Privacy-preserving Prediction

Cynthia Dwork
Harvard University

Vitaly Feldman
Google Brain

Abstract

Ensuring differential privacy of models learned from sensitive user data is an important goal that has been studied extensively in recent years. It is now known that for some basic learning problems, especially those involving high-dimensional data, producing an accurate private model requires much more data than learning without privacy. At the same time, in many applications it is not necessary to expose the model itself. Instead users may be allowed to query the prediction model on their inputs only through an appropriate interface. Here we formulate the problem of ensuring privacy of individual predictions and investigate the overheads required to achieve it in several standard models of classification and regression.

We first describe a simple baseline approach based on training several models on disjoint subsets of data and using standard private aggregation techniques to predict. We show that this approach has nearly optimal sample complexity for (realizable) PAC learning of any class of Boolean functions. At the same time, without strong assumptions on the data distribution, the aggregation step introduces a substantial overhead. We demonstrate that this overhead can be avoided for the well-studied class of thresholds on a line and for a number of standard settings of convex regression. The analysis of our algorithm for learning thresholds relies crucially on strong generalization guarantees that we establish for all differentially private prediction algorithms.

1 Introduction and problem formulation

In machine learning tasks, the training data often consists of information collected from individuals. This data can be highly sensitive, for example in the case of medical or financial information, and therefore privacy-preserving data analysis is becoming an increasingly important area of study in machine learning, data mining and statistics [DS09; SC13; DR14]. We rely on the well-studied differential privacy model of privacy that has become a de facto standard for formal understanding of privacy [DMNS06].

The standard setting of privacy-preserving learning aims to ensure that the model learned from the data is produced in a differently private way. Thus this approach preserves privacy even when a potential adversary has complete access to the description of the predictive model. The downside of this strong guarantee is that for some learning problems, achieving the guarantee is known to have substantial additional costs. More examples are needed to achieve the same level of accuracy (or lower accuracy is achievable for a given number of examples). In addition, private learning may require new and computationally less efficient algorithms.

In this work we consider learning in a setting where the description of the learned model is not accessible to the (potentially adversarial) user(s). Instead the users have access to the model through an interface (often referred to as an API). For an input point the interface provides the value of the predictive model on that point. This view is appropriate for many existing applications where user privacy is a concern. For example, companies that collect data about their users usually expose only a cloud-based interface to the models they train on user data. Credit rating bureaus only allow access to their models through an electronic interface. In addition, it may enable new applications where privacy considerations are currently preventing the use of predictive models trained on sensitive user data. For example, in medical diagnostics a prediction interface would suffice for most applications.

Allowing such restricted access may appear to pose no risk to individual privacy. However, as recently demonstrated by Shokri et al. [SSSS17], blackbox access to Amazon ML and Google prediction APIs suffice for successful membership inference attacks. Membership inference is the task in which given a user's record the goal is to infer whether the record was used for training the model. This information is known to be sensitive in several contexts. Membership inference can also be used to complete partial records revealing the values of sensitive attributes. Even more recently, Long et al. [Lon+18] demonstrated several additional successful membership inference attacks based on blackbox access. Further, Carlini et al. [CLKES18] proposed a more formal way to measure the degree to which sensitive information is memorized by generative sequence models and explored several techniques to extract sensitive information using black box access to such models. The use of differentially private learning algorithms to protect against such attacks has been proposed in [SSSS17] and briefly explored in [CLKES18].

We now describe the setting more formally. For a prediction problem over a domain X and label space Y, a prediction interface is an algorithm that has access to a dataset $S \in (X \times Y)^n$ and given a query point $x \in X$ outputs a value $y \in Y$. The algorithm can be queried multiple times and is stateful (namely, responses can depend on previous queries). We define the privacy of such an interface in the same way as usually done for interactive algorithms. Namely, for a prediction interface M and a stateful query generating algorithm Q we denote by $(Q \rightleftarrows M(S))$ the sequence of queries and responses generated in the interaction of Q and M on dataset S.

Definition 1.1 (Private prediction interface)**.** *A prediction interface M, is (ϵ, δ)-differentially private if for every interactive query generating algorithm Q, the output $(Q \rightleftarrows M(S))$ is (ϵ, δ)-differentially private with respect to dataset S.*

While the problem setting has many facets that merit investigation, we focus on perhaps the most basic question: what is the cost of ensuring privacy of a single prediction. In other words, we focus on the problem of answering a single prediction query. Composition properties of differential privacy imply that such an algorithm can be used to answer multiple queries with privacy parameters that degrade gracefully with the number of queries [DR14]. Therefore such an algorithm is a natural building block for constructing an algorithm that can answer multiple queries. Naturally, better ways of dealing with sequences of queries might exist and the general topic of answering interactive sequences of queries has been studied extensively in the differential privacy literature (see [DR14] for an overview).

An algorithm M that answers a single query x defines a randomized prediction at x and hence such an algorithm implicitly defines a learning algorithm that outputs a randomized predictor $h(x) = M(S, x)$.

Definition 1.2. *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 (ϵ, δ)-*differentially private prediction *algorithm if for every $x \in X$, the output $M(S, x)$ is (ϵ, δ)-differentially private with respect to S. We use $M(S)$ to refer to the (randomized) function $M(S, \cdot)$.*

This definition allows us to treat this building block in the same way as regular learning algorithms and discuss it in the context of standard statistical learning models.

Two standard and closely related models for classification we will look at are PAC (or realizable) learning [Val84] and agnostic [Hau92; KSS94] learning. In the PAC learning model the algorithm is given random examples in which each point is sampled i.i.d. from some unknown distribution over the domain and is labeled by an unknown function from a set of functions C. In the agnostic learning model the algorithm is given examples sampled i.i.d. from an arbitrary (and unknown) distribution over labeled points. The goal of the learning algorithm in both models is to output a hypothesis whose prediction error on the distribution from which examples are sampled is within additive α of the prediction error of the best function in C (which is 0 in the PAC model). See Sec. 3 for formal definitions.

We will also consider a more general regression setting in which we are given a loss function $\ell : \mathbb{R} \times Y \to \mathbb{R}$ and the goal is to design a private prediction algorithm M that minimizes

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

where $\mathcal{P}$ is an unknown probability distribution over $X \times Y$.

2 Overview of the results

We first consider a natural “baseline” approach to this problem based on private aggregation of non-private learning algorithms.

2.1 Private aggregation of non-private models

To produce a prediction differentially privately we partition the dataset S into several subsamples $S_1, \dots, S_r$ and run a non-private learning algorithm on each of those subsamples too obtain predictors $f_1, \dots, f_r$. Now given a point x we use a differentially private aggregation technique on values $f_1(x), \dots, f_r(x)$ and output the result. Several such subsample-and-aggregate techniques are known [NRS07; DL09; ST13; DR14] that carefully exploit properties of the distribution over results on subsamples. A significant advantage of this approach is that it does not require a new learning algorithm and hence is easy to implement (there is an additional computational cost that is easy to parallelize).

Obviously, using r subsamples requires more data than non-private learning and therefore it is natural to ask whether this approach is optimal and how it compares to differentially private learning in the standard setting. We discuss these questions in the context of specific problems below.

PAC Learning: For PAC learning (or realizable case) accurate models $f_1, \dots, f_r$ have to be close to the true labeling function f (that is, they disagree with probability at most α). In particular, the fraction of points on which more than 1/4 of the predictors output the wrong label cannot be more than 4α. Outputting the correct label with privacy is easy in this setting and we do this using a soft majority vote (or, equivalently, the exponential mechanism [MT07] on the label counts). A number of other approaches would give comparable guarantees. A simple analysis shows that using $r = O(\ln(1/\alpha)/\epsilon)$ this reduction ensures ϵ-differentially private prediction (see Thm. 4.1 for a formal statement).

As an immediate corollary of this reduction and standard bounds on the sample complexity of PAC learning we obtain the following upper bound.

Corollary 2.1. *Let C be a class of Boolean functions of VC dimension d. Then for all $\alpha, \beta, \epsilon > 0$, there exists an ϵ-differentially private prediction algorithm M that PAC learns C with error α and confidence $1 - \beta$ given $n = \tilde{O}\left(\frac{d+\log(1/\beta)}{\epsilon\alpha}\right)$ examples.*

It turns out that this simple approach is essentially optimal in the worst case. Specifically, we prove that the sample complexity of this problem is $\Omega(d/(\epsilon\alpha))$ even when δ is as large as $\epsilon/3$.

Theorem 2.2. *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 $n = \Omega(d/(\epsilon\alpha))$ examples.*

For comparison, Kasiviswanathan et al. [KLNRS11] showed that the sample complexity of differentially privately PAC learning a class C over domain X is $O(\log(|C|)/(\epsilon\alpha))$. By Sauer's lemma, $\log(|C|) = O(d \cdot \log(|X|))$ and therefore the multiplicative gap between these two measures can be as large as $\log(|X|)$. The sample complexity of ϵ-differentially private PAC learning was subsequently shown to be $\tilde{\Theta}(R/(\epsilon\alpha))$, where R is the so-called representation dimension of C [BNS13]. However, as shown in [FX15], for many classes the gap between R and the VC dimension is still roughly $\log(|X|)$. For example, the representation dimension of linear threshold functions over $[N]^p$ is $p^2 \cdot \log N$ whereas the VC dimension is just p.

We remark that the technique we use to prove the lower bound in Thm. 2.2 is different from those used for proving lower bounds in the standard setting of learning with privacy.

Agnostic learning: In agnostic learning, the labels $f_1(x), \ldots, f_r(x)$ no do not necessarily agree on most points x and taking the majority vote may even reduce the accuracy. In this setting we predict by first averaging the non-private predictions to obtain $v(x) = \frac{1}{r}(f_1(x) + \cdots + f_r(x))$ and then outputting 1 with probability $v(x) + \zeta$ (truncated to range $[0, 1]$), where ζ is a Laplace noise variable. It is not hard to show that for $r = O(1/(\epsilon\alpha))$, this approach ensures that the prediction will be ϵ-differentially private and the addition of noise increases the prediction error by at most an extra α term (see Cor. 4.7). As a corollary of this reduction, we obtain the following upper-bound on the sample complexity in this setting.

