Adaptive Distributed Differential Privacy with SGD

Junhong Cheng,[1] **Wenyan Liu,**[1] **Xiaoling Wang,**[12*] **Xingjian Lu,**[1]
[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
[2] National Shanghai Institute of Intelligent Science and Technology, Tongji University
{jhcheng, wyliu}@stu.ecnu.edu.cn, {xlwang, xjlu}@cs.ecnu.edu.cn

Abstract

Privacy leakage is an important issue for machine learning. Existing privacy-preserving approaches with differential privacy usually allow the server to fully control users' information. This may be problematic since the server itself may be untrusted, leading to serious privacy leakage. Besides, existing approaches need to choose a fixed number of iterations, so that the total privacy budget is finite. Fine-tuning is a common practice, which needs a lot of opportunities to try. However it is not allowed in the actual environment, because multiple access to user data will bring serious privacy risks. In this paper, we aim to address the problem of achieving privacy-preserving with distributed differential privacy. In this scenario, different from the traditional need to disclose some local data to the centralized server, each participant keeps the data locally, to achieve better privacy protection effect. We propose a novel algorithm for privacy-preserving training with adjustable iteration steps by sampling techniques. The validity of the algorithm is verified by theoretical analysis and experimental evaluation on real-world datasets.

Introduction

In today's intelligent age, distributed data processing and data mining are playing an increasingly important role. The reason is that in recent years, more and more application scenarios involving multiple participants have emerged, such as financial data scattered in multiple banks, medical records in different hospitals, behavior records of each user under a large platform (Pentland 2012) and smart meters, sensors or mobile devices (Rial and Danezis 2011). It is worth noting that these data are usually private or sensitive. In this case, it is of great research value to design an algorithm that can learn a good generalization performance model under the condition of protecting the privacy of all data points.

At the same time, with the increasing attention to privacy in recent years, the technology of satisfying differential privacy(DP) in local environment (Pathak, Rane, and Raj 2010) has been widely studied and applied. Most of the early work on DP is centralized. In centralized scenarios, data managers obtain data from individual users and process data in a way that protects the privacy of individual users. Therefore, data managers must be assumed to be secure and trustworthy. For example, data managers can publish histogram statistics of collected data in order to analyze the data while avoiding the leak of users' personal information. In localized differential privacy scenarios, there are many users and an aggregator. Unlike centralized scenarios, (1) central data managers are no longer trusted, replaced by data aggregators, (2) aggregators do not really see the actual privacy data of each individual, and the individual always keeps the data locally. Such technology allows us to collect information while protecting the privacy of each user, without relying on the data manager's safe and trustworthy assumptions. Companies have adopted localized Differential Privacy like Google (Erlingsson, Pihur, and Korolova 2014) and Apple (Avent et al. 2017). Typical application cases include collecting user web browsing information and user typing habits information.

In prior work with differential privacy on machine learning (Bassily, Smith, and Thakurta 2014), the total number of iterations T must be fixed in advance, because privacy cost ε should be split across the iterations. For any iteration t, the variance of σ_t is a function of $1/\varepsilon_t$ and depends on which version of differential privacy is being used, such as Gaussian Mechanism (Dwork and Roth 2014).

There are some drawbacks to this approach. First, accuracy heavily depends on the pre-specified number of iterations T: if T is too small, the algorithm will stop well short of the optimum; otherwise, large amounts of noise must be added to each gradient. Secondly, a composition that sums the privacy budget simply will result in a waste of privacy budget, and lead to a decline in the accuracy of the model.

Moreover, in the experimental environment, we can adjust the parameters to select a relatively appropriate parameter. But in the real world, every access to the user's data consumes a privacy budget, which allows us to set parameters at a very shallow fault tolerance rate.

To solve these problems, we propose A-DDP algorithm which extends iteration to improve accuracy. Our contributions are summarized in the following:

*Corresponding author. This work was supported by National Key R&D Program of China (No. 2017YFC0803700), NSFC grants (No. 61532021 and 61972155), Shanghai Knowledge Service Platform Project (No. ZF1213).

