**CS 325: Private Data Analysis** **Fall 2022**

# Lecture 13: Differentially Private Empirical Risk Minimization: I

*Lecturer: Di Wang* *Scribes: Di Wang*

**Note**: *LaTeX template courtesy of UC Berkeley EECS dept.*



So far we have shown many relaxations and mechanisms of DP. In the following several lectures we will see how to use them to problems in Machine Learning and Statistics. And we will focus on one of the most fundamental problems in Machine Learning, *i.e.,* Empirical Risk Minimization (ERM). But before we get to that, we need to lay some groundwork in the non-private setting.

## 13.1 Empirical Risk Minimization

Many machine learning models can be expressed as follows. We have an $n$-size dataset $D = \{(x_i, y_i\}_{i=1}^n$, where the $x$'s are feature vectors and $y$'s are labels/responses. There is a specific loss function, which takes in a parameter $\theta$ and a datapoint and output a value:

$$L(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, x_i, y_i). \tag{13.1}$$

We also have a parameter set $\mathcal{C}$ which constraint the parameter $\theta$. The goal of ERM is finding the minimal value of $L(\theta, D)$, *i.e.,* $\theta^* = \arg\min_{\theta\in\mathcal{C}} L(\theta, D)$. Throughout the whole class we will assume the $\mathcal{C} \subseteq \mathbb{R}^d$ and $x_i \in \mathbb{R}^d$.

**Examples:** Lets first see some examples of ERM,

- Linear Regression: We have $\mathcal{C} = \mathbb{R}^d, y \in \mathbb{R}$, and $\ell(\theta, x, y) = (\langle x, \theta\rangle - y)^2$.
- Ridge Regression: We have $\mathcal{C} = \{\theta; \|\theta\|_2 \leq r\}, y \in \mathbb{R}$, and $\ell(\theta, x, y) = (\langle x, \theta\rangle - y)^2$.
- Ridge Regression: We have $\mathcal{C} = \{\theta; \|\theta\|_1 \leq r\}, y \in \mathbb{R}$, and $\ell(\theta, x, y) = (\langle x, \theta\rangle - y)^2$.
- Logistic Regression: This is a classification problem, $y \in \{-1, +1\}$, and $\ell(\theta, x, y) = \log(1 + e^{-y\langle\theta, x\rangle})$.
- SVM: This is a classification problem, $y \in \{-1, +1\}$, $\ell(\theta, x, y) = \max\{0, 1 - y\langle w, x\rangle\}$.
- Geometric median: This is a task in unsupervised learning. We have the loss function $\ell(\theta, x, y) = \|\theta - x\|_2$.

As stated, this formulation is quite general, and trying to solve it without further restrictions would prove to be a fruitless endeavour. We will impose a few restrictions in order to avoid nastiness which would arise otherwise. We explain a few terms before we proceed:

**Terminology**

- The gradient of a function $\ell(\theta) : \mathbb{R}^d \mapsto \mathbb{R}$ at a point $\tilde{\theta}$ is a vector $\nabla\ell(\tilde{\theta}) \in \mathbb{R}^d$ where the $i$-th coordinate is $\frac{\partial\ell(\tilde{\theta})}{\partial\theta_i}$.
- A function $\ell : \mathcal{C} \mapsto \mathbb{R}$ is convex if for all $x, y \in \mathcal{C}$ and for all $t \in [0, 1]$, $f(tx+(1-t)y) \leq tf(x)+(1-t)f(y)$.
- A function $\ell : \mathcal{C} \mapsto \mathbb{R}$ is $L$-Lipschitz if for all $x, y \in \mathcal{C}$ we have $|\ell(x) - \ell(y)| \leq L|x - y|$. This implies that $\|\nabla\ell(\theta)\|_2 \leq L$.
- A function $\ell : \mathcal{C} \mapsto \mathbb{R}$ is $\beta$-smooth if for all $x, y \in \mathcal{C}$ we have
$$\ell(x) \leq \ell(y) + \langle\nabla\ell(y), x - y\rangle + \frac{\beta}{2}\|x - y\|_2^2.$$
- A function $\ell : \mathcal{C} \mapsto \mathbb{R}$ is $\alpha$-strongly convex if for all $x, y \in \mathcal{C}$ we have
$$\ell(x) \geq \ell(y) + \langle\nabla\ell(y), x - y\rangle + \frac{\alpha}{2}\|x - y\|_2^2.$$

**Optimization.** Our entire focus during todays class is essentially an optimization question. How do we find the parameter $\theta^*$ which minimizes the loss function? The most common and flexible method is gradient descent. We will describe it in more detail in the later part of this lecture, as we discuss how to privatize this method. However, until then, we will simply claim there exists some black-box algorithm which is capable of optimizing convex loss functions non-privately, and describe how they can be used to solve ERM problems privately.