Corollary 2.3. *Let C be a class of Boolean functions of VC dimension d. Then for all $\alpha, \beta, \epsilon > 0$ there exists an ϵ-differentially private prediction algorithm M that agnostically learns C with excess error α and confidence $1 - \beta$ given $n = \tilde{O}\left(\frac{d+\log(1/\beta)}{\epsilon\alpha^3}\right)$ examples.*

In this case the upper bound is much worse than the lower bound of $\Omega(d/\alpha^2 + d/(\epsilon\alpha))$ implied by Thm. 2.2. For comparison, ϵ-differentially private agnostic learning can be done using $\tilde{O}(d/\alpha^2 + R/(\epsilon\alpha))$ examples, where R is the representation dimension of C mentioned above [BNS13; FX15]. As a result, for classes such that $R = O(d)$ a differentially private learning algorithm matches the lower bound for private prediction. This leads to a natural question of whether it is possible to match the lower bound for all classes C. While we do not answer this question for arbitrary classes C, we give an example of an algorithm that goes beyond these two approaches. Specifically, it agnostically learns C with ϵ-private prediction using $\tilde{O}(d/\alpha^2 + d/(\epsilon\alpha))$ examples whereas learning C with privacy in the standard model requires an infinite number of examples.

Convex regression: Our analysis of agnostic learning can be seen as a special case of a more general analysis of prediction problems with convex loss functions. Specifically, the aggregation by averaging can be seen as a way to increase the *uniform prediction stability* of a learning algorithm. A learning algorithm is uniformly prediction stable with rate γ if for predictors f_S and $f_{S'}$ produced on any pair of datasets S, S' that differ on a single element and any point x, $|f_S(x) - f_{S'}(x)| \leq \gamma$. As follows immediately from this definition, a uniformly prediction stable learning algorithm can be converted to a differentially private prediction algorithm simply by adding Laplace (or Gaussian) noise to the prediction (see Lem. 4.5). Hence it reduces our problem to the problem of finding a uniformly prediction stable learning algorithm with sufficiently low rate of stability. Aggregation by averaging the predictors obtained by running a learning algorithm on r disjoint datasets can be seen as improving its uniform prediction stability by a factor of r. Convexity of the loss function, in turn, ensures that such averaging preserves the guarantees on the expected loss of the algorithm (see Lem. 4.6 for a formal statement).

We demonstrate how this general approach can be applied to convex regression problems. Specifically, we consider problems in which we have a family of predictors $\{f(w, \cdot)\}_{w \in \mathcal{K}}$ parameterized by a vector $w \in \mathcal{K}$, where $\mathcal{K} \subset \mathbb{R}^d$ is some convex body, ℓ is a convex loss function and $\ell(f(\cdot, x), y)$ is a convex function of w over $\mathcal{K}$ for all $(x, y) \in X \times Y$. The goal is to find $\hat{w}$ such that

$$\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{P}}[\ell(f(\hat{w}, x), y)] \leq \min_{w \in \mathcal{K}} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{P}}[\ell(f(w, x), y)] + \alpha,$$

where $\mathcal{P}$ is an unknown distribution over examples. This setting captures many important learning problems and has also been extensively investigated in the privacy literature (see [CMS11; KST12; BST14; TTZ15; WYX17] and references therein). For the purpose of comparison with sample complexity bounds known in this literature we restrict our attention to a basic setting in which $\mathcal{K}$ is a subset of the unit Euclidean ball and $\ell(f(w, x), y)$ is 1-Lipschitz in w for all (x, y) in support of $\mathcal{P}$. For this setting it is known that $\tilde{O}(d/(\epsilon\alpha^2))$ samples suffice to solve the problem with ϵ-differential privacy and $\tilde{O}(\sqrt{d}\log^4(1/\delta)/(\epsilon\alpha^2))$ samples suffice for (ϵ, δ)-differential privacy [BST14]. Further, such dependence on the dimension is optimal in both settings [BST14].

The dependence on the dimension is not necessary for non-private learning in this setting. In addition, we can exploit known stability analyses to reduce (or even eliminate) the need to use the aggregation step. By plugging the known stability results based on strong convexity and/or [BE02; SSSSS10; HRS16], we demonstrate that convex regression problems of this type can be solved with ϵ-differentially private prediction using $O(1/(\epsilon\alpha^2))$ examples (Cor. 4.11). We also demonstrate that smoothness of the loss function ℓ can be used to improve the dependence on ϵ (Cor. 4.13). We note that stability of the optimal solution of a strongly convex problem has been used to achieve differential privacy in multiple prior works starting with the pioneering work of Chaudhuri et al. [CMS11]. Stability of gradient descent on convex smooth functions has also been recently used to obtain privacy guarantees [WLKCJN17].

2.2 Beyond aggregation: learning thresholds

The class of linear thresholds Thr is defined over a subset of reals and consists of indicator functions of "$x \geq a$" for all $a \in \mathbb{R}$. Without loss of generality, we consider such functions over the set $[N] = \{1, \ldots, N\}$. While the class is very simple, learning it with privacy has proved to be rather challenging and some basic questions are still not fully resolved [BKN10; CH11; BNS13; FX15; BNSV15]. It is known that ϵ-differentially private PAC learning of Thr requires $\Omega(\log(N)/(\epsilon\alpha))$

examples [FX15] and *proper* (ϵ, δ) differentially private PAC learning requires $\Omega(\log^*(N)/(\epsilon\alpha))$ examples [BNSV15] (no lower bounds for non-proper learning and $\delta > 0$ case are known). Note that the VC dimension of this class is just 1.

We give an ϵ-differentially private prediction algorithm for agnostic learning of this class with the following guarantee:

Theorem 2.4. *For any $\alpha, \epsilon > (0, 1]$ and $N \in \mathbb{N}$, there exists an ϵ-differentially private prediction algorithm M that given $n \geq \frac{12\ln(2/\alpha)}{\alpha\epsilon}$ examples from an arbitrary distribution $\mathcal{P}$ over $[N] \times \{0, 1\}$ guarantees:*

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_{\mathcal{P}}(M(S))] \leq e^{\epsilon} \cdot (\mathsf{Opt}_{\mathcal{P}}(\mathsf{Thr}) + \alpha).$$

Note that this statement implies an upper bound of $n = O(\ln(1/\alpha)/(\alpha\epsilon))$ in the realizable case when $\mathsf{Opt}_{\mathcal{P}}(\mathsf{Thr}) = 0$ and also an upper bound of $n = O(\ln(1/\alpha)/(\alpha\epsilon) + \ln(1/\alpha)/\alpha^2)$ in the agnostic setting. The $\tilde{O}(1/\alpha^2)$ term arises from having to set $\epsilon < \alpha$ to ensure that the expected error is at most $\mathsf{Opt}_{\mathcal{P}}(\mathsf{Thr}) + O(\alpha)$. Our algorithm can also handle unions of k intervals (at the expense of an additional factor k in the sample complexity).

At a high level our algorithm works as follows. First, the examples are sorted. To determine the probability with which to output 1 on point x the algorithm traverses the examples on points smaller than x in increasing order. Starting from bias 1/2 the algorithm increases or decreases the current bias by a factor of (roughly) e^{ϵ} for each example it traverses. The bias is increased if the label of the example is 1 and decreased otherwise. Importantly, the bias is projected back to the interval $[\alpha, 1-\alpha]$ after each update. The algorithm outputs 1 with probability obtained at the end of this process. While the prediction privacy of our algorithm is relatively easy to establish, the analysis of its error is more delicate and we are not aware of similar algorithms having been proposed for this problem. Furthermore, our analysis only bounds the empirical error of this algorithm. The hypothesis produced by the algorithm is sufficiently complicated that it would not be possible to ensure generalization using VC dimension or similar techniques. Remarkably, the fact that our algorithm is prediction private allows us to prove that it generalizes.

2.3 Generalization

It has been known for a while that differential privacy is a notion of stability and hence implies bounds on the expectation of generalization error. Recent work in the context of adaptive data analysis has substantially strengthened this connection, proving that differential privacy ensures generalization with high probability [DFHPRR14; BNSSSU16; FS17]. Prediction privacy can also be seen as a notion of stability that is weaker than differential privacy but stronger than uniform prediction stability. We show how to derive relatively strong generalization guarantees from this notion of stability. These guarantees are stronger than those known for classical notions of stability (*e.g.* [BE02; SSSSS10]) but not as strong as those proved for differential privacy. Specifically, our generalization results (Lem. 6.4) imply that for every non-negative loss function ℓ, a moment $k \geq 1$, and an ϵ-differentially private prediction algorithm M:

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

where $\mathcal{E}_S[\ell(M(S)]$ denotes the expected empirical loss of $M(S)$ on S. Note that on the left hand side we are bounding the average loss on an independently drawn set of examples S' which is tightly concentrated around the expected loss $\mathcal{E}_{\mathcal{P}}[\ell(M(S)]$. For comparison, ϵ-differential privacy gives a similar bound with $e^{k\epsilon}$ factor instead of $e^{k^2\epsilon}$ [DFHPRR14]. The bound above is stated using the $k = 1$ version of this result. However this generalization bound implies that loss is also well concentrated. In Lemma 6.5 and Theorem 5.3 we give an example of how to derive high probability bounds on the generalization error from this moment bound.

2.4 Related work

Pathak et al. [PRR10] consider secure and differentially private aggregation of non-private linear models held by multiple mistrusting parties. They achieve it by computing the average model and adding noise to it. They do not consider accuracy guarantees of their approach formally.

To the best of our knowledge, the privacy-preserving aggregation of non-private predictions to produce privacy-preserving predictions was first investigated by Bilenko, Dwork, Muthukrishnan, Rothblum, Thakurta and Wang in 2014[1]. Bilenko *et al.*, obtained high levels of composition by exploiting the frequently high degree of (near) consensus among the predictions of the non-private models via a variant of the sparse-vector technique [DR14]. Our work shares the same goal of generating differentially private predictions. At the same time we formalize the general problem of learning with differentially private predictions and focus on the sample complexity of making a single prediction. In addition, we demonstrate approaches that go beyond privacy-preserving aggregation.

Aggregation of non-private models to produce labels while preserving privacy was also used in recent works of Hamm et al. [HCB16] and, subsequently, Papernot et al. [PAEGT17] and Papernot et al. [PSMRTE18] to give a new semi-supervised approach to differentially private learning. Specifically, their approach is predicated on availability of public *unlabeled* dataset Z. The dataset Z is labeled using differentially-private aggregation of labels provided by models trained on the sensitive dataset S. The labeled data is used to train a new model. Since differential privacy is closed under post-processing, this new model is privacy-preserving for S (but not for Z). The works of Papernot et al. [PAEGT17; PSMRTE18] deal primarily with techniques for accurately bounding the privacy parameters while ensuring accurate prediction on benchmark datasets. Hamm et al. [HCB16] also formally examine additional error that noisy aggregation introduces and explicitly rely on stability of strongly-convex regression problems to provide formal guarantees for their approach. Their framework and the guarantees are incomparable to ours, and, in particular, they do not avoid dependence on the dimension.

