_D^*) + \frac{Tp\sigma^2}{2L}. \tag{16}$$

However, in this paper, we summing the geometric sequence (the second term on the right side) to $\frac{p\sigma^2}{2\lambda}$ in Eq. (15). Obviously, Eq. (15) is tighter than Eq. (16). In this way, we improve the proof process, leading a better excess empirical risk bound by a factor of $\log(n)$.

Our method is better than the methods proposed in distributed setting (Jayaraman et al. (2018)) and centralized setting (Wang et al. (2017)) by factors of $\frac{(mn_{(1)})^2\log(n)}{(\log(mn_{(1)})n)^2}$ and $\log(n)$, respectively. Intuitively, giving weights to parties means data instances owned by all parties are of the same importance,

[2] Here, $|c(o; M, D, D')|$ can be replaced by $c(o; M, D, D')$ because $c(o; M, D, D') = -c(o; M, D', D)$.

[3] Considering $\mathbb{E}(z_w) = 0$ and $\sigma_w^2 = O(1/n^2)$, we omit z_w here, the approximation error is $O(1/n^2)$ at most.

which is similar to the centralized setting. Conversely, simply averaging "over-considers" data instances in smaller data sets, making it more *distributed*. Hence, the theoretical results of our method benefit a lot from *weighted framework*.

4.2. Non-convex

In this part, we generalize Theorem 3 to a more general case: the objective function $L(\cdot)$ is not restricted to *convex* but satisfies the Polyak-Łojasiewicz condition.

Definition 3. For function $L(\cdot)$, denotes $L^* = \min_{\theta\in\mathbb{R}^p} L(\theta)$, if there exists $\mu > 0$ and for every θ,

$$\|\nabla L(\theta)\|^2 \geq 2\mu(L(\theta) - L^*), \tag{17}$$

then function $L(\cdot)$ satisfies the Polyak-Łojasiewicz condition.

Obviously, convex functions also satisfy Eq. (17). In fact, *Polyak-Łojasiewicz condition* is much more general than *convex*. Karimi et al. (2016) claimed that when function $F(\cdot)$ is differentiable and L-smooth under ℓ_2 norm, we have:

Strong Convex ⇒ Essential Strong Convexity ⇒ Weak Strongly Convexity ⇒ Restricted Secant Inequality ⇒ Polyak-Łojasiewicz Inequality ⇔ Error Bound

Theorem 4. *Suppose that $\ell(\theta, x, y)$ is G-Lipschitz and L-smooth over θ, $L_D(\theta)$ satisfies the Polyak-Łojasiewicz condition and differentiable, σ is the same as Eq. (4) and $\eta = \frac{1}{L}$. We have:*

$$\mathbb{E}[L_D(\theta_T)] - L_D^* \leq O\left(\frac{pG^2\ln(1/\delta_1)\log(n)}{n^2\epsilon_1^2}\right),$$

where $T = \tilde{O}\left(\log(\frac{n^2\epsilon_1^2}{pG^2\ln(1/\delta_1)})\right)$, $L_D^* = \min_{\theta\in\mathbb{R}^p} L_D(\theta)$ and p is the number of parameters.

Proof of Theorem 4.. The proof is similar to Theorem 3.

According to updating criteria of gradient descent:

$$\theta_{t+1} - \theta_t = -\eta(\nabla L_D(\theta_t) + z_t) = -\frac{1}{L}(\nabla L_D(\theta_t) + z_t).$$

Function $\ell(\cdot)$ is L-smooth (denoted as L below) and $L_D(\theta)$ is differentiable (denoted as d below), we have:

$$\begin{aligned}
&\mathbb{E}_{z_t}[L_D(\theta_{t+1}) - L_D(\theta_t)] \\
&\overset{(L,d)}{\leq} \mathbb{E}_{z_t}\left[< \nabla L_D(\theta_t), \theta_{t+1} - \theta_t > + \tfrac{L}{2}\|\theta_{t+1} - \theta_t\|^2\right] \\
&\leq -\tfrac{1}{L}\|\nabla L_D(\theta_t)\|^2 - \tfrac{1}{L} < \nabla L_D(\theta_t), z_t > \\
&\quad + \tfrac{1}{2L}\|\nabla L_D(\theta_t)\|^2 + \tfrac{1}{2L}\mathbb{E}_{z_t}\|z_t\|^2 \\
&\quad + \tfrac{1}{L} < \nabla L_D(\theta_t), z_t > \\
&= -\tfrac{1}{2L}\|\nabla L_D(\theta_t)\|^2 + \tfrac{1}{2L}\mathbb{E}_{z_t}\|z_t\|^2.
\end{aligned} \tag{18}$$

Note that $L_D(\theta)$ satisfies the Polyak-Łojasiewicz condition, then we have:

$$\|\nabla L_D(\theta_t)\|^2 \geq 2\mu(L_D(\theta_t) - L_D^*). \tag{19}$$

For random variable X, we have:

$$\mathbb{E}(X^2) = \mathbb{E}^2(X) + v(X), \tag{20}$$

where $v(X)$ denotes variance of X.

By Eq. (19) and Eq. (20), Eq. (18) can be transferred to:

$$\begin{aligned}
&\mathbb{E}_{z_t}[L_D(\theta_{t+1}) - L_D(\theta_t)] \\
&\leq -\tfrac{\mu}{L}(L_D(\theta_t) - L_D^*) + \tfrac{1}{2L}\left(\mathbb{E}_{z_t}^2\|z_t\| + v(\|z_t\|)\right) \\
&= -\tfrac{\mu}{L}(L_D(\theta_t) - L_D^*) + \tfrac{p\sigma^2}{2L}.
\end{aligned}$$

Then, summing over T iterations, we have:

$$\begin{aligned}
&\mathbb{E}[L_D(\theta_T)] - L_D^* \\
&\leq (1 - \tfrac{\mu}{L})^T(L_D(\theta_0) - L_D^*) \\
&\quad + \tfrac{p\sigma^2}{2L}((1 - \tfrac{\mu}{L})^0 + (1 - \tfrac{\mu}{L})^1 + \ldots + (1 - \tfrac{\mu}{L})^{T-1}) \\
&= (1 - \tfrac{\mu}{L})^T(L_D(\theta_0) - L_D^*) + \tfrac{p\sigma^2}{2L}\tfrac{L}{\mu}(1 - (1 - \tfrac{\mu}{L})^T) \\
&\leq (1 - \tfrac{\mu}{L})^T(L_D(\theta_0) - L_D^*) + \tfrac{p\sigma^2}{2\mu}.
\end{aligned}$$

Taking $T = \tilde{O}\left(\log(\frac{n^2\epsilon_1^2}{pG^2\ln(1/\delta_1)})\right)$, we have:

$$\mathbb{E}[L_D(\theta_T)] - L_D^* \leq O\left(\frac{pG^2\ln(1/\delta_1)\log(n)}{n^2\epsilon_1^2}\right).$$

