Efficient Private ERM for Smooth Objectives

Jiaqi Zhang[†], **Kai Zheng**[†], **Wenlong Mou**[†], **Liwei Wang**[†]
[†] Key Laboratory of Machine Perception, MOE,
School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, China
{Zhangjq,wanglw}@cis.pku.edu.cn, {zhengk92, mouwenlong}@pku.edu.cn

Abstract

In this paper, we consider efficient differentially private empirical risk minimization from the viewpoint of optimization algorithms. For strongly convex and smooth objectives, we prove that gradient descent with output perturbation not only achieves nearly optimal utility, but also significantly improves the running time of previous state-of-the-art private optimization algorithms, for both ϵ-DP and (ϵ, δ)-DP. For non-convex but smooth objectives, we propose an RRPSGD (Random Round Private Stochastic Gradient Descent) algorithm, which provably converges to a stationary point with privacy guarantee. Besides the expected utility bounds, we also provide guarantees in high probability form. Experiments demonstrate that our algorithm consistently outperforms existing method in both utility and running time.

1 Introduction

Data privacy has been a central concern in statistics and machine learning, especially when utilizing sensitive data such as financial accounts and health-care data. Thus, it is important to design machine learning algorithms which protect users' privacy. As a rigorous and standard concept of privacy, differential privacy [Dwork *et al.*, 2006] guarantees that the algorithm learns statistical information of the population, but nothing about individual users. In the framework of differential privacy, there has been a long line of research studying differentially private machine learning algorithms, such as [Chaudhuri *et al.*, 2011; Chaudhuri *et al.*, 2013; Jing, 2011; Rubinstein *et al.*, 2012; Talwar *et al.*, 2015].

Among all machine learning models, empirical risk minimization (ERM) plays an important role, as it covers a variety of machine learning tasks. Once we know how to do ERM privately, it is straightforward to obtain differentially private algorithms for a large variety of machine learning problems, such as classification, regression, etc. The earliest representative work of this research line is done by Chaudhuri et al. [Chaudhuri *et al.*, 2011]. They proposed two approaches to guarantee differential privacy of the output of ERM, namely, output perturbation and objective perturbation. Output perturbation is a variant of Laplace (Gaussian) mechanism, where the stability of exact solutions plays a key role in the analysis. Objective perturbation is done by adding noise to ERM objective and solving precise solution to the new problem. In Kifer et at. [Kifer *et al.*, 2012], they extend the method of objective perturbation, and prove similar results for more general case, especially for high-dimensional learning.

Both [Chaudhuri *et al.*, 2011] and [Kifer *et al.*, 2012] were discussed in terms of precise solutions to optimization problems. In reality, however, it is not only intractable but also unnecessary to obtain precise solutions. Instead, we always use some optimization algorithms to obtain approximate solutions. In this context, the interaction between privacy-preserving mechanisms and optimization algorithms has non-trivial implications to both sides: running the algorithm for finite cycles of iteration inherently enhances stability; on the other hand, noise added to preserve privacy introduces new challenges to the convergence rate of optimization algorithms. The purpose of this research is therefore two-fold: both utility and time complexity are of central concern.

In literature, [Bassily *et al.*, 2014] and [Song *et al.*, 2013] use stochastic gradient descent (SGD) as the basic optimization algorithm to solve ERM, and add noise to each iteration to achieve (ε, δ)-differential privacy. Bassily et al. [Bassily *et al.*, 2014] develop an efficient implementation of exponential mechanism to achieve ε-differential privacy. Furthermore, they also prove their algorithms match the lower bounds for corresponding problems (ignoring $\log$ factors). Nearly at the same time of this paper, [Wu *et al.*, 2017] combined output perturbation with permutation SGD according to its stability analysis. However, their utility results only hold for constant number of passes over data, which did not match existing lower bound. Besides these worst-case results, [Talwar *et al.*, 2014] gives a more careful analysis based on constraint set geometry, which leads to better utility bounds in specific problems such as LASSO. Despite the success of previous works in terms of utility, there are still much work to do from a practical perspective.

1. Both of algorithms proposed in [Bassily *et al.*, 2014] and [Talwar *et al.*, 2014] have to run at least $\Omega(n^2)$ iterations to reach ideal accuracy (n is number of data points), which is much slower than non-private version and makes

the algorithm impractical for large data. Can we do faster while still guarantee privacy and accuracy?

2. Note that all existing results only hold for convex ERM, yet non-convex objective functions have been increasingly important, especially in deep neural networks. Can we design an efficient and private optimization algorithms for non-convex ERM with theoretical guarantee?