In a recent and independent work, Bassily et al. [BTT18] consider the formal guarantees for answering a sequence of prediction queries using differentially private aggregation techniques. They demonstrate that given a non-private learning algorithm has error of at most α (such as in the PAC model), there exists an algorithm that answers m prediction queries for points chosen i.i.d. from the same distribution with error $O(\alpha)$ and privacy parameter ϵ scaling as $\sqrt{m\alpha} \cdot \log m$ (for comparison, a direct application of composition theorems for differential privacy implies $\sqrt{m}$ scaling for an arbitrary sequence of queries). They then analyze the sample complexity of semi-supervised (or, equivalently, label-private) learning algorithm that is obtained by labeling a public unlabeled dataset using their algorithm for answering prediction queries.

We remark that all these works do not examine the problem of private prediction itself and

[1] This was the core of a larger project on privacy-preserving click prediction that did not survive the closing of the Silicon Valley lab.

focus on the aggregation-based approaches. Recall that in private prediction, it is the privacy of the training data for the predictor (model) that is being protected.

Organization: In Section 4.1 we provide additional details of our results for PAC learning. Results for agnostic learning and convex regression appear in Section 4.2. Section 5 formally describes our algorithm for agnostic learning of thresholds and unions of intervals. We discuss the generalization properties of private prediction in Section 6.

3 Preliminaries

Differential privacy Differential privacy [DMNS06] relies on bounding the divergence between distributions output by the algorithm on neighboring datasets. Specifically, for two random variables U and V and $\delta > 0$ the β-approximate max-divergence is defined as (*e.g.* [DR14]):

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

A randomized algorithm $M : X^n \to Y$ is said to be (ϵ,δ)-differentially private if for all pairs $S, S' \in X^n$ that differ on a single element, $D_\infty^\delta(M(S)\|M(S')) \leq \epsilon$. We note that our definitions and many of the results can be immediately extended to more refined notions of differential privacy such as those based on Renyi divergence [BS16; Mir].

The group privacy property of differential privacy (*e.g.* [DR14]) implies the that prediction privacy has the analogous property.

Lemma 3.1 (Group privacy)**.** *Let $M : (X\times Y)^n \times X \to Y$ be an (ϵ,δ)-differentially private prediction algorithm and $k\in\mathbb{N}$. For all pairs of data sets $S, S' \in (X\times Y)^n$ differing in at most k elements and all $x\in X$:*

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

3.1 Learning models

Definition 3.2. *An algorithm A PAC learns a concept class C from n examples if for every $\alpha > 0, \beta > 0$, $f\in C$ and distribution $\mathcal{D}$ over X, A given access to $S = \{(x_i,\ell_i)\}_{i\in[n]}$ where each x_i is drawn randomly and independently from $\mathcal{D}$ and $\ell_i = f(x_i)$, outputs, with probability at least $1-\beta$ over the choice of S and the randomness of A, a function $h : X \to \{0,1\}$ such that $\mathbf{Pr}_{x\sim\mathcal{D}}[f(x)\neq h(x)] \leq \alpha$.*

For a Boolean function h and a distribution $\mathcal{P}$ over $X\times\{0,1\}$ let $\mathsf{Err}_{\mathcal{P}}(h) = \mathbf{Pr}_{(x,\ell)\sim\mathcal{P}}[h(x)\neq\ell]$. Define $\mathsf{Opt}_{\mathcal{P}}(C) = \inf_{h\in C}\{\mathsf{Err}_{\mathcal{P}}(h)\}$. Kearns et al. [KSS94] define agnostic learning as follows.

Definition 3.3. *An algorithm A* agnostically *learns a concept class C from n examples if for every $\alpha > 0, \beta > 0$, distribution $\mathcal{P}$ over $X\times\{0,1\}$, A, given access to $S = \{(x_i,\ell_i)\}_{i\in[n]}$ where each (x_i,ℓ_i) is drawn randomly and independently from $\mathcal{P}$, outputs, with probability at least $1-\beta$ over the choice of S and the randomness of A, a function $h : X\to\{0,1\}$ such that $\mathsf{Err}_{\mathcal{P}}(h) \leq \mathsf{Opt}_{\mathcal{P}}(C) + \alpha$.*

4 Prediction privacy via subsampling and uniform stability

In this section we describe two variants of the baseline approach to obtaining prediction privacy. The baseline approach is based on a well-known observation that stability to replacement (or deletion) of a point can be improved by partitioning the dataset S into several subsamples $S_1, \dots, S_\ell$ running a learning algorithm on each of those subsamples to obtain predictors $f_1, \dots, f_\ell$ and then aggregating these predictors in a stable way. The first variant we describe is specialized to the simpler realizable case of classification. The second one is a generic model averaging that works for arbitrary convex loss functions. This case can also be used to obtain guarantees for agnostic learning of Boolean functions. In this case we will also explicitly use uniform prediction stability properties of the algorithm to derive its privacy guarantees.

4.1 PAC Learning

Our algorithm for PAC learning applies a soft majority rule to the outputs of $f_1, \dots, f_r$. Specifically, on a point x it will output a label b with probability proportional to $e^{\epsilon|\{i \in [r] : f_i(x) = b\}|}$. This approach works well for PAC learning since all predictors agree very well with the true labeling function. In particular, if each of the predictors has error of at most α, then the fraction of points on which more than 1/4 of the predictors output the wrong label cannot be more than 4α. Therefore the prediction of the soft majority will be close to the true label on all but the 4α fraction of the points.

Theorem 4.1. *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 ϵ-differentially private prediction algorithm M that PAC learns C using $n = r \cdot m(\alpha/4, \beta/r)$ examples, where $r = \lceil 6 \ln(4/\alpha)/\epsilon \rceil$.*

Proof. We denote by $c \in C$ the unknown labeling function and by $\mathcal{D}$ the unknown distribution over X. We let $r = \lceil 6 \ln(4/\alpha)/\epsilon \rceil$ and $n' = m(\alpha/4, \beta/t)$. Given a set S of $n = r \cdot n'$ examples we split them randomly into r disjoint subsets of size n'. We now run A with error parameter set to $\alpha/4$ and confidence parameter to β/r on each of those sets to obtain r functions $f_1, \dots, f_r$. On an input point x let $v(S,x) = 2|\{i \in [r] : f_i(x) = 1\}| - r$. Our algorithm outputs 1 with probability $\frac{e^{\epsilon v(S,x)/2}}{1+e^{\epsilon v(x)/2}}$ and 0 otherwise.

We first observe that M is an ϵ-differentially private prediction algorithm. This follows easily from observing that changing a single example can change only a single function f_i. Further, such change can change the value $v(S,x)$ by at most 2. Namely, for any pair of neighboring dataset S, S', $|v(S,x) - v(S',x)| \leq 2$. Now the privacy guarantees follow immediately from the definition of the output distribution of our algorithm being as: output 1 with probability $\frac{e^{\epsilon \cdot v(S,x)/2}}{1+e^{\epsilon \cdot v(S,x)/2}}$ and 0 with probability $\frac{1}{1+e^{\epsilon \cdot v(S,x)/2}}$ and the fact that for arbitrary real a, b,

$$\frac{\frac{e^a}{1+e^a}}{\frac{e^b}{1+e^b}} = e^{a-b} \cdot \frac{1+e^b}{1+e^a} \leq e^{|a-b|} \text{ and } \frac{1+e^b}{1+e^a} \leq e^{|a-b|}.$$

We now analyze the accuracy of our algorithm. Using the union bound, we know that with probability at least $1 - \beta$, for every $i \in [r]$, $\mathbf{Pr}_{\mathcal{D}}[f_i(x) \neq c(x)] = \mathbf{E}_{\mathcal{D}}[|f_i(x) - c(x)|] \leq \alpha/4$. This means that

$$\mathop{\mathbf{E}}_{\mathcal{D}}\left[\sum_{i \in [r]} |f_i(x) - c(x)|\right] \leq \alpha r/4.$$

By Markov's inequality this implies that

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

Now we claim that for every x such that $\sum_{i\in[r]}|f_i(x)-c(x)|\leq r/3$ we have that $\mathbf{Pr}_M[M(S,x)\neq c(x)]\leq\alpha/4$. If $c(x)=1$ then

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

This implies that

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

Similarly, if $c(x)=0$ we get that $v(S,x)\leq -r/3$ and $\mathbf{Pr}_M[M(S,x)\neq 0]\leq\alpha/4$.

Combining the last claim with inequality (1) we obtain that $\mathbf{Pr}_{\mathcal{D},M}[c(x)\neq M(S,x)]\leq\alpha$. □

Standard bounds on the sample complexity of PAC learning (e.g. [KV94]) state that $n=O\left(\frac{d\log(1/\alpha)+\log(1/\beta)}{\alpha}\right)$ examples suffice to PAC learn a class C of VC dimension d. Plugging this into our reduction we obtain that for every concept class C there exists a differentially private prediction algorithm that PAC learns the class C given $n=\tilde{O}(d/(\epsilon\alpha))$ examples.

Corollary 4.2 (Cor. 2.1 restated)**.** *Let C be a class of Boolean functions of VC dimension d. Then for all $\alpha,\beta,\epsilon>0$ there exists an ϵ-differentially private prediction algorithm M that PAC learns C with error α and confidence $1-\beta$ given $n=O\left(\frac{d\log^2(1/\alpha)+\log(1/\alpha)\log(\log(1/\alpha)/(\epsilon\beta))}{\epsilon\alpha}\right)$ examples.*

We demonstrate that this upper bound is essentially tight.

Theorem 4.3 (Thm. 2.2 restated)**.** *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 $n\geq d/(32\epsilon\alpha)$ examples.*

Proof. We first deal with the case $\alpha=1/4$. The reduction to general α is standard and is briefly described below.

Let $a_1,\ldots,a_d\in X$ be the set of points shattered by C. For convenience we refer to these points as $\{1,2,\ldots,d\}$. For a vector $b=(b_1,\ldots,b_d)\in\{0,1\}^d$ we denote by f_b the function in C that satisfies: for all $i\in[d], f_b(i)=b_i$. Let $\mathcal{D}$ be the uniform distribution over $[d]$.

Let M be the $(\epsilon,\epsilon/3)$-differentially private prediction algorithm for learning C. Now consider the expected prediction accuracy of M on a point $i\in[d]$, where the expectation is taken over the following process: $b\in\{0,1\}^d$ is chosen randomly, a dataset of size n is generated from $\mathcal{D}$ labeled by f_b and then M is run on S and i. Namely,

$$p_i\doteq\Pr_{b\sim\{0,1\}^d,S\sim(\mathcal{D},f_b)^n,M}[M(S,i)\neq b_i].$$

The accuracy and confidence guarantees of M imply that

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

This means that there exists i such that $p_i \leq 1/3$ and we fix i to this value for the rest of the argument.

