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

Lecture 16: Differentially Private Empirical Risk Minimization: IV

Lecturer: Di Wang *Scribes: Di Wang*

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

So far we have introduced three types of approaches: Output Perturbation, Objective Perturbation and Private Gradient descent. For the utilities, if the loss function is general convex, the best upper bound we have so far is $O(\frac{\sqrt{d\log\frac{1}{\delta}}}{n\epsilon})$ (if we omit other terms) under (ϵ,δ)-DP, where d is the dimensionality of the space/sample and n is the sample size; and if the loss function is strongly convex, the best utility is $O(\frac{d\log\frac{1}{\delta}}{n^2\epsilon^2})$. This means that our methods perform well if $n \gg d$ as the utility will be small. However, in many real-world data, this assumption may not hold, since may datasets are high dimensional where $d \gg n$. Thus, our question is, how about the case in the high dimensional space? In this lecture, we will mainly introduce three general approaches.

16.1 DP-Frank Wolfe

Now we focus on the ERM problem with constraint, *i.e.,*

$$\theta^* = \arg\min_{\theta\in\mathcal{C}} L(\theta, D) = \frac{1}{n}\sum_{i=1}^{n}\ell(\theta, x_i).$$

The motivation of the Frank-Wolfe method comes from the projected gradient descent (PGD), recall that in each iteration of PGD we update:

$$\theta_{t+1} = \Pi_{\mathcal{C}}(\theta_t - \eta\nabla L(\theta_t, D)) = \arg\min_{\theta\in\mathcal{C}}\|\theta - (\theta_t - \eta\nabla L(\theta_t, D))\|_2^2.$$

However, the projection step may be quite costy. The idea of the FW method is that, instead of minimizing a quadratic function in each step for the projection step, we just want to minimize a linear function:

$$\theta_{t+0.5} = \arg\min_{\theta\in\mathcal{C}}\langle\theta, \nabla L(\theta_t, D)\rangle.$$

This step can be interpreted as Minimizing the linear approximation of the problem given by the first-order Taylor approximation of $L(\theta, D)$ around θ_t. The full FW method could be found in Algorithm 7.

Algorithm 1 Frank-Wolfe Method

1: Denote the initial point θ_1.
2: **for** $i = 1, 2, \cdots, T$ **do**
3: $s_t = \arg\min_{\theta\in\mathcal{C}}\langle\theta, \nabla L(\theta_t, D)\rangle$.
4: Set the stepsize $\eta_t = \frac{2}{t+2}$, and update $\theta_{t+1} = (1-\eta_t)\theta_t + \eta_t s_t$.
5: **end for**
6: Return θ_{T+1}.

Now we just give a toy example to show the strength of the FW method. Consider $\mathcal{C}$ is a polyhedron, i.e., it is the convex hull of some finite vertices $\mathcal{S}$ (suppose it is k), such as the ℓ_1-norm ball. Then the minimization step only cost $O(k \cdot d)$ time since the minimal parameter must be one of the vertices (by the theory of linear programming).

Moreover, note that in this case, if we want to privatize this minimization step. Our goal is to select the best vertex privately. That is, our problem now becoming to a **Selection Problem**! Thus, we prefer to use the Exponential Mechanism or the Noisy-max mechanism as their error bound only depends on $\frac{\log k}{\epsilon}$ instead of $\frac{\text{poly}(k)}{\epsilon}$ by using the Laplacian or the Gaussian mechanism. This is the whole idea of DP Frank-Wolfe method, proposed by [2].

Algorithm 2 Frank-Wolfe Method

1: Denote the initial point θ_1.
2: **for** $i = 1, 2, \cdots, T$ **do**
3: For all $s \in \mathcal{S}$, $\alpha_s = \langle s, \nabla L(\theta_t, D)\rangle + \text{Lap}(\frac{2\|\mathcal{C}\|_1 L\sqrt{8T\log 1/\delta}}{n\epsilon})$.
4: Denote $s_t = \min_{s\in\mathcal{S}} \alpha_s$.
5: Set the stepsize $\eta_t = \frac{2}{t+2}$, and update $\theta_{t+1} = (1-\eta_t)\theta_t + \eta_t s_t$.
6: **end for**
7: Return θ_{T+1}.

