Simultaneous Private Learning of Multiple Concepts

Mark Bun[*]   Kobbi Nissim[†]   Uri Stemmer[‡]

November 30, 2015

Abstract

We investigate the *direct-sum* problem in the context of differentially private PAC learning: What is the sample complexity of solving k learning tasks *simultaneously* under differential privacy, and how does this cost compare to that of solving k learning tasks without privacy? In our setting, an individual example consists of a domain element x labeled by k unknown concepts $(c_1, \ldots, c_k)$. The goal of a *multi*-learner is to output k hypotheses $(h_1, \ldots, h_k)$ that generalize the input examples.

Without concern for privacy, the sample complexity needed to simultaneously learn k concepts is essentially the same as needed for learning a single concept. Under differential privacy, the basic strategy of learning each hypothesis independently yields sample complexity that grows polynomially with k. For some concept classes, we give multi-learners that require fewer samples than the basic strategy. Unfortunately, however, we also give lower bounds showing that even for very simple concept classes, the sample cost of private multi-learning must grow polynomially in k.

Keywords: differential privacy, PAC learning, agnostic learning, direct-sum

1 Introduction

The work on *differential privacy* [18] is aimed at providing useful analyses on privacy-sensitive data while providing strong individual-level privacy protection. One family of such analyses that has received a lot of attention is PAC learning [34]. These tasks abstract many of the computations performed over sensitive information [26].

We address the *direct-sum* problem – what is the cost of solving multiple instances of a computational task simultaneously as compared to solving each of them separately? – in the context of differentially private PAC learning. In our setting, individual examples are drawn from domain X and labeled by k unknown concepts $(c_1, \ldots, c_k)$ taken from a concept class $C = \{c : X \to \{0, 1\}\}$, i.e., each example is of the form $(x, y_1, \ldots, y_k)$, where $x \in X$ and $y_i = c_i(x)$. The goal of a multi-learner is to output k hypotheses $(h_1, \ldots, h_k)$ that generalize the input examples while preserving the privacy of individuals.

The direct-sum problem has its roots in complexity theory, and is a basic problem for many algorithmic tasks. It also has implications for the practical use of differential privacy. Consider, for instance, a hospital that collects information about its patients and wishes to use this information for medical research. The hospital records for each patient a collection of attributes such as age, sex, and the results of various diagnostic tests (for each patient, these attributes make up a point x in some domain X) and, for each of k diseases, whether the patient suffers from the disease (the k labels $(y_1, \ldots, y_k)$). Based on this collection of data, the hospital researchers wish to learn good predictors for the k diseases. One option for the researchers is to perform each of the learning tasks on a fresh sample of patients, hence enlarging the number of patient examples needed (i.e. the *sample complexity*) by a factor of k, which can be very costly.

Without concern for privacy, the sample complexity that is necessary and sufficient for performing the k learning tasks is actually fully characterized by the VC dimension of the concept class C – it is independent of the number of learning tasks k. In this work, we set out to examine if the situation is similar when the learning is performed with differential privacy. Interestingly, we see that with differential privacy the picture is quite different, and in particular, the required number of examples can grow polynomially in k.

Private learning. A *private learner* is an algorithm that is given an sample of labeled examples $(x, c(x))$ (each representing the information and label pertaining to an individual) and outputs a generalizing hypothesis h that guarantees differential privacy with respect to its examples. The first differentially private learning algorithms were given by Blum et al. [9] and the notion of *private learning* was put forward and formally researched by Kasiviswanathan et al. [26]. Among other results, the latter work presented a generic construction of differentially private learners with sample complexity $O(\log |C|)$.

In contrast, the sample complexity of (non-private) PAC learning is $\Theta(\mathrm{VC}(C))$, which can be much lower than $\log |C|$ for specific concept classes. This gap led to a line of work examining the sample complexity of private learning, which has revealed a significantly more complex picture than there is for non-private learning. In particular, for *pure* differentially private learners, it is known that the sample complexity of proper learning (where the learner returns a hypothesis h taken from C) is sometimes higher than the sample complexity of improper learners (where h comes from an arbitrary hypothesis class H). The latter is characterized by the *representation dimension* of the concept class C, which is generally higher than the VC dimension [5, 4, 15, 6, 22]. By contrast, a sample complexity gap between proper and improper learners does not exist for non-private learning. In the case of *approximate* differential privacy no such combinatorial characterization is currently known. It is however known that the sample complexity of such learners can be significantly lower than that of pure-differentially private learners and yet higher than the VC dimension of C [6, 22, 13]. Furthermore, there exist (infinite) PAC-learnable concept classes for which no differentially private learner (pure or approximate) exists.

Private multi-learning. In this work we examine the sample complexity of private multi-learning. Our work is motivated by the recurring research theme of the direct-sum, as well as by the need to understand whether multi-learning remains feasible under differential privacy, as it is without privacy constraints.

At first glance, private multi-learning appears to be similar to the query release problem, the goal of which is to approximate the average values of a large collection of predicates on a dataset. One surprising result in differential privacy is that it is possible to answer an exponential number of such queries on a dataset [10, 30, 24]. For example, Blum, Ligett, and Roth [10] showed that given a dataset D and a concept class C, it is possible to generate with differential privacy a dataset $\hat{D}$ such that the average value of c on D approximates the average of c on $\hat{D}$ for every $c \in C$ *simultaneously*. The sample complexity required, i.e., the size of the database D, to perform this sanitization is only logarithmic in $|C|$. Results of this flavor suggest that we can also learn exponentially many concepts simultaneously. However, we give negative results showing that this is not the case, and that multi-learning can have significantly higher sample complexity than query release.

1.1 Our results

Prior work on privately learning the simple concept classes $\texttt{POINT}_X$ (of functions that evaluate to 1 on exactly one point of their domain X and to 0 otherwise) and $\texttt{THRESH}_X$ (of functions that evaluate to 1 on a prefix of the domain X and to 0 otherwise) has demonstrated a rather complex picture, depending on whether learners are proper or improper, and whether learning is performed with pure or approximate differential privacy [5, 4, 6, 7, 13]. We analyze the sample complexity of multi-learning of these simple concept classes, as well as general concept classes. We also consider the class $\texttt{PAR}_d$ of parity functions, but in this case we restrict our attention to uniformly selected examples. We examine both proper and improper PAC and agnostic learning under pure and approximate differential privacy. For ease of reference, we include tables with our results in Section 1.3, where we omit the dependency on the privacy and accuracy parameters.

Techniques for private k-learning. Composition theorems for differential privacy show that the sample complexity of learning k concepts simultaneously is at most a factor of k larger than the sample complexity of learning one concept (and may be reduced to $\sqrt{k}$ for approximate differential privacy). Unfortunately, privately learning one concept from a concept class C can sometimes be quite costly, requiring much higher sample complexity than $\text{VC}(C)$ which is needed to learn non-privately. Building on techniques of Beimel, Nissim, and Stemmer [8], we show that the multiplicative dependence on k can always be reduced to the VC-dimension of C, at the expense of producing a one-time sanitization of the dataset.

Theorem 1.1 (Informal). *Let C be a concept class for which there is pure differentially private sanitizer for $C^{\oplus} = \{f \oplus g : f, g \in C\}$ with sample complexity m. Then there is an pure differentially private agnostic k-learner for C with sample complexity $O(m + k \cdot \text{VC}(C))$.*

Similarly, if $C^{\oplus}$ has an approximate differentially private sanitizer with sample complexity m, then there is an approximate differentially private agnostic k-learner for C with sample complexity $O(m + \sqrt{k} \cdot \text{VC}(C))$.

The best known general-purpose sanitizers require sample complexity $m = O(\text{VC}(C) \log |X|)$ for pure differential privacy [10] and $m = O(\log |C| \sqrt{\log |X|})$ for approximate differential privacy [24]. However, for specific concept classes (such as $\texttt{POINT}_X$ and $\texttt{THRESH}_X$), the sample complexity of sanitization can be much lower.

In the case of approximate differential privacy, the sample complexity of k-learning can be even lower than what is achievable with our generic learner. Using stability-based arguments, we show that point func-

tions and parities under the uniform distribution can be PAC k-learned with sample complexity $O(\mathrm{VC}(C))$ – independent of the number of concepts k (see Theorems 3.13 and 3.12).

Lower bounds. In light of the above results, one might hope to be able to reduce the dependence on k further, or to eliminate it entirely (as is possible in the case of non-private learning). We show that this is not possible, even for the simplest of concept classes. In the case of pure differential privacy, a packing argument [21, 25, 5] shows that any non-trivial concept class requires sample complexity $\Omega(k)$ to privately k-learn (Theorem 5.1). For approximate differential privacy, we use fingerprinting codes [12, 14] to show that unlike points and parities, threshold functions require sample complexity $\tilde{\Omega}(k^{1/3})$ to PAC learn privately (Corollary 4.6). Moreover, any non-trivial concept class requires sample complexity $\tilde{\Omega}(\sqrt{k})$ to privately learn in the agnostic model (Theorem 4.7). In the case of point functions, this matches the upper bound achievable by our generic learner.

We highlight a few of the main takeaways from our results:

A complex answer to the direct sum question. Our upper bounds show that solving k learning problems simultaneously can require substantially lower sample complexity than solving the problems individually. On the other hand, our lower bounds show that a significant dependence on k is generally necessary.

Separation between private PAC and private agnostic learning. Non-privately, the sample complexities of PAC and agnostic learning are of the same order (differing only in the dependency in the accuracy parameters). Beimel et al. [8] showed that this is also the case with differentially private learning (of one concept). Our results on learning point functions show that private PAC and agnostic *multi-learning* can be substantially different (even for learning up to constant error). In the case of approximate differential privacy, $O(1)$ sample suffice to PAC-learn multiple point functions. However, $\tilde{\Omega}(\sqrt{k})$ samples are needed to learn k points agnostically.

Separation between improper learning with approximate differential privacy and non-private learning. Recently, Bun et al. [13] showed that the sample complexity of learning one threshold function with approximate differential privacy exceeds the VC dimension, but only in the case of *proper* learning. Thus it remains possible that improper learning with approximate differential privacy can match the sample complexity of non-private learning. While we do not address this question directly, we exhibit a separation for multi-learning. In particular, learning k thresholds with approximate differential privacy requires $\tilde{\Omega}(k^{1/3})$ samples, even improperly, while $O(1)$ samples suffices non-privately.

