An Equivalence Between Private Classification and Online Prediction

Mark Bun
Department of Computer Science
Boston University
Boston, MA
mbun@bu.edu

Roi Livni
Department of Electrical Engineering
Tel Aviv University
Tel Aviv, Israel
rlivni@tauex.tau.ac.il

Shay Moran
Department of Mathematics
Technion
Haifa, Israel
and
Google Brain
shaymoran1@gmail.com

***Abstract*—We prove that every concept class with finite Littlestone dimension can be learned by an (approximate) differentially-private algorithm. This answers an open question of Alon et al. (STOC 2019) who proved the converse statement (this question was also asked by Neel et al. (FOCS 2019)). Together these two results yield an equivalence between online learnability and private PAC learnability.**

We introduce a new notion of algorithmic stability called "global stability" which is essential to our proof and may be of independent interest. We also discuss an application of our results to boosting the privacy and accuracy parameters of differentially-private learners.

***Keywords*-differential privacy; learning; online learning**

I. Introduction

This paper continues the study of the close relationship between differentially-private learning and online learning.

Differentially-Private Learning: Statistical analyses and computer algorithms play significant roles in the decisions which shape modern society. The collection and analysis of individuals' data drives computer programs which determine many critical outcomes, including the allocation of community resources, decisions to give loans, and school admissions.

While data-driven and automated approaches have obvious benefits in terms of efficiency, they also raise the possibility of unintended negative impacts, especially against marginalized groups. This possibility highlights the need for *responsible* algorithms that obey relevant ethical requirements (see e.g. [O'N16]).

Differential Privacy (DP) [DMNS06] plays a key role in this context. Its initial (and primary) purpose was to provide a formal framework for ensuring individuals' privacy in the statistical analysis of large datasets. But it has also found use in addressing other ethical issues such as *algorithmic fairness* (see, e.g. [DHP$^+$12], [CGKM19]).

Many tasks which involve sensitive data arise in machine learning (e.g. in medical applications and in social networks). Consequently, a large body of practical and theoretical work has been dedicated to understand which learning tasks can be performed by DP learning algorithms. The simplest and most extensively studied model of learning is the private PAC model [Val84], [KLN$^+$11], which captures binary classification tasks under differential privacy. A partial list of works on this topic includes [KLN$^+$11], [BBKN14], [BNSV15], [FX15], [BNS16a], [BDRS18], [BNS19], [ALMM19], [KLM$^+$19]. In this manuscript we make progress towards characterizing what tasks are DP PAC-learnable by demonstrating a qualitative equivalence with online-learnable tasks.

Online Learning: Online learning is a well-studied branch of machine learning which addresses algorithms making real-time predictions on sequentially arriving data. Such tasks arise in contexts including recommendation systems and advertisement placement. The literature on this subject is vast and includes several books, e.g. [CL06], [SS12], [Haz16].

Online Prediction, or *Prediction with Expert Advice* is a basic setting within online learning. Let $\mathcal{H} = \{h : X \to \{\pm 1\}\}$ be a class of predictors (also called experts) over a domain X. Consider an algorithm which observes examples $(x_1, y_1) \dots (x_T, y_T) \in X \times \{\pm 1\}$ in a sequential manner. In each time step t, the algorithm first observes the instance x_t, then predicts a label $\hat{y}_t \in \{\pm 1\}$, and finally learns whether its prediction was correct. The goal is to minimize the *regret*, the number of mistakes compared to the best expert in $\mathcal{H}$:

$$\sum_{t=1}^{T} 1[y_t \neq \hat{y}_t] - \min_{h^* \in \mathcal{H}} \sum_{t=1}^{T} 1[y_t \neq h^*(x_t)].$$

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00044

In this context, a class $\mathcal{H}$ is said to be online learnable if for every T, there is an algorithm that achieves sublinear regret $o(T)$ against any sequence of T examples. The *Littlestone dimension* is a combinatorial parameter associated to the class $\mathcal{H}$ which characterizes its online learnability [Lit87], [BPS09]: $\mathcal{H}$ is online learnable if and only if it has a finite Littlestone dimension $d < \infty$. Moreover, the best possible regret $R(T)$ for online learning of $\mathcal{H}$ satisfies

$$\Omega(\sqrt{dT}) \leq R(T) \leq O(\sqrt{dT \log T}).$$

Furthermore, if it is known that if one of the experts never errs (i.e. in the realizable mistake-bound model), then the optimal regret is exactly d.

Stability: While at a first glance it may seem that online learning and differentially-private learning have little to do with one another, a line of recent works has revealed a tight connection between the two [AS17], [ALMT17], [ALMM19], [BLM19], [NRW19], [JMNR19], [GHM19]. At a high-level, this connection appears to boil down to the notion of stability, which plays a key role in both topics. On one hand, the definition of differential privacy is itself a form of stability; it requires robustness of the output distribution of an algorithm when its input undergoes small changes. On the other hand, stability also arises as a central motif in online learning paradigms such as *Follow the Perturbed Leader* [KV02], [KV05] and *Follow the Regularized Leader* [AHR08], [SSS07], [Haz16].

In their monograph [DR14], Dwork and Roth identified stability as a common factor of learning and differential privacy: *"Differential privacy is enabled by stability and ensures stability... we observe a tantalizing moral equivalence between learnability, differential privacy, and stability."* This insight has found formal manifestations in several works. For example, Abernethy et al. used DP inspired stability methodology to derive a unified framework for proving state of the art bounds in online learning [ALMT17]. In the opposite direction, Agarwal and Singh showed that certain standard stabilization techniques in online learning imply differential privacy [AS17].

Stability plays a key role in this work as well. Our main result, which shows that any class with a finite Littlestone dimension can be privately learned, hinges on the following form of stability: for $\eta > 0$ and $n \in \mathbb{N}$, a learning algorithm $\mathcal{A}$ is (n, η)-*globally stable*[1] with respect to a distribution $\mathcal{D}$ over examples if there exists an hypothesis h whose frequency as an output is at least η. Namely,

$$\Pr_{S \sim \mathcal{D}^n}[\mathcal{A}(S) = h] \geq \eta.$$

We show that every $\mathcal{H}$ can be learned by a globally-stable algorithm with parameters $\eta = \exp(-d), n = \exp(d)$, where d is the Littlestone dimension of $\mathcal{H}$. As a corollary, we get an equivalence between global stability and differential privacy (which can be viewed as a form of local stability). That is, the existence of a globally-stable learner for $\mathcal{H}$ is equivalent to the existence of a differentially-private learner (and both are equivalent to having a finite Littlestone dimension).

Littlestone Classes: It is natural to ask which classes have finite Littlestone dimension. First, note that every finite class $\mathcal{H}$ has Littlestone dimension $d \leq \log|\mathcal{H}|$. There are also many natural and interesting infinite classes with finite Littlestone dimension. For example, let $X = \mathbb{F}^n$ be an n-dimensional vector space over a field $\mathbb{F}$ and let $\mathcal{H} \subseteq \{\pm 1\}^X$ consist of all (indicators of) affine subspaces of dimension $\leq d$. The Littlestone dimension of $\mathcal{H}$ is d. More generally, any class of hypotheses that can be described by polynomial *equalities* of constant degree has finite Littlestone dimension.[2] This can be generalized even further to classes that are definable in *stable theories*. This (different, still) notion of stability is deep and well-explored in model theory. We refer the reader to [CF19], Section 5.1 for more examples of stable theories and the Littlestone classes they correspond to.

Organization: The rest of this manuscript is organized as follows. In Section I-A we formally state our main results and discuss some implications. Section II overviews some of the main ideas in the proofs. Sections III - VI contain complete proofs. In the full version we conclude the paper with some suggestions for future work.

A. Main Results

We next present our main results. We begin with the statements concerning the relationship between on-

[1] The word *global* highlights a difference from other forms of algorithmic stability. Previous forms of stability such as DP and *uniform hypothesis stability* [BE02] are local in that they require output robustness to *local* changes in the input. However, the property required by global stability captures stability with respect to resampling the entire input.

[2] Note that if one replaces "equalities" with "inequalities" then the Littlestone dimension may become unbounded while the VC dimension remains bounded. This is demonstrated, e.g., by halfspaces which are captured by polynomial inequalities of degree 1.

line learning and differentially-private learning. In Section I-A1 we present and discuss the notion of global stability, and finally in Section I-A2 we discuss an implication for private boosting. Throughout this section some standard technical terms are used. For definitions of these terms we refer the reader to Section III.