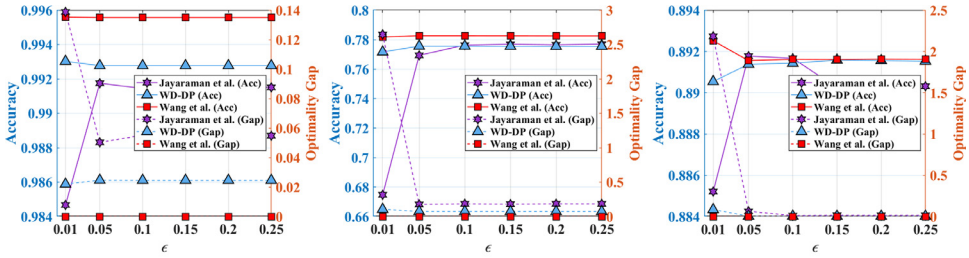
□

The excess empirical risk bounds of our method over both convex case and non-convex case are tighter than the method over convex function proposed by Jayaraman et al. (2018) by a factor of $\frac{(mn_{(1)})^2\log(n)}{(\log(mn_{(1)})n)^2}$, where m is the number of parties and $n_{(1)}$ denotes the smallest data set's size. Under the situation of uneven data scales in real scenarios, the gap between $mn_{(1)}$ and n is huge, our method is extremely superior.

Moreover, as introduced in Remark 2, we prove that the excess empirical risk bound can be tighter than which in Wang et al. (2017) by a factor of $\log(n)$.

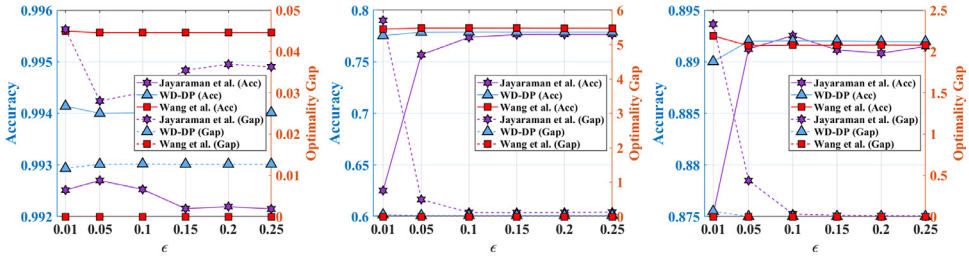
5. Experiments

Experiments are performed on classification task and regression task. We compare our method with the gradient perturbation method proposed by Jayaraman et al. (2018) and the centralized privacy method proposed by Wang et al. (2017). For classification task, logistic regression method is applied on data sets KDDCup99 Hettich and Bay (1999), Adult Dua and Graff (2017), Bank Moro et al. (2014), Breast Cancer Mangasarian and Wolberg (1990) and Credit Card Fraud Bontempi and Worldline (2018), the number of total data instances are 70000, 45222, 41188, 699 and 984, respectively, in which accuracy and optimality gap are used to measure the classification performance. And for regression task, we apply ridge regression method on data sets KDDCup98 Parsa and Howes (1998), Big Mart Sales (BMS) and Black Friday (BF) (data sets BMS and BF are got from Kaggle), the number of total data instances are 70000, 8523 and 100000, respectively, in which Mean Squared Error (MSE) and optimality gap are used to measure the regression performance. Accuracy is defined as $\frac{N_{right}}{m}$, where N_{right} is the number of correct classified samples when testing and m denotes the size of the test set. Optimality gap is defined as $L(\theta) - L(\theta^*)$, where θ^* is centralized optimal non-privacy model. MSE is defined as $\frac{1}{m}\sum_{i=1}^{m}(\hat{y} - y)^2$, where $\hat{y}$ and y



(a) KDDCup99 (b) Adult (c) Bank

Fig. 1 – Accuracy and Optimality Gap on data sets over ϵ, $m = 32$, for classification.



(a) KDDCup99 (b) Adult (c) Bank

Fig. 2 – Accuracy and Optimality Gap on data sets over ϵ, $m = 16$, for classification.

denote the prediction and the true data, respectively. Accuracy and MSE represent the performance on test data and optimality gap denotes excess empirical risk on training data. Higher accuracy and lower MSE mean better classification and regression performance, respectively. And lower optimality gap means that the model is closer to the optimal model.

Training set and testing set are chosen randomly. In all the experiments, total local iteration rounds T and learning rate η are chosen by cross-validation.

5.1. *Convex condition*

In this part, we demonstrate the experimental results when the loss function is convex.

For logistic regression method (for classification task), the loss function is:

$$\ell(\theta, x, y) = -(y \log(h_\theta(x)) + (1-y)\log(1-h_\theta(x))),$$

where $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$.

For ridge regression method (for regression task), the loss function is:

$$\ell(\theta, x, y) = (y - h_\theta(x))^2,$$

where $h_\theta(x) = \theta^T x$.

The loss functions for both tasks are convex.

First, we evaluate the comparisons between our method and previous methods on classification task. We set the number of clients $m = 32, 16, 8, 4, 2$. According to the size of data sets, $m = 32, 16$ are not set for data sets Breast Cancer and Credit Card Fraud. Fig. 1, 2, 3, 4, Fig. 5 show the classification performance over differential privacy budget ϵ, in which dashed lines represent the optimality gap and solid lines represent the accuracy. The numbers of data instances owned by clients are not the same and are set randomly. Moreover, we set a threshold of the smallest data set's size, ensuring effective models are trained by clients. ϵ is set from 0.01 to 0.25 and δ is set according to the size of the data set. In our method, we set $\epsilon_1 + \epsilon_2 = \epsilon$ and $\delta_1 + \delta_2 = \delta$.

It can be observed that on most data sets, the accuracy and the optimality gap of our method: WD-DP are better than the method proposed in Jayaraman et al. (2018), the best distributed method we know, and are similar to the centralized method proposed by Wang et al. (2017), by considering different weights of different clients. In general, the classification performance is becoming better when ϵ increases, which is the same as intuition. However, on some data sets, the accuracy and the optimality gap of our method are worse when ϵ is smaller (such as Fig. 1 (a)). The reason is that the Gaussian noise added to the model has expectation 0 and different variance, higher ϵ means smaller variance and implies smaller amount of noises. When noises are larger, it is more possible for the model to "jump out" from the local optimal solution and to achieve better solutions. Another reason is that in our method, the variance added to the gradient is $c\frac{G^2 T \ln(1/\delta_1)}{n^2\epsilon_1^2}$, term n^2 in the denominator ($n_{(1)}^2$ instead in the method proposed by Jayaraman et al. (2018)) keeps the value of the noise in a low level, avoids large negative perturbation. Meanwhile, it is worth noting that differential privacy is guaranteed by adding random noise, so fluctuations (such as Fig. 3 (c)) are normal.