Let $S^{\oplus i}$ denote the dataset in which all the examples for point i have their label flipped. By group prediction privacy of M (Lemma 3.1) we know that for every $v \in \{0,1\}$,

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

where t is the number of points i in the dataset.

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

$$\begin{aligned}\mathop{\mathbf{Pr}}_{S\sim(\mathcal{D},f_b)^n,M}[M(S,i)=v] &\leq e^{\epsilon s} \mathop{\mathbf{Pr}}_{S\sim(\mathcal{D},f_b)^n,M}[M(S^{\oplus i},i)=v] + s\cdot e^{(s-1)\epsilon}\delta + \frac{1}{24} \\ &< \frac{3}{2}\cdot \mathop{\mathbf{Pr}}_{S\sim(\mathcal{D},f_b)^n,M}[M(S^{\oplus i},i)=v] + \frac{1}{4\epsilon}\cdot\frac{3}{2}\cdot\frac{\epsilon}{3} + \frac{1}{24} \\ &= \frac{3}{2}\cdot \mathop{\mathbf{Pr}}_{S\sim(\mathcal{D},f_b)^n,M}[M(S^{\oplus i},i)=v] + \frac{1}{6}. \end{aligned} \tag{2}$$

Observe that for every b and $S \sim (\mathcal{D},f_b)^n$, $S^{\oplus i}$ is distributed identically to $S \sim (\mathcal{D}, f_{b^{\oplus i}})^n$. This implies that,

$$\begin{aligned}\mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d,\ S\sim(\mathcal{D},f_b)^n,M}[M(S^{\oplus i},i)=b_i] &= \mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d,S\sim(\mathcal{D},f_{b^{\oplus i}})^n,M}[M(S,i)=b_i] \\ &= \mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d,S\sim(\mathcal{D},f_b)^n,M}[M(S,i)\neq b_i] \\ &= p_i. \end{aligned} \tag{3}$$

By plugging equations (2) and (3) into the definition of p_i we obtain that:

$$\begin{aligned}1-p_i &= \mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d,S\sim(\mathcal{D},f_b)^n,M}[M(S,i)=b_i] \\ &< \frac{3}{2}\cdot \mathop{\mathbf{Pr}}_{b\sim\{0,1\}^d,S\sim(\mathcal{D},f_b)^n,M}[M(S^{\oplus i},i)=b_i] + 1/6 \\ &= \frac{3}{2}\cdot p_i + 1/6. \end{aligned}$$

This cannot hold when $p_i \leq 1/3$, implying that $n > d/(8\epsilon)$.

Finally we reduce the general α case to the analysis for $\alpha = 1/4$ in the standard way (*e.g.* [KV94; SSBD14]). We let $\mathcal{D}_\alpha$ be the distribution that outputs the point with index d, with probability $1-4\alpha$ and outputs a uniformly and randomly chosen $i \in [d-1]$ with probability 4α. Achieving error of α on $\mathcal{D}_\alpha$ requires achieving error of $1/4$ on the uniform distribution on $[d-1]$. Only approximately 4α fraction of the examples will be useful for obtaining low error relative to the uniform distribution on $[d-1]$ and therefore the reduction multiplies the lower bound by $\Omega(1/\alpha)$. More formally, we consider only target functions f_b where $b_d = 0$. Therefore for all target functions, examples on point d will be identical. Now given that $n \leq d/(32\alpha\epsilon)$, (and for d larger than some fixed constant) with probability at least $1/24$ over the choice of $S \sim (\mathcal{D}_\alpha, f_b)^n$, S includes at most $1/(4\epsilon)$ points equal to i as before. Hence the rest of the argument is essentially identical. □

4.2 Learning with convex losses and stability

We now deal with the general setting of minimizing convex losses. Specifically, these are problems in which the goal is to minimize the expected loss function $\mathbf{E}_{(x,y)\sim\mathcal{P}}[\ell(f(x), y)]$, where ℓ is a convex function in the first parameter. Note that learning of Boolean functions is a special case in which we use $\ell(a, b) = |a - b|$.

To deal with this case we will rely on (non-private) learning algorithms that are prediction stable in the usual numerical sense. That is

Definition 4.4. *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 $A(S, \cdot)$ denotes the function output by A on dataset S.

This notion is closely-related to the standard uniform replace-one stability [BE02; SSSSS10] which bounds the change in loss $|\ell(A(S, x), y) - \ell(A(S', x), y)| \leq \gamma$ for all x, y. Crucially, the analyses of uniform loss stability that we are aware of implicitly prove bounds on prediction stability. Hence such analyses can be adapted to our applications (we provide some examples below).

Low sensitivity of the value predicted at each point implies that addition of Laplace or Gaussian noise suffices to obtain a differentially private prediction algorithm. The additional error due to noise can be controlled for Lipschitz loss functions. Somewhat stronger bounds on the additional error can be shown if the loss function is smooth (that is, its derivative is Lipschitz-bounded).

Lemma 4.5. *Let $\ell : \mathbb{R} \times Y \to \mathbb{R}$ 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 ϵ-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 $\ell(\cdot, y)$ is L_ℓ-Lipschitz in the first parameter for all $y \in Y$ then*

$$\mathcal{E}[\ell(M(S))] \leq \mathcal{E}[\ell(A(S))] + L_\ell \cdot \gamma/\epsilon.$$

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

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

Proof. Given S and x let v be the output of A on S applied to x. We output $v + \zeta$, where ζ is distributed according to Laplace distribution with scale ϵ/γ. By definition of uniform RO prediction stability and standard properties of the Laplace noise addition (e.g. [DR14]), this algorithm is ϵ-differentially private. To obtain the claimed upper bound on the expected loss observe that if ℓ is L_ℓ-Lipschitz then for any S, x and y,

$$\mathop{\mathbf{E}}_{M}[\ell(A(S, x) + \zeta, y)] \leq \ell(A(S, x), y) + \mathop{\mathbf{E}}_{M}[L_\ell \cdot |\zeta|] = \ell(A(S, x), y) + L_\ell \cdot \gamma/\epsilon,$$

where we have used the fact that $|\zeta|$ is distributed according to an exponential distribution with rate γ/ϵ. This upper bound holds pointwise and therefore for any distribution $\mathcal{P}$ over $X \times Y$.

If ℓ is σ-smooth, then by the definition of smoothness: for any x and y,

$$\ell(A(S,x)+\zeta,y) \le \ell(A(S,x),y) + \ell'(A(S,x),y)\cdot\zeta + \frac{\sigma}{2}\cdot\zeta^2.$$

Using the fact that $\mathbf{E}[\zeta]=0$ and $\mathbf{E}[\zeta^2]=2\gamma^2/\epsilon^2$, we obtain

$$\mathop{\mathbf{E}}_{M}[\ell(A(S,x)+\zeta,y)] \le \ell(A(S,x),y) + \gamma^2\sigma/\epsilon^2.$$

□

Naturally, many learning algorithms are not sufficiently prediction stable to ensure that the additional error due to noise is sufficiently small. However it is easy to see that it is possible to amplify stability by averaging the predictions obtained on disjoint subsamples. Convexity of the loss function then implies that such averaging will preserve the bounds on the expected loss. Specifically, the following lemma follows immediately from the argument above.

Lemma 4.6. *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 $r \in \mathbb{N}$ 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 $\ell(\cdot,\cdot)$ convex in the first parameter, if for a distribution $\mathcal{P}$ over $X\times Y$, A has the guarantee that $\mathbf{E}_{S\sim\mathcal{P}^n}\left[\mathcal{E}_{\mathcal{P}}[\ell(A(S))]\right] \le v$ for some value v that may depend on $\mathcal{P}$ and the parameters of the learning problem then $\mathbf{E}_{S'\sim\mathcal{P}^{rn}}\left[\mathcal{E}_{\mathcal{P}}[\ell(A'(S'))]\right] \le v$. Alternatively, if for some $\beta>0$,*

$$\mathop{\mathbf{Pr}}_{S\sim\mathcal{P}^n}\left[\mathcal{E}_{\mathcal{P}}[\ell(A(S))] \ge v\right] \le \beta$$

then

$$\mathop{\mathbf{Pr}}_{S'\sim\mathcal{P}^{rn}}\left[\mathcal{E}_{\mathcal{P}}[\ell(A'(S'))] \ge v\right] \le r\beta.$$

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

4.3 Agnostic learning

We now spell out immediate corollary of Lemmas 4.6 and 4.5 to agnostic learning of Boolean functions. Agnostic learning of Boolean functions reduces to learning of a real-valued function f with absolute loss $\ell(a,y)=|y-a|$. Note that a real-valued prediction $f(x)$ can also be equivalently thought of as predicting 1 with probability p, where p is equal to $f(x)$ projected to the interval $[0,1]$. The expected disagreement of such prediction on $y\in\{0,1\}$ is upper bounded by $|y-f(x)|$. This loss function is convex and 1-Lipschitz in the first parameter. Further, any learning algorithm that outputs a Boolean function is uniform RO prediction stable at the trivial rate 1. Hence to ensure that the additional error in Lemma 4.5 is at most α we need to amplify the stability to $\alpha\cdot\epsilon$. This requires $r=1/(\alpha\epsilon)$ subsamples. Therefore overall we obtain the following general reduction for agnostic learning of Boolean function.

Corollary 4.7. *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 $n = 2\cdot m(\alpha/2, 2\beta\alpha\epsilon)/(\alpha\epsilon)$ examples.*

As in the case of PAC learning, this reduction allows us to upper bound the sample complexity of private prediction for agnostic learning of C. Specifically, $n = O\left(\frac{d+\log(1/\beta)}{\alpha^2}\right)$ samples suffice to agnostically learn a class VC dimension d (e.g. [SSBD14]). Plugging this into our reduction we get:

Corollary 4.8 (Cor. 2.3 restated)**.** *Let C be a class of Boolean functions of VC dimension d. Then for all $\alpha, \beta, \epsilon \in (0,1]$ there exists an ϵ-differentially private prediction algorithm M that agnostically learns C with excess error α and confidence $1-\beta$ given $n = O\left(\frac{d+\log(1/(\alpha\beta\epsilon))}{\epsilon\alpha^3}\right)$ examples.*

4.4 Applications to convex regression problems

We now apply this general approach to convex regression problems. Specifically, problems of the form:

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

where $\mathcal{K} \subset \mathbb{R}^d$ is some convex body and $\ell(f(\cdot,x),y)$ is a convex function over $\mathcal{K}$ for all $(x,y) \in X \times Y$. For simplicity, we will restrict our attention to the case when $\mathcal{K}$ is a subset of the Euclidean ball of radius R which we denote by $\mathcal{B}_2^d(R)$. Several classes of such problems are known to be solvable efficiently by uniform RO prediction stable algorithms. Our result will be based on the following upper bound on prediction stability of strongly convex optimization that is implicit in [BE02; SSSSS10].

