CS291A Introduction to Differential Privacy **Fall 2021**

Lecture 9: DPML I: Introduction and Posterior Sampling (October 25)

Lecturer: Yu-Xiang Wang *Scribes: Siqi Ouyang*

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

9.1 Introduction to Differentially Private Machine Learning

For typical supervised machine learning problem, labeled data $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ are i.i.d. samples of some data distribution $\mathcal{D}$. From the collected data, we build machine learning models that we hope can generalize to new unseen x. This can also be applied to other unsupervised settings. In general, the input is a dataset and the output is a set of parameters.

Differentially private machine learning basically means that two inputs that differ in only one datapoint lead to similar outputs.

Example 9.1 (Recommendation System)**.** *Your neighbors could know your shopping history by monitoring your network or delivery trucks. Then he/she could create a dummy account to buy the same stuff you bought. A standard recommendation system could identify this dummy account as similar to yours, so this person could know what you like based on the recommended products afterwards.*

However, if the recommendation system is trained differentially privately, then the adversary cannot infer whether a particular person is present and any other information.

Example 9.2 (Federated Learning)**.** *Federated learning requires training a machine learning model without actually gathering the information, which may be quite sensitive, from users. However, the model itself could still contain information about individuals. DP just eliminates the risk from output itself, which could be combined with federated learning techniques.*

9.1.1 Notations and Problem Setup

The data space is usually $\mathcal{X} \times \mathcal{Y} = \mathcal{Z}$ where each datapoint is a pair $(x, y) \in \mathcal{Z}$. The hypothesis space (or model space) is $\mathcal{H}$ which contains all possible classifiers $h \in \mathcal{H}$. $\mathcal{H}$ is defined by the parameter space, which means each classifier h is one-to-one mapped to a set of parameters θ. The loss function can be a 0-1 loss $\mathbb{1}(h(x) \neq y) = \ell(h, (x, y))$ or square loss or some other functions.

The learning algorithm A takes a dataset $(z_1, \cdots, z_n)$ as input and outputs a classifier $h \in \mathcal{H}$. For example, Empirical Risk Minimization (ERM) algorithm finds the empirically optimal classifier

$$\hat{h} = \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{i=1}^{n} \ell(h, z_i). \tag{9.1}$$

The goal of learning is when $n \to \infty$, the excess error $R(A(\text{data})) - \min_{h \in \mathcal{H}} R(h) \to 0$.

Example 9.3 (Linear/Logistic Regression). *For linear regression, the data space is $\mathcal{X} \subseteq \mathbb{R}^d, \mathcal{Y} = \mathbb{R}$. The hypothesis space is described as $h(x) = \theta^\top x$ with $\theta \in \mathbb{R}^d$. The loss function is square loss $(y - \theta^\top x)^2$ and the learning algorithm finds $\min_\theta \sum_i (y_i - \theta^\top x)^2$. For logistic regression, we have $\mathcal{Y} = \{0, 1\}$ and $h(x) = \frac{e^{\theta^\top x}}{1+e^{\theta^\top x}} = \hat{p}_\theta(x)$. The loss function is cross entropy loss $-y \log \hat{p}_\theta(x) - (1-y)\log(1-\hat{p}_\theta(x))$. Additionally, we can add regularization like L2 norm of θ.*

Example 9.4 (PAC Learning/Binary Classification). $\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{0, 1\}, h(x) = \theta^\top x, \ell(h, (x, y)) = \mathbb{1}(h(x) \neq y)$.

Example 9.5 (K-Means Clustering). $\mathcal{X} = \mathbb{R}^d, \theta = (\theta_1, \cdots, \theta_k) \in \mathbb{R}^{d \times k}, \ell(\theta, x) = \min_{1 \leq j \leq k} \|x - \theta_j\|_2^2$. *The learning algorithm can be a approximation since it's not a convex problem.*

Example 9.6 (Recommendation System/Matrix Factorization). *The data is a matrix M with each row describes a user's attributes. Some of the entries are filled and the other are unknown. The hypothesis space is a factorization $UV^\top$ with low rank to approximate the original M. The loss can be square loss on those known entries.*

Definition 9.7 (Learnability). *A problem is learnable if $\exists A$ s.t. $\forall \mathcal{D}$ and $(z_1, \cdots, z_n) \sim \mathcal{D}^n$, we have $R(A(z_1, \cdots, z_n)) - R(h^*) \to 0$ as $n \to \infty$.*

