FAST ESTIMATION FOR PRIVACY AND UTILITY IN DIFFERENTIALLY PRIVATE MACHINE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, differential privacy has been widely studied in machine learning due to its formal privacy guarantees for data analysis. As one of the most important parameters of differential privacy, ϵ controls the crucial tradeoff between the strength of the privacy guarantee and the utility of model. Therefore, the choice of ϵ has a great influence on the performance of differentially private learning models. But so far, there is still no rigorous method for choosing ϵ. In this paper, we deduce the influence of ϵ on utility private learning models through strict mathematical derivation, and propose a novel approximate approach for estimating the utility of any ϵ value. We show that our approximate approach has a fairly small error and can be used to estimate the optimal ϵ according to the expected utility of users. Experimental results demonstrate high estimation accuracy and broad applicability of our approximate approach.

1 INTRODUCTION

In recent years, more and more researches have exposed potential privacy risks in large-scale machine learning tasks Bassily et al. (2014); Fredrikson et al. (2014; 2015); Shokri et al. (2017); Yeom et al. (2018). Therefore, with the broad deployment of machine learning applications and machine learning as a service, the privacy concerns are becoming more and more serious. To address this problem, many recent studies turn to combining machine learning algorithms with the framework of differential privacy Dwork et al. (2014), which guarantees privacy by adding random noise to each of the model parameters. The amount of added noise will directly affect the privacy guarantee and utility of learned model. The more added noise, the stronger the privacy guarantee, and the more serious the degradation of utility. Mathematically, this trade-off is tunable by a parameter ϵ.

The value of ϵ is essential for differentially private learning models. Too small ϵ may lose too much utility that results in a useless model, and too large ϵ may provide meaningless privacy guarantee. Thus, how to choose ϵ is a difficult task for users. The crux of this problem is that users are offered too little insight into how this should be done. For most users, the only way is to try the value of ϵ one by one to seek for the optimal ϵ that provides satisfied privacy-utility trade-off. Note that each attempt requires to train a model to see if this ϵ is available, this will cause a lot of meaningless consumption of resources and time, and usually, it is very difficult for users to find the optimal ϵ that meets their expectations in limited attempts.

One way to address this issue is to make users understand their privacy requirements, so that the corresponding value of ϵ can be directly calculated according to the formula of differential privacy. Some solutions have been put forward based on this point Lee & Clifton (2011); Hsu et al. (2014); Naldi & Dacquisto (2015); Kohli & Laskowski (2018). However, these solutions often rely on strong assumptions, such as users are able to perceive and judge the effects of ϵ, which is often hard to achieve in reality. Especially in DP-ML, where the privacy is a quite abstract and complex concept.

In this paper, we focus on solving the problem from a new angle, which takes advantage of users' familiarity with utility. For most users, the concept of utility is more intuitive than privacy. The requirement of utility can be easily expressed by many indexes, such as accuracy, precision, loss, error, etc. Therefore, choosing ϵ according to the expected utility is more likely to gain better recognition and acceptance in practice. Based on this point, we put forward our approach, which can estimate the optimal ϵ value according to users' expected utility, or, conversely, estimate the utility of model for a given value of ϵ.

The contributions of this paper can be summarized as follows:

- We comprehensively analyze the influence of ϵ on utility in differentially private machine learning. We propose a practical approximate approach for utility estimation, which has a fairly small error on estimation results.
- We show how to use our approximate approach to estimate the optimal ϵ according to the expected utility. We also offer an approximate method for directly obtaining the private model after ϵ estimation.
- Experimental results show that the estimation results of our approximate approach is considerable close to the actual measured results and the error is basically in line with our expectations.

2 RELATED WORKS

The question of how to choose ϵ has always existed since differential privacy was first proposed in Dwork (2006). However, so far, the relevant research is still very limited, most of which mainly focus on the explainability of privacy. Lee & Clifton (2011) analyzes the necessity of privacy guarantee from the perspective of adversary. Naldi & Dacquisto (2015) conducted two new parameters to provide a more precise picture of the level of differential privacy achieved. Hsu et al. (2014) provides some analysis of privacy considerations through the balance between data consumers and data providers. Kohli & Laskowski (2018) uses an information system to incorporate user's privacy preferences by having them vote on the privacy parameters they would like to have. Different from the above researches, we start from a completely new perspective, which allows users to choose the value of ϵ according to their expectations on utility.

Within the research community of DP-ML, the influence of ϵ on utility is also a hot research topic. Especially in recent years, many novel methods Abadi et al. (2016); Geumlek et al. (2017); Zhang et al. (2017); Yu et al. (2019) can significantly reduce the added noise for achieving the same level of differential privacy, which can reduce the effect of noise on utility. However, the utility bound, or error bound, proposed in previous literature can only be used as reference in related researches, which can hardly provide helpful instruction for users in practice.