- We propose a gradient-based algorithm with distributed differential privacy in which try to expand the number of iteration instead of the fixed. Here the Analytic Gaussian Mechanism is used with liner search method, to broaden the definition domain of ε and obtain a relatively compact variance bound.
- The Sampling Composition Theorem under Gaussian mechanism are proposed and proved to be superior to the existing Advanced Composition Theorem in theory and experiments.
- Extensive experiments on real datasets against other recently proposed stochastic gradient decent algorithms with differential privacy empirically show the effectiveness of the proposed algorithm for a wide range of privacy levels.

Related Work

Machine learning with multi-participation is a hot topic in recent years (Shin et al. 2018). (Rajkumar and Agarwal 2012) taking advantage of the fact that the gradient of the summation equals the summation of the gradient, the problem is directly decomposed into multiple forms i.e., $grad(t) = \sum_u^U grad_u(t)$. Therefore, each party only needs to pass gradient information on each round, and then aggregate it into a whole gradient.

At present, there are many works on differential private machine learning. There are two main types of methods: output perturbation (Kifer, Smith, and Thakurta 2012; Jain, Kothari, and Thakurta 2012), target perturbation (Hua, Xia, and Zhong 2015). Output perturbation is the addition of perturbations to the results, in which noise is added to the output of the underlying deterministic algorithm. The form of added noise depends on the sensitivity of the underlying algorithm, which measures the maximum change in the output of the algorithm when an element in the input data set changes. Simply put, the principle of output perturbation is to find the exact convex minimizer and then add noise; the objective perturbation is to add noise to the objective function. Unlike the output perturbation, this is not a random disturbance classifier, but a disturbance to the objective of algorithm minimization. In general, output perturbation is often inaccurate because the noise is calibrated to the worst-case analysis. Although objective perturbation is more effective, its privacy guarantee is based on the premise that the problem can be solved exactly.

In recent year, an approach that has gained popularity is the iterative gradient perturbation method (Bassily, Smith, and Thakurta 2014). (Bassily, Smith, and Thakurta 2014) proposed an (ε, δ)-differentially private version of stochastic gradient descent (SGD) algorithm with the advanced composition and privacy amplification (Beimel et al. 2014). However, all of them need a fixed number in advance. By comparison, when iteration number T is chosen large, we inject less noise, and when iteration number T is chosen small, extra iteration is allowed.

Preliminary

Differential Privacy

(ε, δ)-Differential Privacy (Dwork et al. 2006) is defined as following: A random mechanism $M{:}D \to R^n$ satisfies (ε, δ)-Differential privacy, if and only if for any two adjacent databases $d, d' \in D$ and for any subset of outputs $S \subseteq R^n$ it holds that

$$Pr[M(d) \in S] \leq e^{\varepsilon} Pr[M(d') \in S] + \delta \tag{1}$$

When $\delta = 0$, M achieves pure differential privacy which provides stronger privacy protection. Introduced by (Dwork and Roth 2014), (ε, δ)-differential privacy allows for the possibility that the definition of ε-differential privacy is broken with probability at most δ. Furthermore, it is considered that δ is preferably smaller than $1/d$ (Abadi et al. 2016).

(L_2) sensitivity is defined as following: Let $f : D \to R^n$, Δ is defined as

$$\Delta = \max_{D,D'} \left\| f(D) - f(D') \right\|_2 \tag{2}$$

The sensitivity describes the maximum change in the output value of f, and the higher sensitivity will result in the larger variance of the injection noise.

Differential privacy has several properties that make it particularly useful in applications such as composability, and robustness to auxiliary information. Robustness to auxiliary information means that privacy guarantees are not affected by any side information available to the adversary.

This is an important property because its privacy guarantee degrades gracefully under the composition, which enables the modular design of mechanisms. Because it's widely used, an advanced composition is introduced, where the loss increases sublinearly.

Advanced Composition Theorems (Dwork and Roth 2014) shows that: Let M_i be an (ε, δ)-Differentially Private algorithm. Then for $M_{[k]} = (M_1, M_2, ..., M_k)$, then $M_{[k]}$ is $(\varepsilon', k\delta + \delta')$-Differentially Private, where $\varepsilon' = \sqrt{2k\ln(1/\delta')}\varepsilon + k\varepsilon(e^{\varepsilon} - 1)$.

Frequently-used mechanisms to achieve differential privacy are Laplace Mechanism, Gaussian Mechanism and their variants. The Gaussian Mechanism (Dwork and Roth 2014): Let $\varepsilon \in (0, 1)$, $\delta > 0$, given any function $f : D \to R^n$, the Gaussian mechanism is defined as:

$$M = f + N(0, \sigma^2 I), \tag{3}$$

where $\sigma = \frac{\Delta}{\varepsilon}\sqrt{2\ln(1.25/\delta)}$.