1.2 Related work

Differential privacy was defined in [18] and the relaxation to approximate differential privacy is from [16]. Most related to our work is the work on private learning and its sample complexity [9, 26, 15, 19, 4, 6, 7, 22, 8, 13] and the early work on sanitization [10]. That many "natural" learning tasks can be performed privately was shown in the early work of Blum et al. [9] and Kasiviswanathan et al. [26]. A characterization for the sample complexity of *pure-private* learners was given in [6], in terms of a new combinatorial measure – the *Representation Dimension*, that is, given a class C, the number of samples needed and sufficient for privately learning C is $\Theta(\mathrm{RepDim}(C))$. Building on [6], Feldman and Xiao [22] showed an equivalence between the representation dimension of a concept C and the randomized one-way communication complexity of the evaluation problem for concepts from C. Using this equivalence they separated the sample complexity of pure-private learners from that of non-private ones.

The problem of learning multiple concepts simultaneously (without privacy) has been considered before. Motivated by the problem of bridging computational learning and reasoning, Valiant [35] also observed that (without privacy) multiple concepts can be learned from a common dataset in a data efficient manner.

1.3 Tables of results

The following tables summarize the results of this work. In the tables below C is a class of concepts (i.e., predicates) defined over domain X. Sample complexity upper and lower bounds is given in terms of $|C|$ and $|X|$. Note that for $\texttt{POINT}_X$, $\texttt{THRESH}_X$, and $\texttt{PAR}_d$ we have $|C| = \Theta(|X|)$.

Where not explicitly noted, upper bounds hold for the setting of agnostic learning and lower bounds are for the (potentially easier) setting of PAC learning. Similarly, where not explicitly noted, upper bounds are for proper learning and lower bounds are for the (less restrictive) setting of improper learning. For simplicity, these tables hide constant and logarithmic factors, as well as dependencies on the learning and privacy parameters.

Multi-learning with pure differential privacy.

Upper bounds:

C	PAC learning		Agnostic learning		References
	proper	improper	proper	improper	
$\texttt{POINT}_X$	$k + \log|C|$	k	$k + \log|C|$	k	Thm. 3.1, Cor. 3.3
$\texttt{THRESH}_X$	$k + \log|C|$				Thm. 3.1
General	$\min\{k \log|C|, k\,\mathrm{VC}(C) + \log|X|\,\mathrm{VC}(C)\}$				Thm. 3.1
$\texttt{PAR}_d$ (uniform)	$k \log|C|$				Thm. 3.1

Lower bounds:

C	PAC learning		Agnostic learning		References
	proper	improper	proper	improper	
$\texttt{POINT}_X$	$k + \log|C|$	k	$k + \log|C|$	k	Thm. 5.1, [5]
$\texttt{THRESH}_X$	$k + \log|C|$				Thm. 5.1, [5, 22]
$\texttt{PAR}_d$ (uniform)	$k \log|C|$				Thm. 5.4

Multi-learning with approximate differential privacy.

Upper bounds:

C	PAC learning (proper and improper)	Agnostic learning (proper and improper)
$\texttt{POINT}_X$	1 (Thm. 3.13)	$\sqrt{k}$ (Cor. 3.9)
$\texttt{THRESH}_X$	$2^{\log^* |X|} + \sqrt{k}$ (Cor. 3.10)	
General C	$\min\{\sqrt{k} \log|C|, \sqrt{k}\,\mathrm{VC}(C) + \log|X|\,\mathrm{VC}(C), \sqrt{k}\,\mathrm{VC}(C) + \sqrt{\log|X|}\log|C|\}$ (Thm. 3.1)	
$\texttt{PAR}_d$ (uniform)	$\log|C|$ (Thm. 3.12)	$\sqrt{k} \log|C|$ (Thm. 3.1)

Lower bounds:

C	PAC learning		Agnostic learning		References
	proper	improper	proper	improper	
$\texttt{POINT}_X$	1		$\sqrt{k}$		Cor. 4.10
$\texttt{THRESH}_X$	$\log^* \|X\| + k^{1/3}$	$k^{1/3}$	$\log^* \|X\| + \sqrt{k}$	$\sqrt{k}$	Cor. 4.6, Cor. 4.10, [13]
$\texttt{PAR}_d$ (uniform)	$\log \|C\|$		$\sqrt{k} + \log \|C\|$		Cor. 4.10)

2 Preliminaries

We recall and extend standard definitions from learning theory and differential privacy.

2.1 Multi-learners

In the following X is some arbitrary domain. A concept (similarly, hypothesis) over domain X is a predicate defined over X. A concept class (similarly, hypothesis class) is a set of concepts.

Definition 2.1 (Generalization Error). Let $\mathcal{P} \in \Delta(X \times \{0,1\})$ be a probability distribution over $X \times \{0,1\}$. The *generalization error* of a hypothesis $h : X \to \{0,1\}$ w.r.t. $\mathcal{P}$ is defined as $\mathrm{error}_{\mathcal{P}}(h) = \Pr_{(x,y)\sim\mathcal{P}}[h(x) \neq y]$.

Let $\mathcal{D} \in \Delta(X)$ be a probability distribution over X and let $c : x \to \{0,1\}$ be a concept. The *generalization error* of hypothesis $h : X \to \{0,1\}$ w.r.t. c and $\mathcal{D}$ is defined as $\mathrm{error}_{\mathcal{D}}(c, h) = \Pr_{x\sim\mathcal{D}}[h(x) \neq c(x)]$. If $\mathrm{error}_{\mathcal{D}}(c, h) \leq \alpha$ we say that h is α-*good* for c and $\mathcal{D}$.

Definition 2.2 (Multi-labeled database). A k-*labeled* database over a domain X is a database $S \in (X \times \{0,1\}^k)^*$. That is, S contains $\|S\|$ elements from X, each concatenated with k binary labels.

Let $\mathcal{A} : \left(X \times \{0,1\}^k\right)^n \to \left(2^X\right)^k$ be an algorithm that operates on a k-labeled database and returns k hypotheses. Let C be a concept class over a domain X and let H be a hypothesis class over X. We now give a generalization of the notion of PAC learning [34] to multi-labeled databases (the standard PAC definition is obtained by setting $k = 1$):

Definition 2.3 (PAC Multi-Learner). Algorithm $\mathcal{A}$ is an (α, β)-PAC k-learner for concept class C using hypothesis class H with sample complexity n if for every distribution $\mathcal{D}$ over X and for every fixture of $(c_1, \ldots, c_k)$ from C, given a k-labeled database as an input $S = ((x_i, c_1(x_i), \ldots, c_k(x_i)))_{i=1}^n$ where each x_i is drawn i.i.d. from $\mathcal{D}$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1, \ldots, h_k)$ from H satisfying

$$\Pr\left[\max_{1\leq j\leq k}\left(\mathrm{error}_{\mathcal{D}}(c_j, h_j)\right) > \alpha\right] \leq \beta.$$

The probability is taken over the random choice of the examples in S according to $\mathcal{D}$ and the coin tosses of the learner $\mathcal{A}$. If $H \subseteq C$ then A is called a *proper* learner; otherwise, it is called an *improper* learner.

Definition 2.4 (Agnostic PAC Multi-Learner). Algorithm $\mathcal{A}$ is an (α, β)-PAC agnostic k-learner for C using hypothesis class H and sample complexity n if for every distribution $\mathcal{P}$ over $X \times \{0,1\}^k$, given a k-labeled database $S = ((x_i, y_{1,i}, \ldots, y_{k,i}))_{i=1}^n$ where each k-labeled sample $(x_i, y_{1,i} \ldots, y_{k,i})$ is drawn i.i.d. from $\mathcal{P}$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1, \ldots, h_k)$ from H satisfying

$$\Pr\left[\max_{1\leq j\leq k}\left(\mathrm{error}_{\mathcal{P}_j}(h_j) - \min_{c\in C}\left(\mathrm{error}_{\mathcal{P}_j}(c)\right)\right) > \alpha\right] \leq \beta,$$

where $\mathcal{P}_j$ is the marginal distribution of $\mathcal{P}$ on the examples and the jth label. The probability is taken over the random choice of the examples in S according to $\mathcal{P}$ and the coin tosses of the learner $\mathcal{A}$. If $H \subseteq C$ then A is called a *proper* learner; otherwise, it is called an *improper* learner.

2.2 The Sample Complexity of Multi-Learning

Without privacy considerations, the sample complexities of PAC and agnostic learning are essentially characterized by a combinatorial quantity called the *Vapnik-Chervonenkis (VC) dimension*. We state these characterizations in the context of multi-learning.

2.2.1 The Vapnik-Chervonenkis Dimension

Definition 2.5. Fix a concept class C over domain X. A set $\{x_1, \ldots, x_d\} \in X$ is *shattered* by C if for every labeling $b \in \{0,1\}^d$, there exists $c \in C$ such that $b_1 = c(x_1), \ldots, b_d = c(x_d)$. The *Vapnik-Chervonenkis (VC) dimension* of C, denoted $\text{VC}(C)$, is the size of the largest set which is shattered by C.

The Vapnik-Chervonenkis (VC) dimension is an important combinatorial measure of a concept class. Classical results in statistical learning theory show that the generalization error of a hypothesis h and its empirical error (observed on a large enough sample) are similar.

Definition 2.6 (Empirical Error). Let $S = ((x_i, y_i))_{i=1}^n \in (X \times \{0,1\})^n$ be a labeled sample from X. The *empirical error* of a hypothesis $h : X \to \{0,1\}$ w.r.t. S is defined as $\text{error}_S(h) = \frac{1}{n}|\{i : h(x_i) \neq y_i\}|$.

Let $D \in X^n$ be a (unlabeled) sample from X and let $c : x \to \{0,1\}$ be a concept. The *empirical error* of hypothesis $h : X \to \{0,1\}$ w.r.t. c and D is defined as $\text{error}_D(c, h) = \frac{1}{n}|\{i : h(x_i) \neq c(x_i)]|$.

Theorem 2.7 (VC-Dimension Generalization Bound, e.g. [11]). *Let $\mathcal{D}$ and C be, respectively, a distribution and a concept class over a domain X, and let $c \in C$. For a sample $S = ((x_i, c(x_i)))_{i=1}^n$ where $n \geq \frac{64}{\alpha}(\text{VC}(C)\ln(\frac{64}{\alpha}) + \ln(\frac{8}{\beta}))$ and the x_i are drawn i.i.d. from $\mathcal{D}$, it holds that*

$$\Pr\left[\exists h \in C \text{ s.t. } \text{error}_{\mathcal{D}}(h, c) > \alpha \ \wedge \ \text{error}_S(h) \leq \frac{\alpha}{2}\right] \leq \beta.$$

This generalization argument extends to the setting of *agnostic learning*, where a hypothesis with small empirical error might not exist.

