A Stochastic Gradient Descent Algorithm Based on Adaptive Differential Privacy

Yupeng Deng[1], Xiong Li[2(✉)], Jiabei He[3], Yuzhen Liu[1], and Wei Liang[1]

[1] School of Computer Science and Engineering,
Hunan University of Science and Technology, Xiangtan 411201, China
wliang@hnust.edu.cn

[2] School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu 611731, China
lixiong@uestc.edu.cn

[3] College of Computer Science, Nankai University, Tianjin 300071, China
hejiabei@mail.nankai.edu.cn

Abstract. The application of differential privacy (DP) in federated learning can effectively protect users' privacy from inference attacks. However, privacy budget allocation strategies in most DP schemes not only fail to be applied in complex scenarios but also severely damage the model usability. This paper designs a stochastic gradient descent algorithm based on adaptive DP, which allocates a suitable privacy budget for each iteration according to the tendency of the noise gradients. As the model parameters keep optimizing, the scheme adaptively controls the noise scale to match the decreased gradients, resizing the allocated privacy budget when too small. Compared with other DP schemes, our scheme flexibly reduces the negative effect of added noise on model convergence and consequently improves the training efficiency of the model. We implemented the scheme on five datasets (Adult, BANK, etc.) with three models (SVM, CNN, etc.) and compared it with other popular schemes in the classification accuracy and training time. Our scheme proved to be efficient and practical, which achieved 2% better than the second one in model accuracy, costing merely 4% of its training time with the 0.05 privacy budget.

Keywords: Differential privacy · Stochastic gradient descent · Empirical risk minimization · Machine learning

1 Introduction

In recent years, big data-driven artificial intelligence (AI) has burst into great potential, accomplishing large-scale complex task learning in many fields such as finance, medical care, urban planning, and autonomous driving. Machine learning, as the core technology of AI, is also widely concerned about its performance and privacy. Traditional machine learning requires centralized training by service providers after collecting user data. However, user data is closely related

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 461, pp. 133–152, 2022.
https://doi.org/10.1007/978-3-031-24386-8_8

to the individual user and may contain sensitive information such as personal age, race, and disease. As privacy concerns grow, users are less willing to share their data. Paradoxically, AI technologies must rely on large amounts of data collection and fusion. Without access to the unlimited wealth of information to train models and develop technologies, the development of AI applications will be severely limited.

Federated learning (FL) emerged in the context of the contradiction between data silos and the need for data fusion. In 2017, Google first introduced the concept of federated learning [23], in which multiple data holders (e.g., cell phones, IoT devices, or financial or medical institutions) collaborate to train models without directly sharing data and only exchange training parameters in the intermediate stage. Ideally, federated learning results in a shared model similar to or better than a model trained on a central server with the universal set of all users' data [22]. As a result, companies can fuse data to extract information legally and efficiently. At the same time, individuals or other data-holding organizations can still enjoy the AI services companies provide while retaining data control.

For the sake of dealing with federated learning in data-rich scenarios, the stochastic gradient descent (SGD) algorithm is exploited as a very efficient method due to its ability to extend to the parallel mode in computation. Initially, SGD accesses all data records of a dataset in random order and updates the corresponding model approximation by the local gradient of the loss function associated with each record. Due to its tractability and scalability, SGD has become a method efficient of choice for large-scale data training [5]. To improve the convergence efficiency, some researchers have proposed the mini-batch SGD algorithm [10,11]. It uses partial samples to calculate the gradient instead of a single sample or all samples during the training iteration. Compared with the randomized algorithm, the mini-batch SGD algorithm uses multiple samples to calculate the gradient, which can reduce the variance of the randomized gradient and thus improve the convergence speed of the algorithm to some extent.

Although federated learning achieves natural protection for data privacy by avoiding direct data exposure to third parties, it's still facing three significant risks of privacy leakage: First, federated learning requires the exchange of intermediate parameters for collaborative training, which may compromise privacy. Second, unreliable participants exacerbate the risk of the privacy breach. Finally, attackers can steal user privacy by inferring the user data through the trained model.

To reduce the risk of privacy leakage from passing intermediate parameters, the differential privacy mechanism is introduced. In 2006, Dwork et al. [13,14] proposed differential privacy, which is a practical approach to guarantee a quantifiable level of privacy. Among them, the ϵ-differential privacy model has received extensive research attention. It ensures privacy in theory and is robust to known attacks (e.g., those involving auxiliary information) [16], while implementation requires perturbing the data by adding noise.

Transforming the gradient in the training process can protect data privacy and security but it meanwhile reduces the usability of the model. In earlier work on differential privacy SGD [8,18,25,31], the learning rate was set to a constant

and the privacy budget was equally distributed to each iteration. For example, suppose the budget is ϵ and the maximum number of iterations is T, then the privacy budget allocated to each iteration is $\epsilon_t = \frac{\epsilon}{T}$, $(t = 1, 2, \ldots, T)$. Under this privacy budget allocation scheme, the convergence is strictly limited by the preselection parameter T. If T is too small, the optimal solution has not been obtained at the end of the iterative process; if T is too large, it means that the privacy budget allocated to each iteration is small and the corresponding interference noise added to the gradient is large, and the accuracy cannot be guaranteed.

The optimal problem of differential privacy SGD iterations in the machine learning training process is the empirical risk minimization (ERM): Let $D = \{d_1, \ldots, d_n\}$ be an input database of n independent observations. Each observation $d_i = (x_i, y_i)$ is composed of $x_i \in R_p$ and $y_i \in R$. The ERM problem can be expressed in the following form: $\min_{w \in C} f(w; D) := \frac{1}{n} \sum_{i=1}^{n} L(w, d_i) + \frac{\lambda}{2} \|w\|_2{}^2$, where L is a loss function and C is a convex set. Note that the regularization term $\frac{\lambda}{2} \|w\|_2{}^2$ does not affect the privacy guarantee since it is independent of the data.