The closest research to our work is Ligett et al. (2017), in which they proposed a meta-method to find the empirically strongest privacy level that meets the accuracy constraint. In their method, a very private hypothesis is initially computed, and then the noise is gradually subtracted until a sufficient level of accuracy is achieved. However, this method requires many times of attempts (or noise subtraction), and just as we mentioned in introduction, this will cause a lot of consumption of resources and time. Our method can find the empirically strongest privacy level that meets the accuracy constraint through a few training times, which is more efficient and faster.

3 PRELIMINARY

In the rest of this paper, we use $\|\cdot\|$ to denote the L_2 norm. All vectors will typically be written in boldface. We will use $\boldsymbol{I}$ to denote the identity matrix. Before delving into the details of our approach, we will first recall some basic concepts and expressions of differential privacy and differentially private learning.

3.1 DIFFERENTIAL PRIVACY

Differential privacy is a rigorous mathematical framework for privacy guarantee. It has recently received a significant amount of research attention for its robustness to known attacks. The typical definition of differential privacy is given in Dwork et al. (2014), as follows:

Definition 1 (ϵ-differential privacy)**.** *Given a randomized function $\mathcal{M}: \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$, we say $\mathcal{M}$ is ϵ-differential privacy if for all pairs of neighboring inputs D, D' differing by* *one record, and for any subset of outputs $S \subseteq \mathcal{R}$, we have:*

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{M}(D') \in S].$$

A common way to achieve differential privacy is to add some randomized noise to the output, where the noise is proportional to sensitivity of function $\mathcal{M}$: $\max_{D,D'} \|\mathcal{M}(D) - \mathcal{M}(D')\|$.

3.2 DIFFERENTIALLY PRIVATE LEARNING

For machine learning algorithms, we consider the regularized empirical risk minimization, which is to learn a classifier from labeled examples. Let $\mathcal{X}$ denotes the input space and $\mathcal{Y}$ denotes the output space, given a dataset $\mathcal{D} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n\}$ where $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, the objective is to get a model $\hat{\boldsymbol{\theta}}$ from the following unconstrained optimization problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D}) + \frac{\lambda}{2}||\boldsymbol{\theta}||^2, \tag{1}$$

where $L(\boldsymbol{\theta}, \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{\theta}, \boldsymbol{z}_i)$ is the empirical loss and $\ell(\boldsymbol{\theta}, \boldsymbol{z}_i)$ is the loss function for $\boldsymbol{z}_i$. The term of $\frac{\lambda}{2}||\boldsymbol{\theta}||^2$ is the regularizer that prevents over-fitting.

To solve the privacy issues in machine learning, Chaudhuri et al. (2011) proposed an effective mechanism for learning algorithm to achieve ϵ-differential privacy, called objective perturbation. Formally, the objective perturbation mechanism would be expressed as:

$$\hat{\boldsymbol{\theta}}_{\epsilon} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D}) + \frac{\lambda}{2}||\boldsymbol{\theta}||^2 + \frac{1}{n}\mathbf{b}_{\epsilon}^T\boldsymbol{\theta} + \frac{1}{2}\delta_{\epsilon}||\boldsymbol{\theta}||^2, \tag{2}$$

where $\mathbf{b}_{\epsilon}$ is random noise with density

$$f(\mathbf{b}) = \frac{1}{\alpha}e^{-\beta_{\epsilon}||\mathbf{b}||}, \tag{3}$$

where the parameter α is a normalizing constant. It has been proven that 2 achieves ϵ-differential privacy if $L(\boldsymbol{\theta}, \mathcal{D})$ is 1-strongly convex and twice-differentiable, with $||\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D})|| \leq 1$, $||\nabla^2_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D})|| \leq c$ in $\boldsymbol{\theta}$, and β_{ϵ}, δ_{ϵ} are defined as follows Chaudhuri et al. (2011):

$$\begin{aligned}
\beta_{\epsilon} &= \begin{cases} \frac{1}{4}\epsilon, & \text{if } 0 < \epsilon \leq \log(1 + \frac{2c}{n\lambda} + \frac{c^2}{n^2\lambda^2}), \\ \frac{1}{2}\epsilon - \frac{1}{2}\log(1 + \frac{2c}{n\lambda} + \frac{c^2}{n^2\lambda^2}), & \text{if } \epsilon > \log(1 + \frac{2c}{n\lambda} + \frac{c^2}{n^2\lambda^2}), \end{cases} \\
\delta_{\epsilon} &= \begin{cases} \frac{c}{n(e^{\frac{\epsilon}{4}}-1)} - \lambda, & \text{if } 0 < \epsilon \leq \log(1 + \frac{2c}{n\lambda} + \frac{c^2}{n^2\lambda^2}), \\ 0, & \text{if } \epsilon > \log(1 + \frac{2c}{n\lambda} + \frac{c^2}{n^2\lambda^2}). \end{cases}
\end{aligned} \tag{4}$$

Our analysis in this paper is mainly based on the objective perturbation mechanism in the above form.