Fortunately, the answers to above questions are both "yes". In this paper, we will give two efficient algorithms with privacy and utility guarantees. Throughout this paper, we assume the objective function is β-smooth (See Section 2 for precise definition), which is a natural assumption in optimization and machine learning. Smoothness allows our algorithm to take much more aggressive gradient steps and converge much faster, which is not fully utilized in previous work like [Bassily *et al.*, 2014] and [Talwar *et al.*, 2014]. Moreover, smoothness also makes it possible for non-convex case to have theoretical guarantees around stationary points.

Technically, our work is partially inspired by the work of Hardt et al. [Hardt *et al.*, 2016], in which they established the expected stability $\mathbb{E}\|\mathcal{A}(S) - \mathcal{A}(S')\|$ of SGD ($\mathcal{A}$ is a randomized algorithm, and S, S' are neighboring datasets). Using similar techniques we can derive worst case stability for deterministic algorithms like classical gradient descent, which plays a core role in private algorithm design. For non-convex ERM, we use a variant of Randomized Stochastic Gradient (RSG) algorithm in [Ghadimi and Lan, 2013] to achieve privacy and accuracy at the same time. Our contributions can be summarized as follows:

1. In strongly convex case, by choosing suitable learning rate, basic gradient descent with output perturbation not only runs much faster than private SGD [Bassily *et al.*, 2014], but also improves its utility by a logarithmic factor, which matches lower bound in [Bassily *et al.*, 2014] [1]. Besides, we show its generalization performance.
2. We propose a private optimization algorithm for non-convex function, and prove its utility, both in expectation form and high probability form;
3. Numerical experiments show that our algorithms consistently outperform existing approaches.

In the following, we will give a detailed comparison of our results to existing approaches.

[1]Here we only consider the performance in terms of n and d, which is main concern in learning theory, and regard strongly convex parameter μ as a constant.

Comparison with Existing Results: As the closest work to ours is Bassily et al. [Bassily *et al.*, 2014], and their algorithms also match the lower bound in terms of utility, we mainly compare our results with theirs. Results are summarized in Table 1 (Notations are defined in the next section).

From Table 1, we can see that our algorithm significantly improves the running time for strongly convex objectives, and achieves slightly better utility guarantee with a $\log$ factor. For non-convex functions, our result is the first differentially private algorithm with theoretical guarantee in this case, to the best of our knowledge.

2 Preliminaries

In this section, we provide necessary background for our analyses, including differential privacy and basic assumptions in convex optimization.

2.1 Setting

Throughout this paper, we consider differentially private solutions to the following ERM problem:

$$\min_{w\in\mathbb{R}^d} \quad F(w,S) := \frac{1}{n}\sum_{i=1}^{n} f(w,\xi_i)$$

where $S = \{(x_1,y_1),\dots,(x_n,y_n)\}, \xi_i = (x_i,y_i)$ is training set, and $\hat{w} := \arg\min_w F(w,S)$. The loss function f usually satisfies $f \geq 0$ and we use $f(\cdot)$ to represent $f(\cdot,\xi_i)$ for simplicity.

Assumption 1. *$f(\cdot,\xi_i)$ is β-smooth, i.e*

$$|f(u) - f(v) - \langle \nabla f(v), u-v\rangle| \leq \frac{\beta}{2}\|u-v\|^2$$

If, in addition, $f(\cdot,\xi_i)$ is convex, then above equation reduced to

$$f(u) - f(v) - \langle \nabla f(v), u-v\rangle \leq \frac{\beta}{2}\|u-v\|^2$$

Actually, β-smoothness is a common assumption as in [Nesterov, 2013].

2.2 Differential Privacy

Let S be a database containing n data points in data universe $\mathcal{X}$. Then two databases S and S' are said to be neighbors, if $|S| = |S'| = n$, and they differ in exactly one data point. The concept of differential privacy is defined as follows:

	Ours		Bassily et al. [Bassily *et al.*, 2014]	
	Utility	Runtime	Utility	Runtime
μ-S.C., ϵ-DP	$\mathcal{O}(\frac{d^2}{n^2\varepsilon^2})$	$\mathcal{O}(nd\log(\frac{n\varepsilon}{d}))$	$\mathcal{O}(\frac{\log(n)d^2}{n^2\varepsilon^2})$	$\approx \mathcal{O}(n^3d^3\min\{1,\varepsilon n, d\log(dn)\})$
μ-S.C., (ϵ,δ)-DP	$\mathcal{O}(\frac{d\log(1/\delta)}{n^2\varepsilon^2})$	$\mathcal{O}(nd\log(\frac{n\varepsilon}{\sqrt{d\log(\delta)}}))$	$\mathcal{O}(\frac{d\log^3(n/\delta)}{n^2\varepsilon^2})$	$\mathcal{O}(n^2d)$
Nonconvex	$\mathcal{O}(\frac{\sqrt{d}}{n\epsilon}\log\frac{n}{\delta})$	$\mathcal{O}(n^2d)$	NA	