To solve the private ERM problem, a large number of works have conducted relevant research on privacy budget allocation strategies. Abadi et al. [1] proposed a new mechanism for tracking privacy loss, called moments accountant, which permits tight automated analysis of the privacy loss of complex composite mechanisms and significantly improves the accuracy of the model under a suitable privacy budget. Lee et al. [21] designed an adaptive privacy budget allocation scheme to assign appropriate privacy budgets for each iteration separately to satisfy zero-concentrated differential privacy (zCDP) while improving the accuracy of the model. However, both methods (especially [21]) require more computational overhead to enhance the usability of the model compared to earlier work. Also, in scenarios with small privacy budgets, the scheme of Abadi et al. [1] cannot work due to its mechanism.

It is not hard to discover two critical points in solving the private ERM problem: proper privacy budget strategies and criteria to judge the effectiveness of the noise gradient. Wang et al. [28] proposed a new differential privacy budget allocation scheme in the tree index. The privacy budget allocated during the iteration is an arithmetic progression. Inspired by their scheme, we propose an improved differential privacy budget allocation scheme. When t is small, the gradient value of the model is larger and can withstand greater noise; as t increases, the gradient value of the model decreases, and a larger privacy budget needs to be allocated to reduce the effect of noise and ensure the convergence of the model. And our scheme ensures that each noise gradient conforms to the direction of gradient descent, i.e., that the value of the loss function continues to decrease. It was mentioned in [21] that the noise gradient may be in the direction of descent even if the norm of the noise gradient is much larger than the norm of the true gradient. Our solution is to take in a set of weights obtained from different learning rates, compute their loss function values, and choose the learning rate corresponding to the smallest loss function value. If the selected

learning rate is 0, this noise gradient does not fit the gradient descent direction. The algorithm will discard it and reallocate a larger privacy budget to compute a new noise gradient.

Combining the above solution ideas for the two critical problems, we propose a differential privacy SGD algorithm with adaptive privacy budget allocation. Our contributions are summarized as follows:

- We proposed a stochastic gradient descent algorithm based on adaptive differential privacy. It can flexibly choose a suitable privacy budget for each iteration to guarantee model usability and accelerate the training process.
- Unlike previous works, we designed a learning rate selection mechanism. It can select the optimal learning rate in a set of learning rate, which ensures that the model parameters keep optimizing and ultimately achieve higher accuracy.
- We perform extensive experiments on real datasets against other popular ERM algorithms. Experimental results show that our scheme achieves 2% higher model accuracy than the second one, costing only 4% of its training time with a 0.05 privacy budget, which means that our scheme is efficient and practical.

The rest of this paper is organized as follows. In Sect. 2, we review related work. In Sect. 3, we provide preliminaries on differential privacy and machine learning. We introduce the specific implementation of our scheme in Sect. 4. Section 5 contains the experimental results on real datasets.

2 Related Work

Applying differential privacy in federated learning can effectively protect data privacy, but the usability of models perturbed by noise will unavoidably be destroyed, so solving the private ERM problem becomes a complex problem. So far, many works have been trying to solve the private ERM problem in machine learning, and their methods can be mainly concluded into three types: output perturbation, objective perturbation, and gradient perturbation.

The output perturbation [8,18,32] 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 algorithm's output 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. Chaudhuri et al. [8] implemented the ERM with output perturbation and proved that adding noise to the output result can play a specific role in artifacts. Zhang et al. [32] used algorithmic stability arguments to bound the L_2 sensitivity of the whole batch gradient descent algorithm to determine the amount of noise that must be added to outputs partially optimize the objective function. In general, the noise generated by output perturbation algorithms is usually inaccurate because the noise is calibrated to the worst-case analysis.

The objective perturbation [8,17,19] 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. The study of Chaudhuri et al. [8] showed that objective perturbation performs better than output perturbation. Kifer et al. [19] used approximate differential privacy instead of pure differential privacy, effectively improving the utility of objective perturbation methods. Although objective perturbation is more effective, its privacy guarantee is based on the premise that the problem can be solved exactly, but most time optimization problems are solved approximately in practice.

The iterative gradient perturbation method [3,30] and their variants [27,29, 32] are very popular method. Bassily et al. [3] proposed a (ϵ, δ)-differentially private version of stochastic gradient descent (SGD) algorithm. At each iteration, their algorithm perturbs the gradient with Gaussian noise and applies the advanced composition [15] together with the privacy amplification result [4] to get an upper bound on the total privacy loss. Later, Talwar et al. [26] improved those lower bounds on the utility for the Least absolute shrinkage and selection operator (LASSO) problem. In [27], the gradient perturbation method is combined with the stochastic variance reduced gradient (SVRG) algorithm, and the resulting algorithm is near-optimal with less gradient complexity. Abadi et al. [1] proposed a new mechanism for tracking privacy loss, called moments accountant, which permits tight automated analysis of the privacy loss of complex composite mechanisms and significantly improves the accuracy of the model under a suitable privacy budget. Lee et al. [21] designed an adaptive privacy budget allocation scheme to assign appropriate privacy budgets for each iteration separately to satisfy zero-concentrated differential privacy (zCDP) while improving the accuracy of the model. Du et al. [12] extended the Gaussian DP's central limit theorem (CLT), and proposed a novel dynamic DP-SGD algorithm for this case. Cheng et al. [9] proposed a gradient-based algorithm with distributed differential privacy in which try to expand the number of iterations instead of the fixed. It combines the Analytic Gaussian Mechanism with linear search method to broaden the definition domain of ϵ and obtain a relatively compact variance bound.

3 Preliminaries

In this section, we briefly recall the definition of differential privacy, introduce essential theorems, and then overview the basic principles of machine learning using SGD.

3.1 Differential Privacy

Definition 1 ((ϵ, δ)-Differential Privacy [13]**).** *A randomized mechanism $M : D \to R$ with domain D and range R satisfies (ϵ, δ)-DP if for any two adjacent inputs d,$d' \in D$ and for any subset of outputs $S \subseteq R$ it holds that*

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

The non-negative parameter ϵ is the privacy budget, which indicates the degree of privacy protection. As ϵ is set smaller, the privacy protection becomes greater. When ϵ tends to zero, the probability distribution of M outputting the same result on the adjacent data sets D and D' tends to be the same, and the less information M may disclose. The δ is also a non-negative parameter, which indicates the probability that the difference between the outputs of M on D and D' exceeds e^{ϵ}, i.e., the probability of violating differential privacy protection. Obviously, the smaller the δ, the higher the degree of privacy protection.