We also evaluate the influence caused by the difference of data sets' size owned by different clients. We define the level of non-average u as $\frac{n_{\max}}{n_{\min}}$, where $n_{\max}$ and $n_{\min}$ denote the maximum and minimum data set's size, respectively. In the experiments, considering the number of total data instances, u is set from 1 to 9 on data sets KDDCup99, Adult, Bank, while from 1

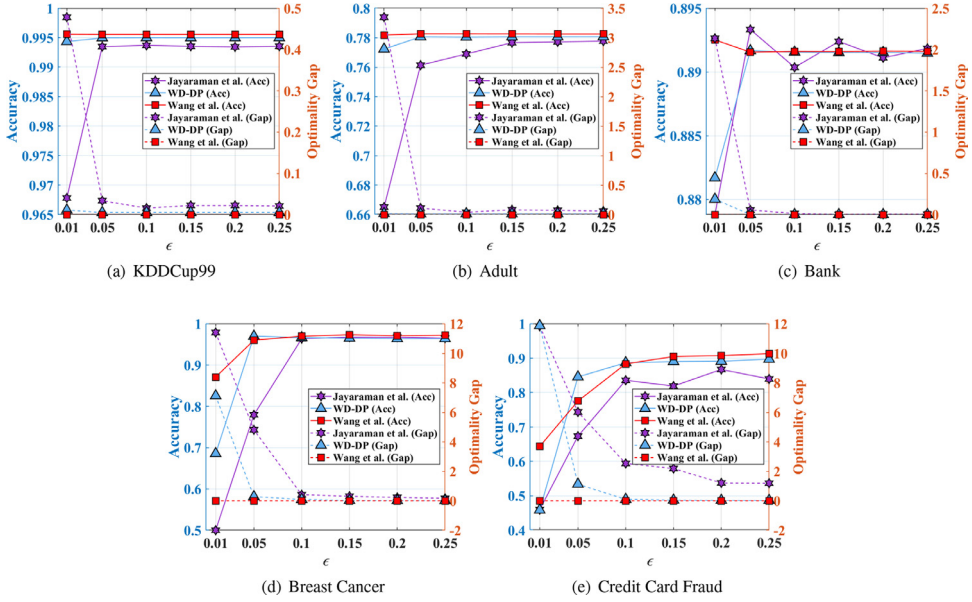


Fig. 3 – Accuracy and Optimality Gap on data sets over ϵ, $m = 8$, for classification.

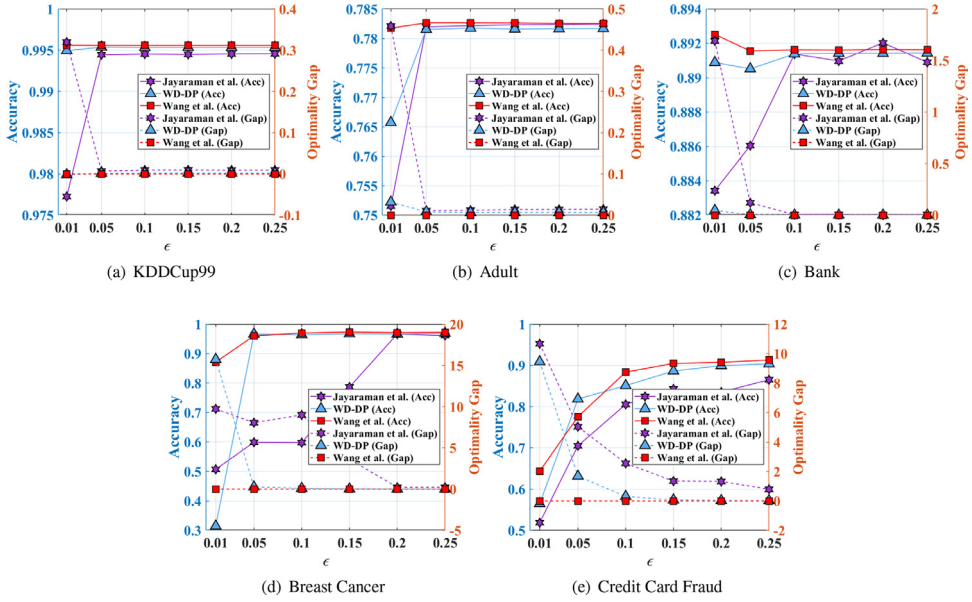


Fig. 4 – Accuracy and Optimality Gap on data sets over ϵ, $m = 4$, for classification.

to 5 on the rest data sets. Particularly, $u = 1$ means average setting. For the sake of simplicity, we divide all the clients into 2 groups, clients in the same group have the same size.

Fig. 6, 7, Fig. 8 show the accuracy and the optimality gap over the level of non-average u, with different privacy budget ϵ and client numbers m, in which dashed lines represent the optimality gap and solid lines represent the accuracy. For average setting, when $u = 1$, the accuracy of our proposed method WD-DP is similar to the method proposed in Jayaraman et al. (2018), which is in line with our theoretical analysis. However, when u increases, which means data scales are more uneven, the accuracy and the optimality gap of our method are steady, but the accuracy and the optimality gap of which proposed by Jayaraman et al. (2018) become worse rapidly or fluctuates sharply. Thus, our method is more reliable, especially in the case of uneven data scales, which is the same as in theoretical analysis. In general, with the increasing of u, classification performance becomes worse. However, on some data sets, larger u leads better accuracy and optimality gap (such as Fig. 6 (c)). The reason is that when u increases from 1, data scales of different clients become more uneven, and the performance becomes worse at first. However, when u is larger than a certain value (the threshold is different according to different data sets), the aggregated model is determined by the clients who own most of the data instances. In other words, training data 'flows to' part of the clients, and those