Theorem 2.8 (VC-Dimension Agnostic Generalization Bound, e.g. [1, 2]). *Let H be a concept class over a domain X, and let $\mathcal{P}$ be a distribution over $X \times \{0,1\}$. For a sample $S = ((x_i, y_i))_{i=1}^n$ containing $n \geq \frac{64}{\alpha^2}(\text{VC}(H)\ln(\frac{6}{\alpha}) + \ln(\frac{8}{\beta}))$ i.i.d. elements from $\mathcal{P}$, it holds that*

$$\Pr\left[\exists h \in H \text{ s.t. } \left|\text{error}_{\mathcal{P}}(h) - \text{error}_S(h)\right| > \alpha\right] \leq \beta.$$

Using theorems 2.11 and 2.8, an upper bound of $O(\text{VC}(C))$ on the sample complexity of learning a concept class C follows by reduction to the *empirical learning* problem. The goal of empirical learning is similar to that of PAC learning, except accuracy is measured only with respect to a fixed input database. Theorems 2.11 and 2.8 state that when an empirical learner is run on sufficiently many samples, it is also accurate with respect to a distribution on inputs.

Definition 2.9 (Empirical Learner). Algorithm $\mathcal{A}$ is an (α, β)-*accurate empirical k-learner* for a concept class C using hypothesis class H with sample complexity n if for every collection of concepts $(c_1, \ldots, c_k)$

from C and database $S = ((x_i, c_1(x_i), \ldots, c_k(x_i)))_{i=1}^n \in (X \times \{0,1\}^k)^n$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1, \ldots, h_k)$ from H satisfying

$$\Pr\left[\max_{1 \leq j \leq k}\left(\text{error}_{S|_j}(h_j)\right) > \alpha\right] \leq \beta,$$

where $S|_j = ((x_i, c_j(x_i)))_{i=1}^n$. The probability is taken over the coin tosses of $\mathcal{A}$.

Definition 2.10 (Agnostic Empirical Learner). Algorithm $\mathcal{A}$ is an *agnostic (α, β)-accurate empirical k-learner* for a concept class C using hypothesis class H with sample complexity n if for every database $S = ((x_i, y_{1,i}, \ldots, y_{k,i}))_{i=1}^n \in (X \times \{0,1\}^k)^n$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1, \ldots, h_k)$ from H satisfying

$$\Pr\left[\max_{1 \leq j \leq k}\left(\text{error}_{S|_j}(h_j) - \min_{c \in C}\left(\text{error}_{S|_j}(c)\right)\right) > \alpha\right] \leq \beta,$$

where $S|_j = ((x_i, y_{j,i}))_{i=1}^n$. The probability is taken over the coin tosses of $\mathcal{A}$.

Theorem 2.11. *Let $\mathcal{A}$ be an (α, β)-accurate empirical k-learner for a concept class C (resp. agnostic empirical k-learner) using hypothesis class H. Then $\mathcal{A}$ is also a $(2\alpha, \beta + \beta')$-accurate PAC learner for C when given at least $\max\{n, \frac{32}{\alpha}(\text{VC}(H \oplus C)\log(32/\alpha) + \log(8/\beta'))\}$ samples (resp. $\max\{n, \frac{64}{\alpha^2}(\text{VC}(H)\log(6/\alpha) + \log(8k/\beta'))$ samples). Here, $H \oplus C = \{h \oplus c : h \in H, c \in C\}$.*

Proof. We begin with the non-agnostic case. Let $\mathcal{A}$ be an (α, β)-accurate empirical k-learner for C. Let $\mathcal{D}$ be a distribution over the example space X. Let S be a random i.i.d. sample of size m from $\mathcal{D}$. The generalization bound for PAC learning (Theorem 2.7) states that if $m \geq \frac{32}{\alpha}(d\log(32/\alpha) + \log(8/\beta'))\}$, then

$$\Pr[\exists c \in C, h \in H : \text{error}_S(c, h) \leq \alpha \wedge \text{error}_{\mathcal{D}}(c, h) > 2\alpha] \leq \beta',$$

where $d = \text{VC}(H \oplus C)$. The result follows by a union bound over the failure probability of $\mathcal{A}$ and the failure of generalization.

Now we turn to the agnostic case. Let $\mathcal{A}$ be an agnostic (α, β)-accurate empirical k-learner for C. Fix an index $j \in [k]$, and let $\mathcal{P}_j$ be a distribution over $X \times \{0,1\}$. Let S be a random i.i.d. sample of size m from $\mathcal{P}_j$. Then generalization for agnostic learning (Theorem 2.8) yields

$$\Pr[\exists h \in H : |\text{error}_S(h) - \text{error}_{\mathcal{P}_j}(h)| > \alpha] \leq \frac{\beta'}{k}$$

for $m \geq \frac{64}{\alpha^2}(\text{VC}(H)\log(6/\alpha) + \log(8k/\beta'))\}$. The result follows by a union bound over the failure probability of $\mathcal{A}$ and the failure of generalization for each of the indices $j = 1, \ldots, k$. $\square$

Applying the above theorem in the special case where $\mathcal{A}$ finds the concept $c \in C$ that minimizes the empirical error on its given sample, we obtain the following sample complexity upper bound for proper multi-learning.

Corollary 2.12. *Let C be a concept class with VC dimension d. There exists an (α, β)-accurate proper PAC k-learner for C using $O(\frac{1}{\alpha}(d\log(1/\alpha) + \log(1/\beta))$ samples. Moreover, there exists an (α, β)-accurate proper agnostic PAC k-learner for C using $O(\frac{1}{\alpha^2}(d\log(1/\alpha) + \log(k/\beta))$ samples.*

Proof. For the non-agnostic case, we simply let $\mathcal{A}$ be the $(0,0)$-accurate empirical learner that outputs any vector of hypotheses that is consistent with its given examples (one is guaranteed to exist, since the target concept satisfies this condition). The claim follows from Theorem 2.11 noting that $\text{VC}(C \oplus C) = O(\text{VC}(C))$.

For the agnostic case, consider the algorithm $\mathcal{A}$ that on input S outputs hypotheses $(h_1, \ldots, h_k)$ that minimize the quantities $\text{error}_{S_j}(h_j)$. Applying the agnostic generalization bound [1], this is an $(\alpha/2, \beta/2)$-accurate agnostic empirical learner given $O(\frac{1}{\alpha^2}(d \log(1/\alpha) + \log(k/\beta)))$ samples. The claim then follows from Theorem 2.11. $\qquad\square$

It is known that even for $k = 1$, the sample complexities of PAC and agnostic learning are at least $\Omega(\text{VC}(C)/\alpha)$ and $\Omega(\text{VC}(C)/\alpha^2)$, respectively. Therefore, the above sample complexity upper bound is tight up to logarithmic factors.

We define a few specific concept classes which will play an important role in this work.

$\texttt{POINT}_X$**:** Let X be any domain. The class of *point functions* is the set of all concepts that evaluate to 1 on exactly one element of X, i.e. $\texttt{POINT}_X = \{c_x : x \in X\}$ where $c_x(y) = 1$ iff $y = x$. The VC-dimension of $\texttt{POINT}_X$ is 1 for any X.

$\texttt{THRESH}_X$**:** Let X be any totally ordered domain. The class of *threshold functions* takes the form $\texttt{THRESH}_X = \{c_x : x \in X\}$ where $c_x(y) = 1$ iff $y \leq x$. The VC-dimension of $\texttt{THRESH}_X$ is 1 for any X.

$\texttt{PAR}_d$**:** Let $X = \{0, 1\}^d$. The class of *parity functions* on X is given by $\texttt{PAR}_d = \{c_x : x \in X\}$ where $c_x(y) = \langle x, y \rangle \pmod 2$. The VC-dimension of $\texttt{PAR}_d$ is d.

In this work, we focus our study of the concept class $\texttt{PAR}_d$ on the problem of learning parities under the uniform distribution. The PAC and agnostic learning problems are defined as before, except we only require a learner to be accurate when the marginal distribution on examples is the uniform distribution U_d over $\{0, 1\}^d$.

Definition 2.13 (PAC Learning $\texttt{PAR}_d$ under Uniform)**.** Algorithm $\mathcal{A}$ is an (α, β)-PAC k-learner for $\texttt{PAR}_d$ using hypothesis class H and sample complexity n if for every fixed $(c_1, \ldots, c_k)$ from C, given a k-labeled database as an input $S = ((x_i, c_1(x_i), \ldots, c_k(x_i)))_{i=1}^n$ where each x_i is drawn i.i.d. from U_d, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1, \ldots, h_k)$ from H satisfying

$$\Pr\left[\max_{1 \leq j \leq k}\left(\text{error}_{U_d}(c_j, h_j)\right) > \alpha\right] \leq \beta.$$

Definition 2.14 (Agnostically Learning $\texttt{PAR}_d$ under Uniform)**.** Algorithm $\mathcal{A}$ is an (α, β)-PAC agnostic k-learner for $\texttt{PAR}_d$ using hypothesis class H and sample complexity n if for every distribution $\mathcal{P}$ over $\{0, 1\}^d \times \{0, 1\}^k$, with marginal distribution U_d over the data universe $\{0, 1\}^d$, given a k-labeled database $S = ((x_i, y_{1,i}, \ldots, y_{k,i}))_{i=1}^n$ where each k-labeled sample $(x_i, y_{1,i} \ldots, y_{k,i})$ is drawn i.i.d. from $\mathcal{P}$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1, \ldots, h_k)$ from H satisfying

$$\Pr\left[\max_{1 \leq j \leq k}\left(\text{error}_{\mathcal{P}_j}(h_j) - \min_{c \in C}\left(\text{error}_{\mathcal{P}_j}(c)\right)\right) > \alpha\right] \leq \beta,$$

where $\mathcal{P}_j$ is the marginal distribution of $\mathcal{P}$ on the examples and the jth label.

2.3 Differential privacy

Two k-*labeled* databases $S, S' \in (X \times \{0, 1\}^k)^n$ are called *neighboring* if they differ on a single (multi-labeled) entry, i.e., $|\{i : (x_i, y_{1,i}, \ldots, y_{k,i}) \neq (x'_i, y'_{1,i}, \ldots, y'_{k,i})\}| = 1$.

