Privacy Preserving Prediction

Vishal Rana

May 2019

1 Introduction

The rapid advancements in machine learning and pattern recognition along with a tremendous increase in the amount of data being collected and processed posses some very fundamental questions about the privacy of the data. These questions becomes even more important when we deal with very sensitive data like medical records, genetic information, financial history, etc. On top of that, the question of what exactly do we mean when we talk about privacy in terms of data used for prediction also has a number of different answers.

Most of us are used to the Alice-Bob-Eve model for a secure cryptosystem. We try to make it difficult for Eve to obtain any new information about the plaintext just from the ciphertext. However, the question of data privacy in context of predictions differs in a big way, Bob and Eve are the same person, denying all the information to Eve will also deny it to Bob. The Curator-Analyst model is more suited. A Curator releases the data or interface to the data and the Analyst uses it to make predictions [2]. The Analyst might use the data in a manner that endangers the privacy of an individual or a group. Differential privacy model of privacy is one popular definition used. In this report we will summarize some of the existing results about differential privacy based heavily on work of Dwork et.al. [2] [1].

The promise that differential privacy makes is "You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available." [2]. This basic idea translates to a divergence between outputs of an algorithm trained on neighbouring datasets.

The results are based on assumption that there is an interface to access the predictions of the model/algorithm and the complete model is not available for scrutiny. Even though this doesn't guarantee protection against privacy breaches like linkage attacks and membership inference, it's still an easier point to start off and differential privacy can give some level of protection [1].

Let Q be a query generating algorithm and M be our prediction algorithm, then the following two definitions can be stated. The exact definition of $(\epsilon, \delta)-$differential privacy is stated later.

Definition. A prediction interface M, is $(\epsilon, \delta)-$differentially private if for every interactive query generating algorithm Q, the output $(Q \rightleftarrows M(S))$ is $(\epsilon, \delta)-$differentially private with respect to dataset S.

We further restrict our attention to the case where a single prediction is allowed and agents cannot collaborate to used the information from multiple queries and we try to study privacy in that setting.

Definition. Let M be an algorithm that given a dataset $S \in (X \times Y)^n$ and a point x produces a value in Y . We say that M is $(\epsilon, \delta)-$differentially private prediction algorithm if for every $x \in X$, the output $M(S, x)$ is $(\epsilon, \delta)-$differentially private with respect to S. We use $M(S)$ to refer to the (randomized) function $M(S, .)$.

1.1 Overview

We start by studying differential privacy formally in terms of divergence between distributions followed by discussing PAC learning in realizable case and the sample complexity. Then we deal with convex losses and agnostic case and supplement it with existing results for convex optimization problems under various constraints on the loss function. We also studied the generalization properties of the private evaluation algorithms.

Two fundamental ideas used in achieving privacy are subsampling and stability, both of which have been studied. Subsampling allows us to obtain private predictions by aggregating non-private ones, while stability allows us to add noise to the predictions in a controlled way, like the Laplace and Exponential mechanism.

2 Differential Privacy

The basic idea behind differential privacy is based on bounding divergence between distributions output by the learning algorithm on neghbouring datasets [1]. To that end β-approximate max-divergence for two random variables U and V and for $\delta \geq 0$ is defined as

$$D_{\infty}^{\delta}(U||V) \doteq \sup_{O \subseteq Range(U); \mathbf{Pr}[U \in O] > \delta} \ln \frac{\mathbf{Pr}[U \in O] - \delta}{\mathbf{Pr}[V \in O]}$$

Definition. A randomized algorithm $M : X^n \to Y$ is said to be $(\epsilon, \delta)-$ differentially private if for all pairs $S, S' \in X^n$ that differ on a single element, $D_{\infty}^{\delta}(M(S)||M(S')) \leq \epsilon$.

The following group privacy property of differential privacy is very useful.