Assumption 16.1 *We assume $\ell(\theta, x)$ is β-smooth w.r.t ℓ_2-norm and L-Lipschitz w.r.t ℓ_1-norm,* i.e., $\|\nabla\ell(\theta, x)\|_\infty \leq L$. *We also assume that $\mathcal{C}$ has bounded ℓ_1-norm diameter, that is $\|\mathcal{C}\|_1 = \max_{s\in\mathcal{C}}\|s\|_1$ is bounded.*

Theorem 16.2 *For any $0 < \epsilon, \delta < 1$, Algorithm 3 is (ϵ, δ)-DP.*

Proof: By the Advanced composition theorem, we can see that it is sufficient to show that each iteration is $(\frac{\epsilon}{2\sqrt{2T\log\frac{1}{\delta}}}, 0)$-DP. Now we study the sensitivity of $g(D) = \langle s, \nabla L(\theta, D)\rangle$ with the fixed s, if $D \sim D'$ are neighboring datasets and differs in the i-th item then we have

$$|g(D) - g(D')| = |\langle s, \frac{1}{n}\ell(\theta, x_i) - \frac{1}{n}\ell(\theta, x_i')\rangle| \leq 2L\|s\|_1 \leq 2L\|\mathcal{C}\|_1.$$

Thus, by the Laplace mechanism, each iteration is $(\frac{\epsilon}{2\sqrt{2T\log\frac{1}{\delta}}}, 0)$-DP. ■

Theorem 16.3 *Under Assumption 16.7, we have with probability at least $1 - \tau$,*

$$L(\theta_{T+1}, D) - L(\theta^*, D) \leq O(\frac{1}{T} + \frac{L\log k\log\frac{T}{\tau}\sqrt{T\log\frac{T}{\delta}}}{n\epsilon}).$$

Specifically, when $T = O((n\epsilon)^{\frac{2}{3}})$, we have $L(\theta_{T+1}, D) - L(\theta^*, D) \leq O(\frac{\log k\sqrt{\log\frac{n}{\delta}\log^2\frac{T}{\tau}}}{(n\epsilon)^{2/3}})$.

Proof: Here we just give a sketch of proof. The proof mainly consists of two parts. The first one is the noisy version of the original FW method, and the second one is the error of s_t w.r.t the optimal one. For fixed θ, we denote $s_t^* = \arg\min_{\theta\in\mathcal{C}}\langle\theta, \nabla L(\theta_t, D)$. That is the optimal solution without noise. Then by the error bound of the noisy-max mechanism (Theorem 5.10 in Lecture 5) we have with probability at least $1 - \tau$

$$\langle s_t, \nabla L(\theta_t, D) - \langle s_t^*, \nabla L(\theta_t, D) \leq O(\frac{\log k\|\mathcal{C}\|_1 L\log\frac{1}{\tau}\sqrt{T\log\frac{1}{\delta}}}{n\epsilon}). \tag{16.1}$$

And by the converge rate of the noisy Frank-Wolf method, we can get the result. See [2] for details. ■

We can see now our utility becomes to $O(\frac{\log k\sqrt{\log\frac{n}{\delta}\log^2\frac{T}{\tau}}}{(n\epsilon)^{2/3}})$, which could be much smaller than the upper bounds in our previous lectures. Consider the LASSO as an example:

LASSO Consider LASSO where $\ell(\theta, x, y) = (\langle\theta, x\rangle - y)^2$ and $\mathcal{C} = \{\theta : \|\theta\|_1 \leq r\}$. Then if each $|x_{i,j}| \leq O(1)$ and $|y_i| \leq O(1)$, then we can verify that Assumption 16.7 holds. Moreover, by the previous Theorem we have the utility will be $O(\frac{\log d\sqrt{\log\frac{n}{\delta}\log^2\frac{T}{\tau}}}{(n\epsilon)^{2/3}})$.