4 APPROXIMATE APPROACH

As we discussed at the beginning of the paper, choosing an appropriate value of ϵ is essential for DP-ML. However, without any instructions, it is also a very difficult job, especially for large-scale learning tasks. Repeated training not only brings huge consumption, but also gets little feedback: looking for the next test point will be as clueless as the previous one, just knowing that it should be larger or smaller.

Our goal is to offer users an efficient way to choose the optimal ϵ for their private learning algorithms. Specifically, we aim to let users know the utility of an arbitrary ϵ, or the optimal ϵ achieves the expected utility in a very convenient way and a very short time, so that they can choose the optimal ϵ more efficiently and purposefully. To this end, our first step will be understanding the influence of ϵ on the model utility in the differentially private learning. In order to facilitate calculation and analysis, we use the empirical loss to represent the model utility.

4.1 APPROXIMATE ANALYSIS

Formally, we denote $\hat{\boldsymbol{\theta}}_{\epsilon'}$ and $\hat{\boldsymbol{\theta}}_{\epsilon}$ as the true minimizers of objective function that achieves ϵ'- and ϵ-differential privacy, respectively. Then, we denote $L(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D}) - L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})$ as the utility difference between $\hat{\boldsymbol{\theta}}_{\epsilon'}$ and $\hat{\boldsymbol{\theta}}_{\epsilon}$. According to the Taylor expression approximation at ϵ, we have

$$L(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D}) - L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D}) \approx \frac{\partial L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \epsilon}(\epsilon' - \epsilon). \tag{5}$$

Apparently, in addition to the change of ϵ, the utility change also denpends on $\frac{\partial L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \epsilon}$. Theorem 1 gives us a further analysis of $\frac{\partial L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \epsilon}$.

Theorem 1. *Assume that $L(\boldsymbol{\theta}, \mathcal{D})$ is twice-differentiable and strictly convex in $\boldsymbol{\theta}$. Let $\boldsymbol{W}_{\epsilon} = \frac{\partial^2 L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \boldsymbol{\theta}^2} + \lambda \boldsymbol{I} + \delta_{\epsilon} \boldsymbol{I}, \mathbf{b}'_{\epsilon} = \frac{\partial \mathbf{b}_{\epsilon}}{\partial \epsilon}, \delta'_{\epsilon} = \frac{\partial \delta_{\epsilon}}{\partial \epsilon}$. Then, we have*

$$\frac{\partial L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \epsilon} = -\left(\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})\right)^T \boldsymbol{W}_{\epsilon}^{-1} \left(\frac{1}{n}\mathbf{b}'_{\epsilon} + \delta'_{\epsilon}\hat{\boldsymbol{\theta}}_{\epsilon}\right). \tag{6}$$

Proof. Due to the fact that the derivative of the objective function 2 at $\hat{\boldsymbol{\theta}}_{\epsilon}$ is $\mathbf{0}$, we have that

$$\frac{\partial L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \boldsymbol{\theta}} + \lambda \hat{\boldsymbol{\theta}}_{\epsilon} + \frac{1}{n}\mathbf{b}_{\epsilon} + \delta_{\epsilon}\hat{\boldsymbol{\theta}}_{\epsilon} = \mathbf{0}. \tag{7}$$

Take the derivative of 7 with respect to ϵ, we have:

$$\frac{\partial^2 L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \boldsymbol{\theta}^2}\frac{\partial \hat{\boldsymbol{\theta}}_{\epsilon}}{\partial \epsilon} + \lambda \frac{\partial \hat{\boldsymbol{\theta}}_{\epsilon}}{\partial \epsilon} + \frac{1}{n}\mathbf{b}'_{\epsilon} + \delta_{\epsilon}\frac{\partial \hat{\boldsymbol{\theta}}_{\epsilon}}{\partial \epsilon} + \delta'_{\epsilon}\hat{\boldsymbol{\theta}}_{\epsilon} = \mathbf{0}.$$

Re-arranging the terms, we get

$$\left(\frac{\partial^2 L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \boldsymbol{\theta}^2} + \lambda \boldsymbol{I} + \delta_{\epsilon}\boldsymbol{I}\right)\frac{\partial \hat{\boldsymbol{\theta}}_{\epsilon}}{\partial \epsilon} = -\frac{1}{n}\mathbf{b}'_{\epsilon} - \delta'_{\epsilon}\hat{\boldsymbol{\theta}}_{\epsilon}.$$

Thus we obtain

$$\frac{\partial \hat{\boldsymbol{\theta}}_{\epsilon}}{\partial \epsilon} = -\boldsymbol{W}_{\epsilon}^{-1}\left(\frac{1}{n}\mathbf{b}'_{\epsilon} + \delta'_{\epsilon}\hat{\boldsymbol{\theta}}_{\epsilon}\right). \tag{8}$$

Finally, we have

$$\frac{\partial L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \epsilon} = \left(\frac{\partial L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \boldsymbol{\theta}}\right)^T \frac{\partial \hat{\boldsymbol{\theta}}_{\epsilon}}{\partial \epsilon} = -\left(\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})\right)^T \boldsymbol{W}_{\epsilon}^{-1}\left(\frac{1}{n}\mathbf{b}'_{\epsilon} + \delta'_{\epsilon}\hat{\boldsymbol{\theta}}_{\epsilon}\right),$$

which concludes the proof. □

The assumptions in Theorem 1 can be satisfied on common used loss functions, such as logistic regression, Huber SVM and quadratic loss. In certain cases, some of these assumptions can be weakened. We will discuss this case in Section 4.4.