Theorem 1 (Littlestone Classes are Privately Learnable)**.** *Let $\mathcal{H} \subseteq \{\pm 1\}^X$ be a class with Littlestone dimension d, let $\varepsilon, \delta \in (0,1)$ be privacy parameters, and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For*

$$n = O\left(\frac{16^d \cdot d^2 \cdot (d + \log(1/\beta\delta))}{\alpha\varepsilon}\right) = O_d\left(\frac{\log(1/\beta\delta)}{\alpha\varepsilon}\right)$$

there exists an (ε, δ)-DP learning algorithm such that for every realizable distribution $\mathcal{D}$, given an input sample $S \sim \mathcal{D}^n$, the output hypothesis $f = \mathcal{A}(S)$ satisfies $\text{loss}_{\mathcal{D}}(f) \leq \alpha$ with probability at least $1 - \beta$, where the probability is taken over $S \sim \mathcal{D}^n$ as well as the internal randomness of $\mathcal{A}$.

A similar result holds in the agnostic setting:

Corollary 2 (Agnostic Learner for Littlestone Classes)**.** *Let $\mathcal{H} \subseteq \{\pm 1\}^X$ be a class with Littlestone dimension d, let ε, and $\delta \in (0,1)$ be privacy parameters, and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For*

$$n = O\left(\frac{16^d \cdot d^2 \cdot (d + \log(1/\beta\delta))}{\alpha\epsilon} + \frac{VC(\mathcal{H}) + \log 1/\beta}{\alpha^2\epsilon}\right)$$

there exists an (ε, δ)-DP learning algorithm such that for every distribution $\mathcal{D}$, given an input sample $S \sim \mathcal{D}^n$, the output hypothesis $f = \mathcal{A}(S)$ satisfies

$$\text{loss}_{\mathcal{D}}(f) \leq \min_{h \in \mathcal{H}} \text{loss}_{\mathcal{D}}(h) + \alpha$$

with probability at least $1 - \beta$, where the probability is taken over $S \sim \mathcal{D}^n$ as well as the internal randomness of $\mathcal{A}$.

Corollary 2 follows from Theorem 1 by Theorem 2.3 in [ABMS20] which provides a general mechanism to transform a learner in the realizable setting to a learner in the agnostic setting[3]. We note that formally the transformation in [ABMS20] is stated for a constant $\varepsilon = O(1)$. Taking $\varepsilon = O(1)$ is without loss of generality as a standard "secrecy-of-the-sample" argument can be used to convert this learner into one which is (ε, δ)-differentially private by increasing the sample size by a factor of roughly $1/\varepsilon$ and running the algorithm on a random subsample. See [KLN+11], [Vad17] for further details.

Theorem 3 (Private PAC Learning $\equiv$ Online Prediction.)**.** *The following statements are equivalent for a class $\mathcal{H} \subseteq \{\pm 1\}^X$:*

1) *$\mathcal{H}$ is online learnable.*
2) *$\mathcal{H}$ is approximate differentially-privately PAC learnable.*

Theorem 3 is a corollary of Theorem 1 (which gives 1 → 2) and the result by Alon et al. [ALMM19] (which gives 2 → 1). We comment that a quantitative relation between the learning and regret rates is also implied: it is known that the optimal regret bound for $\mathcal{H}$ is $\tilde{\Theta}_d(\sqrt{T})$, where the $\tilde{\Theta}_d$ conceals a constant which depends on the Littlestone dimension of $\mathcal{H}$ [BPS09]. Similarly, we get that the optimal sample complexity of agnostically privately learning $\mathcal{H}$ is $\Theta_d(\frac{\log(1/(\beta\delta))}{\alpha^2\varepsilon})$.

We remark however that the above equivalence is mostly interesting from a theoretical perspective, and should not be regarded as an efficient transformation between online and private learning. Indeed, the Littlestone dimension dependencies concealed by the $\tilde{\Theta}_d(\cdot)$ in the above bounds on the regret and sample complexities may be very different from one another. For example, there are classes for which the $\Theta_d(\frac{\log(1/(\beta\delta))}{\alpha\varepsilon})$ bound hides a $\text{poly}(\log^*(d))$ dependence, and the $\tilde{\Theta}_d(\sqrt{T})$ bound hides a $\Theta(d)$ dependence. One example which attains both of these dependencies is the class of thresholds over a linearly ordered domain of size 2^d [ALMM19], [KLM+19].

1) Global Stability: Our proof of Theorem 1, which establishes that every Littlestone class can be learned privately, hinges on an intermediate property which we call *global stability*:

Definition 4 (Global Stability)**.** Let $n \in \mathbb{N}$ be a sample size and $\eta > 0$ be a global stability parameter. An algorithm $\mathcal{A}$ is (n, η)-globally-stable with respect to a distribution $\mathcal{D}$ if there exists an hypothesis h such that

$$\Pr_{S \sim \mathcal{D}^n}[A(S) = h] \geq \eta.$$

While global stability is a rather strong property, it holds automatically for learning algorithms using a finite hypothesis class. By an averaging argument, every learner using n samples which produces a hypothesis in a finite hypothesis class $\mathcal{H}$ is $(n, 1/|\mathcal{H}|)$-globally-stable. The following proposition generalizes "Occam's Razor" for finite hypothesis classes to show that global stability is enough to imply similar generalization bounds in the realizable setting.

[3] Theorem 2.3 in [ABMS20] is based on a previous realizable-to-agnostic transformation from [BNS15] which applies to *proper* learners. Here we require the more general transformation from [ABMS20] as the learner implied by Theorem 1 may be improper.

Proposition 5 (Global Stability $\implies$ Generalization)**.** *Let $\mathcal{H} \subseteq \{\pm 1\}^X$ be a class, and assume that $\mathcal{A}$ is a consistent learner for $\mathcal{H}$ (i.e. $\operatorname{loss}_S(\mathcal{A}(S)) = 0$ for every realizable sample S). Let $\mathcal{D}$ be a realizable distribution such that $\mathcal{A}$ is (n, η)-globally-stable with respect to $\mathcal{D}$, and let h be a hypothesis such that $\Pr_{S\sim\mathcal{D}^n}[A(S) = h] \geq \eta$, as guaranteed by the definition of global stability. Then,*

$$\operatorname{loss}_{\mathcal{D}}(h) \leq \frac{\ln(1/\eta)}{n}.$$

Proof: Let α denote the loss of h, i.e. $\operatorname{loss}_{\mathcal{D}}(h) = \alpha$, and let E_1 denote the event that h is consistent with the input sample S. Thus, $\Pr[E_1] = (1-\alpha)^n$. Let E_2 denote the event that $\mathcal{A}(S) = h$. By assumption, $\Pr[E_2] \geq \eta$. Now, since $\mathcal{A}$ is consistent we get that $E_2 \subseteq E_1$, and hence that $\eta \leq (1-\alpha)^n$. This finishes the proof (using the fact that $1 - \alpha \leq e^{-\alpha}$ and taking the logarithm of both sides). ■

Another way to view global stability is in the context of *pseudo-deterministic algorithms* [GG11]. A pseudo-deterministic algorithm is a randomized algorithm which yields some fixed output with high probability. Thinking of a realizable distribution $\mathcal{D}$ as an instance on which PAC learning algorithm has oracle access, a globally-stable learner is one which is "weakly" pseudo-deterministic in that it produces some fixed output with probability bounded away from zero. A different model of pseudo-deterministic learning, in the context of learning from membership queries, was defined and studied by Oliveira and Santhanam [OS18].

We prove Theorem 1 by constructing, for a given Littlestone class $\mathcal{H}$, an algorithm $\mathcal{A}$ which is globally stable with respect to every realizable distribution.

2) Boosting for Approximate Differential Privacy: Our characterization of private learnability in terms of the Littlestone dimension has new consequences for boosting the privacy and accuracy guarantees of differentially-private learners. Specifically, it shows that the existence of a learner with weak (but non-trivial) privacy and accuracy guarantees implies the existence of a learner with any desired privacy and accuracy parameters — in particular, one with $\delta(n) = \exp(-\Omega(n))$.

Theorem 6. *There exists a constant $c > 0$ for which the following holds. Suppose that for some sample size n_0 there is an $(\varepsilon_0, \delta_0)$-differentially private learner $\mathcal{W}$ for a class $\mathcal{H}$ satisfying the guarantee*

$$\Pr_{S\sim\mathcal{D}^{n_0}}[\operatorname{loss}_{\mathcal{D}}(\mathcal{W}(S)) > \alpha_0] < \beta_0$$

for $\varepsilon_0 = 0.1, \alpha_0 = \beta_0 = 1/16$, and $\delta_0 \leq c/n_0^2 \log n_0$.