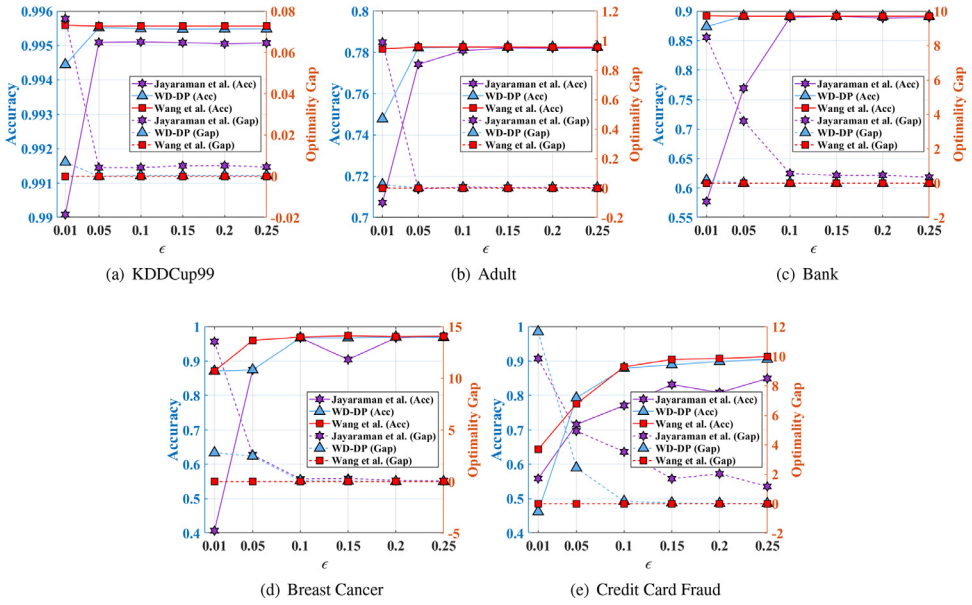


Fig. 5 – Accuracy and Optimality Gap on data sets over ϵ, $m = 2$, for classification.

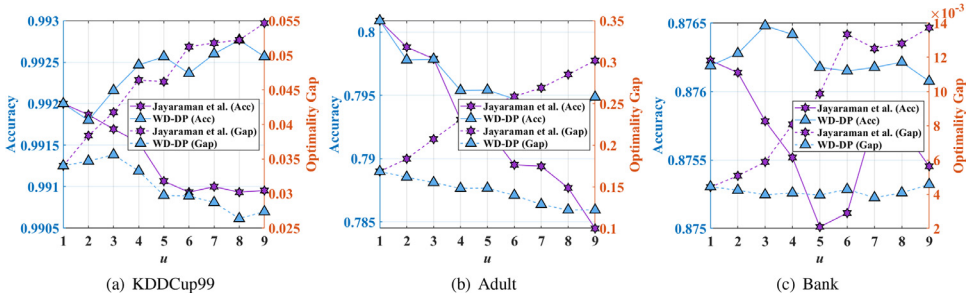


Fig. 6 – Accuracy and Optimality Gap on data sets over the level of non-average u, $m = 32$, $\epsilon = 0.15$, for classification.

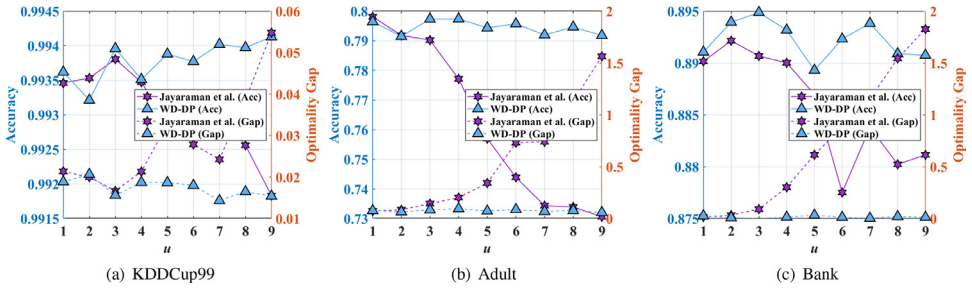
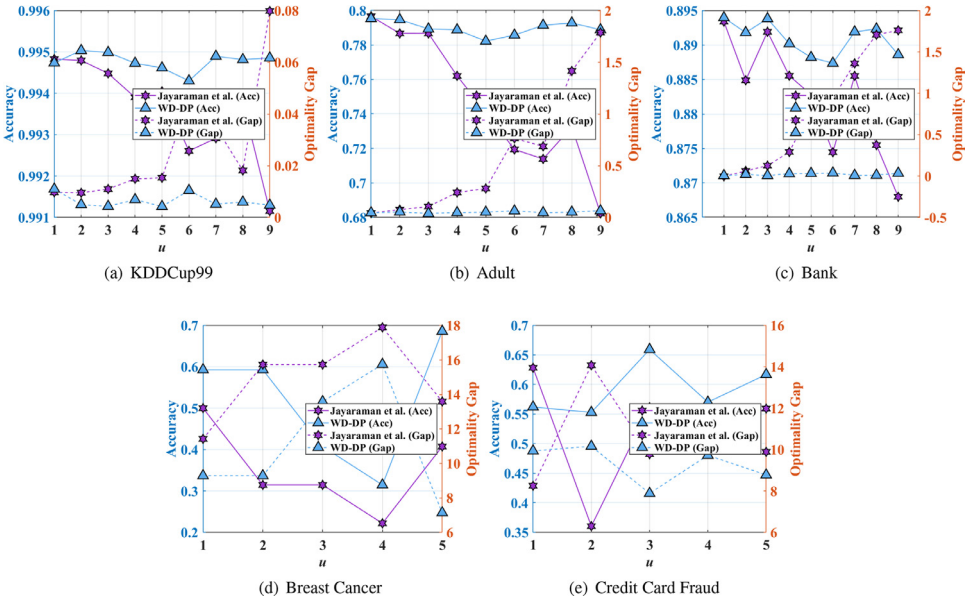


Fig. 7 – Accuracy and Optimality Gap on data sets over the level of non-average u, $m = 16$, $\epsilon = 0.01$, for classification.

clients with more data instances are more 'powerful'. Thus, it is easy to follow that the model trained by more training data is better than which trained by less data.

Then, comparisons between our method: WD-DP and previous methods on regression task are performed. For regression task, we set the number of clients $m = 32, 16, 8$ for all the data sets. Fig. 9, 10, Fig. 11 show the regression performance over differential privacy budget ϵ, in which dashed lines represent the optimality gap and solid lines represent the MSE. Like in classification task, the numbers of data instances owned by clients are set randomly and a threshold of the smallest data set's size is set.

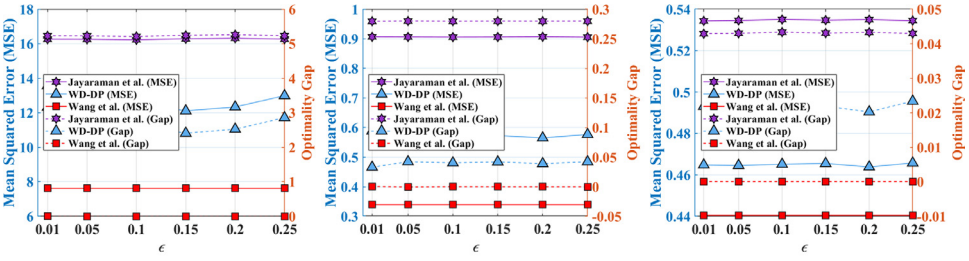
It can be observed that both MSE and optimality gap of our method: WD-DP are better than the method proposed in Jayaraman et al. (2018), although slightly fluctuations exist on some data sets (such as Fig. 10 (a)), which is common, as explained above in classification task. In general, with the increasing of ϵ, the MSE and the optimality gap become smaller. However, on some data sets, the 'common sense' does not hold (such as Fig. 11 (a) when