Definition 2.15 (Differential Privacy [18]). Let $\mathcal{A} : \left(X \times \{0,1\}^k \right)^n \to \left(2^X \right)^k$ be an algorithm that operates on a k-labeled database and returns k hypotheses. Let $\epsilon, \delta \geq 0$. Algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all neighboring S, S' and for all $T \subseteq \left(2^X \right)^k$,

$$\Pr[\mathcal{A}(S) \in T] \leq e^\epsilon \cdot \Pr[\mathcal{A}(S') \in T] + \delta,$$

where the probability is taken over the coin tosses of the algorithm $\mathcal{A}$. When $\delta = 0$ we say that $\mathcal{A}$ satisfies *pure* differential privacy, otherwise (i.e., if $\delta > 0$) we say that $\mathcal{A}$ satisfies *approximate* differential privacy.

Our learning algorithms are designed via repeated applications of differentially private algorithms on a database. Composition theorems for differential privacy show that the price of privacy for multiple (adaptively chosen) interactions degrades gracefully.

Theorem 2.16 (Composition of Differential Privacy [16, 17, 19]). *Let $0 < \epsilon, \delta' < 1$ and $\delta \in [0,1]$. Suppose an algorithm $\mathcal{A}$ accesses its input database S only through m adaptively chosen executions of (ϵ, δ)-differentially private algorithms. Then $\mathcal{A}$ is*

1. *$(m\epsilon, m\delta)$-differentially private, and*

2. *$(\epsilon', m\delta + \delta')$-differentially private for $\epsilon = \sqrt{2m \ln(1/\delta')} \cdot \epsilon + 2m\epsilon^2$.*

2.4   Differentially Private Sanitization

A fundamental task in differential privacy is the *data sanitization* problem. Given a database $D = (x_1, \ldots, x_n) \in X^n$, the goal of a sanitizer is to privately produce a synthetic database $\hat{D} \in X^m$ that captures the statistical properties of D. We are primarily interested in sanitization for boolean-valued functions (equivalently referred to as *counting queries*). Given a function $c : X \to \{0,1\}$ and a database $D = (x_1, \ldots, x_n)$, we write $c(D) = \frac{1}{n} \sum_{i=1}^n c(x_i)$.

Definition 2.17 (Sanitization). An algorithm $\mathcal{A} : X^n \to X^m$ is an (α, β)-accurate sanitizer for a concept class C if for every $D \in X^n$, the algorithm $\mathcal{A}$ produces a database $\hat{D} \in X^m$ such that

$$\Pr[\exists c \in C : |c(D) - c(\hat{D})| > \alpha] \leq \beta.$$

Here, the probability is taken over the coins of $\mathcal{A}$.

In an influential result, Blum, Ligett, and Roth [10] showed that any concept class C admits a differentially private sanitizer with sample complexity $O(\mathrm{VC}(C) \log |X|)$:

Theorem 2.18 ([10]). *For any concept class C over a domain X, there exists an (α, β)-accurate and $(\epsilon, 0)$-differentially private sanitizer $\mathcal{A} : X^n \to X^m$ for C when*

$$n = O\left(\frac{\mathrm{VC}(C) \cdot \log |X| \cdot \log(1/\alpha)}{\alpha^3 \epsilon} + \frac{\log(1/\beta)}{\alpha\epsilon} \right),$$

and $m = O(\mathrm{VC}(C) \log(1/\alpha)/\alpha^2)$.

When relaxing to (ϵ, δ)-differential privacy, the private multiplicative weights algorithm of Hardt and Rothblum [24] can sometimes achieve lower sample complexity (roughly $O(\log |C| \sqrt{\log |X|})$).

Theorem 2.19 ([24])**.** *For any concept class C over a domain X, there exists an (α, β)-accurate and (ϵ, δ)-differentially private sanitizer $\mathcal{A} : X^n \to X^m$ for C when*

$$n = O\left(\frac{(\log|C| + \log(1/\beta)) \cdot \sqrt{\log|X| \cdot \log(1/\delta)}}{\alpha^2 \epsilon}\right),$$

and $m = O(\mathrm{VC}(C) \log(1/\alpha)/\alpha^2)$.

However, for specific concept classes, sanitizers are known to exist with much lower sample complexity. We first give a sanitizer for point functions with essentially optimal sample complexity, which improves and simplifies a result of [7].

Proposition 2.20. *There exists an (α, β)-accurate and (ϵ, δ)-differentially private sanitizer for POINT_X with sample complexity*

$$n = O\left(\frac{\log(1/\alpha\beta\delta)}{\alpha\epsilon}\right).$$

Proof. To give a $(2\alpha, \beta)$-accurate sanitizer, it suffices to produces, for each point function c_x, an approximate answer $a_x \in [0, 1]$ with $|a_x - c_x| \leq \alpha$. This is because given these approximate answers, one can reconstruct a database $\hat{D}$ of size $O(1/\alpha)$ with $|c_x(\hat{D}) - a_x| \leq \alpha$ for every $x \in X$.

The algorithm for producing the answers a_x is as follows.

Algorithm 1 Query release for POINT_X

Input: Privacy parameters (ϵ, δ), database $D \in X^n$

For each $x \in X$, do the following:

1. If $c_x(D) \leq \frac{\alpha}{4}$, release $a_x = 0$

2. Let $\hat{a}_x = c_x(D) + \mathrm{Lap}(2/\epsilon n)$

3. If $\hat{a}_x \leq \frac{\alpha}{2}$, release $a_x = 0$

4. Otherwise, release $a_x = \hat{a}_x$

First, we argue that Algorithm 1 is (ϵ, δ)-differentially private. Below, we write $X \approx_{(\epsilon,\delta)} Y$ to denote the fact that for every measurable set S in the union of the supports of X and Y, we have $\Pr[X \in S] \leq e^\epsilon \Pr[Y \in S] + \delta$.

Let $D \sim D'$ be adjacent databases of size n, with $x \in D$ replaced by $x' \in D'$. Then the output distribution of the mechanism differs only on its answers to the queries c_x and $c_{x'}$. Let us focus on c_x. If both $c_x(D) \leq \alpha/4$ and $c_x(D') \leq \alpha/4$, then the mechanism always releases 0 for both queries. If both $c_x(D) > \alpha/4$ and $c_x(D') > \alpha/4$, then $a_x(D) \approx_{(\epsilon/2,0)} a_x(D')$ by properties of the Laplace mechanism. Finally, if $c_x(D) > \alpha/4$ but $c_x(D') \leq \alpha/4$, then $c_x(D') = 0$ with probability 1. Moreover, we must have $\mathrm{POINT}_x(D) \leq \alpha/4 + 1/n$, so

$$\Pr[a_x(D) = 0] \geq \Pr[Lap(2/\epsilon n) < \alpha/4 - 1/n] = 1 - \frac{1}{2}\exp(-\epsilon n \alpha/8 + \epsilon/2) \geq 1 - \delta/2.$$

So in this case, $a_x(D) \approx_{(0,\delta/2)} a_x(D')$. Therefore, we conclude that overall $a_x(D) \approx_{(\epsilon/2,\delta/2)} a_x(D')$. An identical argument holds for $a_{x'}$, so the mechanism is (ϵ, δ)-differentially private.

Now we argue that the answers a_x are accurate. First, the answers are trivially α-accurate for all queries c_x on which $c_x(D) \leq \alpha/4$. For each of the remaining queries, it is α-accurate with probability at least

$$\Pr[|Lap(2/\epsilon n)| < \alpha/2] = 1 - \exp(-\epsilon n\alpha/4) \geq 1 - \frac{\alpha\beta}{4}.$$

Taking a union bound over the at most $4/\alpha$ queries with $\mathtt{POINT}_x(D) > \alpha/4$, we conclude that the mechanism is α-accurate for all queries with probability at least $1 - \beta$.

$\square$

Bun et al. [13], improving on work of Beimel et al. [7], gave a sanitizer for threshold functions with sample complexity roughly $2^{\log^* |X|}$.

Proposition 2.21 ([13])**.** *There exists an (α, β)-accurate and (ϵ, δ)-differentially private sanitizer for $\mathtt{THRESH}_X$ with sample complexity*

$$n = O\left(\frac{1}{\alpha\epsilon} \cdot 2^{\log^* |X|} \cdot \log^* |X| \cdot \log\left(\frac{\log^* |X|}{\epsilon\delta}\right) \cdot \log(1/\beta) \cdot \log^{2.5}(1/\alpha)\right).$$

2.5 Private learners and multi-learners

Generalizing on the concept of private learners [26], we say that an algorithm $\mathcal{A}$ is $(\alpha, \beta, \epsilon, \delta)$-private PAC k-learner for C using H if $\mathcal{A}$ is (α, β)-PAC k-learner for C using H, and $\mathcal{A}$ is (ϵ, δ)-differentially private (similarly with agnostic private PAC k-learners). We omit the parameter k when $k = 1$ and the parameter δ when $\delta = 0$.

For the case $k = 1$, we have a generic construction with sample complexity proportional to $\log |C|$:

Theorem 2.22 ([26])**.** *Let C be a concept class, and $\alpha, \beta, \epsilon > 0$. There exists an $(\alpha, \beta, \epsilon)$-private agnostic proper learner for C with sample complexity $O\left((\log |C| + \log 1/\beta)(1/(\epsilon\alpha) + 1/\alpha^2)\right)$.*

Beimel, Nissim, and Stemmer [8] gave a generic transformation from data sanitization to private learning, which generally gives improved sample complexity upper bounds.

Theorem 2.23 ([8])**.** *Suppose there exists an (α, β)-accurate and (ϵ, δ)-differentially private sanitizer for $C^{\oplus}$ with sample complexity m. Then there exists a proper $(2\alpha, 2\beta)$-PAC and $(\epsilon + \epsilon', \delta)$-differentially private learner for C with sample complexity*

$$O\left(m + \frac{\mathrm{VC}(C)}{\alpha^3 \epsilon'} \log\left(\frac{1}{\alpha}\right) + \frac{1}{\alpha\epsilon'} \log\left(\frac{1}{\beta}\right)\right).$$

A number of works [5, 6, 7, 22, 13] have established sharper upper and lower bounds for learning the specific concept classes $\mathtt{POINT}_X$ and $\mathtt{THRESH}_X$. In the case of pure differential privacy, $\mathtt{POINT}_X$ requires $\Theta(\log |X|)$ samples to learn properly [5], but can be learned improperly with $O(1)$ samples. On the other hand, the class of threshold functions $\mathtt{THRESH}_X$ require $\Omega(\log |X|)$ samples to learn, even improperly [22]. In the case of approximate differential privacy, $\mathtt{POINT}_x$ and $\mathtt{THRESH}_x$ can be learned properly with sample complexities $O(1)$ [7] and $\tilde{O}(2^{\log^* |X|})$ [13], respectively. Moreover, properly learning threshold functions requires sample complexity $\Omega(\log^* |X|)$.