Then there exists a constant $C_{\mathcal{H}}$ such that for every $\alpha, \beta, \varepsilon, \delta \in (0, 1)$ there exists an (ε, δ)-differentially private learner for $\mathcal{H}$ with

$$\Pr_{S\sim\mathcal{D}^n}[\operatorname{loss}_{\mathcal{D}}(\mathcal{A}(S)) > \alpha] < \beta$$

whenever $n \geq C_{\mathcal{H}} \cdot \log(1/\beta\delta)/\alpha\varepsilon$.

Given a weak learner $\mathcal{W}$ as in the statement of Theorem 6, the results of [ALMM19] imply that $\operatorname{Ldim}(\mathcal{H})$ is finite. Hence Theorem 1 allows us to construct a learner for $\mathcal{H}$ with arbitrarily small privacy and accuracy, yielding Theorem 6. The constant $C_{\mathcal{H}}$ in the last line of the theorem statement suppresses a factor depending on $\operatorname{Ldim}(\mathcal{H})$.

Prior to our work, it was open whether arbitrary learning algorithms satisfying approximate differential privacy could be boosted in this strong a manner. We remark, however, that in the case of *pure* differential privacy, such boosting can be done algorithmically and efficiently. Specifically, given an $(\varepsilon_0, 0)$-differentially private weak learner as in the statement of Theorem 6, one can first apply random sampling to improve the privacy guarantee to $(p\varepsilon_0, 0)$-differential privacy at the expense of increasing its sample complexity to roughly n_0/p for any $p \in (0, 1)$. The Boosting-for-People construction of Dwork, Rothblum, and Vadhan [DRV10] (see also [BCS20]) then produces a strong learner by making roughly $T \approx \log(1/\beta)/\alpha^2$ calls to the weak learner. By composition of differential privacy, this gives an $(\varepsilon, 0)$-differentially private strong learner with sample complexity roughly $n_0 \cdot \log(1/\beta)/\alpha^2\varepsilon$.

What goes wrong if we try to apply this argument using an $(\varepsilon_0, \delta_0)$-differentially private weak learner? Random sampling still gives a $(p\varepsilon_0, p\delta_0)$-differentially private weak learner with sample complexity n_0/p. However, this is not sufficient to improve the δ parameter of the learner *as a function of the number of samples n*. Thus the strong learner one obtains using Boosting-for-People still at best guarantees $\delta(n) = \tilde{O}(1/n^2)$. Meanwhile, Theorem 6 shows that the existence of a $(0.1, \tilde{O}(1/n^2))$-differentially private learner for a given class implies the existence of a $(0.1, \exp(-\Omega(n))$-differentially private learner for that class.

We leave it as an interesting open question to determine whether this kind of boosting for approximate differential privacy can be done algorithmically.

II. Proof Overview

We next give an overview of the main arguments used in the proof of Theorem 1. The proof consist of two parts: (i) we first show that every class with a finite Littlestone dimension can be learned by a globally-stable algorithm, and (ii) we then show how to generically obtain a differentially-private learner from any globally-stable learner.

A. Step 1: Finite Littlestone Dimension $\Longrightarrow$ Globally-Stable Learning

Let $\mathcal{H}$ be a concept class with Littlestone dimension d. Our goal is to design a globally-stable learning algorithm for $\mathcal{H}$ with stability parameter $\eta = \exp(-d)$ and sample complexity $n = \exp(d)$. We will sketch here a weaker variant of our construction which uses the same ideas but is simpler to describe.

The property of $\mathcal{H}$ that we will use is that it can be online learned in the realizable setting with at most d mistakes (see Section III-B for a brief overview of this setting). Let $\mathcal{D}$ denote a realizable distribution with respect to which we wish to learn in a globally-stable manner. That is, $\mathcal{D}$ is a distribution over examples $(x, c(x))$ where $c \in \mathcal{H}$ is an unknown target concept. Let $\mathcal{A}$ be a learning algorithm that makes at most d mistakes while learning an unknown concept from $\mathcal{H}$ in the online model. Consider applying $\mathcal{A}$ on a sequence $S = ((x_1, c(x_1)) \dots (x_n, c(x_n))) \sim \mathcal{D}^n$, and denote by M the random variable counting the number of mistakes $\mathcal{A}$ makes in this process. The mistake-bound guarantee on $\mathcal{A}$ guarantees that $M \le d$ always. Consequently, there is $0 \le i \le d$ such that

$$\Pr[M = i] \ge \frac{1}{d+1}.$$

Note that we can identify, with high probability, an i such that $\Pr[M = i] \ge 1/2d$ by running $\mathcal{A}$ on $O(d)$ samples from $\mathcal{D}^n$. We next describe how to handle each of the $d+1$ possibilities for i. Let us first assume that $i = d$, namely that

$$\Pr[M = d] \ge \frac{1}{2d}.$$

We claim that in this case we are done: indeed, after making d mistakes it must be the case that $\mathcal{A}$ has completely identified the target concept c (or else $\mathcal{A}$ could be presented with another example which forces it to make $d+1$ mistakes). Thus, in this case it holds with probability at least $1/2d$ that $\mathcal{A}(S) = c$ and we are done. Let us next assume that $i = d - 1$, namely that

$$\Pr[M = d - 1] \ge \frac{1}{2d}.$$

The issue with applying the previous argument here is that before making the d'th mistake, $\mathcal{A}$ can output many different hypotheses (depending on the input sample S). We use the following idea: draw two samples $S_1, S_2 \sim \mathcal{D}^n$ independently, and set $f_1 = \mathcal{A}(S_1)$ and $f_2 = \mathcal{A}(S_2)$. Condition on the event that the number of mistakes made by $\mathcal{A}$ on each of S_1, S_2 is exactly $d - 1$ (by assumption, this event occurs with probability at least $(1/2d)^2$) and consider the following two possibilities:

(i) $\Pr[f_1 = f_2] \ge \frac{1}{4}$,
(ii) $\Pr[f_1 = f_2] < \frac{1}{4}$.

If (i) holds then using a simple calculation one can show that there is h such that $\Pr[A(S) = h] \ge \frac{1}{(2d)^2} \cdot \frac{1}{4}$ and we are done. If (ii) holds then we apply the following *"random contest"* between S_1, S_2:

1) Pick x such that $f_1(x) \ne f_2(x)$ and draw $y \sim \{\pm 1\}$ uniformly at random.
2) If $f_1(x) \ne y$ then the output is $\mathcal{A}(S_1 \circ (x, y))$, where $S_1 \circ (x, y)$ denotes the sample obtained by appending (x, y) to the end of S. In this case we say that S_1 "won the contest".
3) Else, $f_2(x) \ne y$ then the output is $\mathcal{A}(S_2 \circ (x, y))$. In this case we that S_2 "won the contest".

Note that adding the auxiliary example (x, y) forces $\mathcal{A}$ to make exactly d mistakes on $S_i \circ (x, y)$. Now, if $y \sim \{\pm 1\}$ satisfies $y = c(x)$ then by the mistake-bound argument it holds that $\mathcal{A}(S_i \circ (x, y)) = c$. Therefore, since $\Pr_{y \sim \{\pm 1\}}[c(x) = y] = 1/2$, it follows that

$$\Pr_{S_1, S_2, y}[\mathcal{A}(S_i \circ (x, y)) = c] \ge \frac{1}{(2d)^2} \cdot \frac{3}{4} \cdot \frac{1}{2} = \Omega(1/d^2),$$

and we are done.

Similar reasoning can be used by induction to handle the remaining cases (the next one would be that $\Pr[M = d - 2] \ge \frac{1}{2d}$, and so on). The proof we present in Section IV is based on a similar idea of performing "random contests," although the construction becomes more complex to handle other issues, such as generalization, which were not addressed here. For more details we refer the reader to the complete argument in Section IV.

B. Step 2: Globally-Stable Learning $\Longrightarrow$ Differentially-Private Learning

Given a globally-stable learner $\mathcal{A}$ for a concept class $\mathcal{H}$, we can obtain a differentially-private learner using standard techniques in the literature on private learning and query release. If $\mathcal{A}$ is a (η, m)-globally stable learner with respect to a distribution $\mathcal{D}$, we obtain a differentially-private learner using roughly m/η samples from that distribution as follows. We first run $\mathcal{A}$ on $k \approx 1/\eta$ independent samples, non-privately producing a list of k hypotheses. We then apply a differentially-private "Stable Histograms" algorithm [KKMN09], [BNS16b] to this list which allows us to privately publish a short list of hypotheses that appear with frequency $\Omega(\eta)$. Global stability of the learner $\mathcal{A}$ guarantees that with high probability, this list contains *some* hypothesis h with small population loss. We can then apply a generic differentially-private learner (based on the exponential mechanism) on a fresh set of examples to identify such an accurate hypothesis from the short list.