Definition 9.8 (Sample Complexity). *For a particular algorithm A, if its sample complexity is $n = O(1/\alpha^2)$, we have $|R(A(Z_{1:n})) - R(h^*)| \leq \alpha$. The sample complexity is the amount of data needed to achieve a particular error rate.*

Definition 9.9 (Uniform Convergence). *With high probability, $\sup_{h \in \mathcal{H}} |\hat{R}_n(h) - R(h)| \leq \alpha(n)$ where $\alpha(n) \to 0$ as $n \to 0$.*

For private learnability, we just need the algorithm A to satisfy (ϵ, δ)-DP. The sample complexity of private learning is the number of samples needed to achieve a particular error rate under some privacy budget.

9.1.2 PAC Learning with Finite Hypothesis Space

We know $\ell(h, (x, y)) \in [0, 1]$. For n i.i.d. samples $(x_1, y_1), \cdots, (x_n, y_n)$, fix $h \in \mathcal{H}$, by Hoeffding's inequality then we know

$$\left| \frac{1}{n} \sum_{i=1}^n \ell(h, (x_i, y_i)) - \mathbb{E}\left[\ell(h, (x, y))\right] \right| \leq \sqrt{\frac{2 \log \frac{1}{\beta}}{n}} \quad \text{w.p. } 1 - \beta. \tag{9.2}$$

Then for all $h \in \mathcal{H}$, we apply union bound and get

$$\sup_{h \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^n \ell(h, (x_i, y_i)) - \mathbb{E}\left[\ell(h, (x, y))\right] \right| \leq \sqrt{\frac{2 \log \frac{|\mathcal{H}|}{\beta}}{n}} \quad \text{w.p. } 1 - \beta. \tag{9.3}$$

Let $\hat{h} = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}_n(h)$ and we can bound the difference between $R(\hat{h})$ and $R(h^*)$,

$$R(\hat{h}) - R(h^*) = (R(\hat{h}) - \hat{R}_n(\hat{h})) + (\hat{R}_n(\hat{h}) - \hat{R}_n(h^*)) + (\hat{R}_n(h^*) - R(h^*)) \tag{9.4}$$

$$\leq \sqrt{\frac{2 \log \frac{|\mathcal{H}|}{\beta}}{n}} + 0 + \sqrt{\frac{2 \log \frac{|\mathcal{H}|}{\beta}}{n}}. \tag{9.5}$$

We can apply exponential mechanism to obtain a set of parameters satisfying ϵ-DP. The hypothesis $\hat{h}$ is sampled with probability proportional to $\exp\left(-\frac{\Delta \hat{R}_n(h)}{2\epsilon}\right)$ where $\Delta = \frac{2}{n}$ for replacement. By applying the utility of exponential mechanism, we know that

$$\hat{R}_n(\hat{h}) - \min_{h\in\mathcal{H}} \hat{R}_n(h) \leq \frac{4}{n\epsilon}\left(\log|\mathcal{H}| + \log\frac{1}{\beta}\right) \quad \text{w.p. } 1-\beta. \tag{9.6}$$

This error goes to 0 much faster than 9.5. Thus we are not losing accuracy if we have enough data.

9.1.3 Continuous Hypothesis Space with Finite VC-dim

Short answer: No DP algorithms can learn the VC class. Here is an example.

Let's say we have $\mathcal{X} = [0,1], \mathcal{Y} = \{0,1\}$ and $h(x) = \mathbb{1}(x \geq h)$. The VC-dim of this hypothesis space is exactly one, which indicates

$$\sup_{h\in\mathcal{H}} \left|\hat{R}_n(h) - R(h)\right| \leq O\left(\sqrt{\frac{1}{n}}\right). \tag{9.7}$$

In nonprivate case, sample complexity $n = O(1/\alpha^2)$. If additionally $\exists h^*$, $R(h^*) = 0$, then $n = O(1/\alpha)$.

Claim 9.10. *For $\epsilon < +\infty$, there is no ϵ-DP algorithm A such that $R(A(Data)) - R(h^*) < 0.9$.*

Proof. The proof is from [CH11,BNS13].