2.6 Private PAC learning vs. Empirical Learning

We saw by Theorem 2.11 that when an empirical k-learner $\mathcal{A}$ for a concept class C is run on a random sample of size $\Omega(\mathrm{VC}(C))$, it is also a (agnostic) PAC k-learner. In particular, if an empirical k-learner $\mathcal{A}$ is differentially private, then it also serves as a differentially private (agnostic) PAC k-learner.

Generalizing a result of [13], the next theorem shows that the converse is true as well: a differentially private (agnostic) PAC k-learner yields a private empirical k-learner with only a constant factor increase in the sample complexity.

Theorem 2.24. *Let $\epsilon \leq 1$. Suppose $\mathcal{A}$ is an (ϵ, δ)-differentially private (α, β)-accurate (agnostic) PAC k-learner for a concept class C with sample complexity n. Then there is an (ϵ, δ)-differentially private (α, β)-accurate (agnostic) empirical k-learner $\tilde{\mathcal{A}}$ for C with sample complexity $m = 9n$. Moreover, if $\mathcal{A}$ is proper, then so is the resulting empirical learner $\tilde{\mathcal{A}}$.*

Proof. We give the proof for the agnostic case; the non-agnostic case is argued identically, and is immediate from [13]. To construct the empirical learner $\tilde{\mathcal{A}}$, we use the fact that the given learner $\mathcal{A}$ performs well on any distribution over labeled examples – in particular, it performs well on the uniform distribution over rows of the input database to $\tilde{\mathcal{A}}$. Consider a database $S = ((x_i, y_{1,i}, \ldots, y_{k,i}))_{i=1}^m \in (X \times \{0,1\}^k)^m$. On input S, define $\tilde{\mathcal{A}}$ by sampling n rows from S (with replacement), and outputting the result of running $\mathcal{A}$ on the sample. Let $\mathcal{S}$ denote the uniform distribution over the rows of S, and let $\mathcal{S}_j$ be its marginal distribution which is uniform over $S|_j = ((x_i, y_{j,i}))_{i=1}^m$. Then sampling n rows from S is equivalent to sampling n rows i.i.d. from $\mathcal{S}$. Hence, if $(h_1, \ldots, h_k)$ is the output of $\mathcal{A}$ on the subsample, we have

$$\Pr\left[\max_{1 \leq j \leq k}\left(\mathrm{error}_{S|_j}(h_j) - \min_{c \in C}\left(\mathrm{error}_{S_j}(c)\right)\right) > \alpha\right] = \Pr\left[\max_{1 \leq j \leq k}\left(\mathrm{error}_{\mathcal{S}_j}(h_j) - \min_{c \in C}\left(\mathrm{error}_{\mathcal{S}_j}(c)\right)\right) > \alpha\right] \leq \beta.$$

To show that $\tilde{\mathcal{A}}$ remains (ϵ, δ)-differentially private, we apply the following "secrecy-of-the-sample" lemma [26, 13], which shows that the sampling procedure does not hurt privacy.

Lemma 2.25. *Fix $\epsilon \leq 1$ and let $\mathcal{A}$ be an (ϵ, δ)-differentially private algorithm with sample complexity n. For $m \geq 2n$, the algorithm $\tilde{\mathcal{A}}$ described above is $(\tilde{\epsilon}, \tilde{\delta})$ for*

$$\tilde{\epsilon} = \frac{6\epsilon m}{n} \quad \text{and} \quad \tilde{\delta} = 4\exp\left(\frac{6\epsilon m}{n}\right) \cdot \frac{m}{n} \cdot \delta.$$

$\square$

3 Upper Bounds on the Sample Complexity of Private Multi-Learners

3.1 Generic Construction

In this section we present the following general upper bounds on the sample complexity of private k-learners.

Theorem 3.1. *Let C be a finite concept class, and let $k \geq 1$. There exists a proper agnostic $(\alpha, \beta, \epsilon)$-private PAC k-learner for C with sample complexity*

$$O_{\alpha,\beta,\epsilon}\left(k \cdot \log k + \min\left\{k \cdot \log|C|, \ (k + \log|X|) \cdot \mathrm{VC}(C)\right\}\right),$$

and there exists a proper agnostic $(\alpha, \beta, \epsilon, \delta)$-private PAC k-learner for C with sample complexity

$$O_{\alpha,\beta,\epsilon,\delta}\left(\sqrt{k} \cdot \log k + \min\left\{\sqrt{k} \cdot \log|C|, \ (\sqrt{k} + \log|X|) \cdot \mathrm{VC}(C), \ \sqrt{k} \cdot \mathrm{VC}(C) + \sqrt{\log|X|} \cdot \log|C|\right\}\right).$$

The straightforward approach for constructing a private k-learner for a class C is to separately apply a (standard) private learner for C for each of the k target concepts. Using composition theorem 2.16 to argue the overall privacy guarantee of the resulting learner, we get the following observation.

Observation 3.2. *Let C be a concept class and let $k \geq 1$. If there is an $(\alpha, \beta, \epsilon, \delta)$-PAC learner for C with sample complexity n, then*

- *There is an $(\alpha, k\beta, k\epsilon, k\delta)$-PAC k-learner for C with sample complexity n.*
- *There is an $(\alpha, k\beta, O(\sqrt{k \log(\frac{1}{\delta})}\epsilon + k\epsilon^2), O(k\delta))$-PAC k-learner for C with sample complexity n.*

Moreover, if the initial learner is proper and/or agnostic, then so is the resulting learner.

In cases where sample efficient private PAC learners exist, it might be useful to apply Observation 3.2 in order to obtain a private k-learner. For example, Beimel et al. [5, 6] gave an improper agnostic $(\alpha, \beta, \epsilon)$-PAC learner for POINT_X with sample complexity $O_\alpha(\frac{1}{\epsilon} \log \frac{1}{\beta})$. Using Observation 3.2 yields the following corollary.

Corollary 3.3. *There exists an improper agnostic $(\alpha, \beta, \epsilon)$-PAC k-learner for POINT_X with sample complexity $O_{\alpha,\beta,\epsilon}(k \log k)$.*

For a general concept class C, we can use Observation 3.2 with the generic construction of Theorem 2.22, stating that for every concept class C there exists a private agnostic proper learner $\mathcal{A}$ that uses $O(\log |C|)$ labeled examples.

Corollary 3.4. *Let C be a concept class, and $\alpha, \beta, \epsilon > 0$. There exists an $(\alpha, \beta, \epsilon)$-private agnostic proper k-learner for C with sample complexity $O_{\alpha,\beta,\epsilon}(k \cdot \log |C| + k \cdot \log k)$. Moreover, there exists an $(\alpha, \beta, \epsilon, \delta)$-private agnostic proper k-learner for C with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(\sqrt{k} \cdot \log |C| + \sqrt{k} \cdot \log k)$.*

Example 3.5. *There exists a proper agnostic $(\alpha, \beta, \epsilon)$-PAC k-learner for PAR_d with sample complexity $O_{\alpha,\beta,\epsilon}(kd + k \log k)$.*

As we will see in Section 5, the bounds of Corollary 3.3 and Example 3.5 on the sample complexity of k-learning POINT_X and PAR_d are tight (up to logarithmic factors). That is, with pure-differential privacy, the direct sum gives (roughly) optimal bounds for improperly learning POINT_X, and for (properly or improperly) learning PAR_d. This is not the case for learning THRESH_X or for *properly* learning learning POINT_X.

In order to avoid the factor $k \log |C|$ (or $\sqrt{k} \log |C|$) in Corollary 3.4, we now show how an idea used in [8] (in the context of semi-supervised learning) can be used to construct sample efficient private k-learners. In particular, this construction will achieve tight bounds for learning THRESH_X and for properly learning learning POINT_X under pure-differential privacy.

Fix a concept class C, target concepts $c_1, \ldots, c_k \in C$, and a k-labeled database S (we use D to denote the unlabeled portion of S). For every $1 \leq j \leq k$, the goal is to identify a hypothesis $h_j \in C$ with low $\text{error}_D(c_j, h_j)$ (such a hypothesis also has good generalization). Beimel et al. [8] observed that given a sanitization $\hat{D}$ of D w.r.t. $C^\oplus = \{f \oplus g : f, g \in C\}$, for every $f, g \in C$ it holds that

$$\text{error}_D(f, g) = \frac{1}{|D|}|\{x \in D : (f \oplus g)(x) = 1\}| \approx \frac{1}{|\hat{D}|}|\{x \in \hat{D} : (f \oplus g)(x) = 1\}| = \text{error}_{\hat{D}}(f, g).$$

Hence, a hypothesis h with low $\text{error}_{\hat{D}}(h, c_j)$ also has low $\text{error}_D(h, c_j)$ and vice versa. Let H be the set of all dichotomies over $\hat{D}$ realized by C. Note that $\exists f_j^* \in H$ that agrees with c_j on $\hat{D}$, i.e., $\exists f_j^* \in H$ s.t. $\text{error}_{\hat{D}}(f_j^*, c_j) = 0$, and hence $\text{error}_D(f_j^*, c_j)$ is also low. The thing that works in our favor here is that H is small – at most $2^{|\hat{D}|} \leq 2^{\text{VC}(C)}$ – and hence choosing a hypothesis out of H is easy. Therefore, for every j we can use the exponential mechanism to identify a hypothesis $h_j \in H$ with low $\text{error}_D(h_j, c_j)$.

13

Lemma 3.6. *Let C be a concept class, and $\alpha, \beta, \epsilon, \delta > 0$. There exists an $(\alpha, \beta, \epsilon)$-private agnostic k-learner for C with sample complexity $O_{\alpha,\beta,\epsilon}(\mathrm{VC}(C) \cdot \log|X| + k \cdot \mathrm{VC}(C) + k \cdot \log k)$. Moreover, there exists an $(\alpha, \beta, \epsilon, \delta)$-private agnostic k-learner for C with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(\min\{\mathrm{VC}(C) \cdot \log|X|, \log|C| \cdot \sqrt{\log|X|}\} + \sqrt{k} \cdot \mathrm{VC}(C) + \sqrt{k} \cdot \log k)$.*

Lemma 3.6 follows from the following lemma.