Theorem 4.9 ([SSSSS10])**.** *Let $\mathcal{K} \subseteq \mathcal{B}_2^d(R)$ be a convex body, $\{f(\cdot,x) \mid x \in X\}$ be a family of L_f-Lipschitz functions over $\mathcal{K}$, $\ell : \mathbb{R}\times Y \to \mathbb{R}$ be convex, L_ℓ-Lipschitz loss function and $\ell(f(\cdot,x),y)$ be λ-strongly convex for all $(x,y) \in X \times Y$. For a dataset $S \in (X\times Y)^n$ let w_S denote the empirical minimizer of loss on S: $w_S = \operatorname{argmin}_{w\in\mathcal{K}} \sum_{(x,y)\in S}[\ell(f(w,x),y)]$. Then the algorithm that given S, outputs a function $f(w_S,\cdot)$ is uniform RO prediction stable with rate $\frac{4L_f^2\cdot L_\ell}{\lambda n}$. Further, for every distribution $\mathcal{P}$ over $X \times Y$:*

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}\left[\mathop{\mathbf{E}}_{\mathcal{P}}[\ell(f(w_S,x),y)]\right] \le \min_{w\in\mathcal{K}} \mathop{\mathbf{E}}_{\mathcal{P}}[\ell(f(w,x),y)] + \frac{4L_f^2\cdot L_\ell^2}{\lambda n}.$$

We remark that this version may appear somewhat different from the results in [SSSSS10] as they consider a single convex loss function with Lipshitz constant L that gives the loss of the model with parameter w on an example. Our statement follows from noting that their work proves that for any pair of datasets S and S' that differ in a single element, $\|w_S - w_{S'}\|_2 \le \frac{4L}{\lambda n}$. This implies that for all x,

$$|f(w_S,x) - f(w_{S'},x)| \le \frac{4L\cdot L_f}{\lambda n} \le \frac{4L_f^2\cdot L_\ell}{\lambda n}.$$

By combining this result with Lemma 4.5 we get the following private prediction algorithm.

Corollary 4.10. *Let $\mathcal{K} \subseteq \mathcal{B}_2^d(R)$ be a convex body, $\{f(\cdot,x) \mid x \in X\}$ be a family of L_f-Lipschitz functions over $\mathcal{K}$, $\ell : \mathbb{R}\times Y \to \mathbb{R}$ be convex, L_ℓ-Lipschitz loss function and $\ell(f(\cdot,x),y)$ be λ-strongly convex for all $(x,y) \in X \times Y$. For every $\epsilon > 0$, there exists an ϵ-differentially private prediction algorithm M that for any probability distribution $\mathcal{P}$ over $X \times Y$ satisfies:*

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

Corollary 4.10 requires strong convexity to obtain meaningful guarantees. However, as pointed out in [SSSSS10], it is possible to add a strongly convex regularizing term $\lambda\|w\|^2$ to the objective function that has sufficiently small effect on the loss function while ensuring stability (and generalization). Specifically, by setting $\lambda = \frac{2L_f L_\ell}{R\sqrt{n\epsilon/(1+\epsilon)}}$ the objective function will change by at most $\frac{2RL_f L_\ell}{\sqrt{n\epsilon/(1+\epsilon)}}$ since w is assumed to be in a ball of radius R. Plugging this value of λ into Corollary 4.10 and accounting for the additional error we get:

Corollary 4.11. *Let $\mathcal{K} \subseteq \mathcal{B}_2^d(R)$ be a convex body, $\{f(\cdot, x) \mid x \in X\}$ be a family of L_f-Lipschitz functions over $\mathcal{K}$, $\ell : \mathbb{R} \times Y \to \mathbb{R}$ be convex, L_ℓ-Lipschitz loss function and $\ell(f(\cdot, x), y)$ be convex for all $(x, y) \in X \times Y$. For every $\epsilon > 0$, there exists an ϵ-differentially private prediction algorithm M that for any probability distribution $\mathcal{P}$ over $X \times Y$ satisfies:*

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

Somewhat stronger results can be obtained for regression problems in which the loss function is also smooth. In this case we can rely on the stability of gradient descent for smooth functions implicit in [HRS16]. This result applies to the projected stochastic gradient descent algorithm. For concreteness, let PSGD_η denote the algorithm that starting from w_0 being the origin, performs the following iterative updates for every $i \in [n]$:

$$w_{i+1} \leftarrow \mathsf{Proj}_\mathcal{K}(w_i + \eta \cdot \nabla\ell(f(w_i, x_i), y_i)),$$

where $\mathsf{Proj}_\mathcal{K}$ denotes projection to $\mathcal{K}$. The algorithm returns the average iterate: $\bar{w}_S \doteq \frac{1}{n}\sum_{i\in[n]} w_i$.

Theorem 4.12 ([HRS16])**.** *Let $\mathcal{K} \subseteq \mathcal{B}_2^d(R)$ be a convex body, $\{f(\cdot, x) \mid x \in X\}$ be a family of L_f-Lipschitz functions over $\mathcal{K}$, $\ell : \mathbb{R} \times Y \to \mathbb{R}$ be convex, L_ℓ-Lipschitz loss function and $\ell(f(\cdot, x), y)$ be convex and σ-smooth for all $(x, y) \in X \times Y$. For a dataset $S \in (X \times Y)^n$ let $\bar{w}_S$ denote the output of $PSGD_\eta$ for $\eta = R/(L_f L_\ell\sqrt{n})$. If $\sigma \le 2/\eta$ then the algorithm that outputs $f(\bar{w}_S, \cdot)$ is uniform RO prediction stable with rate $RL_f/\sqrt{n}$. Further, for every distribution $\mathcal{P}$ over $X \times Y$:*

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}\left[\mathop{\mathbf{E}}_{\mathcal{P}}[\ell(f(w_S,x),y)]\right] \le \min_{w\in\mathcal{K}} \mathop{\mathbf{E}}_{\mathcal{P}}[\ell(f(w,x),y)] + \frac{2L_f \cdot L_\ell \cdot R}{\sqrt{n}}.$$

Plugging this result into our framework we obtain the following stronger bound for convex and smooth functions. We will also additionally assume that the loss function $\ell(\cdot, y)$ is σ_ℓ-smooth in the first parameter for all y.

Corollary 4.13. *Let $\mathcal{K} \subseteq \mathcal{B}_2^d(R)$ be a convex body, $\{f(\cdot, x) \mid x \in X\}$ be a family of L_f-Lipschitz functions over $\mathcal{K}$, $\ell : \mathbb{R}\times Y \to \mathbb{R}$ be convex, L_ℓ-Lipschitz and σ_ℓ-smooth loss function and $\ell(f(\cdot, x), y)$ be convex and σ-smooth for all $(x, y) \in X \times Y$. If $\sigma \le 2L_f L_\ell\sqrt{n}/R$ then for every $\epsilon > 0$, there exists an ϵ-differentially private prediction algorithm M that for any probability distribution $\mathcal{P}$ over $X \times Y$ satisfies:*

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathcal{E}_\mathcal{P}[\ell(M(S))]] \le \min_{w\in\mathcal{K}} \mathop{\mathbf{E}}_{\mathcal{P}}[\ell(f(w,x),y)] + \frac{2 \cdot R \cdot L_f \cdot L_\ell}{\sqrt{n}} + \frac{\sigma_\ell \cdot R^2 \cdot L_f^2}{n\epsilon^2}.$$

One way to interpret this result is that for sufficiently smooth loss functions, the error caused by noise become comparable to the statistical error when ϵ scales as $n^{-1/4}$. In other words, this level of differential privacy is obtained essentially for free. We also remark that the assumption that $\ell(\cdot, y)$ is σ_ℓ-smooth can also be used in Cor. 4.11 to obtain the same bound (up to a constant) as the one we got in Cor. 4.13. Similarly, without this assumption Cor. 4.13 would give essentially the same bound as Cor. 4.11.

5 Learning of thresholds and unions of intervals

We demonstrate a nearly optimal algorithm for agnostically learning the class of threshold functions on a line (and more generally unions of intervals). For $N \in \mathbb{N}$ we consider threshold functions over $[N]$. Specifically, for $a \in [N+1]$, let θ_a denote the threshold function "$x \geq a$" over $[N]$ and let Thr_N denote the set of all $N+1$ threshold functions over $[N]$. More generally, we define a union of intervals by an increasing sequence of interval ends. Specifically, for an increasing sequence of integer numbers $1 \leq a_1 < a_2 < \cdots < a_k \leq N+1$, we define $\theta_{a_{[k]}}$ to be the function defined as follows: given $x \in M$ let $t(x)$ be the largest index such that $x \geq a_t$. Then $\theta_{a_{[k]}}(x)$ is equal to 1 if and only if t is odd. We denote by $\mathsf{Thr}_{N,k}$ the class of all functions of this type.

Our algorithm, referred to as the exponential projected walk, is described below. For convenience, we assume that the dataset $S = (x_1, y_1), \ldots, (x_n, y_n)$ is given in a sorted order, that is $x_i \leq x_j$ for all $i < j$. Further, we define $\mathsf{Proj}_{[A,B]}(x)$ as the projection of x onto interval $[A, B]$.

```
Parameters: $T, \epsilon$
Input: dataset $S = ((x_1, y_1), \ldots, (x_n, y_n)) \in ([N] \times \{0,1\})^n$ in sorted order and a point $x$
  $i = 0$, $v_0 = 0$
  while $x_i \leq x$ do:
    $i$++
    $v'_i = v_{i-1} + (2y_i - 1)$
    $v_i = \mathsf{Proj}_{[-T,T]}(v'_i)$
  $t = i$
  Sample $b$ from Bernoulli distribution with bias $\frac{e^{\epsilon \cdot v_t/2}}{1+e^{\epsilon \cdot v_t/2}}$.
Output $b$
```

Figure 1: ExpPW(T, ϵ): Exponential projected walk algorithm

We first prove that for any setting of parameters and n, the exponential projected walk is an ϵ-differentially private prediction algorithm.

