# Data Heterogeneity Differential Privacy: From Theory to Algorithm [⋆]

Yilin Kang[1,3], Jian Li[1,⋆⋆], Yong Liu[2], and Weiping Wang[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences
{kangyilin,lijian9026,wangweiping}@iie.ac.cn
[2] Gaoling School of Artificial Intellignece, Renmin University of China
liuyonggsai@ruc.edu.cn
[3] School of Cyber Security, University of Chinese Academy of Sciences

**Abstract.** Traditionally, the random noise is equally injected when training with different data instances in the field of differential privacy (DP). In this paper, we first give sharper excess risk bounds of DP stochastic gradient descent (SGD) method. Considering most of the previous methods are under convex conditions, we use Polyak-Łojasiewicz condition to relax it in this paper. Then, after observing that different training data instances affect the machine learning model to different extent, we consider the heterogeneity of training data and attempt to improve the performance of DP-SGD from a new perspective. Specifically, by introducing the influence function (IF), we quantitatively measure the contributions of various training data on the final machine learning model. If the contribution made by a single data instance is so little that attackers cannot infer anything from the model, we do not add noise when training with it. Based on this observation, we design a 'Performance Improving' DP-SGD algorithm: PIDP-SGD. Theoretical and experimental results show that our proposed PIDP-SGD improves the performance significantly.

**Keywords:** Differential privacy · Machine learning · Data heterogeneity.

## 1 Introduction

Machine learning has been widely applied to many fields in recent decades and tremendous data has been collected. As a result, information disclosure becomes a huge problem. Except for the original data, model parameters can reveal sensitive information in an undirect way as well [15, 32].

Differential privacy (DP) [12, 13] is a theoretically rigorous tool to prevent sensitive information [10]. It preserves privacy by introducing random noise, to block adversaries from inferring any single individual included in the dataset by observing the machine learning model. As such, DP has been applied to numerous machine learning methods [31, 47, 6, 39, 17, 43, 35, 2, 9, 21, 33, 5, 40, 44, 30, 41, 36] and three main approaches are studied: output perturbation, objective

[⋆] Supplementary: All the proofs are given in https://arxiv.org/pdf/2002.08578.pdf

[⋆⋆] Corresponding author.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_9

perturbation, and gradient perturbation. However, some problems still exist: First, all data is usually treated equally when training DP model, but in real scenarios, different training data affects the model differently, so treating them all the same lacks 'common sense' and is one of the reasons why low accuracy appears. Meanwhile, previous results always require that the loss function is convex (or even strongly convex), the application scenario is narrow.

To solve the problems, we make the following contributions in this paper: First, we introduce the Polyak-Łojasiewicz (PL) condition [20] to relax the convex (or even strongly convex) assumption. We analyze the excess population risk and give corresponding bounds under PL condition and theoretical results show that our given excess risk bounds are better than previous convex ones. Second, motivated by the definition of DP, we provide a new perspective to improve the performance: treating different data instances differently. In particular, we introduce the Influence Function (IF) [22] to measure the contributions made by different data instances. If the data instance $z$ contributes so little to the machine learning model that the attacker cannot infer anything (represented by the privacy budget $\epsilon$), we do not add noise when training with $z$, rather than treating all of the data instances as being the same. In this way, we propose a 'Performance Improving' algorithm: PIDP-SGD to improve the model performance, by taking data heterogeneity into account.

The rest of the paper is organized as follows. We introduce some related work in Section 2. Preliminaries are presented in Section 3. We analyze the excess risk of DP-SGD and give sharper theoretical bounds in Section 4. The 'Performance Improving' algorithm is given and analyzed in Section 5. In Section 6, we compare our proposed method with previous methods in detail. The experimental results are shown in Section 7 and we conclude the paper in Section 8.

## 2 Related Work

The first method on DP machine learning is proposed in [9], in which output and objective perturbation methods are introduced. Gradient perturbation is proposed in [33] and DP-SGD is analyzed for the first time. The accuracy of the objective perturbation method is improved by [21]. The excess empirical risk bounds of the methods proposed in [9] and [21] are improved by [5]. An output perturbation method is introduced to DP-SGD by [41], in which a novel $\ell_2$ sensitivity is analyzed and better accuracy is achieved. [40] introduces Prox-SVRG [42] to DP and proposes DP-SVRG, in which optimal or near-optimal utility bounds are achieved. Meanwhile, there are also some works concentrated on non-convex analysis. DP is introduced to deep learning by [1], via gradient perturbation method, however, it focuses on the privacy but lacks utility analysis. An output perturbation method is proposed in [44] under non-convex condition. The Polyak-Łojasiewicz condition is introduced in [40] and the excess empirical risk of gradient perturbation method under non-convex condition is analyzed, however, the excess population risk is not discussed. Aiming to achieve better performance, in [30], more noise is added to those features less 'relevant' to the final model. A Laplace smooth operator is introduced to DP-SGD and a new method: DP-LSSGD is proposed in [36], focusing on non-convex analysis. The excess empirical risk bound and the excess population risk bound of DP model under non-convex condition are analyzed by [38], via Gradient Langevin Dynamics. For non-convex condition, the theoretical results are always unsatisfactory.

All the works mentioned above treat all data instances equally, lack 'common sense' and lead unsatisfactory utility. To solve the problems under both convex and non-convex conditions, we take data heterogeneity into account and propose a 'Performance Improving' algorithm. In this way, our method improves the performance of DP model, superior to previous methods in excess risk bounds.

## 3 Preliminaries

### 3.1 Notations and Assumptions

The loss function is $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$, where $\mathcal{C}$ is the parameter space and $\mathcal{D}$ is the data universe. We assume that the parameter space is bounded, whose radius is $r$. Supposing there are $n$ data instances in the dataset $D = \{z_1, \cdots, z_n\} \in \mathcal{D}^n$, where $z_i$ are drawn i.i.d from the underlying distribution $\mathcal{P}$. Besides, for each $z = (x, y)$, $x$ is the feature and $y$ is the label. We assume $\|x\|_2 \leq 1$, i.e. $\mathcal{X}$ is the unit ball. Moreover, for a vector $x = [x_1, \cdots, x_d]$, its $\ell_2$ norm is defined as: $\|x\|_2 = \big(\sum_{i=1}^d x_i^2\big)^{1/2}$, and the $i^{th}$ element is represented by $[x]_i$.