Lemma 1. Let $M : (X \times Y)^n \times X \to Y$ be an $(\epsilon, \delta)-$ differentially private prediction algorithm and $k \in \mathbb{N}$. For all pairs of datasets $S, S' \in (X \times Y)^n$ differing in at most k points, and all $x \in X$

$$D_{\infty}^{ke^{\epsilon(k-1)}\delta}(M(S)||M(S')) \leq k\epsilon$$

.

3 Preserving Privacy via subsampling and uniform stability

The standard baseline technique for approaching the problem of making private predictions is based on subsampling the data S into subsets $S_1, S_2, ..., S_r$ and coming up with a set of non private predictors $f_1, f_2, ..., f_r$ which can then be aggregated in a suitable way to give private predictions [1]. It is based on the idea that replace one stability is improved by such subsampling and aggregation.

We start by studying the simple problem of concept learning in realizable case.

3.1 PAC Learning

The algorithm for PAC learning begins by taking a soft majority over the output of predictors $f_1, f_2, ..., f_r$. It outputs a label b with a probability proportional to $e^{\epsilon|\{i \in r : f_i(x) = b\}|}$. The following theorem gives a proof for the existence of a differentially private algorithm for PAC learning in realizable case.

Theorem 1. [Dwork et.al.'18] Let C be a class of Boolean functions over X. Let A be a PAC learning algorithm for C that uses $m(\alpha, \beta)$ samples to learn with error α and confidence parameter β. For every $\epsilon > 0$, there exists an $\epsilon-$differentially private prediction algorithm M that PAC learns C using $n = r \cdot m(\delta/4, \beta/r)$ examples, where $r = \lceil 6\ln(4/\alpha)/\epsilon \rceil$.

Proof. Let $c \in C$ be the unknown labelling function and $\mathcal{D}$ be the unknown distribution over data. Given a set of samples S of size $n = r \cdot m(\alpha/4, \beta/r)$, the algorithm A is run with $\alpha/4, \beta/r$ as the error and confidence parameters to obtain predictors $f_1, f_2, ..., f_r$.

There are two parts to the proof, one for privacy and the other for accuracy. We will provide a sketch of both here. Define $\nu(S, x) = 2|\{i \in [r] : f_i(x) = 1\}| - r$. Algorithm M outputs 1 with probability $\frac{e^{\epsilon\nu(S,x)/2}}{1+e^{\epsilon\epsilon\nu(S,x)/2}}$ and 0 otherwise. For two datasets S, S' that differ in one data point, $|\nu(S, x) - \nu(S', x)| \leq 2$ holds. Privacy guarantees follow trivially from this.

For accuracy, observe that for all $i \in [r]$, $\mathbf{Pr}_{\mathcal{D}}[f_i(x) \neq c(x)] = \mathbf{E}_{\mathcal{D}}[|f_i(x) - c(x)|] \leq \alpha/4$ with probability at least $1 - \beta$ Using union bound, this gives,

$$E_{\mathcal{D}}\Big[\sum_{i\in[r]} |f_i(x) - c(x)|\Big] \leq \alpha r/4$$

Using Markov's inequality then gives

$$Pr_{\mathcal{D}}\Big[\sum_{i\in[r]} |f_i(x) - c(x)| \geq r/3\Big] \leq 3\alpha/4 \tag{1}$$

If $c(x) = 1$

$$v(S, x) = 2(r - \sum_{i\in[r]} |f_i(x) - c(x)|) - r \geq r/3$$

which then implies

$$\mathbf{Pr}_M[M(S, x) \neq 1] \leq \frac{1}{1 + e^{\epsilon r/6}} \leq e^{-\epsilon r/6} \leq e^{ln(\alpha/4)} = \alpha/4 \tag{2}$$

Similar analysis works for $c(x) = 0$, which together with (1) gives $\mathbf{Pr}_{\mathcal{D},M}[M(S, x) \neq c(x)] \leq \alpha$. □

We need $n = O(\frac{d\log(1/\alpha)+\log(1/\beta)}{\alpha})$ to PAC learn a class C of VC dimension d. [4] This along with the previous theorem can be reduced to show for that for every concept class C there exists a differentially private prediction algorithm that PAC learns the class given $n = \tilde{O}(\frac{d+log(1/\beta)}{\epsilon\alpha})$ training examples.