In particular, when $\delta = 0$, the algorithm M provides the strictest ϵ - differential privacy protection.

Theorem 1 (Privacy amplification via sampling [4]**).** *Over a domain of data sets D^n, if an algorithm M is $\epsilon' < 1$ differentially private, then for any data set $D' \in D^n$, executing M on uniformly random γn entries of D' ensures $2\gamma\epsilon'$-differential privacy.*

Theorem 2 (Navie composition [14]**).** *The class of ϵ-differentially private mechanisms satisfies $k\epsilon$-differential privacy under k-fold adaptive composition.*

Theorem 3 (Strong composition [15]**).** *Let $\epsilon, \delta' > 0$. The class of ϵ-differentially private algorithms satisfies (ϵ', δ')-differential privacy under k-fold adaptive composition for $\epsilon' = \sqrt{2k \ln(1/\delta')} + k\epsilon(e^{\epsilon} - 1)$.*

Lemma 1 [6]**.** *If two mechanisms satisfy (ϵ_1, δ)-DP and (ϵ_2, δ)-DP, then their composition satisfies $(\epsilon_1 + \epsilon_2, \delta)$-differential privacy.*

Definition 2 (L_2sensitivity). *For the adjacent data sets D and D', let $Q : D^n \to R^d$ be the query function, and the L_2 sensitivity of the query function Q is defined as follows:*

$$\triangle_2 (Q) = \max_{D \sim D'} ||Q(D) - Q(D')||_2 \tag{2}$$

The L_2 sensitivity represents the maximum range of variation of the query function Q over the adjacent data set.

Theorem 4 (Gaussian Mechanism [14]**).** *Let $\epsilon \in (0, 1)$ be arbitrary and f be a query function with L_2 sensitivity of $\triangle_2 (Q)$. The Gaussian Mechanism, which returns $f(D) + N(0, \sigma^2)$, with*

$$\sigma \geq \frac{\triangle_2 (Q)}{\epsilon} \sqrt{2 \ln(1.25/\delta)} \tag{3}$$

is (ϵ, δ)- differential privacy.

3.2 Machine Learning

Machine learning is a general term for a class of algorithms that try to attempt to mine large amounts of historical data for implied patterns and use them for

prediction or classification. More specifically, machine learning can be thought of as finding a function where the input is sample data and the output is the desired outcome.

To ensure the correctness of the function found by the machine learning algorithm, we define a loss function L that represents the penalty for mismatching the training data. The loss $L(w)$ on parameters w is the average of the loss over the training examples $\{d_1, \ldots, d_n\}$, so $L(w) = \frac{1}{N} \sum_i L(w; xd_i)$. We want to find w in the training, which produces the minimum loss (though in practice we seldom expect to reach an exact global minimum).

For complex algorithms, like neural networks, the loss function L is usually non-convex and difficult to minimize. In practice, the minimization is often done by the mini-batch stochastic gradient descent (SGD) algorithm. 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. For example, at iteration t, SGD randomly selects a small set of data $D_t \subset D$ and calculates the corresponding gradient $\nabla L(w_t; D_t)$. $-\nabla L(w_t; D_t)$ may not be the correct update direction, but it is the expected update direction since $E(\nabla L(w_t; D_t))|w_t = \nabla L(w_t)$. 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 $g_B = \frac{1}{|B|} \sum_{d \in B} \nabla L(w; d)$ as an estimation to the gradient $\nabla L(w)$. Then w is updated following the gradient direction $-g_B$ towards a local minimum.

4 Proposed Scheme

In this section, we first describe the main idea of the proposed algorithm and then present the details of its four main components.

4.1 The Main Idea

As shown in Algorithm 1, during the t-th epoch training, after obtaining the gradient by training the model on the dataset, the algorithm first clips the gradient to obtain the original gradient g_t. Subsequently, the corresponding noise is calculated according to the assigned privacy budget and added to the original gradient to get the noise gradient $\tilde{g}_t$. Then, it determines whether the noise gradient conforms to the gradient descent direction. If it does, the algorithm calculates the weight w for the next iteration by the corresponding learning rate. If not, the algorithm assigns a larger privacy budget and recalculates the noise gradient until the noise gradient conforms to the gradient descent direction. The above processes repeat until the end of the training, and the notations used in our scheme are shown in Table 1.

Table 1. Notations

Notation	Description
d	The data for training
w	The model weight
$L(\cdot)$	The loss function
$\tilde{g_t}$	The noise gradient
σ	The noise scale
C_{grad}	The gradient norm bound
ϵ	The privacy budget
a	The privacy budget increase rate
ρ	The a increase rate
η	The learning rate
Φ	The set of learning rate
λ	The learning rate decay rate

Algorithm 1: Differential Privacy SGD Algorithm

Input: $\{d_1, \ldots, d_n\}$, $L(w)$, ϵ, C_{grad}, ρ, λ

Initialize w_0 and Φ
while $t \leq T$ **do**
  $g_t \leftarrow \sum_{i=1}^{n} (\nabla L(w_t; d_i) / \max(1, \frac{\|L(w_t)\|_2}{C}))$
  $\epsilon_t \leftarrow A + \frac{B}{1+e^{-at+b}}$
  $\sigma \leftarrow 1/\epsilon_t$
  $\tilde{g_t} \leftarrow \bar{g_t} + N(0, \sigma^2 C_{grad}^2 I)$
  $idx \leftarrow 0$
  while $idx = 0$ & $t < \frac{T+1}{2}$ **do**
    $\Omega = \{L(w_t - \eta \tilde{g_t}) : \eta \in \Phi\}$
    $idx \leftarrow \arg\min\{\Omega\}$
    if $idx > 0$ **then**
      $\eta_t \leftarrow \eta_{idx}$
      $w_{t+1} \leftarrow w_t - \eta_t \tilde{g_t}$
    end
    else
      $a \leftarrow \rho \cdot a$
      $\epsilon_t \leftarrow A + \frac{B}{1+e^{-at+b}}$
      $\sigma \leftarrow 1/\epsilon_t$
      $\tilde{g_t} \leftarrow \bar{g_t} + N(0, \sigma^2 C_{grad}^2 I)$
    end
  end
  if $t \% \tau = 0$ **then**
    $\eta_{max}^{t+1} = \lambda \cdot max(\eta_1, \eta_2, \ldots, \eta_t)$
  end
end
return w

4.2 Private Gradient Approximation

In each iteration, the noise gradient $\tilde{g}_t = \nabla L(w_t) + N(0, \sigma^2)$ can be obtained by adding Gaussian noise with variance σ^2 to $\nabla L(w_t)$. According to Theorem 4, the magnitude of the noise σ^2 depends on the maximum change in the output value of the query function Q when one individual data changes. To Bound this quantity, some works [8,19] assume $\|x\| \leq 1$. Instead, we adopted the scheme of Abadi et al. [1], clip the gradient in L_2 norm by dividing it by $\max(1, \frac{\|L(w_t)\|_2}{C_{grad}})$.