III. Preliminaries

A. PAC Learning

We use standard notation from statistical learning; see, e.g., [SSBD14]. Let X be any "domain" set and consider the "label" set $Y = \{\pm 1\}$. A *hypothesis* is a function $h : X \to Y$, which we alternatively write as an element of Y^X. An *example* is a pair $(x, y) \in X \times Y$. A *sample* S is a finite sequence of examples.

Definition 7 (Population & Empirical Loss)**.** Let $\mathcal{D}$ be a distribution over $X \times \{\pm 1\}$. The population loss of a hypothesis $h : X \to \{\pm 1\}$ is defined by

$$\mathrm{loss}_{\mathcal{D}}(h) = \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y].$$

Let $S = \big((x_i, y_i)\big)_{i=1}^n$ be a sample. The empirical loss of h with respect to S is defined by

$$\mathrm{loss}_S(h) = \frac{1}{n}\sum_{i=1}^{n} 1[h(x_i) \neq y_i].$$

Let $\mathcal{H} \subseteq Y^X$ be a *hypothesis class*. A sample S is said to be *realizable by* $\mathcal{H}$ if there is $h \in H$ such that $\mathrm{loss}_S(h) = 0$. A distribution $\mathcal{D}$ is said to be *realizable by* $\mathcal{H}$ if there is $h \in H$ such that $\mathrm{loss}_{\mathcal{D}}(h) = 0$. A *learning algorithm* A is a (possibly randomized) mapping taking input samples to output hypotheses. We also use the following notation: for samples S, T, let $S \circ T$ denote the combined sample obtained by appending T to the end of S.

B. Online Learning

Littlestone Dimension: The Littlestone dimension is a combinatorial parameter that captures mistake and regret bounds in online learning [Lit87], [BPS09].[4] Its definition uses the notion of *mistake trees*. A mistake tree is a binary decision tree whose internal nodes are labeled by elements of X. Any root-to-leaf path in a mistake tree can be described as a sequence of examples $(x_1, y_1), ..., (x_d, y_d)$, where x_i is the label of the i'th internal node in the path, and $y_i = +1$ if the $(i+1)$'th node in the path is the right child of the i'th node and $y_i = -1$ otherwise. We say that a mistake tree T is *shattered* by $\mathcal{H}$ if for any root-to-leaf path $(x_1, y_1), ..., (x_d, y_d)$ in T there is an $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for all $i \le d$ (see Figure 1). The Littlestone dimension of $\mathcal{H}$, denoted $\mathrm{Ldim}(\mathcal{H})$, is the depth of largest complete tree that is shattered by $\mathcal{H}$. We say that $\mathcal{H}$ is a Littlestone class if it has finite Littlestone dimension.

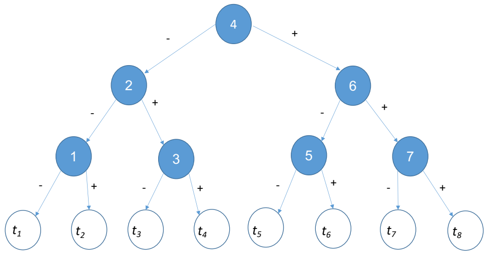

Figure 1: A tree shattered by the class $\mathcal{H} \subseteq \{\pm 1\}^8$ that contains the threshold functions t_i, where $t_i(j) = +1$ if and only if $i \le j$.

*Mistake Bound and the Standard Optimal Algorithm (*SOA*):* The simplest setting in which learnability is captured by the Littlestone dimension is called the *mistake-bound model* [Lit87]. Let $\mathcal{H} \subseteq \{\pm 1\}^X$ be a fixed hypothesis class known to the learner. The learning process takes place in a sequence of trials, where the order of events in each trial t is as follows:

(i) the learner receives an instance $x_t \in X$,
(ii) the learner responses with a prediction $\hat{y}_t \in \{\pm 1\}$, and
(iii) the learner is told whether or not the response was correct.

We assume that the examples given to the learner are realizable in the following sense: For the entire sequence of trials, there is a hypothesis $h \in \mathcal{H}$ such that $y_t = h(x_t)$ for every instance x_t and correct response y_t. An algorithm in this model learns $\mathcal{H}$ with mistake bound M if for every realizable sequence of examples presented to the learner, it makes a total of at most M incorrect predictions.

Littlestone showed that the minimum mistake bound achievable by any online learner is exactly $\mathrm{Ldim}(\mathcal{H})$ [Lit87]. Furthermore, he described an explicit algorithm, called the *Standard Optimal Algorithm* (SOA), which achieves this optimal mistake bound.

[4] It appears that the name "Littlestone dimension" was coined in [BPS09].

```
Standard Optimal Algorithm (SOA)
1) Initialize $\mathcal{H}_1 = \mathcal{H}$.
2) For trials $t = 1, 2, \ldots$:
    (i) For each $b \in \{\pm 1\}$ and $x \in X$, let
        $\mathcal{H}_t^b(x) = \{h \in \mathcal{H}_t : h(x) = b\}$.
        Define $h : X \to \{\pm 1\}$ by $h_t(x) = \mathrm{argmax}_b \, \mathrm{Ldim}(\mathcal{H}_t^b(x))$.
    (ii) Receive instance $x_t$.
    (iii) Predict $\hat{y}_t = h_t(x_t)$.
    (iv) Receive correct response $y_t$.
    (v) Update $\mathcal{H}_{t+1} = \mathcal{H}_t^{y_t}(x_t)$.
```

Extending the SOA *to non-realizable sequences:* Our globally-stable learner for Littlestone classes will make use of an optimal online learner in the mistake bound model. For concreteness, we pick the SOA (any other optimal algorithm will also work). It will be convenient to extend the SOA to sequences which are not necessarily realizable by a hypothesis in $\mathcal{H}$. We will use the following simple extension of the SOA to non-realizable samples:

Definition 8 (Extending the SOA to non-realizable sequences)**.** Consider a run of the SOA on examples $(x_1, y_1), \ldots, (x_m, y_m)$, and let h_t denote the predictor used by the SOA after seeing the first t examples (i.e., h_t is the rule used by the SOA to predict in the $(t+1)$'st trial). Then, after observing both x_{t+1}, y_{t+1} do the following:

- If the sequence $(x_1, y_1), \ldots, (x_{t+1}, y_{t+1})$ is realizable by some $h \in \mathcal{H}$ then apply the usual update rule of the SOA to obtain h_{t+1}.
- Else, set h_{t+1} as follows: $h_{t+1}(x_{t+1}) = y_{t+1}$, and $h_{t+1}(x) = h_t(x)$ for every $x \neq x_{t+1}$.

Thus, upon observing a non-realizable sequence, this update rule locally updates the maintained predictor h_t to agree with the last example.

C. Differential Privacy

We use standard definitions and notation from the differential privacy literature. For more background see, e.g., the surveys [DR14], [Vad17]. For $a, b, \varepsilon, \delta \in [0, 1]$ let $a \approx_{\varepsilon,\delta} b$ denote the statement

$$a \leq e^{\varepsilon} b + \delta \quad \text{and} \quad b \leq e^{\varepsilon} a + \delta.$$

We say that two probability distributions p, q are (ε, δ)-*indistinguishable* if $p(E) \approx_{\varepsilon,\delta} q(E)$ for every event E.

Definition 9 (Private Learning Algorithm)**.** A randomized algorithm

$$A : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$$

is (ε, δ)-differentially-private if for every two samples $S, S' \in (X \times \{\pm 1\})^n$ that disagree on a single example, the output distributions $A(S)$ and $A(S')$ are (ε, δ)-indistinguishable.

We emphasize that (ε, δ)-indistinguishability must hold for every such pair of samples, even if they are not generated according to a (realizable) distribution. The parameters ε, δ are usually treated as follows: ε is a small constant (say 0.1), and δ is negligible, $\delta = n^{-\omega(1)}$, where n is the input sample size. The case of $\delta = 0$ is also referred to as *pure differential privacy*. Thus, a class $\mathcal{H}$ is privately learnable if it is PAC learnable by an algorithm A that is $(\varepsilon(n), \delta(n))$-differentially private with $\varepsilon(n) \leq 0.1$, and $\delta(n) \leq n^{-\omega(1)}$.