Lemma 5.1. *For any $n, T \in \mathbb{N}$, $\epsilon > 0$,* ExpPW(T, ϵ) *is an ϵ-differentially private prediction algorithm.*

Proof. Let $S = (x_1, y_1), \ldots, (x_n, y_n)$ be a dataset in a sorted order and let S' be a dataset that differs from S in a single element. There exist indices i and j such that S' can be seen as removing the example i and then inserting example (x', y') into j-th position so that the resulting sequence of examples is still in the sorted order. Let S^{-i} denote S with i-th element removed. We will prove that for any x, $D_\infty(M(S,x)\|M(S^{-i},x)) \le \epsilon/2$ and $D_\infty(M(S^{-i},x)\|M(S,x)) \le \epsilon/2$. Note that, by removing element j from S' we also obtain S^{-i}. Hence our argument will imply that

$$D_\infty(M(S,x)\|M(S',x)) \le D_\infty(M(S,x)\|M(S^{-i},x)) + D_\infty(M(S^{-i},x)\|M(S',x)) \le \epsilon.$$

Let M denote $\mathrm{ExpPW}(T,\epsilon)$ and let $V(S,x)$ denote the value of v_t at the end of running $M(S,x)$. Note that in order to prove the claim it is sufficient to prove that $|V(S,x) - V(S^{-i},x)| \le 1$. As in the proof of Theorem 4.1, the claim then follows immediately from the definition of the output distribution of ExpPW being as: output 1 with probability $\frac{e^{\epsilon\cdot V(S,x)/2}}{1+e^{\epsilon\cdot V(S,x)/2}}$ and 0 with probability $\frac{1}{1+e^{\epsilon\cdot V(S,x)/2}}$.

To show that $|V(S,x) - V(S^{-i},x)| \le 1$ we observe that: for $x < x_i$ removal of (x_i,y_i) does not affect the output of the algorithm. Hence $V(S,x) = V(S^{-i},x)$. For $x \ge x_i$ the values $v_0,\dots,v_n$ of the walk on S will have an additional step of length at most 1 at index i. After that step the update points $(x_{i+1},y_{i+1}),\dots,(x_t,y_t)$ will be identical for both sequences. Performing such an update step on two different values u and v does not increase the distance between the values. Hence at the end of the walk we obtain that $|V(S,x) - V(S^{-i},x)| \le |V(S,x_i) - V(S^{-i},x_i)| \le 1$. □

We now prove that our algorithm will achieve low empirical error. Let

$$\mathsf{Err}_S(M(S)) \doteq \frac{1}{n}\sum_{i\in[n]} \mathop{\mathbf{Pr}}_{M}[M(S,x_i) \ne y_i]$$

and for a class of functions C let $\mathsf{Opt}_S(C) \doteq \min_{f\in C} \mathsf{Err}_S(f)$.

Lemma 5.2. *Let $S = (x_1,y_1),\dots,(x_n,y_n) \in ([N]\times\{0,1\})^n$ be a set of n examples. Then for $M = \mathrm{ExpPW}(T,\epsilon)$,*

$$\mathsf{Err}_S(M(S)) \le \mathsf{Opt}_S(\mathsf{Thr}_{N,k}) + (k+2)T/n + e^{-\epsilon T/2}.$$

Proof. As before, we assume that examples in S are in the sorted order. Let $V(S,x)$ denote the value of v_t at the end of running $M(S,x)$. Let $f \in \mathsf{Thr}_{N,k}$ be the interval function with the lowest error on S and let $a_{[k]} = a_1,\dots,a_k$ be its parameters.

We first deal with points x_i such that $V(S,x_i) \in \{-T,T\}$. We denote the set of indices of these points by I. Observe that if $V(S,x_i) = T$ then $y_i = 1$. This is true since for $y_i = 0$ the projected walk makes a -1 step and then projects to $[-T,T]$. Such step cannot end in $V(S,x_i) = T$. Similarly, if $V(S,x_i) = -T$ then $y_i = 0$. This means that for $i \in I$, $\mathbf{Pr}_M[M(S,x_i) \ne y_i] = 1/(e^{\epsilon T/2}+1) \le e^{-\epsilon T/2}$. Consequently,

$$\sum_{i\in I} \mathop{\mathbf{Pr}}_{M}[M(S,x_i) \ne y_i] \le |I|\cdot e^{-\epsilon T/2}. \tag{4}$$

We now split the examples into bands according to the endpoints of the step that the projected walk took on them. Specifically, for $v \in \{-T,-T+1,\dots,T-1\}$ let I_v be the set of indices i such that either $V(S,x_{i-1}) = v$ and $V(S,x_i) = v+1$ or $V(S,x_{i-1}) = v+1$ and $V(S,x_i) = v$. Let $J \doteq \bigcup_{-T\le v\le T-1} I_v$. Note that for $u \ne v$, $I_u \cap I_v = \emptyset$ but I_{-T} and I_{T-1} may include some of the points in I. Also the collection of these sets covers all the indices: $I \cup J = [n]$.

We make several simple observations about examples with indices in I_v. Let $i_1 < i_2 < \cdots < i_\ell$ be the indices of points in I_v. The labels of points have to alternate, or $y_{i_j} \ne y_{i_{j+1}}$ for all $j \in [\ell-1]$.

This is due to the fact that the walk cannot traverse the interval of values $[v, v+1]$ twice in a row in the same direction. We use this to compute the total error of both M and f on points with indices in I_v.

If $y_{i_j} = 1$ then the projected walk made $+1$ step at i_j and therefore $\mathbf{Pr}_M[M(S, x_{i_j}) \neq y_{i_j}] = 1/(e^{\epsilon(v+1)/2}+1)$. While if $y_{i_j} = 0$ then the projected walk made -1 step at i_j and therefore $\mathbf{Pr}_M[M(S, x_{i_j}) \neq y_{i_j}] = e^{\epsilon v/2}/(e^{\epsilon v/2}+1)$. Note that

$$1/(e^{\epsilon(v+1)/2}+1) + e^{\epsilon v/2}/(e^{\epsilon v/2}+1) \leq 1.$$

Therefore for any pair of points with opposite labels the sum of expected errors is at most 1. The alternation of labels for examples in I_v then implies that

$$\sum_{i \in I_v} \mathop{\mathbf{Pr}}_{M}[M(S, x_i) \neq y_i] = \sum_{j \in [\ell]} \mathop{\mathbf{Pr}}_{M}[M(S, x_{i_j}) \neq y_{i_j}] \leq \frac{|I_v|+1}{2}, \tag{5}$$

where the additional 1 is the bound on the probability of error on a point that has no pair with the opposite label (which happens when the size of I_v is odd).

Now consider the error of $f = \theta_{a_{[k]}}$ on points in I_v. Note that f splits $[N]$ into at most $k+1$ intervals where the value of f is constant. For $r \in [k+1]$ let J_r denote the r-th interval and $I_{v,r} \doteq I_v \cap J_r$. The alternation of labels in I_v implies that the number of points with indices in $I_{v,r}$ on which f is correct can be larger than the number of points on which f is wrong by at most 1. That is:

$$\sum_{i \in I_{v,r}} |f(x_i) - y_i| \geq \frac{|I_{v,r}|-1}{2}.$$

Hence

$$\sum_{i \in I_v} |f(x_i) - y_i| = \sum_{r \in [k+1]} \sum_{i \in I_{v,r}} |f(x_i) - y_i| \geq \sum_{r \in [k+1]} \frac{|I_{v,r}|-1}{2} = \frac{|I_v|-k-1}{2}. \tag{6}$$

Combining the inequalities (5) and (6), we obtain that

$$\sum_{i \in I_v} \mathop{\mathbf{Pr}}_{M}[M(S, x_i) \neq y_i] \leq \sum_{i \in I_v} |f(x_i) - y_i| + \frac{k+2}{2}.$$

Summing up over all values of $v \in \{-T, -T+1, \ldots, T-1\}$ we get

$$\begin{aligned}
\sum_{i \in J} \mathop{\mathbf{Pr}}_{M}[M(S, x_i) \neq y_i] &= \sum_{v \in \{-T,-T+1,\ldots,T-1\}} \sum_{i \in I_v} \mathop{\mathbf{Pr}}_{M}[M(S, x_i) \neq y_i] \\
&\leq \sum_{v \in \{-T,-T+1,\ldots,T-1\}} \left(\sum_{i \in I_v} |f(x_i) - y_i| \right) + \frac{k+2}{2} \\
&= \sum_{i \in J} |f(x_i) - y_i| + \frac{2T(k+2)}{2} \\
&\leq n \cdot \mathsf{Err}_S(f) + T(k+2)
\end{aligned}$$

Finally, $I \cup J = [n]$ and therefore combining this with equation (4) we get:

$$\begin{aligned} \mathsf{Err}_S(M(S)) &\leq \frac{1}{n}\left(\sum_{i\in J} \mathop{\mathbf{Pr}}_{M}[M(S,x_i) \neq y_i] + \sum_{i\in I} \mathop{\mathbf{Pr}}_{M}[M(S,x_i) \neq y_i]\right) \\ &\leq \mathsf{Err}_S(f) + \frac{T(k+2)}{n} + \frac{|I|}{n} \cdot e^{-\epsilon T/2} \\ &\leq \mathsf{Opt}_S(\mathsf{Thr}_{N,k}) + \frac{T(k+2)}{n} + e^{-\epsilon T/2}. \end{aligned}$$

□

Now by choosing $T = \lceil 2\ln(2/\alpha)/\epsilon \rceil$ we can ensure that the empirical error of $\mathrm{ExpPW}(T,\epsilon)$ is close to the best possible by a function in $\mathsf{Thr}_{N,k}$. To prove that our algorithm generalizes we appeal to generalization properties of differentially private prediction described in Section 6.