4.2 APPROXIMATE CALCULATION

Plug 6 into 5 , we have

$$L(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D}) - L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D}) \approx -\left(\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})\right)^T \boldsymbol{W}_{\epsilon}^{-1}\left(\frac{1}{n}\mathbf{b}'_{\epsilon} + \delta'_{\epsilon}\hat{\boldsymbol{\theta}}_{\epsilon}\right)(\epsilon' - \epsilon). \tag{9}$$

Although Equation 9 seems very complicated, many of the terms are familiar to us, such as $\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})$, $\frac{\partial^2 L(\hat{\boldsymbol{\theta}}_{\epsilon}, \mathcal{D})}{\partial \boldsymbol{\theta}^2}$[1] and $\hat{\boldsymbol{\theta}}_{\epsilon}$. A lot of existing methods can be used to calculate them in the process of model training. Next, we will discuss the calculation of the only new term[2], $\boldsymbol{b}'_{\epsilon}$.

[1] Also known as the Hessian matrix.

[2] δ_{ϵ} and δ'_{ϵ} can be directly calculated by Equation 4

The difficulty of calculating $\mathbf{b}'_\epsilon$ is that the function outputting $\mathbf{b}_\epsilon$ is not explicitly given. Thus, we need to first form a function related to ϵ that outputs $\mathbf{b}_\epsilon$. Let $\mathbf{b}_\epsilon = (\mathrm{b}_1, \ldots, \mathrm{b}_d)$, we have $\mathbf{b}'_\epsilon = (\mathrm{b}'_1, \ldots, \mathrm{b}'_d)$. Since $\mathbf{b}_\epsilon$ is random noise with density in 3, the probability distribution function of each b_i could be given as follows:

$$F(\mathrm{b}) = \begin{cases} (\frac{1}{2} - \frac{1}{\alpha\beta_\epsilon}) + \frac{1}{\alpha\beta_\epsilon} e^{\beta_\epsilon \mathrm{b}}, & \text{if } \mathrm{b} \le 0, \\ (\frac{1}{2} + \frac{1}{\alpha\beta_\epsilon}) - \frac{1}{\alpha\beta_\epsilon} e^{-\beta_\epsilon \mathrm{b}}, & \text{if } \mathrm{b} > 0. \end{cases} \tag{10}$$

Note that $\lim_{x\to\infty} F(\mathrm{b}) = 1$, we have $\frac{1}{\alpha\beta_\epsilon} = \frac{1}{2}$, thus we can get rid of α and re-write the probability distribution function in the following manner:

$$F(\mathrm{b}) = \begin{cases} \frac{1}{2} e^{\beta_\epsilon \mathrm{b}}, & \text{if } \mathrm{b} \le 0, \\ 1 - \frac{1}{2} e^{-\beta_\epsilon \mathrm{b}}, & \text{if } \mathrm{b} > 0. \end{cases} \tag{11}$$

Take the inverse of $F(\mathrm{b})$, we have

$$F^{-1}(\mathrm{c}) = \begin{cases} \frac{1}{\beta_\epsilon} \ln(2\mathrm{c}), & \text{if } 0 < \mathrm{c} \le \frac{1}{2}, \\ -\frac{1}{\beta_\epsilon} \ln(2 - 2\mathrm{c}), & \text{if } \frac{1}{2} < \mathrm{c} < 1. \end{cases} \tag{12}$$

Note that $F^{-1}(\mathrm{c})$ is a function that takes $\mathrm{c} \sim U(0,1)$ as input and outputs random variables follows $F(\mathrm{c})$, it is the function that satisfies our requirement. Thus, we can calculate each b'_i by the following formula:

$$\mathrm{b}'_i = \frac{\partial F^{-1}(\mathrm{c}_i)}{\partial \epsilon} = \begin{cases} -\frac{1}{\beta_\epsilon^2} \ln(2\mathrm{c}_i) \mathrm{d}\beta_\epsilon, & \text{if } 0 < \mathrm{c}_i \le \frac{1}{2}, \\ \frac{1}{\beta_\epsilon^2} \ln(2 - 2\mathrm{c}_i) \mathrm{d}\beta_\epsilon, & \text{if } \frac{1}{2} < \mathrm{c}_i < 1, \end{cases} \tag{13}$$

where each $\mathrm{c}_i \sim U(0,1)$.

Equation 9 provides an approximate approach for us to calculate the change of model's utility when varying the value of ϵ, which offers users a solution for choosing the optimal ϵ. The details of how this could be done will be discussed in Section 5.