IV. Globally-Stable Learning of Littlestone Classes

A. Theorem Statement

The following states that any class $\mathcal{H}$ with a bounded Littlestone dimension can be learned by a globally-stable algorithm.

Theorem 10. *Let $\mathcal{H}$ be a hypothesis class with Littlestone dimension $d \geq 1$, let $\alpha > 0$, and set $m = (8^{d+1} + 1) \cdot \lceil d/\alpha \rceil$. Then there exists a randomized algorithm $G : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$ with the following properties. Let $\mathcal{D}$ be a realizable distribution and let $S \sim \mathcal{D}^m$ be an input sample. Then there exists a hypothesis f such that*

$$\Pr[G(S) = f] \geq \frac{1}{(d+1)2^{d+1}} \text{ and } \mathrm{loss}_{\mathcal{D}}(f) \leq \alpha.$$

B. The distributions $\mathcal{D}_k$

The Algorithm G is obtained by running the SOA on a sample drawn from a carefully tailored distribution. This distribution belongs to a family of distributions which we define next. Each of these distributions can be sampled from using black-box access to i.i.d. samples from $\mathcal{D}$. Recall that for a pair of samples S, T, we denote by $S \circ T$ the sample obtained by appending T to the end of S. Define a sequence of distributions $\mathcal{D}_k$ for $k \geq 0$ as follows:

Distributions $\mathcal{D}_k$

Let n denote an "auxiliary sample" size (to be fixed later) and let $\mathcal{D}$ denote the target realizable distribution over examples. The distributions $\mathcal{D}_k = \mathcal{D}_k(\mathcal{D}, n)$ are defined by induction on k as follows:

1) $\mathcal{D}_0$: output the empty sample $\emptyset$ with probability 1.
2) Let $k \geq 1$. If there exists a f such that
$$\Pr_{S \sim \mathcal{D}_{k-1}, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = f] \geq 2^{-d},$$
or if $\mathcal{D}_{k-1}$ is undefined then $\mathcal{D}_k$ is undefined.
3) Else, $\mathcal{D}_k$ is defined recursively by the following process:
   (i) Draw $S_0, S_1 \sim \mathcal{D}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$ independently.
   (ii) Let $f_0 = \mathsf{SOA}(S_0 \circ T_0)$, $f_1 = \mathsf{SOA}(S_1 \circ T_1)$.
   (iii) If $f_0 = f_1$ then go back to step (i).
   (iv) Else, pick $x \in \{x : f_0(x) \neq f_1(x)\}$ and sample $y \sim \{\pm 1\}$ uniformly.
   (v) If $f_0(x) \neq y$ then output $S_0 \circ T_0 \circ \big((x, y)\big)$ and else output $S_1 \circ T_1 \circ \big((x, y)\big)$.

Please see Figure 2 for an illustration of sampling $S \sim \mathcal{D}_k$ for $k = 3$.

We next observe some basic facts regarding these distributions. First, note that whenever $\mathcal{D}_k$ is well-defined, the process in Item 3 terminates with probability 1.

Let k be such that $\mathcal{D}_k$ is well-defined and consider a sample S drawn from $\mathcal{D}_k$. The size of S is $|S| = k \cdot (n + 1)$. Among these $k \cdot (n + 1)$ examples there are $k \cdot n$ examples drawn from $\mathcal{D}$ and k examples which are generated in Item 3(iv). We will refer to these k examples as *tournament examples*. Note that during the generation of $S \sim \mathcal{D}_k$ there are examples drawn from $\mathcal{D}$ which do not actually appear in S. In fact, the number of such examples may be unbounded, depending on how many times Items 3(i)-3(iii) were repeated. In Section IV-C1 we will define a "Monte-Carlo" variant of $\mathcal{D}_k$ in which the number of examples drawn from $\mathcal{D}$ is always bounded. This Monte-Carlo variant is what we actually use to define our globally-stable learning algorithm, but we introduce the simpler distributions $\mathcal{D}_k$ to clarify our analysis.

The k tournament examples satisfy the following important properties.

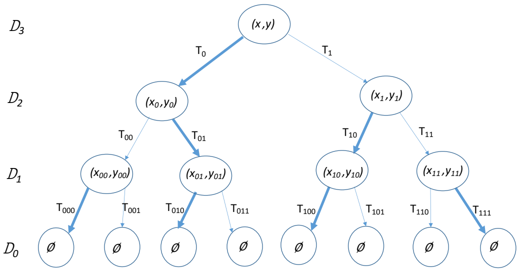

Figure 2: An illustration of the process of generating a sample $S \sim \mathcal{D}_3$. The edge labels are the samples T_b drawn in Item 3(i). The node labels are the tournament examples (x_b, y_b) generated in Item 3(iv). The bold edges indicate which of the samples T_{b0}, T_{b1} was appended to S in step 3(v) along with the corresponding tournament example. The sample S generated in this illustration is $T_{010} \circ (x_{01}, y_{01}) \circ T_{01} \circ (x_0, y_0) \circ T_0 \circ (x, y)$.

Observation 11. *Let k be such that $\mathcal{D}_k$ is well-defined and consider running the* SOA *on the concatenated sample $S \circ T$, where $S \sim \mathcal{D}_k$ and $T \sim \mathcal{D}^n$. Then*

1) *Each tournament example forces a mistake on the* SOA. *Consequently, the number of mistakes made by the* SOA *when run on $S \circ T$ is at least k.*
2) $\mathsf{SOA}(S \circ T)$ *is consistent with T.*

The first item follows directly from the definition of x in Item 3(iv) and the definition of S in Item 3(v). The second item clearly holds when $S \circ T$ is realizable by $\mathcal{H}$ (because the SOA is consistent). For non-realizable $S \circ T$, Item 2 holds by our extension of the SOA in Definition 8.

1) The Existence of Frequent Hypotheses: The following lemma is the main step in establishing global stability.

Lemma 12. *There exists $k \leq d$ and an hypothesis $f : X \to \{\pm 1\}$ such that*
$$\Pr_{S \sim \mathcal{D}_k, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = f] \geq 2^{-d}.$$

Proof: Suppose for the sake of contradiction that this is not the case. In particular, this means that $\mathcal{D}_d$ is well-defined and that for every f:
$$\Pr_{S \sim \mathcal{D}_d, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = f] < 2^{-d}. \qquad (1)$$

We show that this cannot be the case when $f = c$ is the target concept (i.e., for $c \in \mathcal{H}$ which satisfies $\text{loss}_{\mathcal{D}}(c) = 0$). Towards this end, note that with probability 2^{-d} over $S \sim \mathcal{D}_d$ we have that all d tournament examples are consistent with c. Indeed, this follows since in each tournament example (x_i, y_i), the label y_i is drawn independently of x_i and of the sample constructed thus far. So, $y_i = c(x_i)$ with probability $1/2$ independently for each tournament example.

Therefore, with probability 2^{-d} we have that $S \circ T$ is consistent with c (because all examples in $S \circ T$ which are drawn from $\mathcal{D}$ are also consistent with c). Now, since each tournament example forces a mistake on the SOA (Observation 11), and since the SOA does not make more than d mistakes on realizable samples, it follows that if all tournament examples in $S \sim \mathcal{D}_d$ are consistent with c then $\mathsf{SOA}(S) = \mathsf{SOA}(S \circ T) = c$. Thus,

$$\Pr_{S \sim \mathcal{D}_d, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = c] \geq 2^{-d},$$

which contradicts Equation 1 and finishes the proof. ■

2) Generalization: The next lemma shows that only hypotheses f that generalize well satisfy the conclusion of Lemma 12 (note the similarity of this proof with the proof of Proposition 5):

Lemma 13 (Generalization)**.** *Let k be such that $\mathcal{D}_k$ is well-defined. Then every f such that*

$$\Pr_{S \sim \mathcal{D}_k, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = f] \geq 2^{-d}$$

satisfies $\operatorname{loss}_{\mathcal{D}}(f) \leq \frac{d}{n}$.

Proof: Let f be a hypothesis such that $\Pr_{S \sim \mathcal{D}_k, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = f] \geq 2^{-d}$ and let $\alpha = \operatorname{loss}_{\mathcal{D}}(h)$. We will argue that

$$2^{-d} \leq (1-\alpha)^n. \tag{2}$$

Define the events A, B as follows.

1) A is the event that $\mathsf{SOA}(S \circ T) = f$. By assumption, $\Pr[A] \geq 2^{-d}$.
2) B is the event that f is consistent with T. Since $|T| = n$, we have that $\Pr[B] = (1-\alpha)^n$.