The next theorem proves that the above mentioned upper bound is tight.

Theorem 2. [Dwork et.al.'18] Let C be a class of Boolean functions of VC dimension d. Then for all $\alpha, \epsilon > 0$, any $(\epsilon, \epsilon/3)-$ differentially private prediction algorithm M that PAC learns C with error α and confidence $1/12$ requires $n \geq d/(32\alpha\epsilon)$ examples.

Proof. The proof is given for $\alpha = 1/4$ and its extension to arbitrary α is outlined briefly at the end. Let $a_1, ..., a_d \in X$ be a set of points shattered by C which has a VC dimension d and let these points be referred to as $\{1, ..., d\}$ for convenience. Let f_b be a function in class C such that it gives the same labels as the vector $b = (b_1, ..., b_d) \in \{0, 1\}^d$. Let $\mathcal{D}$ be the uniform distribution on $[d]$, M be the $(\epsilon, \epsilon/3)-$differentially private prediction algorithm for C and p_i be the expected prediction accuracy of M on point $i \in [d]$.

$$p_i \doteq \mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d, S\sim(\mathcal{D},f_b)^n, M}[M(S,i) \neq b_i] \tag{3}$$

The accuracy and confidence guarantees of M imply that

$$\mathop{\mathbf{E}}_{i\in\mathcal{D}}[p_i] = \mathop{\mathbf{E}}_{b\sim\{0,1\}^d, S\sim(\mathcal{D},f_b)^n}\left[\mathop{\mathbf{E}}_{i\sim\mathcal{D}}\left[\mathop{\mathbf{Pr}}_{M}[M(S,i)\neq b_i]\right]\right] \leq \alpha+\beta = \frac{1}{4}+\frac{1}{12} = \frac{1}{3} \tag{4}$$

This implies that there exists i such that $p_i \leq 1/2$. Fixing this i we proceed with the rest of the argument. Let $S^{\oplus i}$ be a dataset in which all points same as i have their labels flipped. Let there be t such points. Then from the definition of group prediction privacy it follows that for $v \in \{0,1\}$,

$$\mathbf{Pr}[M(S,i)=v] \leq e^{\epsilon t}\mathbf{Pr}_M[M(S^{\oplus i},i)=v] + te^{(t-1)\epsilon}\delta \tag{5}$$

For the sake of contradiction, we assume that $n \leq d/(8\epsilon)$ for d larger than some fixed constant. Then with probability at least $1/24$ over the choice of S it includes at most $s \doteq 1/4\epsilon$ points equal to i. Using $e^{\epsilon s} = e^{1/4} \leq 3/2$ and $\delta = \epsilon/3$ along with (5) gives

$$\begin{aligned}\mathop{\mathbf{Pr}}_{S\sim(\mathcal{D},f_b)^n}[M(S,i)=v] &\leq e^{\epsilon s}\mathop{\mathbf{Pr}}_{S\sim(\mathcal{D},f_b)^n}[M(S\oplus i,i)] + se^{(s-1)\epsilon}\delta + \frac{1}{24}\\ &< \frac{3}{2}e^{\epsilon s}\mathop{\mathbf{Pr}}_{S\sim(\mathcal{D},f_b)^n}[M(S\oplus i,i)] + \frac{1}{6}\end{aligned}$$

For a fixed b and $S\sim(\mathcal{D},f_b)^n$, $S^{\oplus i}$ and $S\sim(\mathcal{D},f_{b^{\oplus i}})^n$ are distributed identically. This implies,

$$\begin{aligned}\mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d, S\sim(\mathcal{D},f_b)^n, M}[M(S^{\oplus i},i)=b_i] &= \mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d, S\sim(\mathcal{D},f_{b^{\oplus i}})^n, M}[M(S,i)=b_i]\\ &= \mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d, S\sim(\mathcal{D},f_b)^n, M}[M(S,i)\neq b_i]\\ &= p_i\end{aligned}$$

The previous two equations along with the definition of p_i give $p_i > 1/3$ which is a contradiction. Hence, $n > d/8\epsilon$.