4.3 ERROR ANALYSIS

Given the complete Taylor expression of Equation 5 as follows

$$L(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D}) - L(\hat{\boldsymbol{\theta}}_\epsilon, \mathcal{D}) = \frac{\partial L(\hat{\boldsymbol{\theta}}_\epsilon, \mathcal{D})}{\partial \epsilon}(\epsilon' - \epsilon) + r(\epsilon' - \epsilon), \tag{14}$$

where $r(\epsilon' - \epsilon) = \frac{\partial^2 L(\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}, \mathcal{D})}{\partial \epsilon^2}(\epsilon' - \epsilon)^2$ is the Taylor remainder, $\tilde{\epsilon} \in [\epsilon, \epsilon']$, we can see that our error mainly comes from omitting the Taylor remainder. Thus, to derive the error bound of Equation 9, we require the Taylor remainder to be bounded, specifically, we require $\frac{\partial^2 L(\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}, \mathcal{D})}{\partial \epsilon^2}$ to be bounded.

Theorem 2. *Assume that $L(\boldsymbol{\theta}, \mathcal{D})$ is third-differentiable and strictly convex in $\boldsymbol{\theta}$. Let $\mathbf{b}''_{\tilde{\epsilon}} = \frac{\partial \mathbf{b}'_{\tilde{\epsilon}}}{\partial \epsilon}$, $\delta''_{\tilde{\epsilon}} = \frac{\partial \delta'_{\tilde{\epsilon}}}{\partial \epsilon}$, $\boldsymbol{H}_{\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}} = \frac{\partial^2 L(\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}, \mathcal{D})}{\partial \boldsymbol{\theta}^2}$ and $\boldsymbol{T}_{\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}} = \frac{\partial^3 L(\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}, \mathcal{D})}{\partial \boldsymbol{\theta}^3}$. Then, we have*

$$\begin{aligned} \frac{\partial^2 L(\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}, \mathcal{D})}{\partial \epsilon^2} &= \left(\frac{1}{n}\mathbf{b}'_{\tilde{\epsilon}} + \delta'_{\tilde{\epsilon}}\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}\right)^T \boldsymbol{W}_{\tilde{\epsilon}}^{-1} \boldsymbol{H}_{\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}} \boldsymbol{W}_{\tilde{\epsilon}}^{-1} \left(\frac{1}{n}\mathbf{b}'_{\tilde{\epsilon}} + \delta'_{\tilde{\epsilon}}\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}\right) \\ &- \left(\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}, \mathcal{D})\right)^T \boldsymbol{W}_{\tilde{\epsilon}}^{-1} \left(\boldsymbol{T}_{\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}} + \delta'_{\tilde{\epsilon}}\boldsymbol{I}\right) \boldsymbol{W}_{\tilde{\epsilon}}^{-1} \left(\frac{1}{n}\mathbf{b}'_{\tilde{\epsilon}} + \delta'_{\tilde{\epsilon}}\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}\right) \\ &- \left(\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}, \mathcal{D})\right)^T \boldsymbol{W}_{\tilde{\epsilon}}^{-1} \left(\frac{1}{n}\mathbf{b}''_{\tilde{\epsilon}} + \delta''_{\tilde{\epsilon}}\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}\right) \\ &+ \delta'_{\tilde{\epsilon}} \left(\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}, \mathcal{D})\right)^T \boldsymbol{W}_{\tilde{\epsilon}}^{-1} \boldsymbol{W}_{\tilde{\epsilon}}^{-1} \left(\frac{1}{n}\mathbf{b}'_{\tilde{\epsilon}} + \delta'_{\tilde{\epsilon}}\hat{\boldsymbol{\theta}}_{\tilde{\epsilon}}\right). \end{aligned} \tag{15}$$

Proof. See Appendix A.1. □

By bounding each term in 15, we can obtain our error bound as follows:

Theorem 3. *Assume that $L(\boldsymbol{\theta}, \mathcal{D})$ is third-differentiable and strictly convex in $\boldsymbol{\theta}$. The δ_ϵ is defined as 4. If there exist constants c_1, c_2, c_3, c_4 so that for $\forall \boldsymbol{x} \in \mathcal{D}, \boldsymbol{\theta} \in \mathcal{C}, ||\boldsymbol{\theta}|| \leq c_1, ||\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D})|| \leq c_2, ||\nabla^2_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D})|| \leq c_3, ||\nabla^3_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D})|| \leq c_4$. Then, for any $0 < \epsilon, \epsilon' < 1$ we have*

$$|L_{approx}(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D}) - L_{actual}(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D})| = \mathcal{O}\left(\frac{(\epsilon' - \epsilon)^2}{n^2 \epsilon^4}\right), \tag{16}$$

where $L_{approx}(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D})$ is the approximation loss at ϵ' estimated by the loss at ϵ, $L_{actual}(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D})$ is true loss at ϵ' from actual measurement, n is the number of training samples.