Lemma 3.7. *Let $\epsilon' > 0$ and let $\mathcal{A}$ be an $(\frac{\alpha}{5}, \frac{\beta}{5})$-accurate (ϵ, δ)-private sanitizer for $C^\oplus$ with sample complexity m. Then there is an (α, β)-PAC agnostic k-learner for C with sample complexity*

$$O\left(m + \frac{\mathrm{VC}(C)}{\alpha^3 \epsilon'} \log(\frac{1}{\alpha}) + \frac{1}{\alpha \epsilon'} \log(\frac{k}{\beta}) + \frac{1}{\alpha^2} \mathrm{VC}(C) \log(\frac{k}{\alpha\beta}) \right).$$

Moreover, it is both $(\epsilon + k\epsilon', \delta)$ and $(\epsilon + \sqrt{2k\ln(1/\delta)}\epsilon' + 2k\epsilon'^2, 2\delta)$-differentially private.

Using Lemma 3.7 with the generic sanitizer of Theorem 2.18 or Theorem 2.19 results in Lemma 3.6.

An important building block of our generic learner is the exponential mechanism of McSherry and Talwar [27]. A quality function $q : X^* \times \mathcal{F} \to \mathbb{N}$ defines an *optimization problem* over the domain X and a finite solution set $\mathcal{F}$: Given a database $S \in X^*$, choose $f \in \mathcal{F}$ that (approximately) maximizes $q(S, f)$. The exponential mechanism solves such an optimization problem sampling a random $f \in \mathcal{F}$ with probability $\propto \exp\left(\epsilon \cdot q(S, f)/2\Delta q\right)$. Here, the sensitivity of a quality function, Δq, is the maximum over all $f \in \mathcal{F}$ of the sensitivity of the function $q(\cdot, f)$.

Proposition 3.8 (Properties of the Exponential Mechanism [27])**.**

1. *The exponential mechanism is $(\epsilon, 0)$-differentially private.*

2. *Let q be a quality function with sensitivity at most 1. Fix a database $S \in X^n$ and let $\mathrm{OPT} = \max_{f \in \mathcal{F}}\{q(S, f)\}$. Let $t > 0$. Then exponential mechanism outputs a solution f with $q(S, f) \leq \mathrm{OPT} - tn$ with probability at most $|\mathcal{F}| \cdot \exp(-\epsilon tn/2)$.*

Algorithm 2 *GenericLearner*

Input: Concept class C, privacy parameters $\epsilon', \epsilon, \delta$, and a k-labeled database $S = (x_i, y_{i,1}, \ldots, y_{i,k})_{i=1}^n$. We use $D = (x_i)_{i=1}^n$ to denote the unlabeled portion of S.

Used Algorithm: An $(\frac{\alpha}{5}, \frac{\beta}{5})$-accurate (ϵ, δ)-private sanitizer for $C^\oplus$ with sample complexity m.

1. Initialize $H = \emptyset$.

2. Construct an (ϵ, δ)-private sanitization $\widetilde{D}$ of D w.r.t. $C^\oplus$, where $|\widetilde{D}| = O\left(\frac{\mathrm{VC}(C^\oplus)}{\alpha^2} \log(\frac{1}{\alpha})\right) = O\left(\frac{\mathrm{VC}(C)}{\alpha^2} \log(\frac{1}{\alpha})\right)$.

3. Let $B = \{b_1, \ldots, b_{|B|}\}$ be the set of all points appearing at least once in $\widetilde{D}$.

4. For every $(z_1, \ldots, z_{|B|}) \in \Pi_C(B) = \{(c(b_1), \ldots, c(b_{|B|})) : c \in C\}$, add to H an arbitrary concept $c \in C$ s.t. $c(b_\ell) = z_\ell$ for every $1 \leq \ell \leq |B|$.

5. For every $1 \leq j \leq k$, use the exponential mechanism with privacy parameter ϵ' to choose and return a hypothesis $h_j \in H$ with (approximately) minimal error on the examples in S w.r.t. their j^{th} label.

Proof of Lemma 3.7. The proof is via the construction of *GenericLearner* (algorithm 2). Note that *GenericLearner* only accesses S via a sanitizer (on Step 2) and using the exponential mechanism (on Step 5). Composition theorem 2.16 state that *GenericLearner* is both $(\epsilon + k\epsilon', \delta)$-differentially private and $(\epsilon + \sqrt{2k\ln(1/\delta)}\epsilon' + 2k\epsilon'^2, 2\delta)$-differentially private. We, thus, only need to prove that with high probability the learner returns α-good hypotheses.

Fix a distribution $\mathcal{P}$ over $X \times \{0,1\}^k$, and let $\mathcal{P}_j$ denote the marginal distribution of $\mathcal{P}$ on the examples and the jth label. Let S consist of examples $(x_i, y_{i,1}, \ldots, y_{i,k}) \sim \mathcal{P}$. We use $D = (x_i)_{i=1}^n$ to denote the unlabeled portion of S, and use $S|_j = ((x_i, y_{j,i}))_{i=1}^n$ to denote a database containing the examples in S together with their j^{th} label. Define the following three events:

E_1 : For every $f, h \in C$ it holds that $|\text{error}_D(f, h) - \text{error}_{\widetilde{D}}(f, h)| \leq \frac{2\alpha}{5}$.

E_2 : For every $f \in C$ and for every $1 \leq j \leq k$ it holds that $|\text{error}_{S|_j}(f) - \text{error}_{\mathcal{P}_j}(f)| \leq \frac{\alpha}{5}$.

E_3 : For every $1 \leq j \leq k$, the hypothesis h_j chosen by the exponential mechanism is such that $\text{error}_{S|_j}(h_j) \leq \frac{\alpha}{5} + \min_{f \in H}\left\{\text{error}_{S|_j}(f)\right\}$.

We first argue that when these three events happen algorithm *GenericLearner* returns good hypotheses. Fix $1 \leq j \leq k$, and let $c_j^* = \operatorname{argmin}_{f \in C}\{\text{error}_{\mathcal{P}_j}(f)\}$. We denote $\Delta = \text{error}_{\mathcal{P}_j}(c_j^*)$. We need to show that if $E_1 \cap E_2 \cap E_3$ occurs, then the hypothesis h_j returned by *GenericLearner* is s.t. $\text{error}_{\mathcal{P}_j}(h_j) \leq \alpha + \Delta$.

For every $(y_1, \ldots, y_{|B|}) \in \Pi_C(B)$, algorithm *GenericLearner* adds to H a hypothesis f s.t. $\forall 1 \leq \ell \leq |B|$, $f(b_\ell) = y_\ell$. In particular, H contains a hypothesis h_j^* s.t. $h_j^*(x) = c_j^*(x)$ for every $x \in B$, that is, a hypothesis h_j^* s.t. $\text{error}_{\widetilde{D}}(h_j^*, c_j^*) = 0$. As event E_1 has occurred we have that this h_j^* satisfies $\text{error}_D(h_j^*, c_j^*) \leq \frac{2\alpha}{5}$. Using the triangle inequality (and event E_2) we get that this h_j^* satisfies $\text{error}_{S|_j}(h_j^*) \leq \text{error}_D(h_j^*, c_j^*) + \text{error}_{S|_j}(c_j^*) \leq \frac{3\alpha}{5} + \Delta$. Thus, event E_3 ensures that algorithm *GenericLearner* chooses (using the exponential mechanism) a hypothesis $h_j \in H$ s.t. $\text{error}_{S|_j}(h_j) \leq \frac{4\alpha}{5} + \Delta$. Event E_2 ensures, therefore, that this h_j satisfies $\text{error}_{\mathcal{P}_j}(h_j) \leq \alpha + \Delta$. We will now show $E_1 \cap E_2 \cap E_3$ happens with high probability.

Standard arguments in learning theory state that (w.h.p.) the empirical error on a (large enough) random sample is close to the generalization error. Specifically, by setting $n \geq O(\frac{1}{\alpha^2}\text{VC}(C)\log(\frac{k}{\alpha\beta}))$, Theorem 2.8 ensures that Event E_2 occurs with probability at least $(1 - \frac{2}{5}\beta)$.

Assuming that $n \geq m$ (the sample complexity of the sanitizer used in Step 5), with probability at least $(1 - \frac{\beta}{5})$ for every $(h \oplus f) \in C^{\oplus}$ (i.e., for every $h, f \in C$) it holds that

$$
\begin{aligned}
\frac{\alpha}{5} &\geq |Q_{(h\oplus f)}(D) - Q_{(h\oplus f)}(\widetilde{D})| \\
&= \left| \frac{|\{x \in D : (h\oplus f)(x)=1\}|}{|D|} - \frac{|\{x \in \widetilde{D} : (h\oplus f)(x)=1\}|}{|\widetilde{D}|} \right| \\
&= \left| \frac{|\{x \in D : h(x)\neq f(x)\}|}{|D|} - \frac{|\{x \in \widetilde{D} : h(x)\neq f(x)\}|}{|\widetilde{D}|} \right| \\
&= \left| \text{error}_D(h, f) - \text{error}_{\widetilde{D}}(h, f) \right|.
\end{aligned}
$$

Event E_1 occurs therefore with probability at least $(1 - \frac{\beta}{5})$.

The exponential mechanism ensures that the probability of event E_3 is at least $1 - k|H| \cdot \exp(-\epsilon'\alpha m/10)$ (see Proposition 3.8). Note that $\log|H| \leq |B| \leq |\widetilde{D}| = O\left(\frac{\text{VC}(C)}{\alpha^2}\log(\frac{1}{\alpha})\right)$. Therefore, for $n \geq O\left(\frac{\text{VC}(C)}{\alpha^3\epsilon'}\log(\frac{1}{\alpha}) + \frac{1}{\alpha\epsilon'}\log(\frac{k}{\beta})\right)$, Event E_3 occurs with probability at least $(1 - \frac{\beta}{5})$.

15

All in all, setting $n \geq O\left(m + \frac{\mathrm{VC}(C)}{\alpha^3 \epsilon'} \log(\frac{1}{\alpha}) + \frac{1}{\alpha \epsilon'} \log(\frac{k}{\beta}) + \frac{1}{\alpha^2} \mathrm{VC}(C) \log(\frac{k}{\alpha \beta})\right)$, ensures that the probability of *GenericLearner* failing is at most β. □

Theorem 3.1 now follows by combining Lemma 3.6 and Corollary 3.4.