Theorem 5.3 (subsumes Thm. 2.4). *For any $\alpha, \epsilon > 0$ and $k, N \in \mathbb{N}$, $T = \lceil 2\ln(2/\alpha)/\epsilon \rceil$ let $M \doteq \mathrm{ExpPW}(T,\epsilon)$. Then M is an ϵ-differentially private prediction algorithm and given $n \geq \frac{4(k+2)\ln(2/\alpha)}{\alpha\epsilon}$ examples from an arbitrary distribution $\mathcal{P}$ over $[N] \times \{0,1\}$ its output satisfies:*

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_{\mathcal{P}}(M(S))] \leq e^{\epsilon} \cdot (\mathsf{Opt}_{\mathcal{P}}(\mathsf{Thr}_{N,k}) + \alpha).$$

In particular, setting $\epsilon = \alpha/2$ we get that for $n = O(k\ln(1/\alpha)/\alpha^2)$

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_{\mathcal{P}}(M(S))] \leq \mathsf{Opt}_{\mathcal{P}}(\mathsf{Thr}_{N,k}) + \alpha.$$

Further, if $\mathsf{Opt}_{\mathcal{P}}(\mathsf{Thr}_{N,k}) = 0$ and $\epsilon \leq 1/(16\ln(1/\beta))$, then for every $\beta \in (0,1)$,

$$\mathop{\mathbf{Pr}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_{\mathcal{P}}(M(S)) \geq 3\alpha] \leq 2\beta.$$

Proof. Evaluation of the disagreement error $|M(S,x) - y|$ is ϵ-differentially private for all x and y. Therefore we can apply Lemma 6.4 to obtain:

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_{\mathcal{P}}(M(S))] = \mathop{\mathbf{E}}_{S,S'\sim\mathcal{P}^n}[\mathsf{Err}_{S'}(M(S))] \leq e^{\epsilon} \cdot \mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_S(M(S))].$$

By applying Lemma 5.2 we get that

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_S(M(S))] \leq \mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Opt}_S(\mathsf{Thr}_{N,k})] + (k+2)T/n + e^{-\epsilon T/2} \leq \mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Opt}_S(\mathsf{Thr}_{N,k})] + \alpha.$$

Finally, we recall a well known fact that for any class C and n, $\mathbf{E}_{S\sim\mathcal{P}^n}[\mathsf{Opt}_S(C)] \leq \mathsf{Opt}_{\mathcal{P}}(C)$. Hence,

$$\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_S(M(S))] \leq e^{\epsilon} \cdot \left(\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}[\mathsf{Opt}_S(\mathsf{Thr}_{N,k})] + \alpha\right) \leq e^{\epsilon} \cdot (\mathsf{Opt}_{\mathcal{P}}(\mathsf{Thr}_{N,k}) + \alpha).$$

To establish the high probability bounds for the realizable case, we note that if $\mathsf{Opt}_{\mathcal{P}}(\mathsf{Thr}_{N,k}) = 0$ then for every S that includes only the elements in the support of $\mathcal{P}$, $\mathsf{Opt}_S(\mathsf{Thr}_{N,k}) = 0$. Hence $\mathsf{Err}_S(M(S)) \leq \alpha$. Now applying Lemma 6.5, we obtain:

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

Using the condition that $\epsilon \leq 1/(16\ln(1/\beta))$, we get that $e^{2\sqrt{\epsilon\ln(1/\beta)}} \leq e^{1/2} \leq 2$. Hence

$$\mathop{\mathbf{Pr}}_{S,S'\sim\mathcal{P}^n}[\mathsf{Err}_{S'}(M(S)) \geq 2\alpha] \leq \beta.$$

Now if for some S, $\mathsf{Err}_{\mathcal{P}}(M(S)) \geq 3\alpha$, then using the fact that $\mathsf{Err}_{S'}(M(S))$ is the mean of n independent Bernoulli random variables with bias $\mathsf{Err}_{\mathcal{P}}(M(S))$, we get that with high probability $\mathsf{Err}_{S'}(M(S)) \geq 2\alpha$. Specifically, by Chernoff bound, for $n \geq 6\ln(2)/\alpha$ (which is satisfied by the conditions of our theorem),

$$\mathop{\mathbf{Pr}}_{S'\sim\mathcal{P}^n}[\mathsf{Err}_{S'}(M(S)) \geq 2\alpha] \geq 1/2.$$

Thus

$$\beta \geq \mathop{\mathbf{Pr}}_{S,S'\sim\mathcal{P}^n}[\mathsf{Err}_{S'}(M(S)) \geq 2\alpha] \geq \frac{1}{2}\mathop{\mathbf{Pr}}_{S\sim\mathcal{P}^n}[\mathsf{Err}_{\mathcal{P}}(M(S)) \geq 3\alpha].$$

□

Our generalization results in Section 6 are not strong enough to prove that our algorithm has low expected error with high probability over S in the agnostic case (while having asymptotically optimal sample complexity). Establishing such a result is an interesting open problem.

6 Stability and Generalization

We now view prediction privacy as a notion of stability and derive generalization properties of private prediction algorithms. Our results will be stated for a somewhat more general class of algorithms that compute any function of a single data point while satisfying differential privacy.

Definition 6.1 (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.*

In our application the algorithm M will be computing the loss of the prediction produced on S by a prediction algorithm M'. Namely, $Z = X \times Y$ and for some loss function ℓ, $M(S,(x,y)) = \ell(M'(S,x),y)$. Note that by the postprocessing property of differential privacy (*e.g.* [DR14]), this evaluation is differentially private with the same parameters. We state this formally below.

Lemma 6.2 (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 $\ell : \mathbb{R}\times Y \to \mathbb{R}$, $M(S,(x,y)) \doteq \ell(M'(S,x),y)$ is an (ϵ,δ)-differentially private evaluation algorithm.*

We will need the following simple property of D_∞^δ to argue about closeness of expectations.

Lemma 6.3 (e.g.[FS17])**.** *Let U and V be two random variables over $[0,B]$ such that $D_\infty^\delta(U\|V) \leq \epsilon$. Then $\mathbf{E}[U] \leq e^\epsilon \cdot \mathbf{E}[V] + \delta\cdot B$.*

Let $S = (z_1, z_2, \ldots, z_n)$ and $S' = (z'_1, z'_2, \ldots, z'_n)$ be two sequences of samples drawn randomly and independently from some unknown distribution $\mathcal{P}$ over Z. We consider the relationship between the empirical mean of a differentially private evaluation algorithm and its mean on independently drawn samples, namely between $\mathcal{E}_S[M(S)] \doteq \frac{1}{n}\sum_{i\in[n]}\mathbf{E}_M[M(S,z_i)]$ and $\mathcal{E}_{S'}[M(S)] =$

$\frac{1}{n}\sum_{i\in[n]}\mathbf{E}_M[M(S,z'_i)]$. Clearly,

$$\mathop{\mathbf{E}}_{S'\sim\mathcal{P}^n}[\mathcal{E}_{S'}[M(S)]] = \mathop{\mathbf{E}}_{z\sim\mathcal{P},\ M}[M(S,z)] = \mathcal{E}_{\mathcal{P}}[M(S)].$$

Moreover, standard concentration inequalities implies that $\mathcal{E}_{S'}[M(S)]$ is strongly concentrated around $\mathcal{E}_{\mathcal{P}}[M(S)]$. Therefore our bounds on $\mathcal{E}_{S'}[M(S)]$ readily imply bounds on $\mathcal{E}_{\mathcal{P}}[M(S)]$ while being easier to state. Note that for $M(S,(x,y)) = \ell(M'(S,x),y)$, $\mathcal{E}_S[M(S)] = \mathcal{E}_S[\ell(M'(S))]$ and $\mathcal{E}_{\mathcal{P}}[M(S)] = \mathcal{E}_{\mathcal{P}}[\ell(M'(S))]$, in other words these are exactly the empirical and the expected loss of the predictor given by M'. We give the following bound on the k-th moment of $\mathcal{E}_{S'}[M(S)]$.

Lemma 6.4. *Let $M : Z^n \times Z \to [0,B]$ be an (ϵ,δ)-differentially private evaluation algorithm and $\mathcal{P}$ be an arbitrary distribution over Z. Then:*

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

Proof. For a sequence of k indices $I=(i_1,\ldots,i_k)\in[n]^k$ let S_I denote S with every element with index in I replaced with the corresponding element of S'. Namely, for each i, if exists j such that $i=i_j$, then z_i is replaced with z'_i. Now, observe that

$$\mathop{\mathbf{E}}_{S,S'\sim\mathcal{P}^n}\left[\left(\frac{1}{n}\sum_{i\in[n]}\mathop{\mathbf{E}}_{M}[M(S,z'_i)]\right)^k\right] = \frac{1}{n^k}\sum_{I\in[n]^k}\mathop{\mathbf{E}}_{S,S'\sim\mathcal{P}^n}\left[\prod_{i\in I}\mathop{\mathbf{E}}_{M}[M(S,z'_i)]\right]. \tag{7}$$

Using group privacy (Lem. 3.1) we know that $D_\infty^{ke^{\epsilon(k-1)}\delta}(M(S_I,z'_i)\|M(S,z'_i)) \le \epsilon k$. Using Lemma 6.3 we obtain that

$$\mathop{\mathbf{E}}_{M}[M(S,z'_i)] \le e^{\epsilon k}\left(\mathop{\mathbf{E}}_{M}[M(S_I,z'_i)]+k\delta B\right).$$

Consequently,

$$\prod_{i\in I}\mathop{\mathbf{E}}_{M}[M(S,z'_i)] \le e^{k^2\cdot\epsilon}\cdot\prod_{i\in I}\left(\mathop{\mathbf{E}}_{M}[M(S_I,z'_i)]+k\delta B\right)\ .$$

Observe that for $S,S'\sim\mathcal{P}^n$, $\prod_{i\in I}(\mathbf{E}_M[M(S_I,z'_i)]+k\delta B)$ is distributed identically to $\prod_{i\in I}(\mathbf{E}_M[M(S,z_i)]+k\delta B)$. Substituting this into equation (7) we get

$$\begin{aligned}\mathop{\mathbf{E}}_{S,S'\sim\mathcal{P}^n}\left[\left(\frac{1}{n}\sum_{i\in[n]}\mathop{\mathbf{E}}_{M}[M(S,z'_i)]\right)^k\right] &\le e^{k^2\cdot\epsilon}\cdot\frac{1}{n^k}\sum_{I\in[n]^k}\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}\left[\prod_{i\in I}\left(\mathop{\mathbf{E}}_{M}[M(S,z_i)]+k\delta B\right)\right]\\ &= e^{k^2\cdot\epsilon}\cdot\mathop{\mathbf{E}}_{S\sim\mathcal{P}^n}\left[\left(\frac{1}{n}\sum_{i\in[n]}\mathop{\mathbf{E}}_{M}[M(S,z_i)]+k\delta B\right)^k\right].\end{aligned}$$

□

We now give a simple example of how to obtain high probability generalization bounds from Lemma 6.4. For simplicity we consider only the case where $\mathcal{E}_S[M(S)]$ is upper bounded by a fixed value α and $\delta = 0$ (such as in our application for realizable learning in Theorem 5.3). It can be extended relatively easily to the case then such bound holds with sufficiently high probability and $\delta > 0$.