Table 1: Comparison with existing results (S.C. means strongly convex)

Definition 1. *(Differential privacy [Dwork* et al.*, 2006]) A randomized algorithm $\mathcal{A}$ that maps input database into some range $\mathcal{R}$ is said to preserve (ε, δ)-differential privacy, if for all pairs of neighboring databases S, S' and for any subset $A \subset \mathcal{R}$, it holds that*

$$Pr(\mathcal{A}(S) \in A) \leq Pr(\mathcal{A}(S') \in A)e^{\varepsilon} + \delta.$$

In particular, if $\mathcal{A}$ preserves $(\varepsilon, 0)$-differential privacy, we say $\mathcal{A}$ is ε-differentially private.

The core concepte and tools used in DP are sensitivity and Gaussian (or Laplace) mechanism, introduced as follows:

Definition 2. *(L_2-sensitivity) The L_2-sensitivity of a deterministic query $q(\cdot)$ is defined as*

$$\Delta_2(q) = \sup_{S,S'} \|q(S) - q(S')\|_2$$

Similarly, we can defined L_1 sensitivity as $\Delta_1(q) = \sup_{S,S'} \|q(S) - q(S')\|_1$.

Lemma 1. *(Laplace and Gaussian Mechanism [Dwork and Roth, 2014]) Given any function $q : \mathcal{X}^n \to R^k$, the Laplace mechanism is defined as :*

$$\mathcal{M}_L(S, q(\cdot), \epsilon) = q(S) + (Y_1, \ldots, Y_k)$$

where Y_i are i.i.d random variables drawn from $\mathrm{Lap}(\Delta_1(q)/\epsilon)$. This mechanism preserves ε-differential privacy. Similarly, for Gaussian mechanism, each Y_i are $i.i.d$ drawn from $\mathcal{N}(0, \sigma^2)$, and let $\sigma = \sqrt{2\ln(1.25/\delta)}\Delta_2(q)/\epsilon$. Gaussian mechanism preserves (ε, δ)-differential privacy.

3 Main Results

In this section, we present our differentially private algorithms and analyze their utility for strongly convex, general convex and non-convex cases respectively. Because of the limitation of space, we only give proof skecthes of some critical results. For detailed proof, please see the full version of this paper.

3.1 Convex Case

We begin our results with the assumption that each f is μ-strongly convex. Our algorithm is a kind of output perturbation mechanism which is similar to Chaudhuri's [Chaudhuri *et al.*, 2011], but we do not assume an exact minimizer can be accessed. With strong convexity and smoothness, which are the most common assumptions in machine learning, our algorithm runs significantly faster than Bassily et al. [Bassily *et al.*, 2014], and matches their lower bounds for utility. Furthermore, the number of iterations needed in our algorithm is significantly less than previous approaches, making it scalable with large amount of data. From a practical perspective, our algorithm can achieve both ε-DP and (ε, δ)-DP by simply adding Laplacian and Gaussian noise respectively.

As sensitivity serves as an essential technique in the differential privacy analysis, to start with, we will prove the sensitivity of gradient descent based on the idea of Hardt et al. [Hardt *et al.*, 2016]. Let $\Delta_T = \|w_T - w'_T\|_2$ be the L_2-sensitivity of an algorithm, where w_T and w'_T are the variables in T-th round, for two neighboring databases S and S' respectively.

Algorithm 1 Output Perturbation Full Gradient Descent

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, convex loss function $f(\cdot, \cdot)$(with Lipschitz constant L), number of iteration T, privacy parameters (ε, δ), η, Δ, w_0

1: **for** $t = 0$ to $T - 1$ **do**
2: $\quad w_{t+1} := w_t - \frac{\eta}{n}\sum_{i=1}^{n} \nabla f(w_t, \xi_i)$
3: **end for**
4: **if** $\delta = 0$ **then**
5: $\quad$ sample $z \sim P(z) \propto \exp(-\frac{\varepsilon\|z\|_2}{\Delta})$ ▷ (This is for ϵ-DP)
6: **else**
7: $\quad$ sample $z \sim P(z) \propto \exp(-\frac{\varepsilon^2\|z\|_2^2}{4\log(2/\delta)\Delta^2})$ ▷ (This is for (ϵ, δ)-DP)
8: **end if**

Output: $w_{priv} = w_T + z$

Lemma 2. *Assume $f(\cdot)$ is convex, β-smooth and L-Lipschitz. If we run gradient descent(GD) algorithm with constant step size $\eta \leq \frac{1}{\beta}$ for T steps, then the L_2-sensitivity of GD satisfies*

$$\Delta_T \leq \frac{3LT\eta}{n}$$

Proof Sketch. *According te preperties of convex and smooth, one can deduce the following recursion inequalities:*

$$\Delta_{t+1}^2 \leqslant \Delta_t^2 + \frac{4\eta L}{n}\Delta_t + \frac{8\eta^2 L^2}{n^2}$$

Then using an induction argument, one obtain the conclusion.