The population risk over the underlying distribution $\mathcal{P}$ is defined as $L_{\mathcal{P}}(\theta) = \mathbb{E}_{z\sim\mathcal{P}}\big[\ell\big(\theta, z\big)\big]$. However, we cannot achieve $\mathcal{P}$ in practice, so our goal is to find the optimal model that minimizes the empirical risk $L(\theta; D) = \frac{1}{n}\sum_{i=1}^n \ell\big(\theta, z_i\big)$ on dataset $D$, defined as: $\theta^* = \arg\min\big[L\big(\theta; D\big)\big]$, and $L(\theta^*; D)$ is represented by $L^*$. For an algorithm $\mathcal{A} : \mathcal{D}^n \to \mathbb{R}^m$, we denote its output as $\theta_{\mathcal{A}}$. The **excess empirical risk** denotes the gap between $\theta_{\mathcal{A}}$ and $\theta^*$, defined as: $L(\theta_{\mathcal{A}}; D) - L^*$; and the **excess population risk** represents the gap between $\theta_{\mathcal{A}}$ and the optimal model over the underlying $\mathcal{P}$, defined as: $L_{\mathcal{P}}(\theta_{\mathcal{A}}) - \min_{\theta\in\mathcal{C}} L_{\mathcal{P}}(\theta)$. The **generalization error** connects the population risk and the empirical risk, defined as: $L_{\mathcal{P}}(\theta_{\mathcal{A}}) - L(\theta_{\mathcal{A}}; D)$.

Besides, there are some assumptions on the loss function:

**Definition 1 ($G$-Lipschitz)** *Loss $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ is $G$-Lipschitz over $\theta$, if for some constant $G$, any $z \in \mathcal{D}$ and $\theta, \theta' \in \mathcal{C}$, $|\ell(\theta, z) - \ell(\theta', z)| \leq G\|\theta - \theta'\|_2$.*

**Definition 2 ($L$-smooth)** *Loss $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ is $L$-smooth over $\theta$, if for some constant $L$, any $z \in \mathcal{D}$ and $\theta, \theta' \in \mathcal{C}$, $\|\nabla_\theta \ell(\theta, z) - \nabla_\theta \ell(\theta', z)\|_2 \leq L\|\theta - \theta'\|_2$.*

Definitions 1 and 2 upper bound the gradient and the second order gradient, respectively, i.e. $\|\nabla_\theta \ell(\theta, z)\|_2 \leq G$ and $\|\nabla_\theta^2 \ell(\theta, z)\|_2 \leq L$.

### 3.2 Differential Privacy

Two databases $D, D' \in \mathcal{D}^n$ differing by one single element are denoted as $D \sim D'$, called *adjacent databases*.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_9

**Definition 3 (Differential Privacy [13])** *With $S \in range(\mathcal{A})$, the randomized function $\mathcal{A} : \mathcal{D}^n \to \mathbb{R}^m$ is ($\epsilon$,$\delta$)-differential privacy (($\epsilon$,$\delta$)-DP) if:*

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

Differential privacy requires that adjacent datasets $D, D'$ lead to similar distributions on the output of a randomized algorithm $\mathcal{A}$. This implies that an adversary cannot infer whether an individual participates in the training process because essentially the same conclusions about an individual will be drawn whether or not that individual's data was used. Some kind of attacks, such as membership inference attack, attribute inference attack, memorization attack, can be thwarted by differential privacy [3, 7, 19].

## 4 Sharper Utility Bounds for DP-SGD

Considering that SGD naturally fits the condition measuring each data instances differently, before introducing the 'Performance Improving' algorithm in detail, we first analyze the excess risk bounds of DP-SGD.

In DP-SGD, at the $t^{th}$ iteration, we have: $\theta_{t+1} \leftarrow \theta_t - \alpha\left(\nabla_\theta \ell(\theta_t, z_t) + b\right)$, where $z_t$ is the chosen data instance at iteration $t$, $\alpha$ is the learning rate, and $b$ is the sampled random noise. There is a long list of works to analyze the privacy guarantees of DP-SGD. To the best of our knowledge, the moments accountant method proposed by [1] achieves one of the best results. It claimed that if the Gaussian random noise $b \sim \mathcal{N}\left(0, \sigma^2 I_m\right)$ is injected, the loss function is $G$-Lipschitz, and with

$$\sigma \geq c\frac{G\sqrt{T\log(1/\delta)}}{n\epsilon} \tag{1}$$

for some constant $c$, then the algorithm satisfies $(\epsilon, \delta)$-DP, where $T$ is the total number of training iterations and $n$ is the size of the training dataset.

Previous works always discuss the empirical risk but seldom consider the population risk [33, 1, 40, 44, 41]. However, the latter is one of the most concerned terms in machine learning because it demonstrates the gap between the private model and the optimal model over the underlying distribution $\mathcal{P}$.

**Excess Empirical Risk** The excess empirical risk measures the gap between $\theta_{priv}$ and $\theta^*$ over the dataset $D$, where $\theta_{priv}$ denotes the private model. Before the analysis, we first introduce the Polyak-Łojasiewicz (PL) condition [20].

**Definition 4 (Polyak-Łojasiewicz condition)** *$L(\theta; D)$ satisfies the Polyak-Łojasiewicz (PL) condition if there exists $\mu > 0$ for all $\theta$:*

$$\|\nabla_\theta L(\theta; D)\|_2^2 \geq 2\mu(L(\theta; D) - L^*).$$

PL condition is one of the weakest curvature conditions [26], it does not assume the loss function to be convex and it is commonly used in non-convex optimiztion. Many non-convex models satisfy the condition, including deep (linear) [8] and shallow neural networks [25].