Analytic Gaussian Mechanism (Shin et al. 2018) is defined as following: Let $f{:}D \to R^n$ be a function with global l_2 sensitivity Δ. For any $\varepsilon > 0$ and $\delta \in (0, 1)$, the Gaussian output perturbation mechanism $M(d) = f(d) + Z$ with $Z \sim N(0, \sigma^2 I)$ is (ε, δ)-Differential Privacy if and only if

$$\Phi(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}) - e^{\varepsilon}\Phi(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}) \leq \delta \tag{4}$$

This mechanism aims to avoid the suboptimal result due to the slack in these tail bounds in the non-asymptotic regime. Furthermore, a strategy is proposed to find σ using a numerical algorithm by leveraging the fact that the Gaussian CDF can be written as $\Phi(t) = (1 + erf(\frac{t}{\sqrt{2}}))/2$, where erf is the standard error function and Φ is Gaussian Cumulative Distribution Function.

Stochastic Gradient Descent

Stochastic gradient descent (SGD), also known as incremental gradient descent, is an iterative method for optimizing differentiable objective functions. The method iteratively updates weights and biases by calculating the gradient of loss function on small batch data. SGD goes far beyond the naive gradient descent method on the highly nonconvex loss surface, which has dominated most modern machine learning algorithms. In this algorithm, at each step, one forms a batch B of random examples and computes $g_B = 1/|B| \sum_{x \in B} \bigtriangledown_\Theta L(\Theta, x)$ as an estimation to the gradient $\bigtriangledown_\Theta L(\Theta)$. Then θ is updated following the gradient direction $-g_B$ towards a local minimum.

Adaptive Distributed Differential Privacy

We will describe our method. Algorithm 1 shows the each step of the proposed extended differentially private stochastic gradient descent algorithm. Firstly, we compute eps_iter and δ_iter with Advanced Composition(lines 5). Each user compute their privacy δ_u with Sampling Composition. If $\delta_u > \delta$, user will terminate and do not upload his gradient information(lines 7-10). Otherwise, user will compute gradient with a random sample(lines 11-12). Then clip the gradient and inject proper noise into gradient with Analytic Gaussian Mechanism. Finally, server average users' gradient and update model parameter w. The algorithm has five main components: distributed differential privacy, gradient norm clip, the amplification effect of sampling, privacy budget accumulation, and Adaptive iteration steps method.

Algorithm 1: Adaptive Distributed Differential Privacy with SGD

Input : $Users$, privacy budget ε_total and δ_total, estimated number of iterations T,learning rate α, clipping thresholds C, objective loss function l, Analytic Gaussian Mechanism $noise(\Delta, \varepsilon, \delta)$

Output: model parameter w

1 Initialize w;
2 **while** $\exists \delta_u < \delta$ **do**
3 $n = 0$;
4 $grad = 0$;
5 compute eps_iter, δ_iter;
6 **for** *each* $u \in Users$ **do**
7 compute δ_u;
8 **if** $\delta_u > \delta$ **then**
9 continue;
10 **end**
11 sample x from $Users.data$;
12 $gt_u = \bigtriangledown l(w, x)$;
13 $gt_u = gt_u / \max(1, \frac{\|gt_u\|}{C})$;
14 $grad += gt_u + noise(C, eps\_iter, \delta\_iter)$;
15 $n++$;
16 **end**
17 $w = w - \alpha * grad/n$;
18 **end**
19 **return** w;

Distributed Differential Privacy

Unlike traditional machine learning algorithms, each iteration here asks the user for the current gradient information. Then the user injects artificial noise into the calculated results and returns them to the server. Unlike centralization, the gradient submitted by the user is the result, so the output perturbation is used here, not the target perturbation.

Furthermore, in the traditional centralized privacy protection scenario, data managers tend to give each user data the same privacy budget, which ignores the differences between people. Some people want to have better privacy protection, while others are not sensitive to the privacy of some data. In this scenario, because of a distributed architecture, from a large database to many small databases, so for each user, they need to care only about their own privacy. They can set up different privacy budget scenarios, instead of the traditional uniform allocation, and then inject noise in the worst case. So we need to inject less overall noise.