(a) KDDCup99 (b) Adult (c) Bank

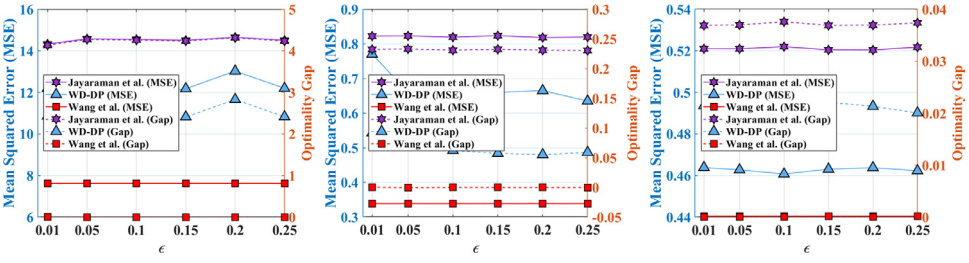
(d) Breast Cancer (e) Credit Card Fraud

Fig. 8 – Accuracy and Optimality Gap on data sets over the level of non-average u, $m = 8$, $\epsilon = 0.01$, for classification.



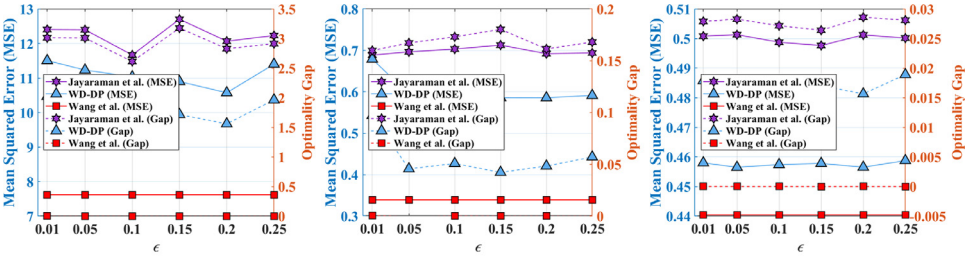
(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 9 – MSE and Optimality Gap on data sets over ϵ, $m = 32$, for regression.



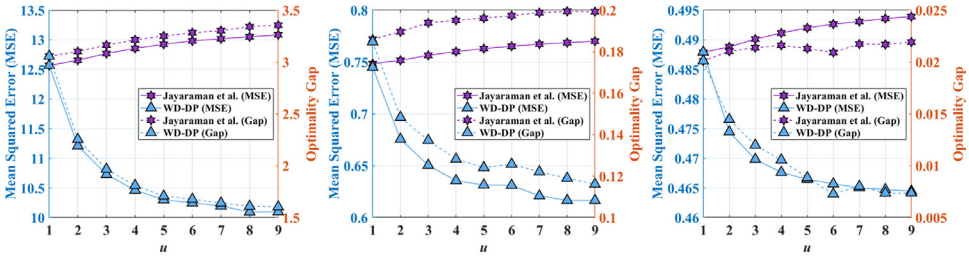
(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 10 – MSE and Optimality Gap on data sets over ϵ, $m = 16$, for regression.



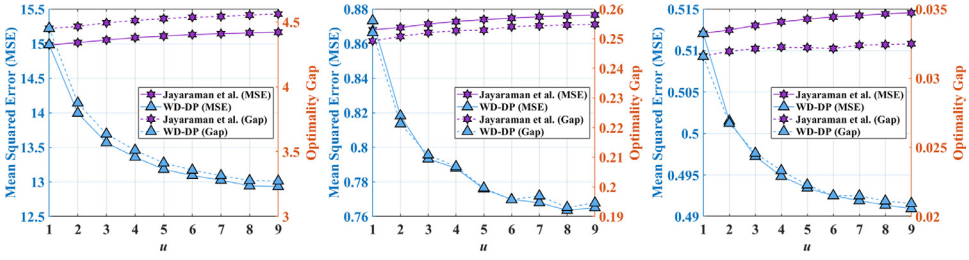
(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 11 – MSE and Optimality Gap on data sets over ϵ, $m = 8$, for regression.



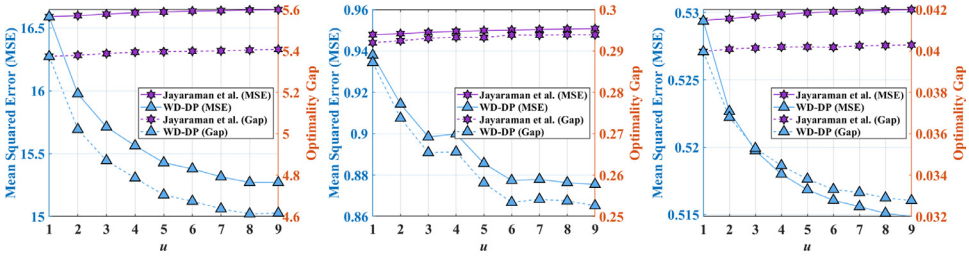
(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 12 – MSE and Optimality Gap on data sets over the level of non-average u, $m = 8$, $\epsilon = 0.15$, for regression.



(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 13 – MSE and Optimality Gap on data sets over the level of non-average u, $m = 16$, $\epsilon = 0.2$, for regression.



(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 14 – MSE and Optimality Gap on data sets over the level of non-average u, $m = 32$, $\epsilon = 0.25$, for regression.

$\epsilon = 0.25$), the reason is similar to which in classification task: higher random noise may make the model 'jump out' from the local optimal solution. Additionally, we find that in regression task, MSE and optimality gap are not so sensitive as in classification task, leading more gentle lines over different ϵ.

Fig. 12, 13, Fig. 14 show the accuracy and the optimality gap over u, in which dashed lines represent the optimality gap and solid lines represent the MSE. When $u = 1$, the MSE and the optimality gap of our method is similar to the method proposed in Jayaraman et al. (2018). As u increases, MSE and optimality gap of our method become better (decrease) and which of previous method become worse (increase), which means that our method is better than the method proposed in Jayaraman et al. (2018). In general, when u increases, the regression performance will decrease. However, as explained above in classification task, with the increasing of u, larger clients own more training data and become more 'powerful'. Training model by larger data sets leads better performance, which is the reason that the regression performance of our method becomes better with the increasing of u.

Practical experiments on convex loss function over real data sets show that our proposed method: WD-DP is more superior than previous methods in the field of distributed differentially private ERM, with satisfactory privacy on the final model and clients, especially in the case that the data scales on clients are uneven. Moreover, the experimental results show that in distributed setting, our method achieves the performance similar to centralized methods, which is an attractive result.