For general α, we define a distribution $\mathcal{D}_\alpha$ on the d chosen points. It assigns a probability of $1-4\alpha$ to last point d and remaining 4α uniformly over the rest of the points $i \in [d-1]$. Relative to this uniform distribution on $[d-1]$ approximately 4α fraction of the training examples will be useful for getting a low error. This gives us the $1/\alpha$ factor in the lower bound of n. Rest of the analysis is similar to that for $\alpha = 1/4$. □

3.2 Learning with convex losses and stability

In this section goal is to study minimization of the expected loss function $\mathbf{E}_{x,y\sim\mathcal{P}}[l(f(x)),y]$ where l is a convex function. The idea is to use (non-private) algorithms that are stable in some sense and add Laplace or Gaussian noise to obtain a differentially private prediction. The additional error is bonded by using loss functions that Lipschitz or smooth [1].

Definition. A learning algorithm A is uniform replace-one (RO) prediction stable with rate γ if for all datasets $S, S' \in (X\times Y)^n$ that differ in a single element and any $x \in X$,

$$|A(S,x) - A(S',x)| \leq \gamma$$

where $A(S,.)$ denotes the function output by A on dataset S.

Before showing the existence of differentially private algorithm in this setting, we will discuss the Laplace mechanism and show that addition of Laplace or Gaussian noise provides differential privacy.

3.2.1 Laplace Mechanism

The Laplace distribution centered at 0 with scale parameter equal to b has a probability density function given as $\text{Lap}(x|b) = \frac{1}{2b}exp(-\frac{|x|}{b})$. It has a variance of $2b^2$. The l_1 sensitivity of the output of an algorithm $A(S, x)$ can be written as

$$\Delta A = \max_{S,S' \in (X \times Y)^n} \max_{x \in X} ||A(S, x) - A(S', x)||_1 \tag{6}$$

where S, S' are neighbouring datasets. We can see that $\Delta A = \gamma$

The Laplace mechanism computes the output of A and preturbs it by drawing noise from Laplace distribution with parameter b set to $\Delta A/\epsilon$.

Theorem 3. [Dwork et.al. '14] The Laplace mechanism preserves $(\epsilon, 0)$-differential privacy.

Proof. Let S, S' be neighbouring datasets and let $L_A(S, .)$ be the output obtained by adding Laplace noise to the output of $A(S, .)$. Also, let p_S be the pdf of output when A is trained on S and $p_{S'}$ be the output when it is trained on S'. For some arbitrary z in the range of L_A,

$$\begin{aligned}
\frac{p_S(z)}{p_{S'}(z)} &= \frac{exp(-\frac{\epsilon|A(S,z)-z|}{\Delta A})}{exp(-\frac{\epsilon|A(S',z)-z|}{\Delta A})} \\
&\leq exp(\frac{\epsilon|A(S, z) - A(S', z)|}{\Delta A}) \\
&\leq exp(\epsilon)
\end{aligned}$$

This shows the Laplace mechanism to be $(\epsilon, 0)-$ differentially private. □

Alternately, we can add Gaussian distributed noise with variance set to $\Delta A \ln(1/\delta)/\epsilon$ to get $(\epsilon, \delta)-$ privacy [2].

3.2.2 Bounds on additional error

Theorem 4. [Dwork et.al.'18] Let $l : \mathbb{R} \times Y \to \mathbb{R}$ be a loss function convex in the first parameter. Let A be a uniform RO prediction stable algorithm with rate γ. For every $\epsilon > 0$, there exists an $\epsilon-$ differentially private prediction algorithm M such that for every dataset $S \in (X \times Y)^n$ and any probability distribution $\mathcal{P}$ over $X \times Y$:

1. if $l(\cdot, y)$ is L_l-Lipschitz in the first parameter for all $y \in Y$ then

$$\mathcal{E}[l(M(S))] \leq \mathcal{E}[l(A(S))] + L_l\gamma/\epsilon$$

2. if $l(\cdot, y)$ is σ-smooth in the first parameter for all $y \in Y$ then

$$\mathcal{E}[l(M(S))] \leq \mathcal{E}[l(A(S))] + \sigma^2\gamma^2/\epsilon^2$$

where $\mathcal{E}[l(M(S))] = \mathbf{E}_{M,(x,y)\sim\mathcal{P}}[l(M(S, x), y)]$.