With the same idea, one can prove following stability results for strongly convex and smooth functions.

Lemma 3. *Assume $f(\cdot)$ is μ-strongly convex, β-smooth and L-Lipschitz. If we run gradient descent(GD) algorithm with constant step size $\eta \leq \frac{1}{\beta+\mu}$ for T steps, then the L_2-sensitivity of GD satisfies*

$$\Delta_T \leq \frac{5L(\mu + \beta)}{n\mu\beta}$$

Theorem 1. *Algorithm 1 is (ε, δ)-differential private for any $\varepsilon > 0$ and $\delta \in [0, 1)$, with concrete setting in below theorems.*

Theorem 2. *If $f(\cdot)$ is μ-strongly convex, β-smooth. Assume $\|\hat{w}\| \leq D$ and $f(\cdot)$ is L-Lipschitz for all $\{w : \|w\| \leq 2D\}$. Let $\eta = \frac{1}{\mu+\beta}$ and $\Delta = \frac{5L(1+\beta/\mu)}{n\beta}$, For w_{priv} output by Algorithm 1, we have the following.*

1. *For ε-differential privacy, if we set $T = \Theta\left(\left[\frac{\mu^2+\beta^2}{\mu\beta}\log(\frac{\mu^2 n^2\varepsilon^2 D^2}{L^2 d^2})\right]\right)$. Then,*

$$\mathbb{E}\, F(w_{priv}, S) - F(\hat{w}, S) \leqslant O\left(\frac{\beta L^2 d^2}{n^2\varepsilon^2\mu^2}\right)$$

2. *For (ε, δ)-differential privacy, if we set $T = \Theta\left(\left[\frac{\mu^2+\beta^2}{\mu\beta}\log(\frac{\mu^2 n^2\varepsilon^2 D^2}{L^2 d\log(1/\delta)})\right]\right)$. Then,*

$$\mathbb{E}\, F(w_{priv}, S) - F(\hat{w}, S) \leqslant O\left(\frac{\beta L^2 d\log(1/\delta)}{n^2\varepsilon^2\mu^2}\right)$$

Proof Sketch. *For ϵ-DP, we have*

$$\begin{aligned}
&\mathbb{E}F(w_{priv}, S) - F(\hat{w}, S) \\
\leq &\mathbb{E}\left[F(w_T, S) + \langle \nabla F(w_T, S), z\rangle + \frac{\beta}{2}\|z\|^2\right] - F(\hat{w}, S) \\
= &(F(w_T, S) - F(\hat{w}, S)) + \frac{\beta}{2}\mathbb{E}\|z\|^2 \\
\leqslant &\frac{\beta}{2}\exp\left(-\frac{2\mu\beta T}{(\mu+\beta)^2}\right)D^2 + \frac{25L^2(\mu+\beta)^2(d+1)d}{n^2\varepsilon^2\mu^2\beta}
\end{aligned}$$

where the last inequality comes from exponential convergence rate of GD and the magnitude of noise, which is closely related to stability results. Thus we obatin desired utility guarantees with above optimal choice of T. The proof of (ϵ, δ)-DP is exactly the same.

It is worth noticing that the results of Bassily et al. [Bassily *et al.*, 2014], hold without smoothness assumption, but their method does not improve too much even with this assumption. This is because they use an SGD-based algorithm, where smoothness could not help in the convergence rate, and where step sizes have to be set conservatively. For strongly convex functions, smoothness assumption is necessary when we use a perturbation-based algorithm. Roughly speaking, a function can become very steep without this assumption, so adding noise to the result of gradient method may cause an unbounded error to the function value.

For the generalization ability of our algorithm, we assume all examples ξ_i are i.i.d drawn from the unknown distribution $\mathcal{D}$, and w^* is the minimizer of population risk $G(w) = \mathbb{E}_\xi f(w, \xi)$. Define excess risk of any w as $\mathrm{ExcessRisk}(w) := G(w) - G(w^*)$. Here we only discuss excess risk of (ε, δ)-differential privacy algorithm, for ε-differential privacy algorithm, the approach is the same.