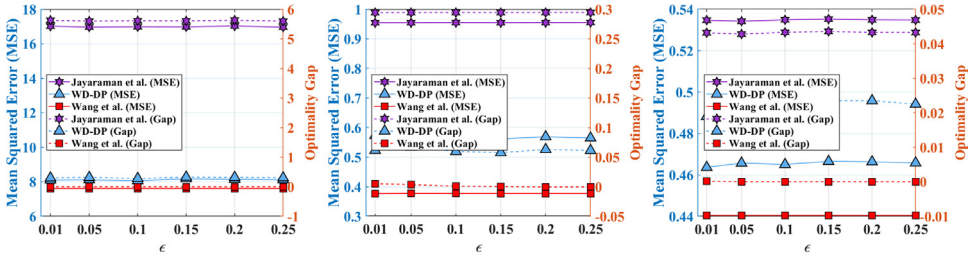
5.2. *Non-Convex condition*

In this part, we perform the experiments on non-convex loss function.

To construct the loss function which satisfies the Polyak-Łojasiewicz condition, we add a regularization term $\|\theta\|^2 + 3\sin^2(\|\theta\|)$ to the loss function. The reason is that function $f(x) = x^2 + 3\sin^2(x)$ is non-convex but satisfies the Polyak-Łojasiewicz inequality (defined in (17)) with $\mu = 1/32$ (Karimi et al. (2016)).

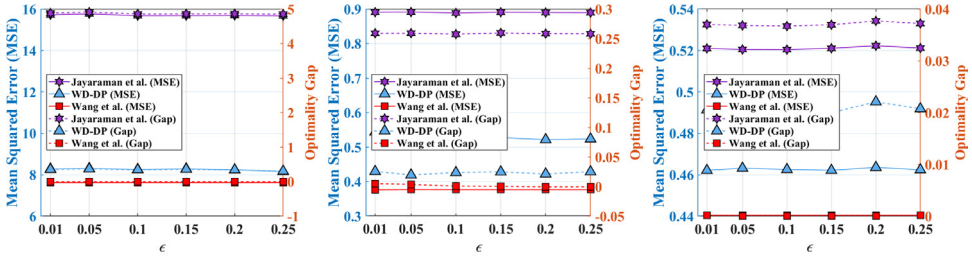
We apply the regularization term to the ridge regression method (introduced in Section 5.1) on regression task. As a result, the loss function is transformed to:

$$\ell(\theta, x, y) = (y - \theta^T x)^2 + \|\theta\|^2 + 3\sin^2(\|\theta\|).$$



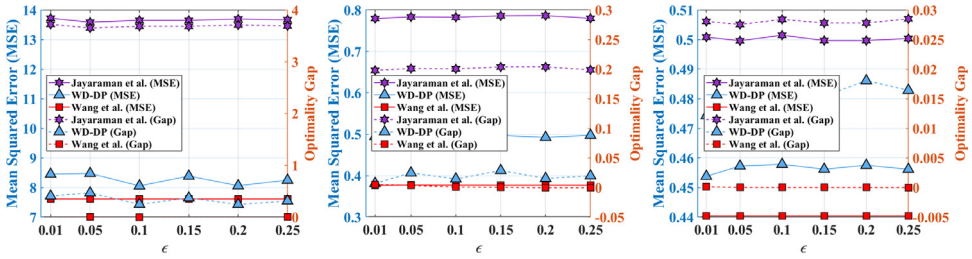
(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 15 – MSE and Optimality Gap on data sets over ϵ, $m = 32$, for regression under Polyak-Łojasiewicz condition.



(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 16 – MSE and Optimality Gap on data sets over ϵ, $m = 16$, for regression under Polyak-Łojasiewicz condition.



(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 17 – MSE and Optimality Gap on data sets over ϵ, $m = 8$, for regression under Polyak-Łojasiewicz condition.

Then the loss function $\ell(\theta, x, y)$ is non-convex but satisfies the Polyak-Łojasiewicz condition.

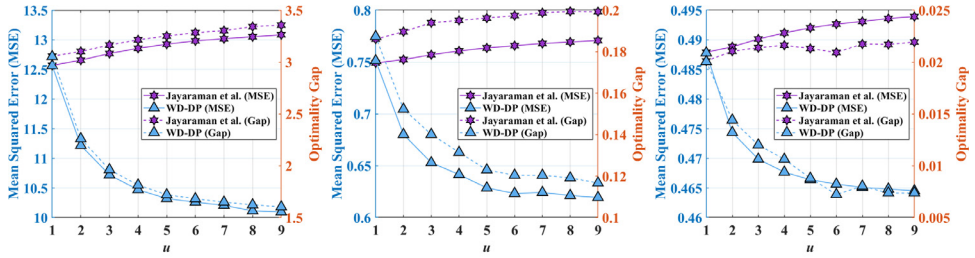
Like in the convex condition, we compare our method to the distributed method proposed by Jayaraman et al. (2018) and the centralized method proposed by Wang et al. (2017). Fig. 15, 16, Fig. 17 show the MSE and the optimality gap over data sets under non-convex condition, the number of clients m are 32, 16 and 8, respectively. In Fig. 15, 16 and 17, dashed lines represent the optimality gap and solid lines represent the MSE.

The experiments show that both MSE and optimality gap of our WD-DP method are better than the distributed method proposed by Jayaraman et al. (2018). The MSE and optimality gap are stable over ϵ because regression task is not sensitive to the privacy budget, as explained in Section 5.1. On some data sets, the performance of our method is similar to the centralized method proposed by Wang et al. (2017). The reason is that by applying 'weighted paradigm', the 'distributed' property of the whole system is reduced a lot, which is detailed explained in the theoretical analysis.

Fig. 18, 19, Fig. 20 show the MSE and the optimality gap over the level of non-average u on different data sets. The number of clients m and the privacy budget ϵ is chosen randomly. The comparison is between our method and the distributed method proposed in Jayaraman et al. (2018). Like in previous figures, dashed lines represent the optimality gap and solid lines represent the MSE.

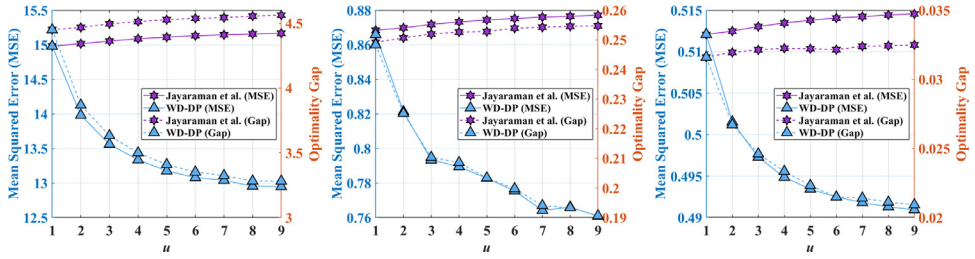
Experimental results show that our method is better than the method proposed by Jayaraman et al. (2018) on both MSE and optimality gap. The difference between our method and previous method is huge. Meanwhile, with the increasing of u, previous method becomes worse and our method becomes better, the gap between previous method and our proposed WD-DP method becomes wider. The reason is similar to which under convex condition. When u increases, data instances 'flow' to larger clients and the model is determined by those clients owning more data.