16.2 DP-IHT

Besides the DP-FW method, there is another method which is called DP Iterative Hard Thresholding (IHT). The IHT method first proposed in the compressed sensing field, which is later extended to problems in statistics. Many high dimensional sparse models in statistics can be formulated as the following:

$$\tilde{\theta} = \min_{\theta\in\mathcal{C}} L(\theta, D) = \frac{1}{n}\sum_{i=1}^{n}\ell(\theta, x_i), \tag{16.2}$$

where $\mathcal{C}$ is some sparsity constraint, *i.e.,* $\mathcal{C} = \{\theta : \|\theta\|_0 \leq s^*, \|\theta\|_2 \leq 1\}$. Note that here $\mathcal{C}$ is non-convex and $\|\theta\|_0$ means the number of non-zero entries in θ. Here we give an example:

Sparse Linear Regression Consider we have the following model, there is a sparse θ^* where $\|\theta^*\|_0 \leq s^*$ and ζ is the random noise with $\mathbb{E}[\zeta] = 0$,

$$y = \langle\theta^*, x\rangle + \zeta. \tag{16.3}$$

Thus, to estimate θ^*, the direct approach is to solve the following problem

$$\tilde{\theta} = \min_{\theta\in\mathcal{C}} L(\theta, D) = \frac{1}{n}\sum_{i=1}^{n}(\langle x_i, \theta\rangle - y_i)^2.$$

The idea of iterative hard thresholding is that we perform the projection step on the set $\mathcal{C}$. However, since $\mathcal{C}$ is non-convex, in general this step could be NP-hard in general.

Algorithm 3 Iterative Hard Thresholding

1: Denote the initial point θ_1.
2: **for** $i = 1, 2, \cdots, T$ **do**
3: $\quad\theta_{t+0.5} = \theta_t - \eta\nabla L(\theta_t, D)$.
4: $\quad$Let $\theta_{t+1} = \text{trunc}(\theta_{t+0.5}, \mathcal{S}^{t+0.5})$.
5: **end for**
6: Return θ_{T+1}.

To ensure the sparsity of our estimator, after getting $\theta_{t+0.5}$, we need to use the hard thresholding operation. More specifically, we first find the set $\mathcal{S}^{t+0.5} \subseteq [d]$ of indices j corresponding to the top s largest $|\theta_{t+0.5,j}|$ (we denote $\mathcal{S}^{t+0.5} = \text{supp}(\theta_{t+0.5}, s)$[1]), and make the value of the remaining entries $\theta_{t+0.5,j}$ for $j \in [d]\backslash\mathcal{S}^{t+0.5}$

[1]In general, given a vector $v \in \mathbb{R}^d$ and an integer s, function $\text{supp}(v, s)$ returns a set of s number ofis indices corresponding to the top s largest value among $\{|v_j|, j \in [d]\}$.

be 0 (we denote $\theta_{t+1} = \text{trunc}(\theta_{t+0.5}, \mathcal{S}^{t+0.5})$[2]). The sparsity level s controls the sparsity of the estimator and the estimation error.

Our goal is to estimate θ^* privately. And the most direct approach is privatize the IHT method. The first approach, given by [4, 5] is based on the gradient perturbation. The idea behind the algorithm is quite straightforward, without hard thresholding we add Gaussian noise to each coordinate. However, if we perform the hard thresholding then the effective dimension now is becoming to s instead of d. That is the variance of noise now depend on $s \log d$ instead of d.

Algorithm 4 DP-Iterative Hard Thresholding

1: Denote the initial point θ_1.
2: **for** $i = 1, 2, \cdots, T$ **do**
3: $\quad \theta_{t+0.5} = \theta_t - \eta(\nabla L(\theta_t, D) + Z_t)$, where $Z_t \sim \mathcal{N}(0, \sigma^2 I_d)$.
4: $\quad$ Let $\theta_{t+1} = \text{trunc}(\theta_{t+0.5}, \mathcal{S}^{t+0.5})$.
5: **end for**
6: Return θ_{T+1}.