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_9

**Remark 1** *If $L(\theta; D)$ satisfies the PL condition, then it satisfies the Quadratic Growth (QG) condition [20], i.e., $L(\theta; D) - L(\theta^*; D) \geq \frac{\mu}{2}\|\theta - \theta^*\|_2^2$, where $\theta^*$ denotes the optimal model over dataset $D$.*

**Theorem 1** *Suppose that $\ell(\theta, z)$ is $G$-Lipschitz, $L$-smooth, and satisfies PL condition over $\theta$. With learning rate $\alpha = \frac{1}{L}$, $\sigma$ given in (1) to guarantee $(\epsilon, \delta)$-DP, and $T = \mathcal{O}(\log(n))$, then with the dimensions of the model $m$:*

$$\mathbb{E}\left[L(\theta_{priv}; D) - L^*\right] \leq \mathcal{O}\left(\frac{mG^2 \log(1/\delta) \log(n)}{n^2 \epsilon^2}\right),$$

*the expectation is taken over the algorithm and dataset $D$.*

**Remark 2** *Many researchers discussed the excess empirical risk in previous works. To the best of our knowledge, one of the best results is given by [40], in which $T$ is multiplied 'rudely' to the noise term when summing the loss over $T$ iterations. As a result, the excess empirical risk bound is $\mathcal{O}\big(\frac{mG^2 \log(1/\delta) \log^2(n)}{n^2 \epsilon^2}\big)$ in [40]. However, we solve a geometric sequence when summing the loss, and get a tighter bound in this paper. As a result, the excess empirical risk bound is improved by a factor of $\log(n)$ overall.*

**Excess Population Risk** To get the excess population risk bound, we first analyze the generalization error, which measures the gap between the performance over the underlying distribution and the dataset $D$ of the private model, connecting the population risk with the empirical risk.

**Theorem 2** *If the loss function is $G$-Lipschitz, $L$-smooth, and satisfies the PL condition over $\theta$, the generalization error bound of $\theta_{priv}$ satisfies:*

$$\begin{aligned} &\mathbb{E}\left[L_{\mathcal{P}}(\theta_{priv}) - L(\theta_{priv}; D)\right] \\ &\leq \inf_{\tau > 0} \left\{ \frac{8(\tau + L)}{\mu} \mathbb{E}[L(\theta_{priv}; D) - L^*] + \frac{16G^2(\tau + L)}{n^2 \mu^2} + \frac{L\mathbb{E}[L(\theta_{priv}; D)]}{\tau} \right\}, \end{aligned} \tag{2}$$

*where the expectation is taken over the algorithm.*

By Theorem 2, one may observe that the generalization error decreases if the optimization error (the excess empirical risk) is smaller, which is in line with the observation in [16, 8, 25]: 'optimization helps generalization'.

Now, we give the excess population risk bound.

**Theorem 3** *If the loss function is $G$-Lipschitz, $L$-smooth, and satisfies the PL condition over $\theta$, with learning rate $\alpha = \frac{1}{L}$, the excess population risk of $\theta_{priv}$ satisfies:*

$$\begin{aligned} \mathbb{E}\left[L_{\mathcal{P}}\left(\theta_{priv}\right) - \min_{\theta} L_{\mathcal{P}}\left(\theta\right)\right] \leq (\tau + L)\left(\frac{(8\tau + \mu)}{\mu\tau} \mathbb{E}[L(\theta_{priv}; D) - L^*] + \frac{16G^2}{n^2\mu^2}\right) \\ + \frac{L}{\tau}\mathbb{E}[L^*]. \end{aligned}$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_9

**Remark 3** *Combining the result given in Theorem 1, if* $\mathbb{E}[L^*] = \mathcal{O}(1/n)$*, taking* $T = \mathcal{O}(\log(n))$*, if we ignore constants and* $\log(\cdot)$ *terms, then for all* $\tau > 0$*, the excess population risk bound comes to:* $\mathcal{O}\big(\frac{m}{\tau n^2 \epsilon^2} + \frac{\tau m}{n^2 \epsilon^2} + \frac{1}{\tau n}\big)$*. If* $\tau = \mathcal{O}(1)$*, it is* $\mathcal{O}\big(\frac{m}{n^2\epsilon^2} + \frac{1}{n}\big)$*. If* $\tau = \mathcal{O}(\sqrt{n}\epsilon/\sqrt{m})$*, the excess population is is* $\mathcal{O}\big(\frac{\sqrt{m}}{n^{1.5}\epsilon} + \frac{m^{1.5}}{n^{2.5}\epsilon^3}\big)$*.*

*Thus, we get the upper bound of the excess population risk:*

$$\mathbb{E}\left[L_{\mathcal{P}}\left(\theta_{priv}\right) - \min_{\theta} L_{\mathcal{P}}\left(\theta\right)\right] = \mathcal{O}\left(\min\left\{\frac{m}{n^2\epsilon^2} + \frac{1}{n}, \frac{\sqrt{m}}{n^{1.5}\epsilon} + \frac{m^{1.5}}{n^{2.5}\epsilon^3}\right\}\right).$$

In Remark 3, to get the better result, we assume $\mathbb{E}[L^*] = \mathcal{O}(1/n)$. It is a small value because $L^*$ is the optimal value over the whole dataset. Besides, it is common to assume the minimal population risk $\mathbb{E}[\min L_{\mathcal{P}}(\theta)] \leq \mathcal{O}(1/n)$ [24, 46, 28, 45, 34]. Moreover, under expectation, considering $\theta^*_{\mathcal{P}} = \arg\min L_{\mathcal{P}}(\theta)$ is independent of dataset, so $\mathbb{E}[L^*] \leq \mathbb{E}[\min L_{\mathcal{P}}(\theta)]$ [23]. Thus, the assumption is reasonable.

All the theorems given above only assume that the loss function $\ell(\cdot)$ is $G$-Lipschitz, $L$-smooth and satisfies PL inequality, without convex assumption. So the results are general and can be applied to some of the non-convex conditions.

## 5 Performance Improving DP-SGD

Motivated by the definition of DP, we focus on the contributions made by data instances on the final model. In particular, if the effects caused by a data instance $z$ on the final machine learning model is so little that the attacker cannot realize it (less than $e^{\epsilon}$), there is no need to add noise to $z$. Now, only one problem is left: How to measure the impact of the data instances on the model? A classic technique, Influence Function (IF), gives us some inspirations.