Moreover, under both convex and non-convex conditions, on some data sets, the performance of our proposed distributed method: WD-DP is similar to the centralized method proposed by Wang et al. (2017). The reason is that 'weighted framework' reduces the negative impacts brought by uneven data scales, which is detailed in the theoretical analysis in Section 3 and Section 4. In WD-DP, by considering weights,



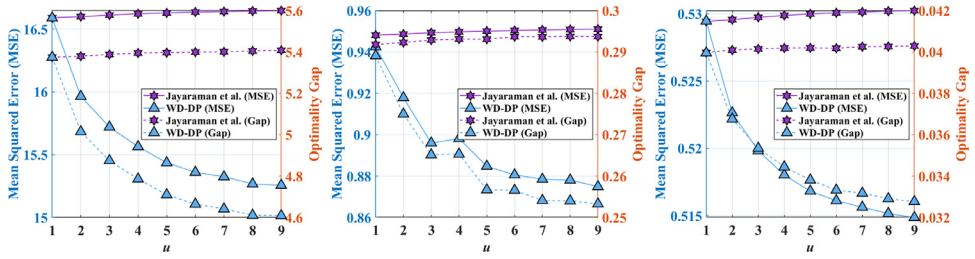
(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 18 – MSE and Optimality Gap on data sets over the level of non-average u, $m = 8$, $\epsilon = 0.15$, for regression under Polyak-Łojasiewicz condition.



(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 19 – MSE and Optimality Gap on data sets over the level of non-average u, $m = 16$, $\epsilon = 0.2$, for regression under Polyak-Łojasiewicz condition.



(a) KDDCup98 (b) Big Mart Sales (c) Black Friday

Fig. 20 – MSE and Optimality Gap on data sets over the level of non-average u, $m = 32$, $\epsilon = 0.25$, for regression under Polyak-Łojasiewicz condition.

clients in the distributed system work as one, making the system more 'centralized'.

6. Conclusions

In this paper, we propose WD-DP, a distributed differential privacy ERM method, providing (ϵ, δ)-differential privacy by gradient perturbation. For the first time, we theoretically analyze the problem: *What can we benefit from the weighted framework?*

Theoretical analysis shows that by considering different weights of different clients, the noise bound and the excess empirical risk bound can be improved in distributed differentially private ERM, comparing with the best previous method. Similar to theoretical results, experimental results on real data sets also show that the performance of our proposed method: WD-DP is better than previous ones, especially under the condition that data scales on different clients are uneven, which is common in real scenarios. It is worth emphasizing that although our method focuses on the distributed setting, it achieves almost the same theoretical and practical results as previous centralized methods, which is an attractive result brought by weighted paradigm.

Moreover, most previous work on differentially private ERM assumes that the loss function is strongly convex and this constraint is not easy to achieve in some situations. So, we generalize our method to a more general condition, in which the loss function satisfies the Polyak-Łojasiewicz condition. For this non-convex case, we also perform experiments to evaluate our method: WD-DP, experimental results are similar to which when the loss function is convex. Thus, both theoretical and experimental results show that our method is adapt to both convex and non-convex conditions. In future work, we will focus on non-convex optimization in more general situ-

ations under distributed setting (e.g. deep learning). How to reduce time complexity of the model is also a problem we focus on, since most models are synchronous in distributed setting.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Yilin Kang: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Yong Liu:** Conceptualization, Methodology, Formal analysis, Funding acquisition, Supervision, Writing - review & editing. **Ben Niu:** Resources, Writing - review & editing. **Weiping Wang:** Resources, Writing - review & editing.

Acknowledgement

This work was supported by Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098; the Open Research Project of the State Key Laboratory of Media Convergence and Communication, Communication University of China, China (No. SKLMCC2020KF004); the National Natural Science Foundation of China [grant numbers 61703396, 62076234]; the Youth Innovation Promotion Association CAS.

REFERENCES

Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016. p. 308–18.

Arora R, Upadhyay J. On Differentially Private Graph Sparsification and Applications. In: Advances in Neural Information Processing Systems 32; 2019. p. 13378–89.

Backes M, Berrang P, Humbert M, Manoharan P. Membership privacy in microrna-based studies. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016. p. 319–30.

Bassily R, Smith A, Thakurta A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science; 2014. p. 464–73.

Bernstein G, Sheldon DR. Differentially private bayesian linear regression. In: Advances in Neural Information Processing Systems; 2019. p. 523–33.

Bontempi, G., Worldline, 2018. ULB the machine learning group.

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning 2011:1–122.

Bun M, Steinke T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Theory of Cryptography Conference; 2016. p. 635–58.

Carlini N, Liu C, Erlingsson U, Kos J, Song D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In: Proceedings of the 28th USENIX Conference on Security Symposium; 2019. p. 267–84.

Chaudhuri K, Monteleoni C. Privacy-preserving logistic regression. In: Advances in neural information processing systems; 2009. p. 289–96.

Chaudhuri K, Monteleoni C, Sarwate AD. Differentially private empirical risk minimization. Journal of Machine Learning Research 2011:1069–109.

Chaudhuri K, Sarwate AD, Sinha K. A near-optimal algorithm for differentially-private principal components. The Journal of Machine Learning Research 2013:2905–43.

Chen K, Ding H, Huo Q. Parallelizing adam optimizer with blockwise model-update filtering. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2020. p. 3027–31.

Chen K, Huo Q. Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2016. p. 5880–4.

Choy CB. Asynchronous distributed neural network training using alternating direction method of multipliers; 2015.

Csiba D, Richtárik P. Global convergence of arbitrary-block gradient methods for generalized polyak-lojasiewicz functions. arXiv preprint arXiv:1709.03014 2017.

Ding J, Errapotu SM, Zhang H, Gong Y, Pan M, Han Z. Stochastic admm based distributed machine learning with differential privacy. In: International Conference on Security and Privacy in Communication Systems; 2019. p. 257–77.

Dua, D., Graff, C., 2017. UCI machine learning repository.

Dwork C. Differential privacy. Encyclopedia of Cryptography and Security 2011:338–40.

Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference; 2006. p. 265–84.

Dwork C, Roth A, et al. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science 2014:211–407.

Dwork C, Rothblum GN, Vadhan S. Boosting and differential privacy. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science; 2010. p. 51–60.

Elgabli A, Park J, Ahmed S, Bennis M. L-Fgadmm: layer-wise federated group admm for communication efficient decentralized deep learning. arXiv preprint arXiv:1911.03654 2019.