Proof. We have already shown that Laplace mechanism gives ϵ-differential privacy. If loss function is L_l-Lipschitz, and ξ is the additional Laplace noise,

$$\underset{M}{\mathbf{E}}[l(A(S,x)+\xi)] \leq l(A(S,x),y) + \underset{M}{\mathbf{E}}[L_l|\xi|)] = l(A(S,x),y) + L_l\gamma/\epsilon \tag{7}$$

where we have used the fact that $|\xi|$ is exponentially distributed. Since this bound hold pointwise, it hold for any distribution $\mathcal{P}$.

If loss is $\sigma-$smooth,

$$l(A(S,x)+\xi,y) \leq l(A(S,x),y) + l'(A(S,x),y)\xi + \sigma\xi^2/2 \tag{8}$$

Taking expectation on both sides and using $\mathbf{E}[\xi] = 0$ and $\mathbf{E}[\xi^2] = 2\gamma^2/\epsilon^2$ for Laplace distribution gives the required inequality. □

If an algorithm is not sufficiently stable to add noise to its predictions, we can amplify the stability by averaging predictions obtained from disjoint subsamples. Since the loss function is convex, it preserves the bounds on the expected loss. The following lemma formalizes this idea.

Lemma 2. [Dwork et.al.'18] Let A be a learning algorithm that outputs a real-valued function on X, is uniform RO prediction stable with rate γ and uses n samples. For any $r \in N$ there exists a learning algorithm A' that is uniform RO prediction stable with rate $\gamma' = \gamma/r$ that uses $n \cdot r$ samples. Further, for any loss function $l(\cdot,\cdot)$ convex in the first parameter, if for a distribution $\mathcal{P}$ over $X \times Y$, A has the guarantee that $\mathbf{E}_{S\in\mathcal{P}^n}[\mathcal{E}_{\mathcal{P}}[l(A(S))]] \geq \nu$ for some value ν that may depend on $\mathcal{P}$ and the parameters of the learning problem then $\mathbf{E}_{S'\in\mathcal{P}^{rn}}[\mathcal{E}_{\mathcal{P}}[l'(A(S'))]] \geq \nu$. Alternatively, if for some $\beta > 0$,

$$\underset{S\sim\mathcal{P}^n}{\mathbf{Pr}}[\mathcal{E}_{\mathcal{P}}[l(A(S))] \geq \nu] \leq \beta$$

then

$$\underset{S'\sim\mathcal{P}^{rn}}{\mathbf{Pr}}[\mathcal{E}_{\mathcal{P}}[l(A'(S'))] \geq \nu] \leq r\beta$$

The running time of A' is r times the running time of A.

3.3 Agnostic Learning

Agnostic learning of Boolean functions can be seen as learning a real-valued function f with absolute loss $l(a,y) = |y-a|$. A real-valued prediction $f(x)$ can also be formulated in a similar way with a prediction of 1 with probability p, with p being projection of $f(x)$ on $[0,1]$. This gives an upper bound on expected disagreement with $y \in \{0,1\}$ as $|y-f(x)|$. This loss function is convex, 1-Lipschitz and for a learning algorithm that outputs a Boolean function, is uniformly stable with rate 1. The stability needs to be amplified to $\alpha\epsilon$ to achieve additional error in (4) of at most α which gives $r = 1/(\alpha\epsilon)$. This gives us the following corollary [1].