### 5.1 Influence Function and Error Analysis

The contribution of data instance $z$ is naturally defined as $\theta^*_{-z} - \theta^*$, where $\theta^*_{-z} = \arg\min_\theta \sum_{z_i \neq z} \ell(\theta, z_i)$. To measure the gap between them, a straight method is to train two models: $\theta^*$, $\theta^*_{-z}$. However, retraining a model for each data instance $z$ is prohibitively slow. To solve the problem, influence [22] measures the contributions on the machine learning model made by data instances:

$$c_z := -\frac{1}{n}\left(-H^{-1}_{\theta^*}\nabla_\theta \ell\left(\theta^*, z\right)\right) \approx \theta^*_{-z} - \theta^*, \tag{3}$$

where $H_{\theta^*} = \frac{1}{n}\sum_{i=1}^{n} \nabla^2_\theta \ell(\theta^*, z_i)$, assumed positive definite. Via (3), we can measure how the model changes if we 'drop' one data instance, naturally in line with the definition of DP.

The influence function $c_z$ is got by Taylor expansion [27], in which Taylor remainders lead an approximation error. However, [22] only gives an approximation via IF, but not discusses the corresponding error. To fill the gap, we analyze the approximation error in this section, via the definition given below.

DOI: 10.1007/978-3-031-35995-8_9

**Algorithm 1** Performance Improving DP-SGD

```
Require: dataset $D$, learning rate $\alpha$, local iteration rounds $T_{local}$, global update rounds $R$
1: function PIDP-SGD($D, \alpha, T_{local}, R$)
2:     Initialize $\theta_0^{(g)}, \theta_0^{(l)} \leftarrow \widetilde{\theta}$.
3:     for $r = 0$ to $R-1$ do
4:         Get $H_{\widetilde{\theta}_r^{(g)}}$ and compute its inverse.
5:         for $t = 0$ to $T_{local}-1$ do
6:             Choose data instance $z_t$ randomly.
7:             Get contribution of $z_t$: $c_t^{(o)} = c_{z_t} + E$, via $H_{\widetilde{\theta}_r^{(g)}}$.
8:             Sample $b^{(c)} \sim \mathcal{N}(0, \sigma_{(c)}^2 I_m)$.
9:             $c_t = 2c_t^{(o)} + b^{(c)}$, if there exists any $i$ that $|[c_t]_i| \geq \frac{2re^{\epsilon_1}\delta_1}{e^{\epsilon_1}-1}$, jump to line 13;
           otherwise, jump to line 10.
10:            if $\ln\left(\frac{2re^{\epsilon_1}\delta_1}{2re^{\epsilon_1}\delta_1-(e^{\epsilon_1}-1)sign([c_t]_i)[c_t]_i}\right) \leq 2\epsilon_1$ for all $i \in [1, m]$, then
11:                $\theta_{t+1}^{(l)} \leftarrow \theta_t^{(l)} - \alpha\nabla_\theta \ell(\theta_t^{(l)}, z_t)$.
12:            else
13:            Sample $b \sim \mathcal{N}(0, \sigma^2 I_m)$,
14:            $\theta_{t+1}^{(l)} \leftarrow \theta_t^{(l)} - \alpha\big(\nabla_\theta \ell(\theta_t^{(l)}, z_t) + b\big)$.
15:            endif
16:        endfor
17:        $\theta_{r+1}^{(g)} = \theta_{T_{local}}^{(l)}$.
18:    endfor
19:    return $\theta_{priv} = \theta_R^{(g)}$.
20: end function
```

**Definition 5 ($C$-Hessian Lipschitz)** *A loss function $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ is $C$-Hessian Lipschitz over $\theta$, if for any $z \in \mathcal{D}$ and $\theta, \theta' \in \mathcal{C}$, we have: $\|\nabla_\theta^2 \ell(\theta, z) - \nabla_\theta^2 \ell(\theta', z)\|_2 \leq C\|\theta - \theta'\|_2$.*

**Remark 4** *$C$-Hessian Lipschitz means that $\|\nabla_\theta^3 \ell(\theta, z)\|_2 \leq C$. For Mean Squared Error, $\|\nabla_\theta^3 \ell(\theta, z)\|_2 = 0$. For logistic regression, elements in $\nabla_\theta^3 \ell(\theta, z)$ are less than 0.097. The examples above show that the assumption is reasonable.*

**Theorem 4** *If $\ell(\theta, z)$ is $G$-Lipschitz, $L$-smooth, $C$-Hessian Lipschitz over $\theta$, and $\|H_{\theta^*}\|_2 \geq \zeta$, then with $c_z$ given in (3) the approximation error satisfies:*

$$E \coloneqq \|(\theta_{-z}^* - \theta^*) - c_z\|_2 \leq \frac{1}{\zeta^2 n^2}\left(2LG + \frac{CG^2}{\zeta}\right).$$

**Remark 5** *Theorem 4 gives a $\mathcal{O}(1/n^2)$ approximation error when applying $c_z$, which means that $c_z$ is precise. Besiders, in Theorem 4, we assume that $\|H_{\theta^*}\|_2 \geq \zeta$. Because most of the algorithms are regularized, the assumption is easy to hold.*

### 5.2 Performance Improving DP-SGD

We set a threshold: $e^\epsilon$ for adding noise by the following observation: the appearance (or absence) of some data affects the model so little that attackers cannot infer anything from them. Changing those data instances cannot threaten $(\epsilon,\delta)$-DP of the model. So, we calculate the 'contribution' of data by IF, only add noise to whom contributes more than $e^{\epsilon}$.

Details of the Performance Improving algorithm are given in Algorithm 1[4]. Different from traditional DP-SGD algorithm, Algorithm 1 applies a decision process before gradient descent (lines 9 and 10), to decide whether to add random noise or not. If the effect made by the chosen $z_t$ is no more than $e^{\epsilon}$, SGD runs; otherwise, we sample Gaussian noise $b$ and run DP-SGD. In other words, lines 9 and 10 connects the value of IF with the privacy loss of DP. Meanwhile, we notice that except the training process, the contribution calculating process may also disclose the sensitive information. So we add noise to the $c_t^{(o)}$ to guarantee the claimed DP (line 9). Noting that the contribution given by (3) is based on Taylor expansion, causing an approximation error (discussed in Section 5.1), we fix it by adding $E$ to $c_t$ in line 7.