Noise Reduction with Larger Batch Size

Like the ordinary SGD algorithm, Algorithm 1 estimates the gradient by computing the gradient of the loss on a batch of examples and taking the average. This average provides an unbiased estimator, and the variance decreases with the size of the batch, whether it's artificially injected noise or noise from the stochastic gradient method. Therefore, we expand the batch size of each iteration to reduce the noise content in each iteration and make the gradient information more accurate. Here, the maximum batch size is the number of users. Therefore, our algorithm allows each user to participate in each iteration. It is noted that the large batch size increases computational complexity, but there is no such concern. In the distributed differential privacy scenario, users leave data locally. In order to achieve better privacy protection, users need to bear some computational costs when computing their gradients. For servers, their job is simply to aggregate all the gradients users upload and update the parameters. Therefore, a huge number of users will not cause too much computation burden on the server. In addition, the large batch size can also speed up convergence, thus decreasing T, thereby reducing the consumption of privacy budget.

Gradient Norm Clip

At each iteration, the algorithm computes the noisy gradient $g = \bigtriangledown f + N(0, \sigma^2 I)$ using the Gaussian mechanism with variance σ^2. The magnitude of noise σ^2 is dependent on the maximum influence one individual can have on g in l_2 norm, that is Δ. Since there is no a priori bound on the size of the gradients, we clip each gradient in l_2 norm. So the gradient vector g is replaced by $g = g/\max(1, \frac{\|g\|}{C})$, for a clipping threshold C. This clipping ensures that if $\|g\| \leq C$, then g is preserved, whereas if $\|g\| > C$, it gets scaled down to be of norm C. We remark that gradient clipping of this form is a popular ingredient of SGD.

The Amplification Effect of Sampling

Much research has been devoted to studying the privacy loss for a particular noise distribution as well as the composi-

tion of privacy losses. For the Analytic Gaussian Mechanism noise that we use, each step is (ε, δ)-differentially privacy. Since the batch itself is a random sample from the database, privacy amplification theorem (Beimel et al. 2014) implies that each step is actually $(\varepsilon p, \delta p)$-differentially private with respect to the full database where p is the sampling ratio.

Why does sampling produce privacy amplification effect? Intuitively, due to the uncertainty of sampling, the whole process will be divided into two kinds of possibilities, being drawn and not being drawn. If it is drawn, then the original privacy guarantee is still valid; if it is not drawn, because of the two adjacent databases, then their outputs will be the same, there will be no privacy disclosure. Therefore, the result distribution distance in adjacent databases will be smaller, thus satisfying the stricter definition of privacy.

Privacy Budget Accumulation

According to the Composition Theorem, the privacy risk of the data queried n times will increase n times. Therefore, we want fewer queries, the better, at least bounded. In the experimental environment, we can determine a relatively ideal number of iterations by several attempts and then distribute the privacy budget equally in each iteration. However, it is difficult to choose the number of iterations in practice because any attempt will add additional privacy risk. When the number is too small, the pre-fitting will occur resulting in poor performance; if the number is too large, the injection noise will be too large, which will affect the accuracy of the model. Another attempt is to use a sequence of equal-ratio increments of injection noise so that no matter how many iterations we have, we can find a finite privacy budget. However, although this method can satisfying ε-differential privacy under infinite sequence with finite ε, it will become meaningless because the gradient information will be completely submerged by noise in the later stage.

Adaptive Iteration Number Method

Gaussian Mechanism Now we analyze the composition problem with the Gauss mechanism. The original Gauss mechanism has a δ parameter, a relaxed version of differential privacy. (Balle and Wang 2018) shows that the traditional Gaussian mechanism is inadequate, and the parameter ε definition domain can only be in (0,1), not R like Laplace Mechanism. Besides, it also proves that the Gaussian Mechanism has the space to improve the noise scale. In order to draw the following conclusions, we can rewrite the form of differential privacy.

Given a vector-valued mechanism $M{:}D \to R^n$. The privacy loss function of M on a pair of neighbouring databases d, d' is defined as

$$L_{M,d,d'}(S) = \frac{Pr[M(d) \in S]}{Pr[M(d') \in S]} \tag{5}$$

It is well-known that the privacy loss random variable is also Gaussian (Dwork and Roth 2014).