Corollary 1. [Dwork et.al.'18] Let C be a class of Boolean functions over X. Let A be an agnostic learning algorithm for C that uses $m(\alpha,\beta)$ samples to learn with excess error α and confidence parameter β. For every $\epsilon \in (0,1]$, there exists an ϵ-differentially private prediction algorithm M that agnostically learns C given $n = 2 \cdot m(\alpha, 2\beta\alpha\epsilon)/(\alpha\epsilon)$ examples.

This corollary along with the result that $n = O(\frac{d+\log(1/\beta)}{\alpha^2})$ samples are sufficient to agnostically learn a class C with VC dimension d [5] shows that for $n = O(\frac{d+\log(1/(\alpha\beta\epsilon))}{\epsilon\alpha^3})$ training examples there exists an ϵ-differentially private prediction algorithm that agnostically learns C with error and confidence parameters as α and β [1].

3.4 Applications to convex regression problems

The approach used in the last section can applied to convex regression problems where the objective is of the form:

$$\min_{w \in \mathcal{K}} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{P}}[l(f(w,x),y)]$$

where $\mathcal{K} \subset \mathbb{R}^d$ is a convex set and l is a convex function over $\mathcal{K}$ for all $(x,y) \in X \times Y$. Using standard results regarding upper bound on prediction stability of strongly convex optimization [6], and (4) the following can be shown.

Corollary 2. [Dwork et.al.'18] Let $\mathcal{K} \subseteq \mathcal{B}_2^d(R)$ be a convex set, $\{f(\cdot,x)|x \in X\}$ be a family of L_f-Lipschitz functions over $\mathcal{K}$, $l : \mathbb{R} \times Y \to \mathbb{R}$ be convex, L_l-Lipschitz loss function and $l(f(\cdot,x),y)$ be λ-strongly convex for all $(x,y) \in X \times Y$. For every $\epsilon > 0$, there exists an $\epsilon-$differentially private prediction algorithm M that for any probability distribution $\mathcal{P}$ over $X \times Y$ satisfies:

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathcal{E}_{\mathcal{P}}[l(M(S))]] \leq \min_{w\in\mathcal{K}} \mathop{\mathbf{E}}_{\mathcal{P}}[l(f(w,x),y)] + \frac{4L_f^2 \cdot L_l^2}{\lambda n} \cdot \left(1 + \frac{1}{\epsilon}\right)$$

where R is the radius of Eucleadian norm ball $\mathcal{B}_2^d(R)$.

Following the standard technique of relaxing the strong convexity requirement on the loss by adding a strongly convex regularizing term $\lambda||w||^2$ with $\lambda = \frac{2L_f L_l}{R\sqrt{n\epsilon/(1+\epsilon)}}$, the following corollary can be obtained from the previous one [1].

Corollary 3. [Dwork et.al.'18] Let $\mathcal{K} \subseteq \mathcal{B}_2^d(R)$ be a convex set, $\{f(\cdot,x)|x \in X\}$ be a family of L_f-Lipschitz functions over $\mathcal{K}$, $l : \mathbb{R} \times Y \to \mathbb{R}$ be convex, L_l-Lipschitz loss function and $l(f(\cdot,x),y)$ be convex for all $(x,y) \in X \times Y$. For every $\epsilon > 0$, there exists an $\epsilon-$differentially private prediction algorithm M that for any probability distribution $\mathcal{P}$ over $X \times Y$ satisfies:

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathcal{E}_{\mathcal{P}}[l(M(S))]] \leq \min_{w\in\mathcal{K}} \mathop{\mathbf{E}}_{\mathcal{P}}[l(f(w,x),y)] + \frac{4R \cdot L_f \cdot L_l}{\sqrt{n\epsilon/(1+\epsilon)}}$$

The next step is to add smoothness to the loss function to obtain stronger results. Using the stability of gradient descent for smooth functions [3] with the general framework we have been using, the following corollary follows