It is easy to follow that if the privacy budget $\epsilon$ is higher, the constraint of adding noise is looser, which means that fewer data instances meet the noise. As a result, the performance of PIDP-SGD will be better if $\epsilon$ is higher.

Besides, the method given in this paper will inspire other researchers to apply it to corresponding fields such as mini-batch gradient descent.

**Remark 6** *The time complexity of Algorithm 1 is $\mathcal{O}\big(Rnm^2+RT_{local}m\big)$. Under the worst case $T_{local}=1$, it becomes $\mathcal{O}\big(Rnm^2\big)$, where $R$ is the total number of iterations. Fortunately, an efficient approach to calculate the Influence Function was given in [22], and the time complexity can be reduced to $\mathcal{O}\big(Rnm\big)$. For some other previous performance method, the time complexity also increases, we take DP-LSSGD as an example here, whose time complexity is $\mathcal{O}\big(Rm^2\big)$. Under the cases $n>m$, our time complexity is larger, but under high dimension cases, when $n\leq m$, our time complexity is better. However, both theoretical and experimental results of our method is much better than DP-LSSGD (see Table 1 and Figures 1 and 2). So under low dimension conditions, the sacrifice on time complexity is a trade-off against the model performance; under high dimension conditions, our method is much better on both the model performance and the time complexity.*

### 5.3 Privacy Guarantees

**Theorem 5** *For $\delta_1,\delta_2>0$ and $\epsilon_1,\epsilon_2>0$, if $\ell(\theta,z)$ is G-Lipschitz over $\theta$, with $\sigma\geq c\frac{G\sqrt{T\log(1/\delta_1)}}{n\epsilon_1}$, $\sigma_{(c)}\geq c'\frac{GR\sqrt{\log(1.25R/\delta_2)}}{n\zeta\epsilon_2}$, where $T=T_{local}*R$. Algorithm 1 is $(\epsilon_1+\epsilon_2,\delta_1+\delta_2)$-DP for some constants $c,c'$.*

Theorem 5 shows that the privacy of Algorithm 1 consists of two parts: (1) computing $c_t$ and (2) training model. Specifically, $\epsilon_1,\delta_1$ $(\sigma)$ are for the privacy when training the model and $\epsilon_2,\delta_2$ $(\sigma_{(c)})$ are for the privacy when computing $c_t$.

In Algorithm 1, noise is only added when training with a partition of data instances, which leads better excess risk bounds. In the following, we suppose

[4] In Algorithm 1, $sign(\cdot)$ is the signum function, $r$ is the radius of the parameter space.

DOI: 10.1007/978-3-031-35995-8_9

that there are $k$ data instances affect the model significantly and measure the improvement brought by our proposed 'Performance Improving' algorithm.

### 5.4 Utility Analysis

We first give the excess empirical risk bound.

**Theorem 6** *Suppose that $\ell(\theta, z)$ is $G$-Lipschitz, $L$-smooth, and satisfies PL condition over $\theta$. With learning rate $\alpha = \frac{1}{L}$ and $k$ data instances affect the model significantly, the excess empirical risk can be improved to:*

$$\mathbb{E}\left[L(\theta_{priv}; D) - L^*\right] \leq \mathcal{O}\left(\frac{kmG^2 \log(1/\delta_1)\log(n)}{n^3\epsilon_1^2}\right).$$

Noting that $\frac{k}{n} < 1$, the excess empirical risk bound brought by Theorem 6 is better than Theorem 1.

Via Theorem 2, we find that the generalization error is only related to $\|\theta_i^T - \theta^T\|_2$ and $L(\theta_{priv}; D)$, and these terms are only determined by the optimization process, so the generalization error of Algorithm 1 is the same as which given in Theorem 2. Then we come to the excess population risk.

**Theorem 7** *If $\ell(\cdot)$ is $G$-Lipschitz, $L$-smooth, and satisfies the PL condition over $\theta$. With learning rate $\alpha = \frac{1}{L}$, $T = \mathcal{O}(\log(n))$ and $k$ data instances affect the model significantly, the excess population risk can be improved to:*

$$\mathcal{O}\left(\min\left\{\frac{km}{n^3\epsilon^2} + \frac{1}{n}, \frac{km}{n^{2.5}\epsilon^2} + \frac{1}{n^{1.5}}, \frac{(km)^{1.5}}{n^2\epsilon^3} + \frac{\epsilon}{\sqrt{kmn}}\right\}\right).$$

The proof is similar to Theorem 3 and the discussion given in Remark 3. The first, second, third and last terms are derived from taking $\tau = \mathcal{O}(1), \mathcal{O}(\sqrt{n})$ and $\mathcal{O}(n\epsilon/\sqrt{km})$, respectively. Noting that $\frac{k}{n} < 1$, the result is better than which given by Theorem 3.

In Theorem 7, we theoretically prove that the excess risk of DP models can be better by considering data heterogeneity. It may give new inspirations to the utility analysis in the future work.

## 6 Comparison with Related Work

Previous works always discuss the excess empirical risk but seldom analyze the excess population risk, so we mainly focus on comparing the excess population risk in this section. Details can be found in Table 1, in which $G, L$, S.C., C, PL represent $G$-Lipschitz, $L$-smooth, strongly convex, convex and PL inequality, respectively and EER, EPR denote the Excess Empirical Risk and the Empirical Population Risk, respectively.