**Generalization.** The focus of todays lecture is ERM that is, we wish to find the parameter that minimizes the loss function on a given dataset. However, this is not typically the end goal of actual machine learning algorithms. In the real world, we often have training data generated from some distribution, and we want the learned classifier to perform well on new data generated from the same distribution. It is frequently the case that this will follow if one simply performs ERM on the training data this phenomenon is known as generalization. That said, this is not the focus of this lecture, and we focus only on the ERM problem.

**Privacy Considerations.** The goal of differentially private machine learning is to output a parameter vector $\theta$ which is differentially with respect to the training dataset $D$. Unfortunately, naively solving an ERM problem may reveal sensitive information. While we have discussed this numerous times in previous lectures, lets use a few more technical examples. Consider the (geometric) median in 1 dimension: if the number of points $n$ is odd, this will exactly output an element of the training dataset, clearly violating this privacy notion. Similar phenomena arise for SVMs: it is well known that the optimal parameter vector $\theta$ is entirely determined by the datapoints which are closest to it (known as the support vectors). Removing one of these points or adding a new point closer than previous support vectors could dramatically shift the parameter vector, again violating differential privacy. Thus we can see that, even from a technical perspective, a naive approach will not work.

**Definition 13.1 (DP-ERM)** *The problem of Differentially Private Empirical Risk Minimization is to design some $(\epsilon, \delta)/\epsilon$-DP algorithm $\mathcal{A}$ to make its output $\theta_{priv}$ to be closed to $\theta^*$. In details, we want to make the* **(expected) excess empirical risk** *($\mathbb{E}[L(\theta_{priv}, D) - L(\theta^*, D)]$) to be as small as possible, the expectation takes over the randomness of the algorithm $\mathcal{A}$.*

To solve DP-ERM, there are three types of approaches; output perturbation; objective perturbation; gradient perturbation. In this lecture, we will focus on the output perturbation. The following methods are proposed in [1, 3]. And we only consider the $(\epsilon, \delta)$-DP.

## 13.2 Output Perturbation

### 13.2.1 Black Box Output Perturbation

Before showing the algorithm we will make some assumptions

**Assumption 13.2** *We assume $\ell(\theta, x, y)$ is L-Lipschitz, $\beta$-smooth and $\alpha$-strongly convex for each $x, y$, and $\mathcal{C} = \mathbb{R}^d$.*

**Algorithm 1** Output Perturbation

1: Solve the exact optimization problem

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^d} L(\theta, D)$$

2: Output $\theta_{priv} = \theta^* + Z$, where $Z \sim \mathcal{N}(0, O(\frac{L^2 \log 1/\delta}{\alpha^2 n^2 \epsilon^2}) I_d)$.

Our first approach is based on the $\ell_2$-norm sensitivity of the black box optimization method. See Algorithm 1 for details.

**Theorem 13.3** *Algorithm 1 is $(\epsilon, \delta)$-DP.*

**Proof:** Our goal is to show the $\ell_2$-norm sensitivity of $\theta^*$ is bounded by $\frac{2L}{n\alpha}$. If we can show this, the proof is just followed by the Gaussian mechanism.

Let $\theta^* = \arg\min_{\theta \in \mathbb{R}^d} L(\theta, D)$ and $\theta^{*'} = \arg\min_{\theta \in \mathbb{R}^d} L(\theta, D')$, where $D$ and $D'$ are neighboring datasets whose $i$-th items are different. We first show the following result.