Farquhar S, Gal Y. Differentially private continual learning. arXiv preprint arXiv:1902.06497 2019.

Fredrikson M, Lantz E, Jha S, Lin S, Page D, Ristenpart T. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In: 23rd {USENIX} Security Symposium ({USENIX} Security 14); 2014. p. 17–32.

Fu J, Huang Y, Xu J, Wu H. Optimization of distributed convolutional neural network for image labeling on asynchronous gpu model. International Journal of Innovative Computing, Information and Control 2019:1145–56.

Ge J, Wang Z, Wang M, Liu H. Minimax-optimal privacy-preserving sparse pca in distributed systems. In: International Conference on Artificial Intelligence and Statistics; 2018. p. 1589–98.

Geyer RC, Klein T, Nabi M. Differentially private federated learning: a client level perspective. arXiv preprint arXiv:1712.07557 2017.

He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1026–34.

Hettich, S., Bay, S. D., 1999. The uci kdd archive.
Huang Y, Tian J, Han L, Wang G, Song X, Su D, Yu D. A random gossip bmuf process for neural language modeling. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2020. p. 7959–63.
Huang Z, Hu R, Guo Y, Chan-Tin E, Gong Y. Dp-admm: admm-based distributed learning with differential privacy. IEEE Trans. Inf. Forensics Secur. 2020:1002–12.
Jayaraman B, Evans D. Evaluating differentially private machine learning in practice. In: Proceedings of the 28th USENIX Conference on Security Symposium; 2019. p. 1895–912.
Jayaraman B, Wang L, Evans D, Gu Q. Distributed learning without distress: Privacy-preserving empirical risk minimization. In: Advances in Neural Information Processing Systems; 2018. p. 6343–54.
Karimi H, Nutini J, Schmidt M. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases; 2016. p. 795–811.
Mangasarian OL, Wolberg WH. In: Technical Report. Cancer diagnosis via linear programming. University of Wisconsin-Madison Department of Computer Sciences; 1990.
McMahan HB, Moore E, Ramage D, Hampson S, et al. Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629 2016.
McMahan HB, Ramage D, Talwar K, Zhang L. Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963 2017.
Moro S, Cortez P, Rita P. A data-driven approach to predict the success of bank telemarketing. Decis. Support Syst. 2014:22–31.
Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: International Conference on the Theory and Applications of Cryptographic Techniques; 1999. p. 223–38.
Parsa, I., Howes, K., 1998. The uci kdd archive.
Pathak M, Rane S, Raj B. Multiparty differential privacy via aggregation of locally trained classifiers. In: Advances in Neural Information Processing Systems; 2010. p. 1876–84.
Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security; 2015. p. 1310–21.
Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP); 2017. p. 3–18.
Smith MT, Álvarez MA, Zwiessele M, Lawrence ND. Differentially private regression with gaussian processes. In: International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9–11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain; 2018. p. 1195–203.
Tian L, Jayaraman B, Gu Q, Evans D. In: NIPS Workshop on Private Multi-Party Machine Learning. Aggregating private sparse learning models using multi-party computation; 2016.
Triastcyn A, Faltings B. Bayesian differential privacy for machine learning.. arXiv: Learning 2019.
Ullman J, Sealfon A. Efficiently Estimating Erdos-renyi Graphs with Node Differential Privacy. In: Advances in Neural Information Processing Systems 32; 2019. p. 3765–75.
Wang D, Xu J. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2019. p. 1182–9.
Wang D, Xu J. Principal component analysis in the local differential privacy model. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI); 2019. p. 4795–801.
Wang D, Ye M, Xu J. Differentially private empirical risk minimization revisited: Faster and more general. In: Advances in Neural Information Processing Systems; 2017. p. 2722–31.
Wang X, Ishii H, Du L, Cheng P, Chen J. Differential privacy-preserving distributed machine learning. In: 2019 IEEE 58th Conference on Decision and Control (CDC); 2019. p. 7339–44.
Wang Y, Zhou W, Zhang Q, Li H. Convolutional neural networks with generalized attentional pooling for action recognition. In: 2018 IEEE Visual Communications and Image Processing (VCIP); 2018. p. 1–4.
Wu B, Zhao S, Chen C, Xu H, Wang L, Zhang X, Sun G, Zhou J. Generalization in Generative Adversarial Networks: A Novel Perspective from Privacy Protection. In: Advances in Neural Information Processing Systems 32; 2019. p. 306–16.
Xu C, Ren J, Zhang D, Zhang Y, Qin Z, Ren K. Ganobfuscator: mitigating information leakage under gan via differential privacy. IEEE Trans. Inf. Forensics Secur. 2019:2358–71.
Xu J, Ni B, Yang X. Video prediction via selective sampling. In: Advances in Neural Information Processing Systems; 2018. p. 1705–15.
Yu Z, Shi X, Yan L, Li W. Distributed stochastic admm for matrix factorization; 2014. p. 1259–68.
Zhang H, Wheldon C, Dunn AG, Tao C, Huo J, Zhang R, Prosperi M, Guo Y, Bian J. Mining twitter to assess the determinants of health behavior towards human papillomavirus vaccination in the united states. arXiv preprint arXiv:1907.11624 2019.
Zhang J, Zhang Z, Xiao X, Yang Y, Winslett M. Functional mechanism: regression analysis under differential privacy. Proceedings of the VLDB Endowment 2012:1364–75.
Zhang J, Zheng K, Mou W, Wang L. Efficient private erm for smooth objectives. arXiv preprint arXiv:1703.09947 2017.
Zhang R, Kwok J. Asynchronous distributed admm for consensus optimization. In: International conference on machine learning; 2014. p. 1701–9.
Zhao L, Ni L, Hu S, Chen Y, Zhou P, Xiao F, Wu L. Inprivate digging: Enabling tree-based distributed data mining with differential privacy. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications; 2018. p. 2087–95.

Yilin Kang was born in 1996. He is currently an Ph.D. at Institute of Information Engineering, Chinese Academy of Sciences. His main research interests include differential privacy, machine learning.

Yong Liu received his Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2016. He is currently an assistant professor at Institute of Information Engineering, Chinese Academy of Sciences. His research interests include differential privacy, large-scale kernel methods, machine learning. He has published more than 20 papers in top journals and conferences by first author or corresponding author.

Ben Niu received his Ph.D. degrees in the school of Telecommunications Engineering from Xidian University, China, in 2014. He is currently an associate professor in Institute of Information Engineering, Chinese Academy of Sciences. From 2011, he is with Department of CSE, the Pennsylvania State University as a visiting Ph.D. student for two years. His research interests include network security and applied cryptography, with focus on the security and privacy in mobile social networks.

Weiping Wang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2006. He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His research interests include big data, data security, database, and storage systems. He has more than 70 publications in major journals and international conferences.