For the excess population risk, the best previous result is $\mathcal{O}\big(\frac{m}{n^2\epsilon^2} + \frac{1}{n}\big)$, given in [14], under strongly convex condition. As shown in Table 1, our result is better

DOI: 10.1007/978-3-031-35995-8_9

**Table 1.** Comparisons on excess risk bounds between our method and other methods.

| | $G$ | $L$ | S.C. | C | PL | EPR |
|---|---|---|---|---|---|---|
| [4] | ✓ | ✓ | × | ✓ | × | $\mathcal{O}\left(\frac{\sqrt{m}}{n\epsilon}+\frac{1}{\sqrt{n}}\right)$ |
| [14] | ✓ | × | ✓ | ✓ | × | $\mathcal{O}\left(\frac{m}{n^2\epsilon^2}+\frac{1}{n}\right)$ |
| [14] | ✓ | × | × | ✓ | × | $\mathcal{O}\left(\frac{\sqrt{m}}{n\epsilon}+\frac{1}{\sqrt{n}}\right)$ |
| Ours | ✓ | ✓ | × | × | ✓ | $\mathcal{O}\left(\min\left\{\frac{m}{n^2\epsilon^2}+\frac{1}{n}, \frac{\sqrt{m}}{n^{1.5}\epsilon}+\frac{m^{1.5}}{n^{2.5}\epsilon^3}\right\}\right)$ |
| Ours (PIDP-SGD) | ✓ | ✓ | × | × | ✓ | $\mathcal{O}\left(\min\left\{\frac{km}{n^3\epsilon^2}+\frac{1}{n}, \frac{km}{n^{2.5}\epsilon^2}+\frac{1}{n^{1.5}}, \frac{(km)^{1.5}}{n^2\epsilon^3}+\frac{\epsilon}{\sqrt{kmn}}\right\}\right)$ |

by a factor up to $\mathcal{O}\big(\frac{1}{\sqrt{n}}\big)$. When it comes to our proposed PIDP-SGD method, the result is further improved by a factor up to $\mathcal{O}\big(\frac{k}{n}\big)$. Noting that the best result given in [14] requires the loss function to be strongly convex, which means that our result is not only better but also strictly more general than which given in [14]. For the best results under convex condition [4, 14]: $\mathcal{O}\big(\frac{\sqrt{m}}{n\epsilon}+\frac{1}{\sqrt{n}}\big)$, our results (both the original one and the one given by PIDP-SGD) are much better. Although it is hard to compare convexity with the PL condition, our results can be applied to some of the non-convex models (shown in Definition 4).

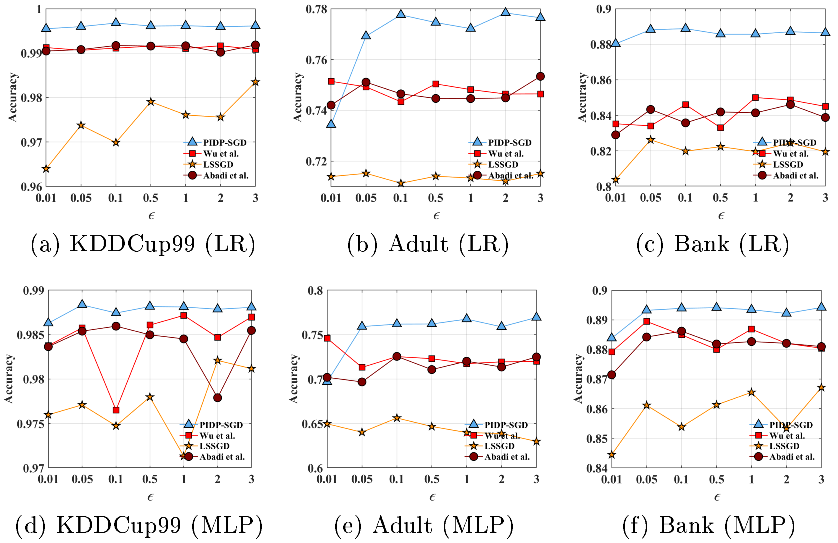
Besides, for the excess empirical risk, our analyzed bound is better than which proposed by [5, 41, 37, 4] and achieves the best result $\mathcal{O}\big(\frac{m}{n^2\epsilon^2}\big)$. For our proposed 'performance improving' method: PIDP-SGD, our analyzed excess empirical risk bound is further tighter by a factor of $\mathcal{O}\big(\frac{k}{n}\big)$. It is worth emphasizing that most of the methods proposed previously assume that the loss function is convex, which is not required in our method. Under this circumstance, we achieve a better result, which is attractive. Additionally, for the non-convex analysis given in [40], the excess empirical risk bound of our analyzed DP-SGD method is better by a factor of $\mathcal{O}\big(\log(n)\big)$ (as discussed in Remark 2) and the PIDP-SGD method is better by a factor of $\mathcal{O}\big(\frac{k\log(n)}{n}\big)$.

## 7 Experimental Results

Experiments on several real datasets are performed on the classification task. Since our method is based on SGD, we compare our method with previous DP-SGD methods. Specifically, we compare our method with the gradient perturbation method proposed in [1], the output perturbation method proposed in [41] and the DP-LSSGD method proposed in [36]. The performance is measured in terms of classification accuracy and the optimality gap. The accuracy represents the performance on the testing set, and the optimality gap represents the excess empirical risk on the training set. The optimality gap is denoted by $L(\theta_{priv}; D) - L^*$.

We use both logistic regression model and deep learning model on the datasets KDDCup99 [18], Adult [11] and Bank [29], where the total number of data instances are 70000, 45222, and 41188, respectively. In the experiments, to make

DOI: 10.1007/978-3-031-35995-8_9



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

(d) KDDCup99 (MLP) (e) Adult (MLP) (f) Bank (MLP)

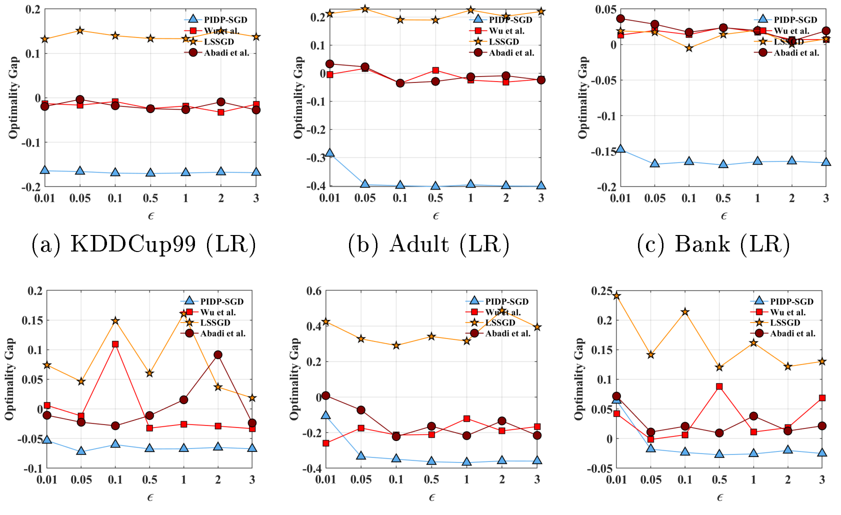
**Fig. 1.** Accuracy over $\epsilon$, LR denotes logistic regression model and MLP denotes the deep learning model.