4.3 Adaptive Privacy Budget Allocation

Varying from the equal allocation in previous privacy budget allocation schemes, our scheme considers the privacy protection needs of each iteration. It continuously adjusts the size of the allocated privacy budget as the training progresses. In the model training process, as the training progresses, the model's accuracy increases, and the impact of adding noise becomes greater, so more privacy budget is needed to ensure that the model's accuracy is not affected.

We divide the training process into three stages: Model initialization, Model training, and Model convergence. In the first stage, the model is just trained, and the accuracy requirement is not high, so we can allocate a small privacy budget and increase the privacy budget by a small amount, and finally obtain a model with low accuracy. After entering the second stage, the privacy budget is increased by a large margin so that the model can be steadily improved until it is close to convergence. Finally, in the third stage, the near-convergence model can converge faster under the maximum privacy budget.

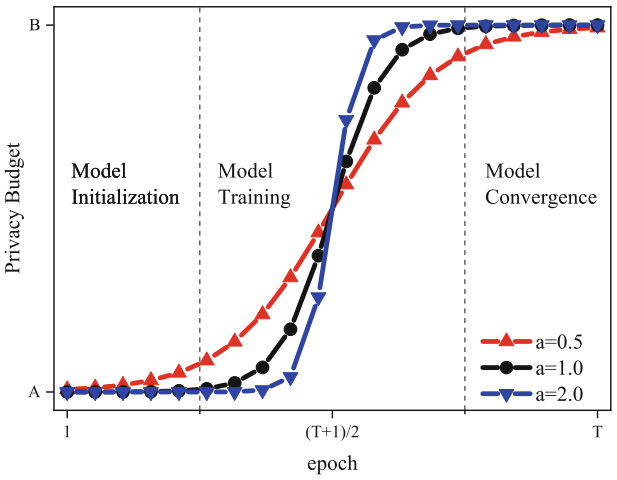


Fig. 1. Allocated privacy budget per epoch

For the t-th iteration, the allocated privacy budget is calculated according to Eq. 4, i.e., the privacy budgets $\epsilon_1, \epsilon_2, \ldots, \epsilon_T$ are assigned to the iterations from 1 to T respectively (assuming the total number of iterations is T).

$$\epsilon_t = \epsilon_{min} + \frac{\epsilon_{max}}{1 + e^{-at+b}}, 1 \leq t \leq T \tag{4}$$

where ϵ_t is the privacy budget at iteration t, the constants ϵ_{min} and ϵ_{max} jointly control the maximum and minimum values of the privacy budget during training, and the constants a and b control the growth rate of the privacy budget.

From Eq. 4 and Fig. 1, we can find that the value of a determines the trend of each privacy budget value assigned throughout the training process when the constants ϵ_{min}, ϵ_{max} and b are fixed. The smaller the value of a, the closer the difference between neighboring privacy budgets; conversely, the larger the value of a, the larger the difference between neighboring privacy budgets. Therefore, an appropriate a value can assign a reasonable privacy budget for each iteration and ensure the improvement of model accuracy.

Our scheme changes the size of the allocated privacy budget by dynamically adjusting the value of a during the training process. As shown in Fig. 1, when $t < \frac{T+1}{2}$, the smaller a is in the same iteration, the larger the corresponding privacy budget is. So when the algorithm finds that the current privacy budget is too small, it updates $a = \rho \cdot a, 0 < \rho < 1$, where ρ is the decay coefficient, and reallocates the privacy budget. After half of the training process is completed, i.e., $t > \frac{T+1}{2}$, when the privacy budget has reached a high level, keep a constant and complete the subsequent privacy budget allocation according to Eq. 4.

4.4 Learning Rate Selection

In the previous section on SGD, we learned that the descent direction of the gradient g_t during SGD training is not the correct gradient update direction but it is the expected update direction. While in a private setup, the guarantee in expectation becomes less reliable due to the addition of noise. Thus the per-iteration privacy budget needs to be used more efficiently.

For this purpose, we create a learning rate set Φ containing k candidate learning rates that follow a uniform distribution from 0 to η_{max}. The η_{max} denotes the maximum upper bound of the learning rate.

The updated gradient is obtained after the tth iteration of training $\nabla L(w_t)$, and the perturbed updated gradient is obtained by adding noise according to the assigned privacy budget $\nabla L(w_t) + N_t(0, \lambda)$. We use $\tilde{g}$ to denote the update gradient after perturbation, so the iterative update process has the following simplified form:

$$w_{t+1} = w_t - \eta_t \tilde{g} \tag{5}$$

For the best case, the model parameters are close to the optimal solution for each iteration. Therefore, we need to select the most appropriate learning rate from the learning rate candidate set Φ for the current noise level.

In our scheme, the element in Ω is the value of the loss function $L(w_t - \eta\tilde{g})$, i.e. $\Omega = \{L(w_t - \eta\tilde{g}) : \eta \in \Phi\}$. Then we locate the minimum value $L(w_t - \eta_{idx}\tilde{g})$ in Ω, and the corresponding learning rate η_{idx} is the best learning rate we aim to find. If the learning rate η_{idx} is 0, it means that this noise gradient does not match the gradient descent direction; the algorithm will reallocate a larger privacy budget and recalculate the noise gradient. Until the learning rate η_{idx} is not 0, the model is updated using the selected learning rate.