Theorem 16.4 *If we assume that each $\ell(\theta, x_i)$ is L-Lipschitz, then for any $0 < \epsilon, \delta < 1$, Algorithm 4 is (ϵ, δ)-DP if for some constant $c > 0$,*

$$\sigma^2 = c\frac{T \log \frac{1}{\delta}}{n^2\epsilon^2}. \tag{16.4}$$

The proof is the same as the DP Gradient Descent since the posts-processing step keeps DP.

Theorem 16.5 *For sparse linear regression, for some T, s, η we have*

$$\|\theta_{T+1} - \theta^*\|_2^2 \leq O(\frac{s^* \log d}{n} + \frac{s^2 \log^2 d}{n^2\epsilon^2}).$$

The second approach, given by [1], is to privatize the hard thresholding step. Note that the hard thresholding step is selecting s items. Thus, we can use s-times private selection algorithms. That is Algoritm 5. In total, the second version of DP-IHT is in Algorithm 6. Here we just consider the sparse linear regression as an example.

Algorithm 5 Peeling [1]

1: **Input:** Vector $v = v(D) \in \mathbb{R}^d$ which depends on the data D, sparsity s, privacy parameter ϵ, δ, and noise scale λ.
2: Initialize $S = \emptyset$.
3: **for** $i = 1 \cdots s$ **do**
4: $\quad$ Generate $w_i \in \mathbb{R}^d$ with $w_{i,1}, \cdots, w_{i,d} \sim \text{Lap}(\frac{2\lambda\sqrt{3s \log \frac{1}{\delta}}}{\epsilon})$.
5: $\quad$ Append $j^* = \arg\max_{j \in [d] \setminus S} |v_j| + w_{i,j}$ to S.
6: **end for**
7: Generate $\tilde{w} \in \mathbb{R}^d$ with $\tilde{w}_1, \cdots, \tilde{w}_d \sim \text{Lap}(\frac{2\lambda\sqrt{3s \log \frac{1}{\delta}}}{\epsilon})$.
8: **return** $v_S + \tilde{w}_S$.

[2] In general, given a vector $v \in \mathbb{R}^d$ and a set of indices $\mathcal{S} \subseteq [d]$, function $\text{trunc}(v, \mathcal{S}) \in \mathbb{R}^d$, where $[\text{trunc}(v, \mathcal{S})]_j = v_j$ if $j \in \mathcal{S}$ and $[\text{trunc}(v, \mathcal{S})]_j = 0$ otherwise.

Algorithm 6 Heavy-tailed Private Sparse Linear Regression

1: Split the data $\tilde{D}$ into T parts $\{\tilde{D}_t\}_{t=1}^T$, each with $m = \frac{n}{T}$ samples.
2: **for** $t = 1, \cdots, T$ **do**.
3: Denote $\theta^{t+0.5} = \theta^t - \frac{\eta_0}{m}\sum_{x\in\tilde{D}_t}\tilde{x}(\langle\tilde{x}, \theta^t\rangle - \tilde{y})$
4: Let $\theta^{t+1} = \text{Peeling}(\theta^{t+0.5}, D_t, s, \epsilon, \delta, \frac{2K^2C\eta_0(\sqrt{s}+1)}{m})$.
5: Project θ^{t+1} to the ℓ_2-norm ball with radius C.
6: **end for**
7: **return** θ^{T+1}.

Theorem 16.6 *If we assume each* $\|x\|_{\max} \leq K$ *and* $|y| \leq K$*, then for any* $0 < \epsilon, \delta < 1$*, Algorithm 6 is* (ϵ, δ)*-DP. Moreover, with some* s, T, η *we have with high probability*

$$\|\theta_{T+1} - \theta^*\|_2^2 \leq O(\frac{s^* \log d}{n} + \frac{s^2 \log^2 d}{n^2\epsilon^2}).$$

16.3 Exploring the Geometric Structure

The previous two approaches all are based on the selection mechanism or projecting on a lower dimensional space. In this section, we provide another approach which use the Gaussian width of the underlying convex set $\mathcal{C}$. We will assume that the set $\mathcal{C}$ is closed and convex. The following results are given by [3].