Note that $A \subseteq B$: Indeed, $\mathsf{SOA}(S \circ T)$ is consistent with T by the second item of Observation 11. Thus, whenever $\mathsf{SOA}(S \circ T) = f$, it must be the case that f is consistent with T. Hence, $\Pr[A] \leq \Pr[B]$, which implies Inequality 2 and finishes the proof (using the fact that $1 - \alpha \leq 2^{-\alpha}$ and taking logarithms on both sides). ■

C. The Algorithm G

1) A Monte-Carlo Variant of $\mathcal{D}_k$: Consider the following first attempt of defining a globally-stable learner G: (i) draw $i \in \{0 \ldots d\}$ uniformly at random, (ii) sample $S \sim \mathcal{D}_i$, and (iii) output $\mathsf{SOA}(S \circ T)$, where $T \sim \mathcal{D}^n$. The idea is that with probability $1/(d+1)$ the sampled i will be equal to a number k satisfying the conditions of Lemma 12, and so the desired hypothesis f guaranteed by this lemma (which also has low population loss by Lemma 13) will be outputted with probability at least $2^{-d}/(d+1)$.

The issue here is that sampling $f \sim \mathcal{D}_i$ may require an unbounded number of samples from the target distribution $\mathcal{D}$ (in fact, $\mathcal{D}_i$ may even be undefined). To circumvent this possibility, we define a Monte-Carlo variant of $\mathcal{D}_k$ in which the number of examples drawn from $\mathcal{D}$ is always bounded.

The Distributions $\tilde{\mathcal{D}}_k$ (a Monte-Carlo variant of $\mathcal{D}_k$)

1) Let n be the auxiliary sample size and N be an upper bound on the number of examples drawn from $\mathcal{D}$.
2) $\tilde{\mathcal{D}}_0$: output the empty sample $\emptyset$ with probability 1.
3) For $k > 0$, define $\tilde{\mathcal{D}}_k$ recursively by the following process:
   (*) **Throughout the process, if more than N examples from $\mathcal{D}$ are drawn (including examples drawn in the recursive calls), then output "Fail".**
   (i) Draw $S_0, S_1 \sim \tilde{\mathcal{D}}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$ independently.
   (ii) Let $f_0 = \mathsf{SOA}(S_0 \circ T_0)$, $f_1 = \mathsf{SOA}(S_1 \circ T_1)$.
   (iii) If $f_0 = f_1$ then go back to step (i).
   (iv) Else, pick $x \in \{x : f_0(x) \neq f_1(x)\}$ and sample $y \sim \{\pm 1\}$ uniformly.
   (v) If $f_0(x) \neq y$ then output $S_0 \circ T_0 \circ \big((x,y)\big)$ and else output $S_1 \circ T_1 \circ \big((x,y)\big)$.

Note that $\tilde{\mathcal{D}}_k$ is well-defined for every k, even for k such that $\mathcal{D}_k$ is undefined (however, for such k's the probability of outputting "Fail" may be large).

It remains to specify the upper bound N on the number of examples drawn from $\mathcal{D}$ in $\tilde{\mathcal{D}}_k$. Towards this end, we prove the following bound on the expected number of examples from $\mathcal{D}$ that are drawn during generating $S \sim \mathcal{D}_k$:

Lemma 14 (Expected Sample Complexity of Sampling From $\mathcal{D}_k$)**.** *Let k be such that $\mathcal{D}_k$ is well-defined, and let M_k denote the number of examples from $\mathcal{D}$ that are drawn in the process of generating $S \sim \mathcal{D}_k$. Then,*

$$\mathbb{E}[M_k] \leq 4^{k+1} \cdot n.$$

Proof: Note that $\mathbb{E}[M_0] = 0$ as $\mathcal{D}_0$ deterministically produces the empty sample. We first show that for all $0 < i < k$,

$$\mathbb{E}[M_{i+1}] \leq 4\mathbb{E}[M_i] + 4n, \tag{3}$$

and then conclude the desired inequality by induction.

To see why Inequality 3 holds, let the random variable R denote the number of times Item 3(i) was executed during the generation of $S \sim \mathcal{D}_{i+1}$. That is, R is the number of times a pair $S_0, S_1 \sim \mathcal{D}_i$ and a pair $T_0, T_1 \sim \mathcal{D}^n$ were drawn. Observe that R is distributed geometrically with success probability θ, where:

$$\begin{aligned}\theta &= 1 - \Pr_{S_0,S_1,T_0,T_1}\big[\mathsf{SOA}(S_0 \circ T_0) = \mathsf{SOA}(S_1 \circ T_1)\big]\\ &= 1 - \sum_f \Pr_{S,T}\big[\mathsf{SOA}(S \circ T) = f\big]^2\\ &\geq 1 - 2^{-d},\end{aligned}$$

where the last inequality follows because $i < k$ and hence $\mathcal{D}_i$ is well-defined, which implies that $\Pr_{S,T}\big[\mathsf{SOA}(S \circ T) = f\big] \leq 2^{-d}$ for all h.

Now, the random variable M_{i+1} can be expressed as follows:

$$M_{i+1} = \sum_{j=1}^{\infty} M_{i+1}^{(j)},$$

where

$$M_{i+1}^{(j)} = \begin{cases} 0 & \text{if } R < j,\\ \text{\# of examples drawn from } \mathcal{D} \text{ in} & \\ \text{the } j\text{'th execution of Item 3(i)} & \text{if } R \geq j.\end{cases}$$

Thus, $\mathbb{E}[M_{i+1}] = \sum_{j=1}^{\infty} \mathbb{E}[M_{i+1}^{(j)}]$. We claim that

$$\mathbb{E}[M_{i+1}^{(j)}] = (1-\theta)^{j-1} \cdot (2\mathbb{E}[M_i] + 2n).$$

Indeed, the probability that $R \geq j$ is $(1-\theta)^{j-1}$ and conditioned on $R \geq j$, in the j'th execution of Item 3(i) two samples from $\mathcal{D}_i$ are drawn and two samples from $\mathcal{D}^n$ are drawn. Thus

$$\begin{aligned}\mathbb{E}[M_{i+1}] &= \sum_{j=1}^{\infty} (1-\theta)^{j-1} \cdot (2\mathbb{E}[M_i] + 2n)\\ &= \frac{1}{\theta} \cdot (2\mathbb{E}[M_i] + 2n) \leq 4\mathbb{E}[M_i] + 4n,\end{aligned}$$

where the last inequality is because $\theta \geq 1 - 2^{-d} \geq 1/2$. This gives Inequality 3. Next, using that $\mathbb{E}[M_0] = 0$, a simple induction gives

$$\mathbb{E}[M_{i+1}] \leq (4 + 4^2 + \ldots + 4^{i+1})n \leq 4^{i+2}n,$$

and the lemma follows by taking $i + 1 = k$. ■

2) Completing the Proof of Theorem 10: We define our globally-stable learning algorithm G as follows.

```
Algorithm $G$
1) Consider the distribution $\tilde{\mathcal{D}}_k$, where the auxiliary sample size is set to $n = \lceil \frac{d}{\alpha} \rceil$ and the sample complexity upper bound is set to $N = 8^{d+1} \cdot n$.
2) Draw $k \in \{0, 1, \ldots, d\}$ uniformly at random.
3) Output $h = \mathsf{SOA}(S \circ T)$, where $T \sim \mathcal{D}^n$ and $S \sim \tilde{\mathcal{D}}_k$.
```

First note that the sample complexity of G is $|S| + |T| \leq N + n = (8^{d+1} + 1) \cdot \lceil \frac{d}{\alpha} \rceil$, as required. It remains to show that there exists a hypothesis f such that:

$$\Pr[G(S) = f] \geq \frac{2^{-(d+1)}}{d+1} \text{ and } \operatorname{loss}_{\mathcal{D}}(f) \leq \alpha.$$

By Lemma 12, there exists $k^* \leq d$ and f^* such that

$$\Pr_{S \sim \mathcal{D}_{k^*}, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = f^*] \geq 2^{-d}.$$

By Lemma 13,

$$\operatorname{loss}_{\mathcal{D}}(f^*) \leq \frac{d}{n} \leq \alpha.$$

We claim that G outputs f^* with probability at least $2^{-(d+1)}$. To see this, let M_{k^*} denote the number of examples drawn from $\mathcal{D}$ during the generation of $S \sim \mathcal{D}_{k^*}$. Lemma 14 and an application of Markov's inequality yield

$$\begin{aligned}\Pr\big[M_{k^*} > 8^{d+1} \cdot n\big] &\leq \Pr\big[M_{k^*} > 2^{d+1} \cdot 4^{k^*+1} \cdot n\big]\\ &\qquad\qquad \text{(because } k^* \leq d\text{)}\\ &\leq 2^{-(d+1)}.\\ &\text{(by Markov's inequality, since } \mathbb{E}[M_{k^*}] \leq 4^{k^*+1} \cdot n\text{)}\end{aligned}$$

Therefore,

$$\begin{aligned}&\Pr_{S \sim \tilde{\mathcal{D}}_{k^*}, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = f^*]\\ =&\Pr_{S \sim \mathcal{D}_{k^*}, T \sim \mathcal{D}^n}[\mathsf{SOA}(S \circ T) = f^* \text{ and } M_{k^*} \leq 8^{d+1} \cdot n]\\ \geq& 2^{-d} - 2^{-(d+1)} = 2^{-(d+1)}.\end{aligned}$$

Thus, since $k = k^*$ with probability $1/(d+1)$, it follows that G outputs f^* with probability at least $\frac{2^{-(d+1)}}{d+1}$ as required. ■

V. Globally-Stable Learning Implies Private Learning

In this section we prove that any globally-stable learning algorithm yields a differentially-private learning algorithm with finite sample complexity.