**Lemma 13.4** *Consider two differentiable functions $G(\theta)$ and $g(\theta)$, where $G(\theta)$ and $g(\theta)$ all are $\alpha$-strongly convex. If $\theta_1 = \arg\min G(\theta)$ and $\theta_2 = \arg\min G(\theta) + g(\theta)$. Then*

$$\|\theta_1 - \theta_2\|_2 \leq \frac{1}{\alpha} \max \|\nabla g(\theta)\|_2.$$

Before showing the proof of Lemma 13.4 we see how to use it to proof our main theorem. Let $G(\theta) = L(\theta, D)$ and we can see that $L(\theta, D') = L(\theta, D) + \frac{1}{n}(\ell(\theta, x_i', y_i') - \ell(\theta, x_i, y_i)) = G(\theta) + g(\theta)$ with $g(\theta) = \frac{1}{n}(\ell(\theta, x_i', y_i') - \ell(\theta, x_i, y_i))$. Thus we have

$$\|\theta^* - \theta^{*'}\|_2 \leq \frac{1}{\alpha} \max \frac{1}{n} \|(\nabla \ell(\theta, x_i', y_i') - \nabla \ell(\theta, x_i, y_i))\|_2 \leq \frac{2L}{n\alpha},$$

where the last inequality is due to the $L$-Lipschitz property.

Now we will proof Lemma 13.4. By the definition of $\theta_1$ and $\theta_2$ we have

$$\nabla G(\theta_1) = \nabla G(\theta_2) + \nabla g(\theta_2) = 0.$$

Since $G(\theta)$ is $\alpha$-strongly convex, we have

$$(G(\theta_1) - G(\theta_2))^T (\theta_1 - \theta_2) \geq \alpha \|\theta_1 - \theta_2\|_2^2.$$

Thus,

$$\alpha\|\theta_1-\theta_2\|_2^2\leq (G(\theta_1)-G(\theta_2))^T(\theta_1-\theta_2)\leq \|G(\theta_1)-G(\theta_2)\|_2\|\theta_1-\theta_2\|_2\leq \max\|g(\theta)\|_2\|\theta_1-\theta_2\|_2.$$

■

**Theorem 13.5** *For the expected excess empirical risk we have*

$$\mathbb{E}L(\theta_{priv},D)-L(\theta^*,D)\leq O(\frac{d\beta L^2\log\frac{1}{\delta}}{\alpha^2n^2\epsilon^2}). \tag{13.2}$$

**Proof:** By the $\beta$-smoothness property and the form of $\theta_{priv}$ we have

$$\begin{aligned}\mathbb{E}L(\theta_{priv},D)-L(\theta^*,D)&=\mathbb{E}L(\theta^*+Z,D)-L(\theta^*,D)\\&\leq\mathbb{E}[(\nabla L(\theta^*,D)^TZ+\frac{\beta\|Z\|_2^2}{2}]\\&=O(\frac{d\beta L^2\log\frac{1}{\delta}}{\alpha^2n^2\epsilon^2}).\end{aligned}$$

■

### 13.2.2 (Stochastic) Gradient Descent Perturbation

We can see that Algorithm 1 is just based on the sensitivity of $\theta^*$. The strength is it could be seen as a postprocessing step on the result of some optimization method, which is quite easy to implement. However, based on the proof we can see that there are so many shortness:

1. The conditions on the loss function in Assumption 13.2 is necessary. However, in practice, they may not be held, such as the strongly convex property. Thus, how to extend to the general convex loss function case?
2. The constraint set $\mathcal{C}=\mathbb{R}^d$ is quite strong. How to generalize to any bounded convex set?
3. In Algorithm 1 we need to get the exact optimal parameter $\theta^*$, which is impractical in practice.

In this section, we will focus on the term 2 and 3, and we will also try to relax the strongly convex property. The key observation is that, Algorithm 1 is based on the stability of black box optimization method, which is quite general. Thus, can we replace the black box optimization method to some specific one? For example, if we replace it by the (Stochastic) Gradient Descent, can we do better? Thus, the problem now is becoming analyzing the stability, or the $\ell_2$-norm sensitivity of the SGD method. For simplicity here we just focus on the GD method, see [2] for the stability of SGD. The general framework of this method can be found in Algorithm 2.

**Algorithm 2** Output Perturbation base on GD

1: Initialize a start parameter $\theta_0$
2: **for** $t=0,\cdots T-1$ **do**
3: $\quad\theta_t=\Pi_{\mathcal{C}}(\theta_{t-1}-\eta_{t-1}\nabla L(\theta_{t-1},D))$.
4: **end for**
5: Return $\theta_{priv}=\theta_T+Z$, where $Z\sim\mathcal{N}(0,O(\frac{\Delta_T^2\log\frac{1}{\delta}}{\epsilon^2})I_d)$ and $\Delta_T$ is the $\ell_2$-norm sensitivity of $\theta_T$.

**Theorem 13.6** *Algorithm 2 is $(\epsilon, \delta)$-DP.*

We will first consider the case where the loss function is strongly convex.