Minkowski norm w.r.t a set $\mathcal{C}$: For any vector $v \in \mathbb{R}^p$, the Minkowski norm (denoted by $\|v\|_{\mathcal{C}}$) w.r.t a centrally symmetric convex set $\mathcal{C}$ is defined as follows.

$$\|v\|_{\mathcal{C}} = \min\{r \in \mathbb{R} : v \in r\mathcal{C}\}.$$

Dual of Minkowski norm w.r.t a set $\mathcal{C}$: For any vector $v \in \mathbb{R}^p$, the dual of Minkowski norm (denoted by $\|v\|_{\mathcal{C}^*}$) is defined by

$$\|v\|_{\mathcal{C}^*} = \max_{\theta\in\mathcal{C}} |\langle v, w\rangle|.$$

Note that if $\mathcal{C}$ is the unit ℓ_p-norm ball, then the Minkowski norm w.r.t a set $\mathcal{C}$ is just the ℓ_p-norm. And the dual norm will be the ℓ_q-norm where p, q satisfies $\frac{1}{p} + \frac{1}{q} = 1$ (if $p = 1$ then $q = \infty$). The following inequality is the key property we will use:

$$\langle x, y\rangle \leq \|x\|_{\mathcal{C}}\|y\|_{\mathcal{C}^*}.$$

Gaussian Width of a set $\mathcal{C}$: Let $z \sim \mathcal{N}(0, I_d)$ be a Gaussian vector in $\mathbb{R}^d$. The Gaussian width of a set $\mathcal{C}$ is defined as $G_{\mathcal{C}} = \mathbb{E}\|z\|_{\mathcal{C}^*} = \mathbb{E}[\sup_{w\in\mathcal{C}}\langle z, w\rangle]$.

Note that for some $\mathcal{C} \subseteq \mathbb{R}^d$, its Gaussian width could be much smaller than d. For example:

Fact 1: If $\mathcal{C}$ is a symmetric polytope with k vertices. Then we have $G_{\mathcal{C}} = O(\|\mathcal{C}\|_2\sqrt{\log k})$.

Fact 2: For a vector $\theta \in \mathbb{R}^d$ and a parameter k, the grouped ℓ_1-norm is defined as

$$\|\theta\|_{k,\ell_{1,2}} = \sum_{i=1}^{\frac{d}{k}} \sqrt{\sum_{j=(i-1)k+1}^{\min\{ik,d\}} |\theta_j|^2}.$$

Let $\mathcal{C}$ be the unit ball w.r.t the grouped ℓ_1-norm. Then we have $G_{\mathcal{C}} = \sqrt{k \log \frac{d}{k}}$.

Actually, we can think the Gaussian width as an effective dimension of the set $\mathcal{C}$. In the following, we will show that for the objective perturbation method, we can get an upper bound of utility that only depends on the Gaussian width.

Assumption 16.7 *We assume $\ell(\cdot, x, y)$ is L-Lipschitz, twice differentiable and β-smooth.* **Moreover, we assume that the rank of the Hessian matrix for each $\ell(\theta, x, y)$ is at most 1**.

Algorithm 7 Objective Perturbation

1: For (ϵ, δ)-DP, sample $b \sim \mathcal{N}(0, \frac{10L^2 \log \frac{1}{\delta}}{\epsilon^2} I_d)$. For ϵ-DP, sample b from the Gamma distribution with density $h(b) \propto \exp(-\frac{\epsilon}{2L}\|b\|_2)$
2: Get the exact solution of the following optimization problem

$$\theta_{priv} = \arg\min_{\theta \in \mathcal{C}} L^+(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, x_i, y_i) + \frac{\lambda}{2n}\|\theta\|_2^2 + \frac{1}{n}\langle b, \theta\rangle,$$

where λ is some parameter.