The most usual technique to obtain excess risk is to use Theorem 5 and inequality (18) in [Shalev-Shwartz *et al.*, 2009]. In this case, we assume loss function $f(w, \xi)$ is μ-strongly convex and L-Lipschitz continuous (w.r.t w) within a ball of radius R, which includes the population minimizer w^*. Thus, by substituting our utility bound in Theorem 2, we can obtain: with probability at least $1-\gamma$, $\mathrm{ExcessRisk}(w_{priv}) \leqslant \tilde{\mathcal{O}}(\frac{L\sqrt{\beta d}}{n\epsilon\mu\gamma})^2$ ($\tilde{\mathcal{O}}$ means we ignore all $\log$ factors). Another method to obtain excess risk is to directly use the relation between the stability of gradient descent and its excess risk, as shown in [Hardt *et al.*, 2016]. Then we have:

$$\begin{aligned}
\mathrm{ExcessRisk}(w_{priv}) &= G(w_{priv}) - G(w_T) + G(w_T) - G(w^*) \\
&\leqslant L\|z\| + \mathrm{Error}_{\mathrm{opt}}(w_T) + L\Delta_T
\end{aligned}$$

where $\mathrm{Error}_{\mathrm{opt}}(w_T)$ represents the empirical optimization error. Note $\|z\|$ term in above inequality can be bounded through tail bound of χ^2 distribution, hence, it will lead to nearly same excess risk bound as the first method.

If we remove the strong convexity property of our loss function, we have the following theoretical guarantee of Algorithm 1.

[2]Note $\frac{1}{\gamma}$ dependence on failure probability γ can be improved to $\log\frac{1}{\gamma}$ by boosting the confidence method used in [Shalev-Shwartz *et al.*, 2010]

Theorem 3. *If $f(\cdot)$ is L-Lipschitz, convex and β-smooth on $\mathbb{R}^d$. Assume $\|\hat{w}\| \leq D$ and let $\eta = \frac{1}{\beta}$ and $\Delta = \frac{3LT}{\beta n}$, then for w_{priv} output by Algorithm 1, we have the following.*

1. *For ε-differential privacy, if we set $T = \Theta\left(\left[\frac{\beta^2 n^2\varepsilon^2 D^2}{L^2 d^2}\right]^{\frac{1}{3}}\right)$, then,*

$$\mathbb{E}\, F(w_{priv}, S) - F(\hat{w}, S) \leqslant O\left(\left[\frac{\sqrt{\beta}Ld\|\hat{w}\|^2}{n\varepsilon}\right]^{\frac{2}{3}}\right)$$

2. *For (ε, δ)-differential privacy, if we set $T = \Theta\left(\left[\frac{\beta^2 n^2\varepsilon^2 D^2}{L^2 d\log(1/\delta)}\right]^{\frac{1}{3}}\right)$ then,*

$$\mathbb{E}F(w_{priv}, S) - F(\hat{w}, S) \leqslant O\left(\left[\frac{L\sqrt{\beta d\log(1/\delta)}\|\hat{w}\|^2}{n\varepsilon}\right]^{\frac{2}{3}}\right)$$

Though the utility guarantee is weaker than Bassily et al. [Bassily *et al.*, 2014] in general convex case by a factor of $O(\frac{1}{\sqrt[3]{n}})$, but when d is smaller than n, then both bounds are below the typical $\tilde{\Theta}(n^{-\frac{1}{2}})$ generalization error in learning theory.[3] So our algorithm does not harm accuracy of machine learning task indeed. Furthermore, compared with [Bassily *et al.*, 2014], our algorithm runs uniformly faster for pure ϵ-DP, and also faster for (ϵ, δ)-DP for high-dimensional problems. This acceleration is mainly due to smoothness of objective function. Moreover, our experimental results show that our algorithm is significantly better than [Bassily *et al.*, 2014] under both convex and strongly convex settings, in the sense that our algorithm not only achieves a lower empirical error but also runs faster than theirs (See Section 4 for more details). As for generalization property for general convex loss, we can solve it along the same road as strongly convex case by adding a regularization term $\frac{\mu}{2}\|w\|_2^2$ (where $\mu = \frac{\sqrt{2}L^{1/2}(\beta d)^{1/4}}{\sqrt{n\epsilon\gamma R}}$). Therefore, in convex case, we can obtain: with probability at least $1-\gamma$, $\mathrm{ExcessRisk}(w_{priv}) \leqslant \tilde{\mathcal{O}}(\frac{RL^{1/2}(\beta d)^{1/4}}{\sqrt{n\epsilon\gamma}})$.

3.2 Nonconvex Case

In this section, we propose a random round private SGD which is similar with private SGD in [Bassily *et al.*, 2014]. We will show that our algorithm can differential privately (we only focus on (ε, δ)-DP this time) find a stationary point in expectation with diminishing error. To the best of our knowledge, this is the first theoretical result about differentially private non-convex optimization problem and this algorithm also achieve same utility bound with [Bassily *et al.*, 2014], which are known to be near optimal for more restrictive convex case. Our algorithm is inspired by the work of Bassily et al. [Bassily *et al.*, 2014] and Ghadimi et al. [Ghadimi and Lan, 2013].