For certain concept classes, there are sanitizers with substantially lower sample complexity than the generic sanitizers. Combining Lemma 3.6 with Proposition 2.20, we obtain:

Corollary 3.9. *There is an (α, β)-PAC agnostic k-learner for* $\texttt{POINT}_X$ *with sample complexity*

$$O\left(\frac{\log(1/\alpha \beta \delta)}{\alpha \epsilon} + \frac{\log(1/\alpha)}{\alpha^3 \epsilon'} + \frac{\log(k/\beta)}{\alpha \epsilon'} + \frac{\log(k/\alpha \beta)}{\alpha^2}\right).$$

Moreover, it is both $(\epsilon + k\epsilon', \delta)$ and $(\epsilon + \sqrt{2k \ln(1/\delta)}\epsilon' + 2k\epsilon'^2, 2\delta)$-differentially private.

Similarly, combining Lemma 3.6 with Proposition 2.21, we obtain:

Corollary 3.10. *There is an (α, β)-PAC agnostic k-learner for* $\texttt{THRESH}_X$ *with sample complexity*

$$O\left(\frac{2^{\log^* |X|} \cdot \log^* |X| \cdot \log\left(\frac{\log^* |X|}{\epsilon \delta}\right) \cdot \log(1/\beta) \cdot \log^{2.5}(1/\alpha)}{\alpha \epsilon} + \frac{\log(1/\alpha)}{\alpha^3 \epsilon'} + \frac{\log(k/\beta)}{\alpha \epsilon'} + \frac{\log(k/\alpha \beta)}{\alpha^2}\right).$$

Moreover, it is both $(\epsilon + k\epsilon', \delta)$ and $(\epsilon + \sqrt{2k \ln(1/\delta)}\epsilon' + 2k\epsilon'^2, 2\delta)$-differentially private.

3.2 Upper Bounds for Approximate Private Multi-Learners

In this section we give two examples of cases where the sample complexity of private k-learning is of the same order as that of non-private k-learning (the sample complexity does not depend on k). Our algorithms are (ϵ, δ)-differentially private, and rely on stability arguments: the identity of the best k concepts, as an entire vector, is unlikely to change on nearby k-labeled databases. Hence, it can be released privately.

The main technical tool we use is the $\mathcal{A}_{\mathrm{dist}}$ algorithm of Smith and Thakurta [31]. Our discussion follows the treatment of [7].

Recall that a quality function $q : X^* \times \mathcal{F} \to \mathbb{N}$ defines an *optimization problem* over the domain X and a finite solution set $\mathcal{F}$: Given a database $S \in X^*$, find $f \in \mathcal{F}$ that (approximately) maximizes $q(S, f)$. The sensitivity of a quality function, Δq, is the maximum over all $f \in \mathcal{F}$ of the sensitivity of the function $q(\cdot, f)$. The algorithm $\mathcal{A}_{\mathrm{dist}}$ privately identifies the *exact* maximizer as long as it is sufficiently stable.

Algorithm 3 $\mathcal{A}_{\mathrm{dist}}$

Input: Privacy parameters ϵ, δ, database $S \in X^*$, sensitivity-1 quality function q

1. Let $f_1, f_2 \in \mathcal{F}$ be the highest scoring and second-highest scoring solutions to $q(S, \cdot)$, respectively.

2. Let $\mathrm{gap} = q(S, f_1) - q(S, f_2)$, and $\widehat{\mathrm{gap}} = \mathrm{gap} + \mathrm{Lap}(1/\epsilon)$.

3. If $\widehat{\mathrm{gap}} < \frac{1}{\epsilon} \log \frac{1}{\delta}$, output $\perp$. Otherwise, output f_1.

Proposition 3.11 (Properties of $\mathcal{A}_{\mathrm{dist}}$ [31])**.**

1. *Algorithm $\mathcal{A}_{\mathrm{dist}}$ is (ϵ, δ)-differentially private.*

2. *When run on a database S with $\mathrm{gap} > \frac{1}{\epsilon} \log \frac{1}{\delta \beta}$, Algorithm $\mathcal{A}_{\mathrm{dist}}$ outputs the highest scoring solution f_1 with probability at least $1 - \beta$.*

3.2.1 Learning Parities under the Uniform Distribution

Theorem 3.12. *For every k, d there exists an $(\alpha{=}0, \beta, \epsilon, \delta)$-PAC (non-agnostic) k-learner for $\mathtt{PAR}_d$ under the uniform distribution with sample complexity $O(d \log(\frac{1}{\beta}) + \frac{1}{\epsilon} \log(\frac{1}{\beta\delta}))$.*

Recall that (even without privacy constraints) the sample complexity of PAC learning $\mathtt{PAR}_d$ under the uniform distribution is $\Omega(d)$. Hence the sample complexity of privately k-learning $\mathtt{PAR}_d$ (non-agnostically) under the uniform distribution is of the same order as that of non-private k-learning.

For the intuition behind Theorem 3.12, let $c_1, \ldots, c_k$ denote the k target concepts, and consider the quality function $q(D, (h_1, \ldots, h_k)) = \max_{1 \leq j \leq k} \{\mathrm{error}_D(h_j, c_j)\}$. On a large enough sample D we expect that $q(D, (h_1, \ldots, h_k)) \approx \frac{1}{2}$ for every $(h_1, \ldots, h_k) \neq (c_1, \ldots, c_k)$, while $q(D, (c_1, \ldots, c_k)) = 0$. The k target concepts can hence be privately identified (exactly) using stability techniques.

In order to make our algorithm computationally efficient, we apply the "subsample and aggregate" idea of Nissim et al. [28]. We divide the input sample into a small number of subsamples, use Gaussian elimination to (non-privately) identify a candidate hypothesis vector on each subsample, and then select from these candidates privately.

Algorithm 4 *ParityLearner*

Input: Parameters ϵ, δ, and a k-labeled database S of size $n = O(\frac{d}{\epsilon} \log(\frac{1}{\beta\delta}))$.
Output: Hypotheses $h_1, \ldots, h_k$.

1. Split S into $m = O(\frac{1}{\epsilon} \log(\frac{1}{\beta\delta}))$ disjoint samples $S_1, \ldots, S_m$ of size $O(d)$ each. Initiate Y as the empty multiset.

2. For every $1 \leq t \leq m$:

   (a) For every $1 \leq j \leq k$ try to use Gaussian elimination to identify a parity function y_j that agrees with the labels of the jth column of S_t.

   (b) If a parity is identified for every j, then set $Y = Y \cup \{(y_1, ..., y_k)\}$. Otherwise set $Y = Y \cup \{\perp\}$.

3. Use algorithm $\mathcal{A}_{\mathrm{dist}}$ with privacy parameters ϵ, δ to choose and return a vector of k parity functions $(h_1, \ldots, h_k) \in (\mathtt{PAR}_d)^k$ with a large number of appearances in Y.

Proof of Theorem 3.12. The proof is via the construction of *ParityLearner* (algorithm 4). First note that changing a single input element in S can change (at most) one element of Y. Hence, applying (the (ϵ, δ)-private) algorithm $\mathcal{A}_{\mathrm{dist}}$ on Y preserves privacy (applying *ParityLearner* on neighboring inputs amounts to executing $\mathcal{A}_{\mathrm{dist}}$ on neighboring inputs).

Now fix k target concepts $c_1, \ldots, c_k \in \mathtt{PAR}_d$ and let S be a random k-labeled database containing n i.i.d. elements from the uniform distribution U_d over $X = \{0, 1\}^d$, each labeled by $c_1, \ldots, c_k$. Observe that (for every $1 \leq t \leq m$) we have that S_t contains i.i.d. elements from U_d labeled by $c_1, \ldots, c_k$. We use D_t to denote the unlabeled portion of S_t. Standard arguments in learning theory (cf. Theorem 2.7) state that for $|S_t| \geq O(d)$,

$$\Pr \left[\exists h, f \in \mathtt{PAR}_d \text{ s.t. } \mathrm{error}_{U_d}(h, f) \geq \frac{1}{4} \quad \wedge \quad \mathrm{error}_{D_t}(h, f) \leq \frac{1}{40} \right] \leq \frac{1}{8}.$$

The above inequality holds, in particular, for every hypothesis $h \in \mathtt{PAR}_d$ and every target concept c_j, and hence,

$$\Pr\left[\exists h \in \mathtt{PAR}_d \text{ and } j \text{ s.t. } \mathrm{error}_{U_d}(h, c_j) \geq \frac{1}{4} \quad \wedge \quad \mathrm{error}_{D_t}(h, c_j) \leq \frac{1}{40}\right] \leq \frac{1}{8}.$$

Recall that under the uniform distribution, the only $h \in \mathtt{PAR}_d$ s.t. $\mathrm{error}_{U_d}(h, c_j) \neq \frac{1}{2}$ is c_j itself, and hence

$$\Pr\left[\exists h \in \mathtt{PAR}_d \text{ and } j \text{ s.t. } h \neq c_j \quad \wedge \quad \mathrm{error}_{D_t}(h, c_j) \leq \frac{1}{40}\right] \leq \frac{1}{8}.$$

So, for every $1 \leq t \leq m$, with probability $7/8$ we have that for every label column j the only hypothesis with empirical error less than $\frac{1}{40}$ on S_t is the jth target concept itself (with empirical error 0). In such a case, step 2a (Gaussian elimination) identifies exactly the vector of k target concepts $(c_1, \ldots, c_k)$. Since $m \geq O(\log(\frac{1}{\beta}))$, the Chernoff bound ensures that except with probability $\beta/2$, the vector $(c_1, \ldots, c_k)$ is identified in at least $3/4$ of the iterations of step 2. Assuming that this is the case, the vector $(c_1, \ldots, c_k)$ appears in Y at least $3m/4$ times, while every other vector can appear at most $m/4$ times. Provided that $m \geq O(\frac{1}{\epsilon}\log(\frac{1}{\beta\delta}))$, algorithm $\mathcal{A}_{\text{dist}}$ ensures that the k target concepts are chosen with probability $1 - \beta/2$.

All in all, algorithm $ParityLearner$ identifies the k target concepts (exactly) with probability $1 - \beta$, provided that $n \geq O(\frac{d}{\epsilon}\log(\frac{1}{\beta\delta}))$. □

3.2.2 Learning Points

We next show that the class of $\mathtt{POINT}_X$ can be (non-agnostically) k-learned using constant sample complexity, matching the non-private sample complexity.