4.5 Adjusting the Learning Rate Selection Range

Generally, the SGD algorithm with a constant learning rate does not guarantee convergence to the optimum. In order to control the variance of private gradient estimation, the range of learning rate selection needs to be updated. After every τ iteration, we update η_{max} according to the following rule:

$$\eta_{max}^{t+1} = \lambda \cdot max(\eta_1, \eta_2, \ldots, \eta_t), 0 < \lambda < 1 \tag{6}$$

where η_{max}^{t+1} is the maximum learning rate at the $t+1$th iteration, λ is the decay coefficient, η_t denotes the learning rate chosen at iteration t. This allows our algorithm to adaptively change the learning rate range based on the relative position of the current iteration.

4.6 Correctness of Privacy

The correctness of the algorithm depends on (ϵ, δ)-differential privacy composition (Lemma 1) and accounting for the privacy cost of each primitive.

Proof To maintain the total privacy budget as our pre-set value ϵ, let $\epsilon_t + \epsilon_{T-t+1} = 2 \cdot \frac{\epsilon}{T}$. According to the Lemma 1, the total privacy budget ϵ can be expressed as:

$$\epsilon_{total} = \sum_{t=1}^{T} \epsilon_t = \frac{T}{2}(\epsilon_t + \epsilon_{T-t+1}) = T \cdot \frac{\epsilon}{T} = \epsilon \tag{7}$$

To keep the correctness of Eq. 7, the sum of parameters ϵ_{min} and ϵ_{max} will be constant as $2 \cdot \frac{\epsilon}{T}$, and parameter b will be a fixed value $\frac{aT}{2}$.

Since our scheme adjusts the value of a during training, we need to discuss whether $\epsilon_{total} \leq \epsilon$ holds in this scenario. We assume that k operations to update the value of a occur, where $k \leq \lfloor \frac{T+1}{2} \rfloor$. When $k = 0$, Eq. 7 still holds. And when $k > 0$, take $k = 1$ as an example, the value of a is updated from a_0 to a_1, where $a_1 < a_0$. Figure 1 shows that when $t > \frac{T+1}{2}$, the smaller a is, the smaller the corresponding privacy budget is; thus, we have $\epsilon_t(a_1) < \epsilon_t(a_0)$ for each iteration.

Therefore, the total privacy budget satisfies the following condition:

$$\begin{aligned}
\epsilon_{total} &= \sum_{t=1}^{T} \epsilon_t \\
&= \sum_{1}^{\frac{T+1}{2}} \epsilon_t + \sum_{\frac{T+1}{2}+1}^{T} \epsilon_t \\
&= \sum_{1}^{\frac{T+1}{2}} \epsilon_t + \epsilon_{\frac{T+1}{2}+1}(a_1) + \ldots + \ldots + \epsilon_T(a_1) \\
&< \sum_{1}^{\frac{T+1}{2}} \epsilon_t + \epsilon_{\frac{T+1}{2}+1}(a_0) + \ldots + \epsilon_{t'}(a_0) + \epsilon_{t'+1}(a_1) + \ldots + \epsilon_T(a_1) \\
&= \epsilon
\end{aligned}$$

In summary, the privacy budget consumed in our scenario does not exceed the total set privacy budget.

5 Experimental Results

In this section, we apply the scheme to the classical machine learning algorithms: support vector machines (SVM) and logistic regression. We compare our scheme with other popular schemes in model classification accuracy and training time to show that our scheme has better accuracy and higher efficiency. Finally, we apply our scheme to a more complex convolutional neural network (CNN) to illustrate that our scheme is also excellently suited for complex models.

5.1 Experiment on SVM and Logistic Regression

In this part, we experiment with SVM and Logistic Regression on the four UCI datasets and compare them with other schemes. The dataset we use is as follows: (i) Adult [2,7] dataset contains 48,842 records of individuals from the 1994 US Census. (ii) BANK [2] contains 45,211 records of marketing campaign-related information about customers of a Portuguese banking institution. (iii) IPUMS-BR and (iv) IPUMS-US datasets are also Census data extracted from IPUMS-International [24], and they contain 38,000 and 40,000 records, respectively. Table 2 summarizes the characteristics of datasets used in our experiments.

Baseline Model. We compare our scheme against four baseline algorithms, namely, SGD-Adv [3], SGD-MA [1], DP-AGD [21] and NonPrivate. SGD-Adv is a differentially private version of SGD algorithm, it implements differential privacy protection under SGD algorithm by advanced composition theorem with privacy amplification results. SGD-MA also implements differential privacy protection under SGD, except that it uses a combinatorial approach called moments

Table 2. Characteristics of datasets

Dataset	Size (n)	Dime	Label
Adult	48,842	124	Is annual income >50k?
BANK	45,211	33	Is the product subscribed?
IPUMS-BR	38,000	53	Is monthly income >$300?
IPUMS-US	40,000	58	Is annual income >25k?

accountant, tailored for Gaussian noise distribution. DP-AGD is an adaptive privacy budget allocation scheme that dynamically adjusts the size of the privacy budget for the current iteration during the training process. NonPrivate is an optimization algorithm that does not satisfy differential privacy.

In the comparison experiment, we report the classification accuracy (i.e., the proportion of correctly classified examples in the test set) and the time spent on training the model. All reported results are averaged over 10 iterations of 5-fold cross-validation.

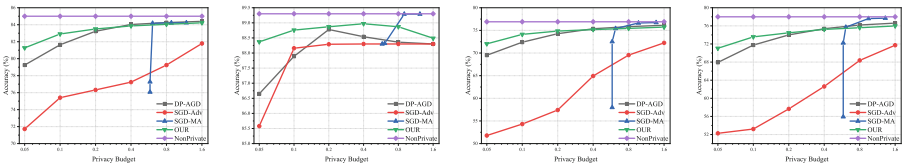
Parameter Setting. When there are known default parameter settings for the prior works, we use the same settings. Throughout all the experiments, the value of privacy parameter δ is fixed to 10^{-8} for the Adult, BANK, IPUMS-BR, and IPUMS-US datasets. According to the common practice in optimization, the sizes of mini-batches for SGD-Adv and SGD-MA are set to $\sqrt{n}$. The parameter settings of DP-AGD refer to the values given in the article [3], where the gradient crop value C_{grad} is set to 3.0, the objective function crop value C_{obj} is set to 3.0, and the attenuation parameter γ is set to 0.1.