(Balle and Wang 2018) shows that the privacy loss $L_{M,d,d'}$ of a Gaussian output perturbation mechanism follows a distribution $N(\eta, 2\eta)$ with $\eta = H^2/2\sigma^2$, where $H = \|f(d) - f(d')\|$.

Privacy loss obeys the Gaussian distribution, which means that we can analyze the composition properties directly at the level of privacy loss, because the sum of Gaussian distribution is also Gaussian distribution.

Theorem 1. *Gaussian Sampling Composition Theorem: Let M_i be an $(\varepsilon_i, \delta_i)$-Differentially Private algorithm. Then for $M_{[k]} = (M_1, M_2, ..., M_k)$, then $M_{[k]}$ is (ε, δ)-Differentially Private, where*

$$\delta = \sum_{i=1}^{k} B(i,k,p)[\Phi(\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}}) - e^{\varepsilon}\Phi(-\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}})]$$

Proof. We apply Bayes Rule to compute δ. So, we can get

$$\delta = \sum_{i=1}^{k} B(i,k,p)[Pr[L_{d,d'} * i > \varepsilon] - e^{\varepsilon} Pr[L_{d',d} * i < -\varepsilon]]$$

Note that the privacy loss random variables $L_{M,d,d'}$ and $L_{M,d',d}$ both follow the same distribution $N(\eta, 2\eta)$ with $\eta = H^2/2\sigma^2$. This allows us to write privacy loss in terms of the Gaussian CDF as follows:

$$\begin{aligned} &Pr[L_{d,d'} * i > \varepsilon] = Pr[N(\eta i, 2\eta i) > \varepsilon] \\ =&Pr[N(0,1) > \frac{-\eta i + \varepsilon}{\sqrt{2\eta i}}] = Pr[N(0,1) < \frac{\eta i - \varepsilon}{\sqrt{2\eta i}}] \\ =&Pr[N(0,1) < \sqrt{\frac{\eta i}{2}} - \frac{\varepsilon\sigma}{\sqrt{2\eta i}}] = \Phi(\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}}) \end{aligned}$$

$$Pr[L_{d',d} * i > \varepsilon] = Pr[N(\eta i, 2\eta i) < -\varepsilon] = \Phi(-\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}})$$

So, we can get

$$\delta = \sum_{i=1}^{k} B(i,k,p)[\Phi(\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}}) - e^{\varepsilon}\Phi(-\frac{H\sqrt{i}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{i}})]$$

□

To compare with the former works, we try to give a lower bound with respect to fixed T. Here, we denote σ as the smallest noise scale to achieve (ε, δ)-differential privacy. Because of $\Phi(\frac{H\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{T}}) - e^{\varepsilon}\Phi(-\frac{H\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{T}})$ is increasing with respect to T and decreasing with respect to σ. So we can get

$$\begin{aligned} &\sum_{i=1}^{T} B(i,T,p)[Pr[L_{d,d'} * i > \varepsilon] - e^{\varepsilon} Pr[L_{d',d} * i < -\varepsilon]] \\ \le&\sum_{i=1}^{T} B(i,T,p)[Pr[L_{d,d'} * T > \varepsilon] - e^{\varepsilon} Pr[L_{d',d} * T < -\varepsilon]] \\ \le&Pr[L_{d,d'} * T > \varepsilon] - e^{\varepsilon} Pr[L_{d',d} * T < -\varepsilon] \end{aligned}$$

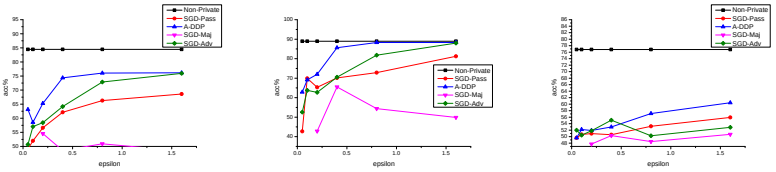
So with the same δ, $\sigma_1 \le \sigma_2 \le \sigma_3$.

$$\begin{aligned} &Pr[L_{d,d'} * T > \varepsilon] - e^{\varepsilon} Pr[L_{d',d} * T < -\varepsilon] \\ &= \Phi(\frac{H\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{T}}) - e^{\varepsilon}\Phi(-\frac{H\sqrt{T}}{2\sigma} - \frac{\varepsilon\sigma}{H\sqrt{T}}) \le \delta \end{aligned}$$