Proof. See Appendix A.2. □

From Equation 16, we can see that our error bound is proportional to the difference between ϵ and ϵ'. That is, the closer the measuring ϵ to the estimation target ϵ', the higher the estimation accuracy. In addition, our error bound is also inversely proportional to n and measuring ϵ, which indicates that more training samples and large measuring ϵ values would help to produce more accurate estimates. For example, if we set $n = 10000$ and use $\epsilon = 0.1$ as measuring point to estimate the loss at $\epsilon = 0.2$, then the error is only about 10^{-6}, which is quite small as the range of loss is 0 to 1.

However, sometimes the targets ϵ' we want to estimate may be quite small. In this case, we can divide the estimate targets into groups with similar ϵ' values and use the mean value of each group as the measuring ϵ, which may minimize the estimation error.

4.4 ASSUMPTION VIOLATION

Equation 5 relies on several assumptions that may be violated in practice. The first would be non-convex or non-convergent objectives. In this case, the obtained parameters $\tilde{\boldsymbol{\theta}}_\epsilon$ may not be the global minimum, thus Equation 7 may not hold. To address this issue, we can form a convex quadratic approximation of the loss around $\tilde{\boldsymbol{\theta}}_\epsilon$, i.e.,

$$L(\boldsymbol{\theta}, \mathcal{D}) \approx L(\tilde{\boldsymbol{\theta}}_\epsilon, \mathcal{D}) + \nabla_{\boldsymbol{\theta}} L(\tilde{\boldsymbol{\theta}}_\epsilon, \mathcal{D})(\tilde{\boldsymbol{\theta}}_\epsilon - \boldsymbol{\theta}) + (\tilde{\boldsymbol{\theta}}_\epsilon - \boldsymbol{\theta})^T \nabla^2_{\boldsymbol{\theta}} L(\tilde{\boldsymbol{\theta}}_\epsilon, \mathcal{D})(\tilde{\boldsymbol{\theta}}_\epsilon - \boldsymbol{\theta}).$$

The second violation would be non-differentiable losses, in which neither $\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_\epsilon, \mathcal{D})$ nor $\nabla^2_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_\epsilon, \mathcal{D})$ exists. To address this issue, we can approximate the loss function by a different one, which is doubly differentiable.

Remark 1. *The above analysis shows that our approximate approach can be extended to the non convex, non-convergent and non-differentiable situations. However, in this paper, we mainly focus on verifying the effectiveness of our approach. Therefore, we only consider the normal case in our evaluations. We leave the important analysis of our approximate approach for non-convex, non-convergent and non-differentiable cases to our future work.*

5 CHOOSING ϵ WITH OUR APPROXIMATION APPROACH

With the approximation approach and its error, we can achieve our goal: to offer users an efficient way to choose the optimal ϵ. The simplest way is to select a suitable initial point ϵ, train a model with ϵ, calculate all coefficients in 9, and then use 9 to estimate the utility of other points. It should be noted that when selecting initial point, it is better to refer to the considerations mentioned in Section 4.3.

Users can also choose ϵ by setting a baseline of utility. To this end, we need to re-arrange the terms in 9 as follows:

$$\epsilon' \approx \frac{L(\hat{\boldsymbol{\theta}}_{\epsilon'}, \mathcal{D}) - L(\hat{\boldsymbol{\theta}}_\epsilon, \mathcal{D})}{\left(\nabla_{\boldsymbol{\theta}} L(\hat{\boldsymbol{\theta}}_\epsilon, \mathcal{D})\right)^T \boldsymbol{W}_\epsilon^{-1} \left(\frac{1}{n}\mathbf{b}'_\epsilon + \delta'_\epsilon \hat{\boldsymbol{\theta}}_\epsilon\right)} + \epsilon. \tag{17}$$

Similarly, users only need to train the model once at an initial point to calculate all coefficients in 9, and then the optimal ϵ can be obtained according to the users' base line of utility. The baseline of utility could be given in various forms, for example, the user can give the difference from the utility of initial point, such as setting the baseline to be 10% higher.

Note that we only need to train the algorithm on the full dataset only once, our optimization effect is considerable, especially in large-scale learning tasks. Once the optimal ϵ' is estimated, the user can use it to train the final model directly. Of course, if the user has very strict requirements on the accuracy, he/she can also use our approximation results as instructions for choosing the test ϵ during repeated training, which can save a lot of training times to help him/her find the optimal ϵ faster.