Note our iteration times R satisfies $R \leq n^2$, so the same argument with bassily et al. [Bassily *et al.*, 2014] can be applied

[3]Actually without any other assumption, the performances of almost all private algorithms have polynomial dependence over d, which will hurt generalization error in some degree for large d.

Algorithm 2 Random Round Private Stochastic Gradient Descent

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, loss function $f(\cdot, \cdot)$ (with Lipschitz constant L), privacy parameters $(\varepsilon, \delta)(\delta > 0)$, a probability distribution $\mathbb{P}$ (See distribution setting in the Theorem 5) over $[n^2]$, learning rate $\{\eta_k\}$

1: draw R from $\mathbb{P}$
2: **for** $t = 0$ to $R - 1$ **do**
3: sample $\xi \sim U(S)$
4: sample $z_t \sim \exp(-\frac{\varepsilon^2 \|z\|_2^2}{8L^2 \log(3n/\delta) \log(2/\delta)})$
5: $w_{t+1} := w_t - \eta_t(\nabla f(w_t, \xi) + z_t)$
6: **end for**

Output: $w_{priv} = w_R$

to ensure the DP property of Algorithm 2. The technical details for proofs are deferred to appendix. The utility guarantee mainly comes from the convergence result of SGD (Ghadimi et al. [Ghadimi and Lan, 2013]) under non-convex setting.

Theorem 4. *(Privacy guarantee) Algorithm 2 is (ε, δ) differential private for any $\varepsilon \in (0, 1]$ and $\delta \in (0, 1)$.*

Theorem 5. *(Utility guarantee) If $f(\cdot)$ is L-Lipschitz and β-smooth, and we choose $\mathbb{P}$ which satisfies*

$$\mathbb{P}(k+1) := Pr(R = k+1) = \frac{2\eta_k - \beta\eta_k^2}{\sum_{r=0}^{n^2-1} 2\eta_r - \beta\eta_r^2}$$
$$\textit{for} \quad k = 0, 1, \ldots, n^2 - 1.$$

Assume η_k are chosen such that $\eta_k < \frac{2}{\beta}$. Let $\sigma^2 = 4L^2 + \frac{4dL^2 \log(3n/\delta) \log(2/\delta)}{\varepsilon^2}$, then for w_{priv} output by Algorithm 2, we have the following (the expectation is taken w.r.t $\mathbb{P}$ and ξ_i)

$$\mathbb{E}\|\nabla F(w_{priv}, S)\|^2 \leq \frac{\beta[D_F^2 + \sigma^2 \sum_{k=0}^{n^2-1} \eta_k^2]}{\sum_{r=0}^{n^2-1} 2\eta_r - \beta\eta_r^2}$$

where $D_F = \sqrt{2(F(w_0, S) - F^*)/\beta}$ and F^* is a global minimum of F, note that $F^* \geq 0$ in our settings.

What's more, if we take $\eta_k := \min\{\frac{1}{\beta}, \frac{D_F}{\sigma n}\}$ then we get,

$$\mathbb{E}\|\nabla F(w_{priv}, S)\|^2 = O\left(\frac{\beta L \sqrt{d \log(n/\delta) \log(1/\delta)} D_F}{n\varepsilon}\right)$$

If in addition, $f(\cdot)$ is convex and $\|\hat{w}\| \leq D$, then we have,

$$\mathbb{E}\, F(w_{priv}, S) - F(\hat{w}, S) = O\left(\frac{L \sqrt{d \log(n/\delta) \log(1/\delta)} D}{n\varepsilon}\right)$$

Proof Sketch. *Let $G(w_t) = \nabla f(w_t, \xi) + z_t$. Note that over the randomness of ξ and z_t, we have $\mathbb{E}G(w_t) = \nabla F(w_t, S)$ and $\mathbb{E}\|G(w_t) - \nabla F(w_t, S)\|^2 \leq 4L^2 + \frac{8L^2 \log(3n/\delta) \log(2/\delta)}{\varepsilon^2}$. Thus the theorem holds immediately based on convergence results of [Ghadimi and Lan, 2013].*

As in convex and strongly convex cases, we are using output perturbation to protect privacy, so it is straightforward to obtain high probability version of this bound based on tail bounds for Laplacian and Gaussian distribution. Thus we only consider high probability bounds for non-convex case. The following lemma serves as an important tool for our high-probability analysis.

	n	d	type
BANK	45211	42	classification
ADULT	32561	110	classification
CreditCard	30000	34	classification
WINE	6497	12	regression
BIKE	17379	62	regression

Table 2: Dataset information

Lemma 4. *[Lan* et al.*, 2012] Let $X_1, \ldots, X_T$ be a martingale difference sequence, i.e., $\mathbb{E}_{t-1}[X_t] = 0$ (where $\mathbb{E}_{t-1}[\cdot]$ denotes the expectation conditioned on all the randomness till time $t-1$) for all t. Suppose that for some values σ_t, for $t = 1, 2, \ldots, T$, we have $\mathbb{E}_{t-1}[\exp(\frac{X_t^2}{\sigma_t^2})] \leqslant \exp(1)$. Then with probability at least $1 - \delta$, we have*

$$\sum_{t=1}^{T} X_t \leqslant \sqrt{3 \log(\frac{1}{\delta}) \sum_{t=1}^{T} \sigma_t^2}$$

Now, we can proceed to prove the following theorem about high probability bound.