Corollary 4. [Dwork et.al.'18] Let $\mathcal{K} \subseteq \mathcal{B}_2^d(R)$ be a convex set, $\{f(\cdot,x)|x \in X\}$ be a family of L_f-Lipschitz functions over $\mathcal{K}$, $l : \mathbb{R} \times Y \to \mathbb{R}$ be convex, L_l-Lipschitz and $\sigma-$ smooth loss function and $l(f(\cdot,x),y)$ be λ-strongly convex and $\sigma-$ smooth for all $(x,y) \in X \times Y$. If $\sigma \leq 2L_f L_l\sqrt{n}/R$ then for every $\epsilon > 0$, there exists an $\epsilon-$differentially private prediction algorithm M that for any probability distribution $\mathcal{P}$ over $X \times Y$ satisfies:

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathcal{E}_{\mathcal{P}}[l(M(S))]] \leq \min_{w\in\mathcal{K}} \mathop{\mathbf{E}}_{\mathcal{P}}[l(f(w,x),y)] + \frac{2 \cdot R \cdot L_f \cdot L_l}{\sqrt{n}} + \frac{\sigma \cdot R^2 \cdot L_f^2}{n\epsilon^2}$$

What this last result says is that for sufficiently smooth loss function, ϵ needs to scale as $n^{-1/4}$ for the additional error due to added noise becomes comparable to statistical error, that is, we obtain this level of differential privacy just by virtue of statistical errors [1].

4 Stability and Generalization

Prediction privacy can be viewed as one notion of stability and we can talk about generalization properties of private prediction algorithms in that context.

Before getting into the details of it, we will mention a few simple and useful lemmas without giving the proofs for them, which can be found in [1], [2].

Lemma 3 (Postprocessing)**.** For $Z = X \times Y$ let $M' : Z^n \times X \to \mathbb{R}$ be an (ϵ, δ)-differentially private prediction algorithm. Then for every loss function $l : \mathbb{R} \times Y \to \mathbb{R}$, $M(S, (x, y)) \doteq l(M'(S, x), y)$ is an (ϵ, δ)-differentially private evaluation algorithm.

Lemma 4 (Composition)**.** Let M_1 and M_2 be ϵ_1 and ϵ_2-differentially private algorithms. Then $M_{1,2} = (M_1, M_2)$ is $\epsilon_1 + \epsilon_2$-differentially private.

Lemma 5. Let U and V be two random variables over $[0, B]$ such that $D_\infty^\delta(U||V) \leq \epsilon$. Then $\mathbf{E}[U] \leq e^\epsilon \cdot \mathbf{E}[V] + \delta \cdot B$.

Now we will look at the generalization properties and use a more general setting for the same which is defined as follows

Definition. (Private evaluation algorithm) Let M be an algorithm that given a dataset $S \in Z^n$ and a value $z \in Z$ produces a value in a set W. We say that M is an (ϵ, δ)-differentially private evaluation algorithm if for every $z \in Z$, the output $M(S, z)$ is (ϵ, δ)-differentially private with respect to S.

From postprocessing property mentioned above we can see that a differentially private evaluation is differentially private with the same set of parameters. Here the private evaluation M will compute the loss of prediction given by a prediction algorithm M', i.e. for some loss function l, $M(S, (x, y)) = l(M'(S, x), y)$.

Let $S = (z_1, ..., z_n)$ and $S' = (z'_1, ..., z'_n)$ be independently and randomly drawn samples from an unknown distribution $\mathcal{P}$ over Z. The empirical mean is given by $\mathcal{E}_S[M(S)] = \frac{1}{n}\sum_{i\in[n]} \mathbf{E}_M[M(S, z_i)]$ and the mean on independent samples is given as $\mathcal{E}_{S'}[M(S)] = \frac{1}{n}\sum_{i\in[n]} \mathbf{E}_M[M(S, z'_i)]$. Further, $\mathbf{E}_{S'\sim\mathcal{P}^n}[\mathcal{E}_{S'}[M(S)]] = \mathcal{E}_P[M(S)]$, $\mathcal{E}_\mathcal{P}[M(S)] = \mathcal{E}_\mathcal{P}[l(M'(S))]$ and $\mathcal{E}_S[M(S)] = \mathcal{E}_S[l(M'(S, x), y)]$.