Sometimes, it may be inconvenient for users to re-train the model, for example, the training data may have been lost, or the computing resource authorization may have been expired. To solve the problems of such users, we also provide a model estimation method as follows:

Theorem 4. *Assume that $L(\boldsymbol{\theta}, \mathcal{D})$ is twice-differentiable and strictly convex in $\boldsymbol{\theta}$. Let $\boldsymbol{W}_\epsilon = \frac{\partial^2 L(\hat{\boldsymbol{\theta}}_\epsilon, \mathcal{D})}{\partial \boldsymbol{\theta}^2} + \lambda \boldsymbol{I} + \delta_\epsilon \boldsymbol{I}, \mathrm{b}'_\epsilon = \frac{\partial \mathrm{b}_\epsilon}{\partial \epsilon}, \delta'_\epsilon = \frac{\partial \delta_\epsilon}{\partial \epsilon}$. Then, we have*

$$\hat{\boldsymbol{\theta}}_{\epsilon'} \approx \hat{\boldsymbol{\theta}}_\epsilon - \boldsymbol{W}_\epsilon^{-1} \left(\frac{1}{n} \mathrm{b}'_\epsilon + \delta'_\epsilon \hat{\boldsymbol{\theta}}_\epsilon \right). \tag{18}$$

Apparently, Equation 18 is derived from the following formula:

$$\hat{\boldsymbol{\theta}}_{\epsilon'} - \hat{\boldsymbol{\theta}}_\epsilon \approx \frac{\partial \hat{\boldsymbol{\theta}}_\epsilon}{\partial \epsilon} (\epsilon' - \epsilon),$$

which is the Taylor expression approximation of $\hat{\boldsymbol{\theta}}_\epsilon$ at ϵ. However, we still suggest that users use the estimated ϵ to re-train the model if conditions permit, since the superposition of estimation methods may bring about large errors.

6 EVALUATIONS

6.1 EVALUATION SETUP

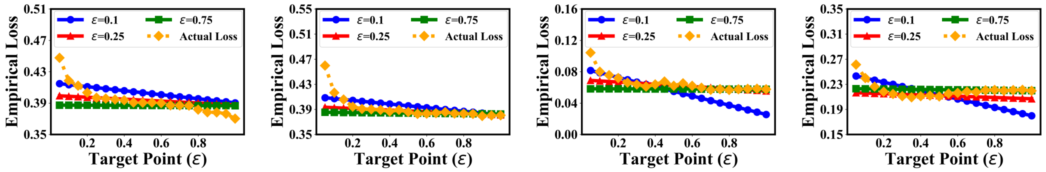
In this section, we will empirically analyze the performance of our approximation approach. We implement the output perturbation mechanism of Chaudhuri et al. (2011) based on the open source released by Iyengar et al. (2019), using Laplace distribution for noise sampling. Our evaluation considers the loss functions for two commonly used models: logistic regression (LR) and Huber SVM (SVM), which are defined as follows:

$$\begin{aligned} \ell_{LR}(\boldsymbol{z}) &= \log(1 + e^{-\boldsymbol{z}}), \\ \ell_{SVM}(\boldsymbol{z}) &= \begin{cases} 0, & \text{if } z > 1 + h, \\ \frac{1}{4h}(1 + h - z)^2, & \text{if } |1 - z| \leq h, \\ 1 - z, & \text{if} z < 1 - h, \end{cases} \end{aligned} \tag{19}$$

both of which are twice-differentiable. The Adult Dua & Graff (2017), Kddcup99 Stamper and Gisette Guyon et al. (2004) datasets are used in our experiments, each of which is randomly partitioned into 80% training samples and 20% testing samples. We use stochastic gradient descent algorithm to minimize 1 and set the iterations and learning rate to be 100 and 0.01. Due to the random noise addition, all the experiments are repeated ten times and the average results are reported. We tune the hyperparameter λ by training a non-private model on each dataset and find the optimal value to be $\lambda = 10^{-6}$. The constant c in Equation 4 is set to be 0.25 for both loss functions according to Chaudhuri et al. (2011).

6.2 EVALUATION RESULTS AND ANALYSIS

The first experiment is to verify our approximation approach. In this experiment, we fix the sample number $n = 10000$. The measuring points are chosen as $\epsilon = 0.1, \epsilon = 0.25, \epsilon = 0.75$ and the target points vary from $\epsilon = 0.05$ to $\epsilon = 1.0$. The experimental results on Adult and Kddcup99 are shown in Figure 1, and the experimental results on Gisette are shown in Appendix A.3 due to the limitation of space. The orange line, which represents actual empirical loss of each ϵ is called the actual line, and the other lines, which represent the estimate of loss, are called fitting line. Each fitting line is drawn by the estimation results of the loss of each ϵ, which are called the target points, based on the actual measurement under a specific ϵ, which is called the measuring point. It can be seen that our



(a) Adult-LR (b) Adult-SVM (c) Kddcup99-LR (d) Kddcup99-SVM

Figure 1: Performance of our approximation approach on datasets of Adult and Kddcup99 with logistic regression (LR) loss and Huber SVM (SVM) loss.

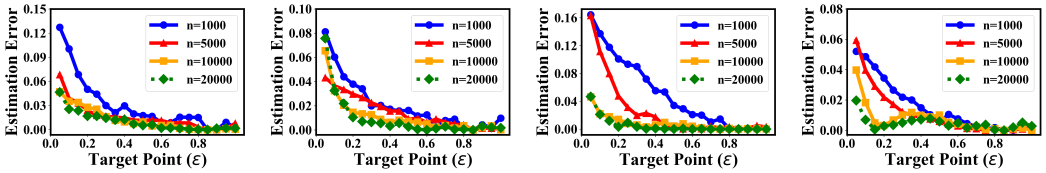
Table 1: Average error of estimated loss for each measuring point.