Theorem 6. *When in the same condition of Theorem 5, by setting $\eta_k := \min\{\frac{1}{\beta}, \frac{D_F}{\sigma n}\}$, then with probability at least $1-\gamma$ (Note this probability is over the noise and the randomness of choosing point in each round), there is*

$$\mathbb{E}\|\nabla F(w_{priv}, S)\|^2 \leqslant O\left(\frac{\sqrt{d \log(1/\gamma) \log(n/\delta) \log(1/\delta)}}{n\varepsilon}\right)$$

4 Experimental Results

To show the effectiveness of our algorithm in real world data, we experimentally compare our algorithm with Bassily et al. [Bassily *et al.*, 2014] for convex and strongly convex loss function. To be more specific, we consider (regularized) logistic regression on 3 UCI [Lichman, 2013] binary classification datasets and (regularized) Huber regression on 2 UCI regression datasets (see Table 2 for more details[4]).

The loss function for logistic regression is $f(w, \xi) = \log(1 + \exp(1 + y\langle w, x\rangle))$. And for Huber regression, the loss function $f(w, \xi; \delta) = h_\delta(\langle w, x\rangle - y)$, where [5]

$$h_\delta(u) = \begin{cases} \frac{1}{2}u^2 & for \quad |u| \leq \delta, \\ \delta(|u| - \frac{1}{2}\delta) & otherwise. \end{cases}$$

All parameters are chosen as stated in theorems in both papers, except that we use a mini-batch version of SGD in

[4]Note all category variables in these datasets are translated into binary features.

[5]For loss functions in above problems, we add an additional square regularization term with parameter μ to make them strongly convex.

Dataset	μ	ε	Error		Runtime(CPU time)	
			ours,(ε,δ)	Bassily,(ε,δ)	ours,(ε,δ)	Bassily,(ε,δ)
BANK	0	0.1	**0.3983**	2.2552	**12.613**	518.67
		0.5	**0.2231**	1.4585	**36.796**	519.33
		1	**0.1459**	1.0203	**58.305**	519.02
		2	**0.0838**	0.7824	**92.501**	518.27
	0.1	0.1	**0.2566**	0.4829	**20.483**	518.03
		0.5	**0.0106**	0.4090	**40.541**	519.44
		1	**0.0025**	0.3387	**49.311**	516.73
		2	**0.0005**	0.2475	**57.947**	520.17
ADULT	0	0.1	**0.0499**	0.6229	**23.813**	250.50
		0.5	**0.0208**	0.6081	**69.536**	254.14
		1	**0.0122**	0.4781	**110.20**	254.18
		2	**0.0065**	0.3691	**175.01**	253.72
	0.1	0.1	**3.2039**	5.2166	**112.09**	256.70
		0.5	**0.1287**	5.1532	**193.98**	255.36
		1	**0.0309**	5.1148	**229.23**	255.69
		2	**0.0080**	5.1009	264.23	**257.23**
CreditCard	0	0.1	**0.0293**	0.4106	**4.9595**	190.30
		0.5	**0.0102**	0.4220	**14.591**	190.89
		1	**0.0053**	0.3140	**22.983**	188.67
		2	**0.0024**	0.2708	**36.721**	188.86
	0.1	0.1	**0.3643**	1.3271	**13.664**	190.36
		0.5	**0.0141**	1.2973	**22.012**	189.97
		1	**0.0035**	1.2792	**25.743**	188.81
		2	**0.0008**	1.2501	**29.256**	187.97
WINE	0	0.1	**0.6061**	6.1755	**0.1672**	6.3859
		0.5	**0.2487**	4.1900	**0.4328**	6.3828
		1	**0.1713**	3.0972	**0.7469**	6.4234
		2	**0.1110**	1.3609	**1.1719**	6.3016
	0.5	0.1	**1.0842**	8.2900	**0.0922**	6.4328
		0.5	**0.0364**	7.9584	**0.1437**	6.3625
		1	**0.0101**	6.5471	**0.1891**	6.5391
		2	**0.0024**	5.3811	**0.1812**	6.4484
BIKE	0	0.1	**5.4659**	35.279	**0.1531**	6.4953
		0.5	**4.0404**	30.822	**0.4375**	6.2375
		1	**3.2768**	27.196	**0.6922**	6.2734
		2	**2.4081**	23.865	**1.1766**	6.3969
	0.5	0.1	**0.0555**	3.0770	**0.1031**	6.5766
		0.5	**0.0301**	3.0448	**0.1578**	6.5094
		1	**0.0242**	2.1792	**0.1625**	6.4094
		2	**0.0232**	1.0406	**0.1984**	6.3625