Concentration inequalities also state that $\mathcal{E}_{S'}[M(S)]$ is strongly concentrated around $\mathcal{E}_P[M(S)]$, so bounds on first can be used to imply bounds on the second. The following bound on the $k-$th moment of $\mathcal{E}_{S'}[M(S)]$ holds.

Lemma 6. [Dwork et.al.'18][Dwork et.al.'18] Let $M : Z^n \times Z \to [0, B]$ be an $(\epsilon, \delta)-$differentially private evaluation algorithm and $\mathcal{P}$ be an arbitrary distribution over Z. Then:

$$\mathop{\mathbf{E}}_{S,S'\sim\mathcal{P}^n}\left[(\mathcal{E}_{S'}[M(S)])^k\right] \geq e^{k^2\cdot\epsilon} \cdot \mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}\left[(\mathcal{E}_S[M(S)] + k\delta B)^k\right]$$

Proof. Let $I = (i_1, ..., i_k) \in [n]^k$ be a set of k indices and S_I be the set obtained from S by replacing elements corresponding to these in S by the corresponding ones in S'. We can write,

$$\mathop{\mathbf{E}}_{S,S'\sim\mathcal{P}^n}\left[\left(\frac{1}{n}\sum_{i\in[n]} \mathbf{E}_M[M(S, z'_i)]\right)^k\right] = \frac{1}{n^k}\sum_{I\in[n]^k} \mathop{\mathbf{E}}_{S,S'\sim\mathcal{P}^n}\left[\prod_{i\in I} \mathop{\mathbf{E}}_{M}[M(S, z'_i)]\right] \tag{9}$$

From definition of group privacy and the lemma (5) we can write

$$\prod_{i\in I} \mathop{\mathbf{E}}_{M}[M(S, z_i')] = e^{k^2\epsilon} \prod_{i\in I} \Big(\mathbf{E}_M[M(S, z_i')] + k\delta B\Big)$$

For $S, S' \sim \mathcal{P}^n$, the term of right hand side of the previous equation $\prod_{i\in I} \Big(\mathbf{E}_M[M(S, z_i')] + k\delta B\Big)$ is distributed identically to $\prod_{i\in I} \Big(\mathbf{E}_M[M(S, z_i)] + k\delta B\Big)$. Substituting this along with the previous equation into (9) gives the desired inequality. □

This lemma can be used to obtain high probability generalization bounds. For example the next lemma, which we state without proof.

Lemma 7. Let $M : Z^n \times Z \to [0, B]$ be an $(\epsilon, \delta)-$differentially private evaluation algorithm and $\mathcal{P}$ be an arbitrary distribution over Z. Assume that for every S, $\mathcal{E}_S[M(S)] \leq \alpha$. Then for every $\beta \in (0, 1)$,

$$\mathop{\mathbf{Pr}}_{S,S'\sim\mathcal{P}^n} \left[\mathcal{E}'_S[M(S)] \geq \alpha \cdot e^{2\sqrt{\epsilon \ln(1/\beta)}}\right] \leq \beta$$

5 Conclusion

There is a fundamental trade off between the amount of data available for training our prediction models, the accuracy that is desired and the level of privacy that needs to be ensured. There is no free lunch in data privacy. Subsampling clearly needs more data for training while additional noise for privacy can reduce accuracy of the model. Thus privacy-preserving aggregations generally lead to suboptimal solutions. Dwork et.al.[1] provide an example of algorithm for learning thresholds that improves on aggregation based methods. However, finding such algorithms in a more general agnostic setting is an open and challenging problem. Further, dealing with the case where multiple queries are allowed or when multiple users collaborate is an interesting and active direction of work.

References

[1] Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. *arXiv preprint arXiv:1803.10266*, 2018.

[2] 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.

[3] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

[4] Michael J Kearns, Umesh Virkumar Vazirani, and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[5] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[6] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11(Oct):2635–2670, 2010.