Let $\eta = e^{-\epsilon n}$ where n is the size of dataset and $k = 1/\eta$. Let $h_i = (i - \frac{1}{2})\eta$ for $1 \leq i \leq k$. We construct the i_{th} dataset $D_i = D_i^- \cup D_i^+$ where

$$D_i^- \sim \left\{(x,-) \mid x \in (h_i - \frac{1}{3}\eta, h_i)\right\} \tag{9.8}$$

$$D_i^+ \sim \left\{(x,+) \mid x \in (h_i, h_i + \frac{1}{3}\eta)\right\}. \tag{9.9}$$

If A is successful for D_i, then w.h.p. $A(D_i) \in \left(h_i - \frac{\eta}{3}, h_i + \frac{\eta}{3}\right)$. Or in another word,

$$\mathbb{E}\left[\hat{R}_n(A(D_i), D_i)\right] \leq 0.1. \tag{9.10}$$

Assume A is successful on $D_2, \cdots, D_k$. Then the probability it fails on D_1 equals

$$\Pr\left[A(D_1) \notin \left(h_1 - \frac{\eta}{3}, h_1 + \frac{\eta}{3}\right)\right] \geq \sum_{i=2}^{k} \Pr\left[A(D_1) \in \left(h_i - \frac{\eta}{3}, h_i + \frac{\eta}{3}\right)\right] \tag{9.11}$$

$$\geq \sum_{i=2}^{k} e^{-\epsilon n} \Pr\left[A(D_i) \in \left(h_i - \frac{\eta}{3}, h_i + \frac{\eta}{3}\right)\right] \tag{9.12}$$

$$\geq (k-1)e^{-\epsilon n} \cdot 0.9 \approx 0.9. \tag{9.13}$$

□

This proof shows that the ϵ-DP cannot learn well on the first dataset if it learns well on all the other dataset.

Is this the issue of pure DP mechanism? It depends. [BNSV15] says this problem is differentially privately learnable once $\delta(n) = \Omega(1/n)$, but this is a rather bad δ.

There are remedies to the problem. As stated in [WLF16], if we require the loss function to be Lipschitz or some regularity conditions on the probability distribution of data, most of the problems are learnable by an exponential mechanism.

9.2 Posterior Sampling

From Baysian perspective, I have prior belief $\pi(\theta)$. When I collect some i.i.d. data $z_1, \cdots, z_n \sim D$, I update our belief $\pi(\theta \mid z_1, \cdots, z_n)$ by applying Bayesian rule

$$\pi(\theta \mid z_{1:n}) = \frac{\Pr[z_{1:n} \mid \theta]\,\pi(\theta)}{\int_{\theta'} \Pr[z_{1:n} \mid \theta']\,\pi(\theta')} \tag{9.14}$$

$$= \frac{\pi(\theta)\prod_{i=1}^n \Pr[z_i \mid \theta]}{\int_{\theta'} \Pr[z_{1:n} \mid \theta']\,\pi(\theta')} \tag{9.15}$$

$$= \frac{e^{\sum_{i=1}^n \log \Pr[z_i \mid \theta] + \log \pi(\theta)}}{\int_{\theta'} \Pr[z_{1:n} \mid \theta']\,\pi(\theta')} \tag{9.16}$$

$$= \frac{e^{-\sum_{i=1}^n \ell(z_i,\theta) + r(\theta)}}{\int_{\theta'} \Pr[z_{1:n} \mid \theta']\,\pi(\theta')} \tag{9.17}$$

If the loss function is bounded by B, then sampling from this distribution is the same as exponential mechanism with L2 sensitivity B or $2B$ depeneding on the definition of neighboring dataset.

The advantages of posterior sampling include applicability and strong bounds under weak assumptions. But it is also computationally inefficient, requires bounded loss functions and can get only one sample.

References

[BNS13] Beimel, A., Nissim, K., Stemmer, U. (2013, January). Characterizing the sample complexity of private learners. In Proceedings of the 4th conference on Innovations in Theoretical Computer Science (pp. 97-110).

[CH11] Chaudhuri, K., Hsu, D. (2011, December). Sample complexity bounds for differentially private learning. In Proceedings of the 24th Annual Conference on Learning Theory (pp. 155-186). JMLR Workshop and Conference Proceedings.

[BNSV15] Bun, M., Nissim, K., Steemmer, U., Vadhan, S. (2015, October). Differentially private release and learning of threshold functions. In 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (pp. 634-649). IEEE.

[WLF16] Wang, Y. X., Lei, J., Fienberg, S. E. (2016). Learning with differential privacy: Stability, learnability and the sufficiency and necessity of ERM principle. The Journal of Machine Learning Research, 17(1), 6353-6392.