**Theorem 13.7 ([3])** *If the loss function is L-Lipschitz, $\beta$-smooth and $\alpha$-strongly convex. If we run gradient descent(GD) algorithm with constant step size $\eta \leq \frac{1}{\alpha+\beta}$ for $T$ steps, then the $\ell_2$-norm-sensitivity of GD for any bounded convex set $\mathcal{C}$,*

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

Based on this, we can get the expected excess risk. To show it, followed by the proof of Theorem 11.5 we have

$$\begin{aligned}
\mathbb{E}L(\theta_{priv}, D) - L(\theta^*, D) &= \mathbb{E}L(\theta_T + Z, D) - L(\theta_T, D) + L(\theta_T, D) - L(\theta^*, D) \\
&\leq \mathbb{E}[(\nabla L(\theta_T, D)^T Z + \frac{\beta\|Z\|_2^2}{2}] + L(\theta_T, D) - L(\theta^*, D) \\
&\leq O(\frac{d\beta\Delta_T^2 \log\frac{1}{\delta}}{\epsilon^2}) + L(\theta_T, D) - L(\theta^*, D) \\
&= O(\frac{d\log\frac{1}{\delta}}{n^2\epsilon^2}) + L(\theta_T, D) - L(\theta^*, D).
\end{aligned}$$

Note that $L(\theta_T, D) - L(\theta^*, D)$ is just the converge rate of the Gradient Descent method, which is $O(\beta\exp(-\frac{2\alpha\beta T}{(\alpha+\beta)^2}))$. Thus in total we have

**Theorem 13.8** *Under the same assumption as in Theorem 13.7*

$$\mathbb{E}L(\theta_{priv}, D) - L(\theta^*, D) \leq O(\frac{d\log\frac{1}{\delta}}{n^2\epsilon^2} + \exp(-\frac{2\alpha\beta T}{(\alpha+\beta)^2})).$$

*Moreover, if we take $T = O(\frac{\beta}{\alpha}\log n)$ then we have*

$$\mathbb{E}L(\theta_{priv}, D) - L(\theta^*, D) \leq O(\frac{d\log\frac{1}{\delta}}{n^2\epsilon^2}).$$

Next, we extend to the general convex setting.

**Theorem 13.9 ([3])** *If the loss function is L-Lipschitz and $\beta$-smooth. If we run gradient descent(GD) algorithm with constant step size $\eta \leq \frac{1}{\beta}$ for $T$ steps, then the $\ell_2$-norm-sensitivity of GD for any bounded convex set $\mathcal{C}$,*

$$\Delta_T \leq \frac{3LT}{n\beta}.$$

The similar as the strongly convex case, for the excess empirical risk we have

$$\mathbb{E}L(\theta_{priv}, D) - L(\theta^*, D) \leq O(\frac{dT^2\log\frac{1}{\delta}}{n^2\epsilon^2}) + L(\theta_T, D) - L(\theta^*, D).$$

For smooth and Lipschitz loss function, the converge rate of the Gradient Descent is bounded by $O(\frac{\beta}{T})$. Thus we have

**Theorem 13.10** *Under the same assumption as in Theorem 13.9*

$$\mathbb{E}L(\theta_{priv}, D) - L(\theta^*, D) \leq O(\frac{dT^2 \log \frac{1}{\delta}}{n^2\epsilon^2} + \frac{1}{T}).$$

*Set* $T = O((\frac{n^2\epsilon^2}{d \log 1/\delta})^{\frac{1}{3}})$ *we have*

$$\mathbb{E}L(\theta_{priv}, D) - L(\theta^*, D) \leq O((\frac{d \log 1/\delta}{n^2\epsilon^2})^{\frac{1}{3}}).$$

We can see that now we extend to general convex set $\mathcal{C}$ and the he convex case. Compared with the bound of $O(\frac{d \log 1/\delta}{n^2\epsilon^2})$ for the strongly convex case, the bound of $O((\frac{d \log 1/\delta}{n^2\epsilon^2})^{\frac{1}{3}})$ is larger. However, there are still some issues on this approach:

1. Although we can handle general convex set $\mathcal{C}$, the perturbed parameter $\theta_{priv}$ may not lie in the convex set $\mathcal{C}$.
2. As we will see later, the upper bound in the convex can be further improved.
3. In practice, the iteration number is quite hard to select for general convex case.

We will see how to address these two issues by using the second type of approach: Objective Perturbation Method.

# References

[1] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[2] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

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