Since the parameters of our scheme can be dynamically adjusted during training, we just need to consider a few parameters, and the most important of which is the gradient clipping norm C_{grad}. If C_{grad} is set to a too low value, it significantly reduces the sensitivity, but at the same time, it can cause too much information loss in the estimates. Conversely, if C_{grad} is set too high, the sensitivity becomes high, resulting in adding too much noise to the estimates. On the other hand, increasing the norm bound C forces us to add more noise to the gradients (and hence the parameters) since we add noise based on σC. In practice, a good way to choose a value for C is by taking the median of the norms of the unclipped gradients over the course of training. Based on experiments in testing the effect of different C_{grad} on classification accuracy, we finally determined the value of C_{grad} to be 3.0.

Experimental Results on SVM. Support vector machine (SVM) is one of the most effective tools for classification problems. The SVM classification problem is formulated as an optimization problem:

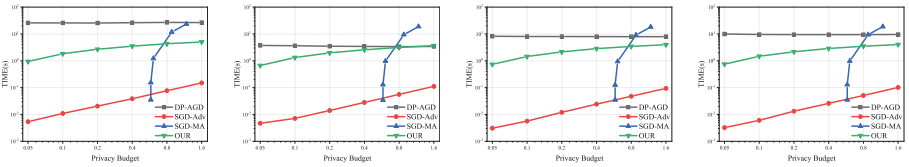
$$\min_{w} \frac{\lambda}{2}\|w\|_2^2 + \frac{1}{n}\sum_{i=1}^{n} \max\{1 - y_i w^T x_i, 0\},$$

where $x_i \in R^{p+1}$ and $y_i \in -1, +1$ for $i \in [n]$.



(a) Adult (b) BANK (c) IPUMS-BR (d) IPUMS-US

Fig. 2. Classification accuracies of SVM by varying the ϵ

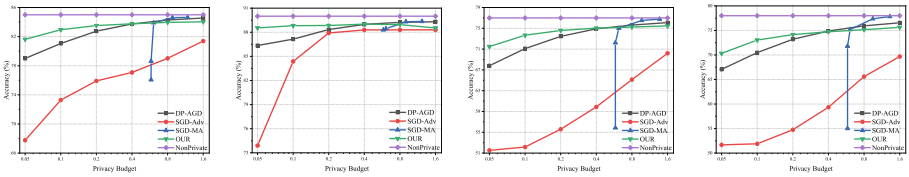


(a) Adult (b) BANK (c) IPUMS-BR (d) IPUMS-US

Fig. 3. Training time of SVM by varying ϵ

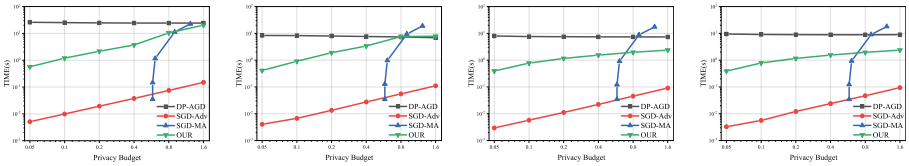
Figure 2 shows the classification accuracy of the SVM model on the four real data sets, and Fig. 3 shows the training time of the SVM. Taking the experiments on the Adult dataset as an example, in Fig. 2(a), we can see that when the privacy budget was small (e.g., $\epsilon = 0.05$ and $\epsilon = 0.1$), our scheme was able to achieve a higher classification accuracy compared to the other three schemes. The reason was that adaptive privacy budget allocation provides a tighter privacy loss and enables our scheme to withstand a higher number of iterations, this can also be reflected in Fig. 3(a), our scheme took a longer time than SGD-Adv to complete an iteration. However, we improved the classification accuracy by about 10% compared to SGD-Adv. Also, compared to DP-AGD, which also performed well with a small privacy budget, our scheme achieved higher classification accuracy in less time. When $\epsilon = 0.05$, our scheme spent 0.94 s to obtain 71.05% classification accuracy, while the DP-AGD scheme spent 25.86s to obtain 67.93 % classification accuracy.

As the privacy budget increases, we no longer have a clear advantage in classification accuracy. After $\epsilon \geq 0.2$, DP-AGD obtained an approximate accuracy with our scheme. When $\epsilon = 0.2$, our scheme spent 2.68s to access 83.50% classification accuracy, while the DP-AGD scheme spent 25.55s to get 83.23 % classification accuracy. SGD-MA worked properly when $\epsilon = 0.58$ and led other differential privacy schemes at $\epsilon = 0.95$ and $\epsilon = 1.32$. When $\epsilon = 0.8$, our scheme spent 4.30 s to obtain 84.05% classification accuracy, while the DP-AGD scheme spent 26.93 s to access 84.25% classification accuracy, and the SGD-MA scheme spent 11.76 s to acquire 84.22% classification accuracy. The reason was that the moments accountant mechanism used by SGD-MA can provide a tight bound on the privacy loss, which, combined with the privacy amplification effect due to subsampling, enables SGD-MA to withstand a higher number of iterations.



(a) Adult (b) BANK (c) IPUMS-BR (d) IPUMS-US

Fig. 4. Classification accuracies of logistic regression by varying ϵ



(a) Adult (b) BANK (c) IPUMS-BR (d) IPUMS-US

Fig. 5. Training time of logistic regression by varying ϵ

However, the disadvantage is that when the privacy budget is small, it is difficult to use the moment accountant scheme due to the insufficient number of iterations. When δ was fixed as 10^{-8}, a single iteration of SGD-MA consumed a privacy budget of about 0.58. And a lower privacy budget implies a higher level of privacy protection, so it is difficult for the moments accountant scheme to achieve a high level of privacy protection. However, our scheme still had significant strength in training time.

The experimental results on other datasets were basically consistent with the performance on the Adult dataset as well. It is worth noting that in Fig. 2(b) and Fig. 3(b), we noticed that when conducting experiments on BANK dataset, the classification accuracy decreases and the training time was large when the privacy budget was large (e.g., $\epsilon = 0.8$ and $\epsilon = 1.6$). After analysis, we believed that it is because the structure of the BANK dataset is very simple. Hence, the model reaches convergence too early, and finding a noisy gradient to match the gradient descent direction can be more difficult. Maybe this problem can be improved by adjusting the number of training iterations.