A. Tools from Differential Privacy

We begin by stating a few standard tools from the differential privacy literature which underlie our construction of a learning algorithm.

Let X be a data domain and let $S \in X^n$. For an element $x \in X$, define $\mathrm{freq}_S(x) = \frac{1}{n} \cdot \#\{i \in [n] : x_i = x\}$, i.e., the fraction of the elements in S which are equal to x.

Lemma 15 (Stable Histograms [KKMN09], [BNS16b]). *Let X be any data domain. For*

$$n \geq O\left(\frac{\log(1/\eta\beta\delta)}{\eta\varepsilon}\right)$$

there exists an (ε, δ)-differentially private algorithm Hist *which, with probability at least $1-\beta$, on input $S = (x_1, \ldots, x_n)$ outputs a list $L \subseteq X$ and a sequence of estimates $a \in [0,1]^{|L|}$ such that*

- *Every x with $\mathrm{freq}_S(x) \geq \eta$ appears in L and*
- *For every $x \in L$, the estimate a_x satisfies $|a_x - \mathrm{freq}_S(x)| \leq \eta$.*

Using the Exponential Mechanism of McSherry and Talwar [MT07], Kasiviswanathan et al. [KLN$^+$11] described a generic differentially-private learner based on approximate empirical risk minimization.

Lemma 16 (Generic Private Learner [KLN$^+$11]). *Let $H \subseteq \{\pm 1\}^X$ be a collection of hypotheses. For*

$$n = O\left(\frac{\log|H| + \log(1/\beta)}{\alpha\varepsilon}\right)$$

there exists an ε-differentially private algorithm GenericLearner : $(X \times \{\pm 1\})^n \to H$ *such that the following holds. Let $\mathcal{D}$ be a distribution over $(X \times \{\pm 1\})$ such that there exists $h^* \in H$ with*

$$\mathrm{loss}_{\mathcal{D}}(h^*) \leq \alpha.$$

Then on input $S \sim \mathcal{D}^n$, algorithm GenericLearner *outputs, with probability at least $1-\beta$, a hypothesis $\hat{h} \in H$ such that*

$$\mathrm{loss}_{\mathcal{D}}(\hat{h}) \leq 2\alpha.$$

Our formulation of the guarantees of this algorithm differ slightly from those of [KLN$^+$11], so we give its standard proof for completeness.

Proof of Lemma 16: The algorithm GenericLearner(S) samples a hypothesis $h \in H$ with probability proportional to $\exp(-\varepsilon n \, \mathrm{loss}_S(h)/2)$. This algorithm can be seen as an instantiation of the Exponential Mechanism [MT07]; the fact that changing one sample changes the value of $\mathrm{loss}_S(h)$ by at most 1 implies that GenericLearner is ε-differentially private.

We now argue that GenericLearner is an accurate learner. Let E denote the event that the sample S satisfies the following conditions:

1) For every $h \in H$ such that $\mathrm{loss}_{\mathcal{D}}(h) > 2\alpha$, it also holds that $\mathrm{loss}_S(h) > 5\alpha/3$, and
2) For the hypothesis $h^* \in H$ satisfying $\mathrm{loss}_{\mathcal{D}}(h^*) \leq \alpha$, it also holds that $\mathrm{loss}_S(h^*) \leq 4\alpha/3$.

We claim that $\Pr[E] \geq 1 - \beta/2$ as long as $n \geq O(\log(|H|/\beta)/\alpha)$. To see this, let $h \in H$ be an arbitrary hypothesis with $\mathrm{loss}_D(h) > 2\alpha$. By a multiplicative Chernoff bound[5] we have $\mathrm{loss}_S(h) > 7\alpha/4$ with probability at least $1 - \beta/(4|H|)$ as long as $n \geq O(\log(|H|/\beta)/\alpha)$. Taking a union bound over all $h \in H$ shows that condition 1. holds with probability at least $1-\beta/4$. Similarly, a multiplicative Chernoff bound ensures that condition 2 holds with probability at least $1-\beta/4$, so E holds with probability at least $1-\beta/2$.

Now we show that conditioned on E, the algorithm GenericLearner(S) indeed produces a hypothesis h with $\mathrm{loss}_D(\hat{h}) \leq 2\alpha$. This follows the standard analysis of the accuracy guarantees of the Exponential Mechanism. Condition 2 of the definition of event E guarantees that $\mathrm{loss}_S(h^*) \leq 4\alpha/3$. This ensures that the normalization factor in the definition of the Exponential Mechanism is at least $\exp(-2\varepsilon\alpha n/3)$. Hence by a union bound,

$$\begin{aligned}\Pr[\mathrm{loss}_S(\hat{h}) > 5\alpha/3] &\leq |H| \cdot \frac{\exp(-5\varepsilon\alpha n/6)}{\exp(-2\varepsilon\alpha n/3)} \\ &= |H| e^{-\varepsilon\alpha n/6}.\end{aligned}$$

Taking $n \geq O(\log(|H|/\beta)/\alpha\varepsilon)$ ensures that this probability is at most $\beta/2$. Given that $\mathrm{loss}(\hat{h}) \leq 5\alpha/3$, Condition 1 of the definition of event E ensures that $\mathrm{loss}_{\mathcal{D}}(\hat{h}) \leq 2\alpha$. Thus, for n sufficiently large as described, we have overall that $\mathrm{loss}_{\mathcal{D}}(\hat{h}) \leq 2\alpha$ with probability at least $1-\beta$. ■

B. Construction of a Private Learner

We now describe how to combine the Stable Histograms algorithm with the Generic Private Learner to convert any globally-stable learning algorithm into a differentially-private one.

Theorem 17. *Let $\mathcal{H}$ be a concept class over data domain X. Let $G : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$ be a randomized algorithm such that, for $\mathcal{D}$ a realizable distribution and $S \sim \mathcal{D}^m$, there exists a hypothesis h such that $\Pr[G(S) = h] \geq \eta$ and $\mathrm{loss}_{\mathcal{D}}(h) \leq \alpha/2$.*

[5] I.e., for independent random variables $Z_1, \ldots, Z_n$ whose sum Z satisfies $\mathbb{E}[Z] = \mu$, we have for every $\delta \in (0,1)$ that $\Pr[Z \leq (1-\delta)\mu] \leq \exp(-\delta^2\mu/2)$ and $\Pr[Z \geq (1+\delta)\mu] \leq \exp(-\delta^2\mu/3)$.

Then for some

$$n = O\left(\frac{m \cdot \log(1/\eta\beta\delta)}{\eta\varepsilon} + \frac{\log(1/\eta\beta)}{\alpha\varepsilon}\right)$$

there exists an (ε, δ)*-differentially private algorithm* $M : (X \times \{\pm 1\})^n \to \{\pm 1\}^X$ *which, given* n *i.i.d. samples from* $\mathcal{D}$*, produces a hypothesis* $\hat{h}$ *such that* $\mathrm{loss}_{\mathcal{D}}(\hat{h}) \leq \alpha$ *with probability at least* $1 - \beta$*.*

Theorem 17 is realized the learning algorithm M described below. Here, the parameter

$$k = O\left(\frac{\log(1/\eta\beta\delta)}{\eta\varepsilon}\right)$$

is chosen so that Lemma 15 guarantees Algorithm Hist succeeds with the stated accuracy parameters. The parameter

$$n' = O\left(\frac{\log(1/\eta\beta)}{\alpha\varepsilon}\right)$$

is chosen so that Lemma 16 guarantees that GenericLearner succeeds on a list L of size $|L| \leq 2/\eta$ with the given accuracy and confidence parameters.