Theorem 16.8 *Under Assumption 16.7, Algorithm 7 is $\ln(1 + \frac{2\beta}{\lambda}) + \epsilon$-DP or $(\frac{\epsilon}{2} + \ln(1 + \frac{2\beta}{\lambda}), \delta)$-DP.*

Theorem 16.9 *Under Assumption 16.7, for (ϵ, δ)-DP, we have*

$$\mathbb{E}L(\theta_{priv}, D) - \min_{\theta \in \mathcal{C}} L(\theta, D) \leq O(\frac{LG_{\mathcal{C}}\sqrt{\log 1/\delta} + \beta\|\mathcal{C}\|_2^2}{\epsilon n}). \tag{16.5}$$

Proof: Before that we recall some definitions

$$\theta^* = \arg\min_{\theta \in \mathcal{C}} L(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, x_i, y_i)$$

$$\theta_{priv} = \arg\min_{\theta \in \mathcal{C}} L^+(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, x_i, y_i) + \frac{\lambda}{2n}\|\theta\|_2^2 + \frac{1}{n}\langle b, \theta\rangle$$

$$\bar{\theta} = \arg\min_{\theta \in \mathcal{C}} \bar{L}(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta, x_i, y_i) + \frac{\lambda}{2n}\|\theta\|_2^2$$

To show the utility, we have

$$\begin{aligned} L(\theta_{priv}, D) - L(\theta^*, D) &\leq \bar{L}(\theta_{priv}, D) - \bar{L}(\bar{\theta}, D) + \bar{L}(\bar{\theta}, D) - \bar{L}(\theta^*, D) + \frac{\lambda}{2n}\|\theta^*\|_2^2 - \frac{\lambda}{2n}\|\theta_{priv}\|_2^2 \\ &\leq \bar{L}(\theta_{priv}, D) - \bar{L}(\bar{\theta}, D) + \frac{\lambda}{2n}\|\theta^*\|_2^2 - \frac{\lambda}{2n}\|\theta_{priv}\|_2^2. \end{aligned}$$

Moreover we have

$$\begin{aligned} & L^+(\bar{\theta}, D) \geq L^+(\theta_{priv}, D) \\ \iff & \bar{L}(\bar{\theta}, D) + \frac{1}{n}\langle b, \bar{\theta}\rangle \geq \bar{L}(\theta_{priv}, D) + \frac{1}{n}\langle b, \theta_{priv}\rangle \\ \iff & \bar{L}(\theta_{priv}, D) - \bar{L}(\bar{\theta}, D) \leq \frac{1}{n}\langle b, \bar{\theta} - \theta_{priv}\rangle. \end{aligned}$$

Taking the expectation we have

$$\mathbb{E}L(\theta_{priv}, D) - L(\theta^*, D) \leq \frac{1}{n}\mathbb{E}\langle b, \bar{\theta} - \theta_{priv}\rangle + \frac{\lambda}{2n}\|\theta^*\|_2^2 \leq O(\frac{\sigma G_{\mathcal{C}}}{n} + \frac{\beta\|\mathcal{C}\|_2^2}{n\epsilon}) = O(\frac{LG_{\mathcal{C}}\sqrt{\log 1/\delta} + \beta\|\mathcal{C}\|_2^2}{\epsilon n}).$$

■

References

[1] T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *arXiv preprint arXiv:1902.04495*, 2019.

[2] Kunal Talwar, Abhradeep Guha Thakurta, and Li Zhang. Nearly optimal private lasso. *Advances in Neural Information Processing Systems*, 28:3025–3033, 2015.

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

[4] Di Wang and Jinhui Xu. On sparse linear regression in the local differential privacy model. In *International Conference on Machine Learning*, pages 6628–6637. PMLR, 2019.

[5] Lingxiao Wang and Quanquan Gu. Differentially private iterative gradient hard thresholding for sparse learning. In *28th International Joint Conference on Artificial Intelligence*, 2019.