Table 3: Summary of experimental results

[Bassily *et al.*, 2014] with batch size $m = 50$, since their algorithm in its original version requires prohibitive n^2 time of iterations for real data, which is too slow to run. This conversion is a natural implication of amplification lemma, which preserves the same order of privacy and affects utility with constant ratio. We evaluate the minimization error $\mathbb{E}F(w_{priv}, S) - F(\hat{w}, S)$ and running time of these algorithms under different $\varepsilon = \{0.1, 0.5, 1, 2\}$ and $\delta = 0.001$. The experimental results are averaged over 100 independent rounds. Table 3 illustrates the experimental results of both methods.

From Table 3, we can see our algorithm outperforms existing one on both optimization error and runtime under almost all settings.

5 Conclusion

We study differentially private ERM for smooth loss function under (strongly) convex and non-convex situation. Though output perturbation has been well studied before, our results show that adding noise to approximate solutions instead of exact solutions has important implications to both privacy and running time. Our work is inspired by [Hardt *et al.*, 2016], whose technique for stability analysis of SGD can be applied to deterministic gradient descent algorithms. We show that for strongly convex and smooth objectives, our output perturbation gradient descent achieves optimal utility and runs much faster than the existing private SGD in Bassily et al. [Bassily *et al.*, 2014]. And for general convex objectives, it is also an efficient practical algorithm due to its fast convergence and reasonable utility. From the experimental results, our algorithm achieves lower optimization error and runtime in almost all cases compared to private SGD. For non-convex objectives, by carefully chosen parameters, we show that a random rounds private SGD can reach a stationary point in expectation. This is first theoretical bound for differentially private non-convex optimization to the best of our knowledge.

Acknowledgments

This work was partially supported by National Basic Research Program of China (973 Program) (grant no. 2015CB352502), NSFC (61573026) and the MOE-Microsoft Key Laboratory of Statistics and Machine Learning, Peking University. We would like to thank the anonymous reviewers for their valuable comments on our paper.

References

[Bassily *et al.*, 2014] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 464–473, New York, USA, October 2014. IEEE.

[Chaudhuri *et al.*, 2011] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *The Journal of Machine Learning Research*, 12:1069–1109, March 2011.

[Chaudhuri *et al.*, 2013] Kamalika Chaudhuri, Anand D. Sarwate, and Kaushik Sinha. A near-optimal algorithm for differentially-private principal components. *The Journal of Machine Learning Research*, 14(1):2905–2943, September 2013.

[Dwork and Roth, 2014] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, August 2014.

[Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adame Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography*, pages 265–284, Berlin, Germany, March 2006. Springer.

[Ghadimi and Lan, 2013] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, December 2013.

[Hardt *et al.*, 2016] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1225–1234, Princeton, New Jersey, June 2016. The International Machine Learning Society.

[Jing, 2011] Lei Jing. Differentially private m-estimators. In *Advances in Neural Information Processing Systems*, pages 361–369, California, USA, December 2011. Neural Information Processing Systems Foundation.

[Kifer *et al.*, 2012] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory (COLT)*, 2012.

[Lan *et al.*, 2012] Guanghui Lan, Arkadi Nemirovski, and Alexander Shapiro. Validation analysis of mirror descent stochastic approximation method. *Mathematical programming*, 134(2):425–458, September 2012.

[Lichman, 2013] Moshe Lichman. Uci machine learning repository, 2013.

[Nesterov, 2013] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, Berlin, Germany, 2013.

[Rubinstein *et al.*, 2012] Benjamin IP Rubinstein, Peter L. Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 4(1):65–100, August 2012.

[Shalev-Shwartz *et al.*, 2009] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *Conference on Learning Theory (COLT)*, 2009.

[Shalev-Shwartz *et al.*, 2010] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, October 2010.

[Song *et al.*, 2013] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 245–248, New York, USA, December 2013. IEEE.

[Talwar *et al.*, 2014] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry. *arXiv preprint arXiv:1411.5417*, 2014.

[Talwar *et al.*, 2015] Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, pages 3007–3015, California, USA, December 2015. Neural Information Processing Systems Foundation.

[Wu *et al.*, 2017] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1307–1322. ACM, 2017.