Experimental Results on Logistic Regression. We also applied our algorithm to a regularized logistic regression model in which the goal is

$$\min_{w} \frac{\lambda}{2}\|w\|_2^2 + \frac{1}{n}\sum_{i=1}^{n} \log(1 + exp(-y_i w^T x_i)),$$

where $x_i \in R^{p+1}$, $y_i \in -1, +1$ and $\lambda > 0$ is a regularization coefficient.

The performance of the logistic regression model on the four datasets in Fig. 4 and Fig. 5 was generally consistent with that of the SVM model. Under

a small privacy budget, our scheme still performed the best among the four privacy schemes, it used less time and obtained a higher accuracy rate. Similarly, taking the experiments on the Adult dataset as an example. When $\epsilon = 0.05$, our scheme spent 0.5679 s to obtain 81.63% classification accuracy, while the DP-AGD scheme spent 25.91736 s to obtain 79.00% classification accuracy.

By experimenting with SVM models and logistic regression models, we can find that our scheme performed well under various sizes of privacy budgets, especially under small privacy budgets, it can get a higher classification accuracy in a shorter training time. This shows that our scheme is a practical and efficient differential privacy SGD scheme.

5.2 Experiment on CNN

We conduct experiments on the standard MNIST dataset for handwritten digit recognition consisting of 60,000 training examples and 10,000 testing examples [20]. Each example is a 28×28 size gray-level image. We use a CNN with ReLU units and softmax of 10 classes (corresponding to the 10 digits) with cross-entropy loss.

Baseline Model. Our baseline model is a CNN model consisting of two convolutional layers and one Fully-Connected layer, using ReLU as the activation function. Using the batch size 128, we can reach an accuracy of 98.34% in about 10 epochs as shown in Fig. 6(a).

Parameter Setting. In this part of the experiment, the key parameter we need to determine remains the gradient clipping norm C_{grad}. Based on experiments in testing the effect of different C_{grad} on classification accuracy, we finally determined the value of C_{grad} to be 5.0.

Experiment Results on CNN. For a differentially private model, we experiment with the same model on MNIST. To limit sensitivity, we clip the gradient norm of each layer at 5.0. We report results for three choices of the privacy budget, which we call small ($\epsilon = 0.5$), medium ($\epsilon = 2$), and large ($\epsilon = 8$). Meanwhile, the δ-values are all set to 10^{-5}.

Figure 6(b) 6(c) 6(d) shows the results for different privacy budget. Each plot shows the evolution of training accuracy and loss function value as a function of the number of epochs. We achieved 89%, 95%, and 96% test set accuracy for (0.5, 10^{-5}) (2, 10^{-5}), and (8, 10^{-5})-differential privacy respectively. In Fig. 6(b) we can see that the loss function value fluctuates more during the descent and the final model had a lower classification accuracy at $\epsilon = 0.5$ compared to the other two cases. This was because using a small privacy budget requires adding a larger amount of noise to the gradient. Benefiting from the fact that our algorithm determines whether the noise gradient according to the direction of gradient descent during training, the value of the loss function decreased and the accuracy

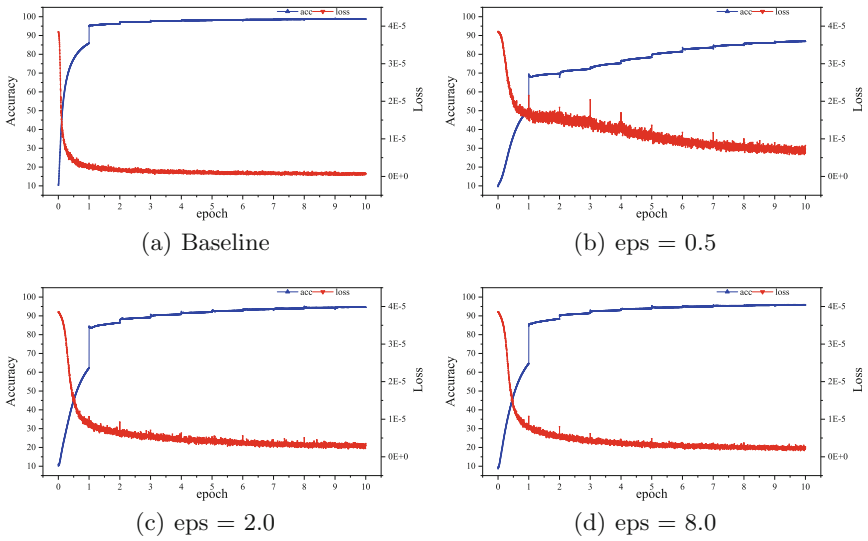


Fig. 6. Results on the accuracy for different noise levels on the MNIST dataset

improved. Also thanks to the adaptive privacy budget allocation, a larger privacy budget can be allocated to ensure the convergence of the model in the later stages of training. Combining Fig. 6(c) and Fig. 6(d), the training process of the model is smoother and the final accuracy is higher after increasing the privacy budget. The final model accuracy of our scheme was only 1% away from the SGD-MA scheme, which performed well on CNN.

In addition, in our experiments, a substantial improvement in accuracy occurred at the start of a new epoch of training, which can be also demonstrated in Fig. 6. The problem also appeared in non-privacy-preserving training and was slowly corrected in subsequent training, so we believe it caused little influence on the final result.

By experimenting on a CNN, we are able to find that our scheme still has excellent performance even on a more complex model, which indicates the suitable generalization of our scheme.

6 Conclusion

The private ERM problem in machine learning has been a tough problem since it sacrifices too much utility and accuracy for privacy preservation. Solving the privacy ERM problem can enhance the usability of the model while protecting users' privacy, which has considerable positive significance for the popularization and practicality of federated learning. According to the problem above, we design a stochastic gradient descent algorithm based on adaptive differential privacy: it allocates a privacy budget for each iteration incrementally based on the current iteration and evaluates the suitability of each added noise. Thus, in our scheme,

we can train models with higher accuracy for a limited number of iterations. In our experiments, our algorithm has been shown to train models with higher accuracy in less time and performs well, especially in scenarios with a small privacy budget. However, privacy budget allocation in our scheme cannot fully exhaust the given privacy budget. Accordingly, we may further optimize the privacy budget allocation strategy to improve the model accuracy in future work.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (62072078) and the Natural Science Foundation of Sichuan, China (2022NSFSC0550).