ϵ	0.10	0.25	0.30	0.35	0.45	0.75	0.85
Adlut-LR	0.00823	**0.00003**	0.00033	0.00004	0.00293	0.00649	0.01480
Adlut-SVM	0.02052	0.00445	0.00367	0.00385	**0.00298**	0.00770	0.00840
Kddcup-LR	0.02393	0.00763	0.00845	0.00960	**0.00283**	0.01186	0.01200
Kddcup-SVM	0.00866	0.00851	0.01052	0.01143	0.00848	0.00112	**0.00038**
Gisette-LR	0.01482	0.01931	**0.00302**	0.00810	0.00394	0.04637	0.03693
Gisette-SVM	0.00571	0.00567	0.01023	**0.00550**	0.00823	0.01745	0.02911

approximation results basically fit the empirical loss for each ϵ. In addition, we can see that each fitting line has a different slope. The larger the value of ϵ used in the actual measurement, the greater the slope of the fitting line. This is consistent with our previous analysis. Specially, each fitting line will cross the actual line at its measured ϵ. In addition, it can be seen that each fitting line is straight. The reason for this is that after training at the measuring ϵ, the coefficients in 9 are determined, which makes a linear relationship between the value of target ϵ and its estimated empirical loss. We will explore the non-linear relationship between the two in our future work.

In order to further analyze the impact of the selection of measuring point on the estimation results, we calculate the average error of estimated loss when each ϵ is used as the measuring point and represent the results in Table 1. The experimental results are abridged, and the rest can be found in Appendix A.3. For each experiment, the result in bold represents the measuring ϵ with minimum error. Although the ϵ with minimum error of each results are slightly different, ϵ in the middle tends to have lower average error. It is their sum of squares to other points is smaller. Thus, when using our approximate approach for choosing optimal ϵ, we suggest to begin with ϵ in the middle.

Finally, we evaluate the effect of sample number on estimation error. Due to the limited sample number of Gisette, we only use Adult and Kddcup99 in this experiment. We vary the sample number from 1000 to 20000 and fix ϵ for each evaluation to be the one that achieves minimum error in Table 1. The results are shown in Figure 2. Obviously, with the increase of sample number, the estimation error decreases, which is consistent with our previous analysis.



(a) Adult-LR (b) Adult-SVM (c) Kddcup99-LR (d) Kddcup99-SVM

Figure 2: Affect of sample number on estimation error. The measuring point for each evaluation, from left to right, is set to be $\epsilon = 0.25, \epsilon = 0.45, \epsilon = 0.45, \epsilon = 0.85$, respectively, which achieves the minimum error in Table 1.

7 CONCLUSION

In this paper, we focus on solving the problem of choosing optimal ϵ in DP-ML. We start by analyzing the influence of ϵ on the utility of learned model. Then we put forward our approximate approach for estimating the utility difference between the private models trained with any two ϵ values. We show how to use our approximate approach to solve the problem we mentioned at the beginning. We conduct several experiments to verify our analysis. Experimental results demonstrate the good estimation accuracy and broad applicability of our approximate approach.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318. ACM, 2016.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. 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, 2014.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd international conference on Automata, Languages and Programming - Volume Part II*, 2006.

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

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333. ACM, 2015.

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pp. 17–32, 2014.

Joseph Geumlek, Shuang Song, and Kamalika Chaudhuri. Renyi differential privacy mechanisms for posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 5289–5298, 2017.

Isabelle Guyon, S R Gunn, Asa Benhur, and Gideon Dror. Result analysis of the nips 2003 feature selection challenge. pp. 545–552, 2004.

Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. pp. 398–410, 2014.

Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. pp. 299–316, 2019.

N. Kohli and P. Laskowski. Epsilon voting: Mechanism design for parameter selection in differential privacy. In *2018 IEEE Symposium on Privacy-Aware Computing (PAC)*, pp. 19–30, 2018.

Jaewoo Lee and Chris Clifton. How much is enough? choosing ϵ for differential privacy. pp. 325–340, 2011.

Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Zhiwei Steven Wu. Accuracy first: Selecting a differential privacy level for accuracy-constrained erm. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 2563–2573, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Maurizio Naldi and Giuseppe Dacquisto. Differential privacy: An estimation theory-based method for choosing epsilon. *arXiv: Cryptography and Security*, 2015.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Niculescu-Mizil A. Ritter S. Gordon G.J. & Koedinger K.R. Stamper, J. Challenge data set from kdd cup 2010 educational data mining challenge.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. pp. 268–282, 2018.

Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning. *arXiv preprint arXiv:1904.02200*, 2019.

Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. *arXiv preprint arXiv:1703.09947*, 2017.