the model satisfies the assumptions (such as PL condition) mentioned in the theoretical part, the deep learning model is denoted by Multi-layer Perceptron (MLP) with one hidden layer whose size is the same as the input layer. Training and testing datasets are chosen randomly. In all the experiments, total iteration rounds $T$ is chosen by cross-validation. For PIDP-SGD, we set $RT_{local} = T$. We evaluate the performance of our proposed PIDP-SGD method and some of previous algorithms over the differential privacy budget $\epsilon$. For $\epsilon$, we set it from 0.01 to 3, and in the PIDP-SGD method, we set $\epsilon_1 = 3\epsilon_2 = 3\epsilon/4$ to guarantee $\epsilon_1 + \epsilon_2 = \epsilon$. The results are shown in Figure 1 and Figure 2.

Figure 1 shows that as the privacy budget $\epsilon$ increases, so does the accuracy, which follows the intuition. When applying the PIDP-SGD algorithm, the accuracy rises on most datasets, which means that our proposed 'performance improving' method is effective. Meanwhile, when $\epsilon$ is small, the difference (on accuracy) between traditional methods and 'performance improving' method is also small. However, as $\epsilon$ increases, the 'performance improving' method becomes more and more competitive. The reason is that larger $\epsilon$ means that more data instances 'escape' the injected noise, leading to better accuracies.

Figure 2 shows that on some datasets, by applying PIDP-SGD algorithm, the optimality gap of our method is almost 0, which means that it achieves almost

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_9



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

(d) KDDCup99 (MLP) (e) Adult (MLP) (f) Bank (MLP)

**Fig. 2.** Optimality gap over $\epsilon$, LR denotes logistic regression model and MLP denotes the deep learning model.

the same performance as the model without privacy in some scenarios[5]. Besides, similar to the accuracy in Figure 1, the optimality gap decreases as $\epsilon$ increases, which follows our intuition. Moreover, on some datasets, the performance of some of the methods fluctuates. The reason is that in the setting of differential privacy, random noise is injected into the model, so it is a common phenomenon.

Additionally, on some datasets, the performance of our 'performance improving' method is worse when $\epsilon$ is small, the reason is that part of the privacy budget is allocated to $c_t$, which means 'pure privacy budget' on the model is smaller. Thus, with the increase of $\epsilon$, the 'performance improving' method becomes more competitive, which has been analyzed before in Section 5. Experimental results show that our proposed PIDP-SGD algorithm significantly improves the performance under most circumstances.

## 8 Conclusions

In this paper, we give sharper excess risk empirical and population risk bounds of traditional DP-SGD paradigm. Theoretical results show that our given excess risk bounds are better than previous methods under both convex and nonconvex conditions. Meanwhile, based on DP-SGD, we attempt to improve the

[5] The optimal model (derives $L^*$) is trained, rather than numerical solutions, and random noise may make the model escape from local minima, so negative optimality gaps appear.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_9

performance from a new perspective: considering data heterogeneity, rather than treating all data the same. In particular, we introduce the influence function (IF) to analyze the contribution of each data instance to the final model, and the approximation error analysis shows that IF is reasonable to approximate the contribution. In this way, we propose PIDP-SGD: only adding noise to the data demonstrating significant contributions (more than $e^{\epsilon}$) when training. Detailed theoretical analysis and experimental results show that our proposed PIDP-SGD achieves better performance, without the convexity assumption. Moreover, the new perspective of treating different data instances differently may give new inspirations to future work, including the privacy analysis and the utility analysis. In future work, we will focus on improving the time complexity of PIDP-SGD and applying the algorithm to larger datasets.

**Acknowledgements** This work was supported in part by the Excellent Talents Program of IIE, CAS, the Special Research Assistant Project of CAS, the Beijing Outstanding Young Scientist Program (No. BJJWZYJH012019100020098), Beijing Natural Science Foundation (No. 4222029), and National Natural Science Foundation of China (No. 62076234, No. 62106257).

## References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318 (2016)
2. Arora, R., Upadhyay, J.: On differentially private graph sparsification and applications. In: Advances in Neural Information Processing Systems. pp. 13378–13389 (2019)
3. Backes, M., Berrang, P., Humbert, M., Manoharan, P.: Membership privacy in microrna-based studies. In: ACM SIGSAC Conference on Computer and Communications Security. p. 319–330 (2016)
4. Bassily, R., Feldman, V., Talwar, K., Guha Thakurta, A.: Private stochastic convex optimization with optimal rates. In: Advances in Neural Information Processing Systems. pp. 11279–11288 (2019)
5. Bassily, R., Smith, A., Thakurta, A.: Private empirical risk minimization: Efficient algorithms and tight error bounds. In: IEEE Annual Symposium on Foundations of Computer Science. pp. 464–473 (2014)
6. Bernstein, G., Sheldon, D.R.: Differentially private bayesian linear regression. In: Advances in Neural Information Processing Systems. pp. 523–533 (2019)
7. Carlini, N., Liu, C., Erlingsson, U., Kos, J., Song, D.: The secret sharer: Evaluating and testing unintended memorization in neural networks. In: USENIX Conference on Security Symposium. p. 267–284 (2019)
8. Charles, Z., Papailiopoulos, D.: Stability and generalization of learning algorithms that converge to global optima. In: International Conference on Machine Learning. pp. 745–754 (2018)
9. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. Journal of Machine Learning Research pp. 1069–1109 (2011)