References

1. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS 2016, pp. 308–318. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2976749.2978318
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
3. Bassily, R., Smith, A., Thakurta, A.: Private empirical risk minimization: efficient algorithms and tight error bounds. In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, pp. 464–473. IEEE, October 2014. https://doi.org/10.1109/FOCS.2014.56
4. Beimel, A., Kasiviswanathan, S.P., Nissim, K.: Bounds on the sample complexity for private learning and private data release. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 437–454. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_26
5. Bottou, L., Cun, Y.: Large scale online learning. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems, vol. 16. MIT Press, Cambridge (2003)
6. Bun, M., Steinke, T.: Concentrated differential privacy: simplifications, extensions, and lower bounds. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9985, pp. 635–658. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53641-4_24
7. Chang, C.C., Lin, C.J.: LibSVM: a library for support vector machines **2**(3) (2011). https://doi.org/10.1145/1961189.1961199
8. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. J. Mach. Learn. Res. **12**(null), 1069–1109 (2011). https://doi.org/10.5555/1953048.2021036
9. Cheng, J., et al.: Adaptive distributed differential privacy with SGD. In: Workshop on Privacy-Preserving Artificial Intelligence, vol. 6 (2020)
10. Cotter, A., Shamir, O., Srebro, N., Sridharan, K.: Better mini-batch algorithms via accelerated gradient methods. In: Proceedings of the 24th International Conference on Neural Information Processing Systems. NIPS 2011, pp. 1647–1655. Curran Associates Inc., Red Hook (2011)
11. Dekel, O., Gilad-Bachrach, R., Shamir, O., Xiao, L.: Optimal distributed online prediction using mini-batches. J. Mach. Learn. Res. **13**(null), 165–202 (2012). https://doi.org/10.5555/2188385.2188391
12. Du, J., Li, S., Feng, M., Chen, S.: Dynamic differential-privacy preserving sgd. arXiv preprint arXiv:2111.00173 (2021)

13. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
14. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014). https://doi.org/10.1561/0400000042
15. Dwork, C., Rothblum, G.N., Vadhan, S.: Boosting and differential privacy. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, pp. 51–60. IEEE, October 2010. https://doi.org/10.1109/FOCS.2010.12
16. Ganta, S.R., Kasiviswanathan, S.P., Smith, A.: Composition attacks and auxiliary information in data privacy. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2008, pp. 265–273. Association for Computing Machinery, New York (2008). https://doi.org/10.1145/1401890.1401926
17. Hua, J., Xia, C., Zhong, S.: Differentially private matrix factorization. IJCAI 2015, pp. 1763–1770. AAAI Press (2015). https://doi.org/10.5555/2832415.2832494
18. Jain, P., Kothari, P., Thakurta, A.: Differentially private online learning. In: Conference on Learning Theory, pp. 24–31. JMLR Workshop and Conference Proceedings (2012)
19. Kifer, D., Smith, A., Thakurta, A.: Private convex empirical risk minimization and high-dimensional regression. In: Mannor, S., Srebro, N., Williamson, R.C. (eds.) Proceedings of the 25th Annual Conference on Learning Theory. Proceedings of Machine Learning Research, vol. 23, pp. 25.1-25.40. PMLR, Edinburgh, 25–27 June 2012
20. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
21. Lee, J., Kifer, D.: Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2018, pp. 1656–1665. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3219819.3220076
22. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: challenges, methods, and future directions. IEEE Sig. Process. Mag. **37**(3), 50–60 (2020). https://doi.org/10.1109/MSP.2020.2975749
23. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: International Conference on Artificial Intelligence and Statistics (2016)
24. Ruggles, S., Genadek, K., Goeken, R., Grover, J., Sobek, M.: Integrated public use microdata series, Minnesota Population Center (2018)
25. Song, S., Chaudhuri, K., Sarwate, A.D.: Stochastic gradient descent with differentially private updates. In: 2013 IEEE Global Conference on Signal and Information Processing, pp. 245–248. IEEE, December 2013. https://doi.org/10.1109/GlobalSIP.2013.6736861
26. Talwar, K., Thakurta, A., Zhang, L.: Nearly-optimal private lasso. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. NIPS 2015, vol. 2, pp. 3025–3033. MIT Press, Cambridge (2015). https://doi.org/10.5555/2969442.2969577
27. Wang, D., Ye, M., Xu, J.: Differentially private empirical risk minimization revisited: faster and more general. In: Advances in Neural Information Processing Systems, vol. 30 (2017). https://doi.org/10.48550/arXiv.1802.05251

28. Wang, X., Han, H., Zhang, Z., Yu, Q., Zheng, X.: Budget allocation method for tree index data differential privacy. Comput. Appl. **38**(7), 1960–1966 (2008)
29. Wang, Y.X., Fienberg, S.E., Smola, A.J.: Privacy for free: posterior sampling and stochastic gradient Monte Carlo. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning. ICML 2015, vol. 37, pp. 2493–2502. PMLR, JMLR.org (2015). https://doi.org/10.5555/3045118.3045383
30. Williams, O., McSherry, F.: Probabilistic inference and differential privacy. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems. NIPS 2010, vol. 2, pp. 2451–2459. Curran Associates Inc., Red Hook (2010). https://doi.org/10.5555/2997046.2997169
31. Wu, X., Li, F., Kumar, A., Chaudhuri, K., Jha, S., Naughton, J.: Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In: Proceedings of the 2017 ACM International Conference on Management of Data. SIGMOD 2017, pp. 1307–1322. Association for Computing Machinery, New York (2017). https://doi.org/10.1145/3035918.3064047
32. Zhang, J., Zheng, K., Mou, W., Wang, L.: Efficient private ERM for smooth objectives. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. IJCAI 2017, pp. 3922–3928. AAAI Press (2017). https://doi.org/10.5555/3172077.3172437