Differentially-Private Learner M

1) Let $S_1, \ldots, S_k$ each consist of m i.i.d. samples from $\mathcal{D}$. Run G on each batch of samples producing $h_1 = G(S_1), \ldots, h_k = G(S_k)$.
2) Run the Stable Histogram algorithm Hist on input $H = (h_1, \ldots, h_k)$ using privacy parameters $(\varepsilon/2, \delta)$ and accuracy parameters $(\eta/8, \beta/3)$, producing a list L of frequent hypotheses.
3) Let S' consist of n' i.i.d. samples from $\mathcal{D}$. Run GenericLearner(S') using the collection of hypotheses L with privacy parameter $\varepsilon/2$ and accuracy parameters $(\alpha/2, \beta/3)$ to output a hypothesis $\hat{h}$.

Proof of Theorem 17: We first argue that the algorithm M is differentially private. The outcome L of step 2 is generated in a $(\varepsilon/2, \delta)$-differentially-private manner as it inherits its privacy guarantee from Hist. For every fixed choice of the coin tosses of G during the executions $G(S_1), \ldots, G(S_k)$, a change to one entry of some S_i changes at most one outcome $h_i \in H$. Differential privacy for step 2 follows by taking expectations over the coin tosses of all the executions of G, and for the algorithm as a whole by simple composition.

We now argue that the algorithm is accurate. Using standard generalization arguments, we have that with probability at least $1 - \beta/3$,

$$\left|\mathrm{freq}_H(h) - \Pr_{S \sim \mathcal{D}^m}[G(S) = h]\right| \leq \frac{\eta}{8}$$

for every $h \in \{\pm 1\}^X$ as long as $k \geq O(\log(1/\beta)/\eta)$. Let us condition on this event. Then by the accuracy of the algorithm Hist, with probability at least $1 - \beta/2$ it produces a list L containing h^* together with a sequence of estimates that are accurate to within additive error $\eta/8$. In particular, h^* appears in L with an estimate $a_{h^*} \geq \eta - \eta/8 - \eta/8 \geq 3\eta/4$.

Now remove from L every item h with estimate $a_h < 3\eta/4$. Since every estimate is accurate to within $\eta/8$, this leaves a list with $|L| \leq 2/\eta$ that contains h^* with $\mathrm{loss}_{\mathcal{D}}(h^*) \leq \alpha$. Hence, with probability at least $1 - \beta/3$, step 3 succeeds in identifying h^* with $\mathrm{loss}_{\mathcal{D}}(h^*) \leq \alpha/2$.

The total sample complexity of the algorithm is $k \cdot m + n'$ which matches the asserted bound. ■

VI. Wrapping up (Proof of Theorem 1)

We now combine Theorem 10 (finite Littlestone dimension $\Longrightarrow$ global stability) with Theorem 17 (global stability $\Longrightarrow$ private learnability) to prove Theorem 1.

Proof of Theorem 1: Let $\mathcal{H}$ be a hypothesis class with Littlestone dimension d and let $\mathcal{D}$ be any realizable distribution. Then Theorem 10 guarantees, for $m = O(8^d \cdot d/\alpha)$, the existence of a randomized algorithm $G : (X \times \{\pm 1\})^m \to \{\pm 1\}^X$ and a hypothesis f such that

$$\Pr[G(S) = f] \geq \frac{1}{(d+1)2^{d+1}} \text{ and } \mathrm{loss}_{\mathcal{D}}(f) \leq \alpha/2.$$

Taking $\eta = 1/(d+1)2^{d+1}$, Theorem 17 gives an (ε, δ)-differentially private learner with sample complexity

$$\begin{aligned} n &= O\left(\frac{m \cdot \log(1/\eta\beta\delta)}{\eta\varepsilon} + \frac{\log(1/\eta\beta)}{\alpha\varepsilon}\right) \\ &= O\left(\frac{16^d \cdot d^2 \cdot (d + \log(1/\beta\delta))}{\alpha\varepsilon}\right). \end{aligned}$$

■

Acknowledgments

We would like to thank Amos Beimel and Uri Stemmer for pointing out, and helping to fix, a mistake in the derivation of Corollary 2 in a previous version. We also thank Yuval Dagan for providing useful comments and for insightful conversations.

M.B. is supported by NSF grant CCF-1947889. Part of this work was done while the author was at the Simons Institute for the Theory of Computing, supported by a Google Research Fellowship. R.L is supported by the Israel Science Foundation (grant No. 2188/20). S.M is supported by the Israel Science Foundation (grant No. 1225/20), by an Azrieli Faculty Fellowship, and by a grant from the United States - Israel Binational Science Foundation (BSF). Part of this work was done while the author was a Google AI Princeton.

References

[ABMS20] Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. Closure properties for private classification and online prediction. *arXiv preprint arXiv:2003.04509*, 2020.

[AHR08] Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 263–274, 2008.

[ALMM19] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite Littlestone dimension. In *Proceedings of the 51st Annual ACM Symposium on the Theory of Computing*, STOC '19, New York, NY, USA, 2019. ACM.

[ALMT17] Jacob D. Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via differential privacy. *CoRR*, abs/1711.10019, 2017.

[AS17] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 32–40. PMLR, 2017.

[BBKN14] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94(3):401–437, 2014.

[BCS20] Mark Bun, Marco L. Carmosino, and Jessica Sorrell. Efficient, noise-tolerant, and private learning via boosting. *CoRR*, abs/2002.01100, 2020.

[BDRS18] Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated cdp. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing*, STOC '18, pages 74–86, New York, NY, USA, 2018. ACM.

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *J. Mach. Learn. Res.*, 2:499–526, 2002.

[BLM19] Olivier Bousquet, Roi Livni, and Shay Moran. Passing tests without memorizing: Two models for fooling discriminators, 2019.

[BNS15] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Learning privately with labeled and unlabeled examples. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 461–477. Society for Industrial and Applied Mathematics, 2015.

[BNS16a] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. *Theory of Computing*, 12(1):1–61, 2016.

[BNS16b] Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In *Proceedings of the 7th Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 369–380, New York, NY, USA, 2016. ACM.

[BNS19] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of pure private learners. *Journal of Machine Learning Research*, 20(146):1–33, 2019.

[BNSV15] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '15, pages 634–649, Washington, DC, USA, 2015. IEEE Computer Society.

[BPS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, 2009.

[CF19] Hunter Chase and James Freitag. Model theory and machine learning. *The Bulletin of Symbolic Logic*, 25(03):319–332, Feb 2019.

[CGKM19] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, UMAP'19 Adjunct, pages 309–315, New York, NY, USA, 2019. ACM.

[CL06] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[DHP$^+$12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 214–226. ACM, 2012.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the*

3rd Conference on Theory of Cryptography, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Machine Learning*, 9(3–4):211–407, 2014.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60, Washington, DC, USA, 2010. IEEE Computer Society.

[FX15] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. *SIAM Journal on Computing*, 44(6):1740–1764, 2015.

[GG11] Eran Gat and Shafi Goldwasser. Probabilistic search algorithms with unique answers and their cryptographic applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:136, 2011.

[GHM19] Alon Gonen, Elad Hazan, and Shay Moran. Private learning implies online learning: An efficient reduction. *NeurIPS*, 2019.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3-4):157–325, August 2016.

[JMNR19] Matthew Joseph, Jieming Mao, Seth Neel, and Aaron Roth. The role of interactivity in local differential privacy. In *FOCS*, 2019.

[KKMN09] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 171–180, Madrid, Spain, 2009. ACM.

[KLM$^+$19] Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap, 2019.

[KLN$^+$11] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[KV02] Adam Kalai and Santosh Vempala. Geometric algorithms for online optimization. In *Journal of Computer and System Sciences*, pages 26–40, 2002.

[KV05] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, October 2005.

[Lit87] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.

[NRW19] Seth Neel, Aaron Roth, and Zhiwei Steven Wu. How to use heuristics for differential privacy. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 72–93, 2019.

[O'N16] Cathy O'Neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown, New York, first edition edition, 2016.

[OS18] Igor Carboni Oliveira and Rahul Santhanam. Pseudo-derandomizing learning and approximation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, RANDOM '18, pages 55:1–55:19, 2018.

[SS12] Shai Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, February 2012.

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

[SSS07] Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2):115–142, 2007.

[Vad17] Salil Vadhan. The complexity of differential privacy. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, chapter 7, pages 347–450. Springer International Publishing AG, Cham, Switzerland, 2017.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.