Define $\sigma = \alpha H\sqrt{T}/\sqrt{2\epsilon}$ and substitute the σ,we can get

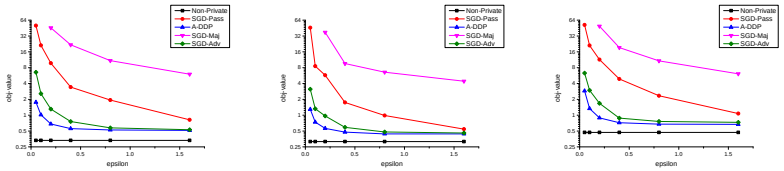
$$\Phi(\sqrt{\epsilon/2}(1/\alpha - \alpha)) - e^{\varepsilon}\Phi(-\sqrt{\epsilon/2}(1/\alpha + \alpha)) \le \delta$$

Now, it has nothing to do with T. The remaining α can be derived as (Shin et al. 2018). So $\sigma_1 \le \sigma_3 = O(\sqrt{T})$, compared with advanced composition $\sigma = O(\sqrt{T\log(T)})$, it can save $\sqrt{\log(T)}$ and (Abadi et al. 2016) $\sigma = O(\sqrt{T})$, but it is central.



(a) Adult (b) Bank (c) IPUMS

Figure 1: The accurate by varying epsilon



(a) Adult (b) Bank (c) IPUMS

Figure 2: The object value by varying epsilon

Table 1: Characteristics of datasets

Datasets	Size	Dime.	Label
Adult	48,842	124	income $>$ 50k
BANK	45,211	33	product subscribed
IPUMS-BR	38,000	53	income $>$ \$300

Experiment

Datasets

We evaluate the performance of our algorithm on 3 real datasets: (i) Adult dataset contains 48,842 records of individuals from 1994 US Census. (ii) BANK contains marketing campaign related information about customers of a Portuguese banking institution. (iii) IPUMS-BR is also Census data extracted from IPUMS-International, which contains 38,000 records. Data sets have been pre-processed by (Lee and Kifer 2018). Table 1 summarizes the characteristics of datasets used in our experiments.

Contrast algorithms

We compare four baseline algorithms:

- Non-Private: a non-private SGD algorithm.
- SGD-Maj: a differentially private version of SGD algorithm that noise is injected into gradient violently.
- SGD-Adv (Bassily, Smith, and Thakurta 2014): a differentially private SGD algorithm with advanced composition theorem and privacy amplification.
- SGD-Pass: a differentially private SGD algorithm that each sample is visited T times , trying to reduce the waste of privacy budgets in this way instead of benefit from sampling.
- A-DDP: a differentially private version of SGD algorithm we propose.

We report both classification accuracy and final objective value . All the reported numbers are averaged over 5-fold cross-validation.

Logistic Regression

We have experimented on logistic regression problems with high generalization.

$$\min_w \frac{1}{n}\sum_{i=1}^{n} log(1+exp(-y_i w^T x_i))$$

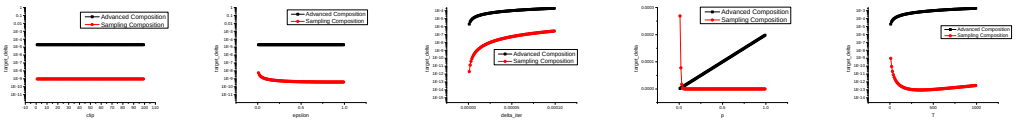
where $y_i \in \{-1, 1\}$.

When there are known default parameter settings for the prior works, we used the same settings. We set the value of parameter $\delta = 1e-8$ and $clip = 3$ for all three datasets. We evenly assigned 100 samples to each user, so the parameter $p = 0.01$. We set T a large number for non-private, while set parameter $T = 10$ for private method.

Figure 1 shows the classification accuracies of the logistic regression model on three different datasets. It can be seen from the experiments that A-DDP are far superior to other methods in a wide range of ε. This is because they benefit from the nature of sampling, which reduces the waste of ε or additional training rounds are allowed. On the other hand, Analytic Gaussian Mechanism gives a relatively compact lower bound on σ. We can get better gradient information under the same privacy settings.