10. Chen, Z., Ni, T., Zhong, H., Zhang, S., Cui, J.: Differentially private double spectrum auction with approximate social welfare maximization. IEEE Transactions on Information Forensics and Security pp. 2805–2818 (2019)
11. Dua, D., Graff, C.: UCI machine learning repository (2017)
12. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference. pp. 265–284 (2006)
13. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science pp. 211–407 (2014)
14. Feldman, V., Koren, T., Talwar, K.: Private stochastic convex optimization: Optimal rates in linear time. In: Annual ACM SIGACT Symposium on Theory of Computing. p. 439–449 (2020)
15. 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: USENIX Conference on Security Symposium. pp. 17–32 (2014)
16. Hardt, M., Recht, B., Singer, Y.: Train faster, generalize better: Stability of stochastic gradient descent. In: International Conference on Machine Learning. pp. 1225–1234 (2016)
17. Heikkilä, M., Jälkö, J., Dikmen, O., Honkela, A.: Differentially private markov chain monte carlo. In: Advances in Neural Information Processing Systems, pp. 4115–4125 (2019)
18. Hettich, S., Bay, S.D.: The uci kdd archive (1999)
19. Jayaraman, B., Evans, D.: Evaluating differentially private machine learning in practice. In: USENIX Conference on Security Symposium. pp. 1895–1912 (2019)
20. 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. pp. 795–811 (2016)
21. Kifer, D., Smith, A., Thakurta, A.: Private convex empirical risk minimization and high-dimensional regression. In: Conference on Learning Theory. pp. 25–1 (2012)
22. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: International Conference on Machine Learning. pp. 1885–1894 (2017)
23. Lei, Y., Ledent, A., Kloft, M.: Sharper generalization bounds for pairwise learning. Advances in Neural Information Processing Systems (2020)
24. Lei, Y., Ying, Y.: Fine-grained analysis of stability and generalization for stochastic gradient descent. In: International Conference on Machine Learning. pp. 5809–5819 (2020)
25. Lei, Y., Ying, Y.: Sharper generalization bounds for learning with gradient-dominated objective functions. In: International Conference on Learning Representations (2021)
26. Li, S., Liu, Y.: Improved learning rates for stochastic optimization: Two theoretical viewpoints (2021)
27. Linnainmaa, S.: Taylor expansion of the accumulated rounding error. BIT Numerical Mathematics pp. 146–160 (1976)
28. Liu, M., Zhang, X., Zhang, L., Jin, R., Yang, T.: Fast rates of ERM and stochastic approximation: Adaptive to error bound conditions. In: Advances in Neural Information Processing Systems. pp. 4683–4694 (2018)
29. Moro, S., Cortez, P., Rita, P.: A data-driven approach to predict the success of bank telemarketing. Decision Support Systems pp. 22–31 (2014)
30. Phan, N., Wu, X., Hu, H., Dou, D.: Adaptive laplace mechanism: Differential privacy preservation in deep learning. In: IEEE International Conference on Data Mining. pp. 385–394 (2017)

31. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: ACM SIGSAC Conference on Computer and Communications Security. pp. 1310–1321 (2015)
32. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: IEEE Symposium on Security and Privacy. pp. 3–18 (2017)
33. Song, S., Chaudhuri, K., Sarwate, A.D.: Stochastic gradient descent with differentially private updates. In: IEEE Global Conference on Signal and Information Processing. pp. 245–248 (2013)
34. Srebro, N., Sridharan, K., Tewari, A.: Optimistic rates for learning with a smooth loss. arXiv preprint arXiv:1009.3896 (2010)
35. Ullman, J., Sealfon, A.: Efficiently estimating erdos-renyi graphs with node differential privacy. In: Advances in Neural Information Processing Systems. pp. 3765–3775 (2019)
36. Wang, B., Gu, Q., Boedihardjo, M., Barekat, F., Osher, S.J.: Dp-lssgd: A stochastic optimization method to lift the utility in privacy-preserving erm. arXiv preprint arXiv:1906.12056 (2019)
37. Wang, B., Gu, Q., Boedihardjo, M., Barekat, F., Osher, S.J.: Dp-lssgd: A stochastic optimization method to lift the utility in privacy-preserving erm. CoRR (2019)
38. Wang, D., Chen, C., Xu, J.: Differentially private empirical risk minimization with non-convex loss functions. In: International Conference on Machine Learning. pp. 6526–6535 (2019)
39. Wang, D., Xu, J.: Principal component analysis in the local differential privacy model. Theoretical Computer Science (2019)
40. Wang, D., Ye, M., Xu, J.: Differentially private empirical risk minimization revisited: Faster and more general. In: Advances in Neural Information Processing Systems. pp. 2722–2731 (2017)
41. Wu, X., Li, F., Kumar, A., Chaudhuri, K., Jha, S., Naughton, J.: Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In: ACM International Conference on Management of Data. pp. 1307–1322 (2017)
42. Xiao, L., Zhang, T.: A proximal stochastic gradient method with progressive variance reduction. SIAM Journal on Optimization pp. 2057–2075 (2014)
43. Xu, C., Ren, J., Zhang, D., Zhang, Y., Qin, Z., Ren, K.: Ganobfuscator: Mitigating information leakage under gan via differential privacy. IEEE Transactions on Information Forensics and Security pp. 2358–2371 (2019)
44. Zhang, J., Zheng, K., Mou, W., Wang, L.: Efficient private erm for smooth objectives. arXiv preprint arXiv:1703.09947 (2017)
45. Zhang, L., Yang, T., Jin, R.: Empirical risk minimization for stochastic convex optimization: $o(1/n)$-and $o(1/n^2)$-type of risk bounds. In: Conference on Learning Theory. pp. 1954–1979 (2017)
46. Zhang, L., Zhou, Z.H.: Stochastic approximation of smooth and strongly convex functions: Beyond the $o(1/t)$ convergence rate. In: Conference on Learning Theory. pp. 3160–3179 (2019)
47. 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 Conference on Computer Communications. pp. 2087–2095 (2018)

DOI: 10.1007/978-3-031-35995-8_9