Lemma 6.5. *Let $M : Z^n \times Z \to \mathbb{R}^+$ be an ϵ-differentially private evaluation algorithm and $\mathcal{P}$ be an arbitrary distribution over Z. Assume that for every S, $\mathcal{E}_S[M(S)] \le \alpha$. Then for every $\beta \in (0,1)$,*

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

Proof. By Markov's inequality and Lemma 6.4, for every $t \ge 1$

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{S,S' \sim \mathcal{P}^n}\left[\mathcal{E}_{S'}[M(S)] \ge t \cdot \alpha \cdot e^{k\epsilon}\right] &= \mathop{\mathbf{Pr}}_{S,S' \sim \mathcal{P}^n}\left[(\mathcal{E}_{S'}[M(S)])^k \ge \left(t \cdot \alpha \cdot e^{k\epsilon}\right)^k\right] \\
&\le \frac{\mathbf{E}_{S,S' \sim \mathcal{P}^n}\left[(\mathcal{E}_{S'}[M(S)])^k\right]}{(t \cdot \alpha \cdot e^{k\epsilon})^k} \\
&\le \frac{e^{k^2 \cdot \epsilon} \cdot \mathbf{E}_{S \sim \mathcal{P}^n}\left[(\mathcal{E}_{S}[M(S)])^k\right]}{(t \cdot \alpha \cdot e^{k\epsilon})^k} \\
&\le \frac{e^{k^2 \cdot \epsilon} \cdot \alpha^k}{(t \cdot \alpha \cdot e^{k\epsilon})^k} \le t^{-k}.
\end{aligned}$$

Setting $k = \sqrt{\ln(1/\beta)/\epsilon}$ and $t = \beta^{-1/k}$, we obtain that $t^{-k} = \beta$ and

$$t \cdot \alpha \cdot e^{k\epsilon} = \alpha \cdot e^{\ln(1/\beta)/\sqrt{\ln(1/\beta)/\epsilon}} \cdot e^{\epsilon \cdot \sqrt{\ln(1/\beta)/\epsilon}} = \alpha \cdot e^{2\sqrt{\epsilon \ln(1/\beta)}}.$$

□

7 Discussion

Several recent works point out risks to the privacy of personal data used to train a predictive model even when the attacker is given only black-box access to the model [SSSS17; CLKES18; Lon+18]. In a number of application such access is provided via a prediction interface. While the risks can be mitigated by training the model in a differentially private way [CLKES18], known theoretical and practical results show that this may substantially reduce the accuracy of the model (*e.g.* [BST14]). In this work we formulated and examined an alternative approach that only aims to ensure that the predictions themselves are differentially private. Further, we focused on understanding of making a single prediction with differential privacy (on an arbitrarily chosen point).

As we have demonstrated, simple privacy-preserving aggregation of labels created by non-private models allows to avoid some of the overheads of training the model differentially privately. Most notably, it removes the dependence on the dimension of the data for some classification and regression problems. Further, we show that algorithms satisfying uniform prediction stability can be used to reduce the overheads of aggregation. Yet, it appears that for many problems, privacy-preserving aggregation leads to a suboptimal algorithm. We therefore ask which other algorithmic approaches can be used to address the problem. Our algorithm for learning thresholds gives an example of an approach that improves on the aggregation-based learning. Finding a more general approach to agnostic learning with prediction privacy that achieves optimal sample complexity is a natural open problem.

Differentially private prediction is also a natural notion of stability. We demonstrate that it leads to (relatively) strong generalization guarantees and exploit these results to analyze our algorithm for learning thresholds. Still, the guarantees we prove are not as strong as those proved for models trained with differentially privacy. An interesting open problem is whether our results can be improved. Specifically, whether the factor $e^{\sqrt{\epsilon}}$ in Lemma 6.5 can be improved to $e^{O(\epsilon)}$.

A learning algorithm that ensures differential privacy of a single prediction would be suitable for applications in which each user asks few queries and it can be assumed that users do not share their predictions. Finding approaches for dealing with multiple prediction queries (that go beyond the composition properties of differential privacy) is an important direction for further research (in the context of privacy-preserving aggregation this question has been considered in several work described in Section 2.4).

References

[BE02] O. Bousquet and A. Elisseeff. “Stability and generalization”. In: *JMLR* 2 (2002), pp. 499–526.

[BKN10] A. Beimel, S. P. Kasiviswanathan, and K. Nissim. “Bounds on the Sample Complexity for Private Learning and Private Data Release”. In: *TCC*. 2010, pp. 437–454.

[BNS13] A. Beimel, K. Nissim, and U. Stemmer. “Characterizing the sample complexity of private learners”. In: *ITCS*. 2013, pp. 97–110.

[BNSSSU16] R. Bassily, K. Nissim, A. D. Smith, T. Steinke, U. Stemmer, and J. Ullman. “Algorithmic stability for adaptive data analysis”. In: *STOC*. 2016, pp. 1046–1059.

[BNSV15] M. Bun, K. Nissim, U. Stemmer, and S. P. Vadhan. “Differentially Private Release and Learning of Threshold Functions”. In: *FOCS*. 2015, pp. 634–649.

[BS16] M. Bun and T. Steinke. “Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds”. In: *CoRR* abs/1605.02065 (2016). arXiv: `1605.02065`.

[BST14] R. Bassily, A. Smith, and A. Thakurta. “Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds”. In: *FOCS*. 2014, pp. 464–473.

[BTT18] R. Bassily, O. Thakkar, and A. Thakurta. “Model-Agnostic Private Learning via Stability”. In: *CoRR* abs/1803.05101 (2018). arXiv: `1803.05101`.

[CH11] K. Chaudhuri and D. Hsu. “Sample Complexity Bounds for Differentially Private Learning”. In: *COLT*. 2011, pp. 155–186.

[CLKES18] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song. “The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets”. In: *CoRR* abs/1802.08232 (2018).

[CMS11] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. “Differentially Private Empirical Risk Minimization”. In: *Journal of Machine Learning Research* 12 (2011), pp. 1069–1109.

[DFHPRR14] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. “Preserving Statistical Validity in Adaptive Data Analysis”. In: *CoRR* abs/1411.2664 (2014). Extended abstract in STOC 2015.

[DL09] C. Dwork and J. Lei. “Differential Privacy and Robust Statistics”. In: *STOC*. 2009, pp. 371–380.

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. “Calibrating noise to sensitivity in private data analysis”. In: *TCC*. 2006, pp. 265–284.

[DR14] C. Dwork and A. Roth. *The Algorithmic Foundations of Differential Privacy*. Vol. 9. 3-4. 2014, pp. 211–407.

[DS09] C. Dwork and A. Smith. “Differential Privacy for Statistics: What we Know and What we Want to Learn”. In: *Journal of Privacy and Confidentiality* 1(2) (2009), pp. 135–154.

[FS17] V. Feldman and T. Steinke. “Generalization for Adaptively-chosen Estimators via Stable Median”. In: *COLT*. 2017, pp. 728–757.

[FX15] V. Feldman and D. Xiao. “Sample Complexity Bounds on Differentially Private Learning via Communication Complexity”. In: *SIAM J. Comput.* 44.6 (2015), pp. 1740–1764.

[Hau92] D. Haussler. “Decision theoretic generalizations of the PAC model for neural net and other learning applications”. In: *Information and Computation* 100.1 (1992), pp. 78–150. ISSN: 0890-5401.

[HCB16] J. Hamm, Y. Cao, and M. Belkin. “Learning privately from multiparty data”. In: *ICML*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. 2016, pp. 555–563.

[HRS16] M. Hardt, B. Recht, and Y. Singer. “Train faster, generalize better: Stability of stochastic gradient descent”. In: *ICML*. 2016, pp. 1225–1234.

[KLNRS11] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. “What Can We Learn Privately?” In: *SIAM J. Comput.* 40.3 (June 2011), pp. 793–826.

[KSS94] M. Kearns, R. Schapire, and L. Sellie. “Toward Efficient Agnostic Learning.” In: *Machine Learning* 17.2-3 (1994), pp. 115–141.

[KST12] D. Kifer, A. D. Smith, and A. Thakurta. “Private Convex Optimization for Empirical Risk Minimization with Applications to High-dimensional Regression”. In: *COLT*. 2012, pp. 25.1–25.40.

[KV94] M. Kearns and U. Vazirani. *An introduction to computational learning theory*. Cambridge, MA: MIT Press, 1994.

[Lon+18] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen. “Understanding Membership Inferences on Well-Generalized Learning Models”. In: *CoRR* abs/1802.04889 (2018). arXiv: 1802.04889.

[Mir] I. Mironov. “Rényi Differential Privacy”. In: *Computer Security Foundations Symposium, CSF, pages = 263–275, year = 2017.*

[MT07] F. McSherry and K. Talwar. “Mechanism Design via Differential Privacy”. In: *FOCS*. 2007, pp. 94–103.

[NRS07] K. Nissim, S. Raskhodnikova, and A. D. Smith. “Smooth sensitivity and sampling in private data analysis”. In: *STOC*. 2007, pp. 75–84.

[PAEGT17] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar. “Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data”. In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. 2017.

[PRR10] M. A. Pathak, S. Rane, and B. Raj. “Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers”. In: *NIPS*. 2010, pp. 1876–1884.

[PSMRTE18] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson. “Scalable Private Learning with PATE”. In: *International Conference on Learning Representations* (2018).

[SC13] A. D. Sarwate and K. Chaudhuri. “Signal Processing and Machine Learning with Differential Privacy: Algorithms and Challenges for Continuous Data”. In: *IEEE Signal Process. Mag.* 30.5 (2013), pp. 86–94.

[SSBD14] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[SSSS17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. “Membership Inference Attacks Against Machine Learning Models”. In: *2017 IEEE Symposium on Security and Privacy, SP 2017*. 2017, pp. 3–18.

[SSSSS10] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. “Learnability, stability and uniform convergence”. In: *The Journal of Machine Learning Research* 11 (2010), pp. 2635–2670.

[ST13] A. Smith and A. G. Thakurta. “Differentially Private Feature Selection via Stability Arguments, and the Robustness of the Lasso”. In: *COLT*. 2013.

[TTZ15] K. Talwar, A. Thakurta, and L. Zhang. “Nearly Optimal Private LASSO”. In: *NIPS*. 2015, pp. 3025–3033.

[Val84] L. G. Valiant. “A theory of the learnable”. In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142.

[WLKCJN17] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. F. Naughton. “Bolt-on Differential Privacy for Scalable Stochastic Gradient Descent-based Analytics”. In: *SIGMOD*. 2017, pp. 1307–1322.

[WYX17] D. Wang, M. Ye, and J. Xu. “Differentially Private Empirical Risk Minimization Revisited: Faster and More General”. In: *NIPS*. 2017, pp. 2719–2728.