In all of the baseline algorithms, the value of T needs to be determined before the execution of the algorithms. They may need a lot of opportunities to try and error when tuning, which is not allowed in the actual environment because multiple access to user data will bring serious privacy risks. Besides, they need to control the parameters ε of each iteration between (0,1) to satisfy the definition of the Gaussian mechanism, which is completely not limited in our algorithm.

Figure 2 illustrates how the final objective value achieved by each algorithm changes as the value of ε increases. As it



(a) Effect of clip (b) Effect of epsilon (c) Effect of delta (d) Effect of p (e) Effect of T

Figure 3: Effects of parameters

was observed in the experiments on classification accuracy, A-DDP get close to the best achievable objective values for a wide range of ε values considered in the experiments.

Moreover, Figure 1 and Figure 2 show that the A-DDP perform better especially when the parameter ε is small. This is because the noise variance of Gauss Mechanism is $\propto 1/\varepsilon$, instead of $\propto 1/\varepsilon^2$. This means that compared with other medthed, A-DDP is more suitable for dealing with such high-privacy scenarios with lower noise scale.

Sampling Composition and Advanced Composition

Now, we compare the Sampling Composition Theorem with the Advanced Composition Theorem. The experiment is compared by the method of controlling variables. In general, we set $p = 0.1$, $clip = 3$, $\delta = 1e-5$, $\varepsilon = 0.1$ and $T = 10$. On the whole, we first determine a pair of privacy parameter ε and δ, then use the Advanced Composition Theorem to get the total privacy parameter ε, and finally use the Sampling Composition Theorem to get its δ value. We compare the two theorems by comparing the magnitude of δ.

It can be seen from the Figure 3 that in most cases the sampling composition theorem is much better than the advanced composition theorem because the advanced composition theorem has room for optimization . It is interesting that when the probability of sampling becomes extremely small, Advanced Composition perform better. There may be a few samples in each user in the real world.

Conclusion

This paper has developed an iterative optimization algorithm for distributed differential privacy, in which the number of iteration is not be restricted to a fixed number without prior knowledge in general. Existing private algorithms need a number of iteration to be chosen before training, which is very difficult in the real world. To address this significant drawback, we presented a general algorithm to extend the iteration number and proposed Sampling Composition Theorem. The validity of the algorithm is verified by theoretical analysis and experimental evaluation on real-world datasets.

References

Abadi, M.; Chu, A.; Goodfellow, I. J.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *SIGSAC*, 308–318.

Avent, B.; Korolova, A.; Zeber, D.; Hovden, T.; and Livshits, B. 2017. BLENDER: enabling local search with a hybrid differential privacy model. In *USENIX Security*, 747–764.

Balle, B., and Wang, Y. 2018. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *ICML*, 403–412.

Bassily, R.; Smith, A. D.; and Thakurta, A. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, 464–473.

Beimel, A.; Brenner, H.; Kasiviswanathan, S. P.; and Nissim, K. 2014. Bounds on the sample complexity for private learning and private data release. *Machine Learning* 94(3):401–437.

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

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. D. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*, 265–284.

Erlingsson, Ú.; Pihur, V.; and Korolova, A. 2014. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *CCS*, 1054–1067.

Hua, J.; Xia, C.; and Zhong, S. 2015. Differentially private matrix factorization. In *IJCAI*, 1763–1770.

Jain, P.; Kothari, P.; and Thakurta, A. 2012. Differentially private online learning. In *COLT*, 24.1–24.34.

Kifer, D.; Smith, A. D.; and Thakurta, A. 2012. Private convex optimization for empirical risk minimization with applications to high-dimensional regression. In *COLT*, 25.1–25.40.

Lee, J., and Kifer, D. 2018. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *SIGKDD*, 1656–1665.

Pathak, M. A.; Rane, S.; and Raj, B. 2010. Multiparty differential privacy via aggregation of locally trained classifiers. In *NIPS*, 1876–1884.

Pentland, A. 2012. Society's nervous system: Building effective government, energy, and public health systems. *IEEE Computer* 45(1):31–38.

Rajkumar, A., and Agarwal, S. 2012. A differentially private stochastic gradient descent algorithm for multiparty classification. In *AISTATS*, 933–941.

Rial, A., and Danezis, G. 2011. Privacy-preserving smart metering. In *WPES*, 49–60.

Shin, H.; Kim, S.; Shin, J.; and Xiao, X. 2018. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Trans. Knowl. Data Eng.* 30(9):1770–1782.