Theorem 3.13. *For every domain X and every $k \in \mathbb{N}$ there exists an $(\alpha, \beta, \epsilon, \delta)$-PAC (non-agnostic) k-learner for $\mathtt{POINT}_X$ with sample complexity $O(\frac{1}{\alpha\epsilon}\log(\frac{1}{\alpha\beta\delta}))$.*

The proof is via the construction of Algorithm 5. The algorithm begins by privately identifying (using sanitization) a set of $O(1/\alpha)$ "heavy" elements in the input database, appearing $\Omega(\alpha)$ times. The k labels of such a heavy element can be privately identified using stability arguments (since their duplicity in the database is large). The labels of a "non-heavy" element can be set to 0 since a target concept can evaluate to 1 on at most one such non-heavy element, in which case the error is small.

Notation. We use $\#_S(x)$ to denote the duplicity of a domain element x in a database S. For a distribution μ we denote $\mu(x) = \Pr_{\hat{x} \sim \mu}[\hat{x} = x]$.

Proof. The proof is via the construction of $PointLearner$ (algorithm 5). First note the algorithm only access the input database using sanitization on step 1, and using algorithm $\mathcal{A}_{\text{dist}}$ on step 4. By composition theorem 2.16, algorithm $PointLearner$ is (ϵ, δ)-differentially private.

Let μ be a distribution over X, and let $c_1, \ldots, c_k \in \mathtt{POINT}_X$ be the fixed target concepts. Consider the execution of $PointLearner$ on a database $S = (x_i, y_{i,1}, \ldots, y_{i,k})_{i=1}^n$ sampled from μ and labeled by $c_1, \ldots, c_k$. We use D to denote the unlabeled portion of S, $\hat{D}$ for the sanitization of D constructed on step 1, and write $m = |\hat{D}|$. Define the following good events.

E_1: For every $x \in X$ s.t. $\mu(x) \geq \alpha$ it holds that $\frac{1}{n}\#_S(x) \geq \alpha/10$.

E_2: For every $x \in X$ we have that $|\frac{1}{m}\#_{\hat{D}}(x) - \frac{1}{n}\#_S(x)| \leq \alpha/30$.

Algorithm 5 *PointLearner*

Input: Privacy parameters ϵ, δ, and a k-labeled database $S = (x_i, y_{i,1}, \ldots, y_{i,k})_{i=1}^n$. We use $D = (x_i)_{i=1}^n$ to denote the unlabeled portion of S.

Output: Hypotheses $h_1, \ldots, h_k$.

1. Let $\hat{D} \in X^m$ be an $(\frac{\epsilon}{2}, \frac{\delta}{2})$-private $(\frac{\alpha}{30}, \frac{\beta}{4})$-accurate sanitization of D w.r.t. $\texttt{POINT}_X$ (e.g., using Proposition 2.20).

2. Let $G = \{x \in X : \frac{1}{m}\#_{\hat{D}}(x) \geq \alpha/15\}$ be the set of all "$\frac{\alpha}{15}$-heavy" domain elements w.r.t. the sanitization $\hat{D}$. Note that $|G| \leq 15/\alpha$.

3. Let q be the quality function that on input a k-labeled database S, a domain element x, and a binary vector $\vec{v} \in \{0,1\}^k$, returns the number of appearances of $(x, \vec{v})$ in S. That is, $q(S, x, (v_1, \ldots, v_k)) = |\{i : x_i = x \wedge y_{i,1} = v_1 \wedge \cdots \wedge y_{i,k} = v_k\}|$.

4. Use algorithm $\mathcal{A}_{\text{dist}}$ with privacy parameters $\frac{\epsilon}{2}, \frac{\delta}{2}$ to choose a set of vectors $V = \{\vec{v}_x \in \{0,1\}^k : x \in G\}$ maximizing $Q(S, V) = \min_{\vec{v}_x \in V}\{q(S, x, \vec{v}_x)\}$. That is, we use algorithm $\mathcal{A}_{\text{dist}}$ to choose a set of $|G|$ vectors – a vector $\vec{v}_x$ for every $x \in G$ – such that the minimal number of appearances of an entry $(x, \vec{v}_x)$ in the database S is maximized.

5. For $1 \leq j \leq k$: If the jth entry of every $\vec{v}_x \in V$ is 0, then set $h_j \equiv 0$. Otherwise, let x be s.t. $\vec{v}_x \in V$ has 1 as its jth entry, and define $h_j : X \to \{0,1\}$ as $h_j(y) = 1$ iff $y = x$.

6. Return $h_1, \ldots, h_k$.

E_3 : Algorithm $\mathcal{A}_{\text{dist}}$ returns a vector set V s.t. $q(S, x, \vec{v}_x) \geq 1$ for every $x \in G$.

We now argue that when these three events happen algorithm *PointLearner* returns good hypotheses. First, observe that the set G contains every element x s.t. $\mu(x) \geq \alpha$: Let x be s.t. $\mu(x) \geq \alpha$. As event E_1 has occurred, we have that $\frac{1}{n}\#_S(x) \geq \alpha/10$. As event E_2 has occurred, we have that $\frac{1}{m}\#_{\hat{D}}(x) \geq \alpha/15$, and therefore $x \in G$.

Note that if $q(S, x, \vec{v}) \geq 1$ then the example x is labeled as $\vec{v}$ by the target concepts. Thus, as event E_3 has occurred, for every $\vec{v}_x \in V$ it holds that $\vec{v}_x = (c_1(x), \ldots, c_k(x))$. Now let h_j be the jth returned hypothesis. We next show that h_j is α-good. If $h \not\equiv 0$, then let x be the unique element s.t. $h_j(x) = 1$, and note that (according to step 5) the jth entry of $\vec{v}_x$ is 1, and hence, $c_j(x) = 1$. So $h_j = c_j$ (since c_j is a concept in $\texttt{POINT}_X$).

If $h_j \equiv 0$ then the jth entry of every $\vec{v}_x \in V$ is 0. Note that in such a case h_j only errs on the unique element x s.t. $c_j(x) = 1$, and it suffices to show that $\mu(x) < \alpha$. Assume towards contradiction that $\mu(x) \geq \alpha$. As before, event $E_1 \cap E_2$ implies that $x \in G$. As event E_3 has occurred, we also have that $\vec{v}_x \in V$ is s.t. $q(S, x, \vec{v}_x) \geq 1$, and the example x is labeled as $\vec{v}_x$ by the target concepts. This contradicts the assumption that the jth entry of $\vec{v}_x \in V$ is 0.

Thus, whenever $E_1 \cap E_2 \cap E_3$ happens, algorithm *PointLearner* returns α-good hypotheses. We will now show $E_1 \cap E_2 \cap E_3$ happens with high probability. Provided $n \geq O(\frac{1}{\alpha\epsilon}\log(\frac{1}{\alpha\delta}))$, event E_2 is guaranteed to hold with all but probability $\beta/4$ by the utility properties of the sanitizer used on step 1. See Proposition 2.20.

Theorem 2.7 (VC bound) ensures that event E_1 holds with probability $1 - \beta/4$, provided that $n \geq O(\frac{1}{\alpha}\log(\frac{1}{\alpha\beta}))$. To see this, let $z \equiv 0$ denote the constant 0 hypothesis, and consider the class $C = \texttt{POINT}_X \cup \{z\}$. Note that $\text{VC}(C) = 1$. Hence, Theorem 2.11 states that, with all but probability $1 - \beta/4$, for every $c \in \texttt{POINT}_x$ s.t. $\text{error}_\mu(c, z) \geq \alpha$ it holds that $\text{error}_D(c, z) \geq \alpha/10$. That is, with all but probability

19

$1 - \beta/4$, for every $x \in X$ s.t. $\mu(x) \geq \alpha$ it holds that $\frac{1}{n}\#_D(x) = \frac{1}{n}\#_S(x) \geq \alpha/10$.

Before analyzing event E_3, we show that if E_2 occurs, then every $x \in G$ is s.t. $\#_S(x) \geq \alpha/30$. Let $x \in G$, that is, x s.t. $\frac{1}{m}\#_{\hat{D}}(x) \geq \alpha/15$. Assuming event E_2 has occurred, we therefore have that $\frac{1}{n}\#_S(x) \geq \alpha/30$. So every $x \in G$ appears in S at least $\alpha n/30$ times with the labels $(c_1(x), \ldots, c_k(x)) \triangleq \vec{c}(x)$. Thus, $q(S, x, \vec{c}(x)) \geq \alpha n/30$. In addition, for every $\vec{v} \neq \vec{c}(x)$ it holds that $q(S, x, \vec{v}) = 0$, since *every* appearance of the example x is labeled by the target concepts. Hence, provided that $n \geq O(\frac{1}{\alpha \epsilon} \log(\frac{1}{\beta \delta}))$, algorithm $\mathcal{A}_{\text{dist}}$ ensures that event E_3 happens with probability at least $1 - \beta/2$.

Overall, $E_1 \cap E_2 \cap E_3$ happens with probability at least $1 - \beta$. $\qquad\square$

4   Approximate Privacy Lower Bounds from Fingerprinting Codes

In this section, we show how fingerprinting codes can be used to obtain $\text{poly}(k)$ lower bounds against privately learning k concepts, even for very simple concept classes. Fingerprinting codes were introduced by Boneh and Shaw [12] to address the problem of watermarking digital content. The connection between fingerprinting codes and differential privacy lower bounds was established by Bun, Ullman, and Vadhan [14] in the context of private query release, and has since been extended to a number of other differentially private analyses [3, 20, 32, 13].

A (fully-collusion-resistant) fingerprinting code is a scheme for distributing codewords $w_1, \ldots, w_n$ to n users that can be uniquely traced back to each user. Moreover, if any group of users combines its codewords into a pirate codeword w', then the pirate codeword can still be traced back to one of the users who contributed to it. Of course, without any assumption on how the pirates can produce their combined codeword, no secure tracing is possible. To this end, the pirates are constrained according to a *marking assumption*, which asserts that the combined codeword must agree with at least one of the pirates' codeword in each position. Namely, at an index j where $w_{ij} = b$ for every $i \in b$, the pirates are constrained to output w' with $w'_j = b$ as well.

To illustrate our technique, we start with an informal discussion of how the original Boneh-Shaw fingerprinting code yields an $\tilde{\Omega}(k^{1/3})$ sample complexity lower bound for multi-learning threshold functions. For parameters n and k, the (n, k)-Boneh-Shaw codebook is a matrix $W \in \{0, 1\}^{n \times k}$, whose rows w_i are the codewords given to users $i = 1, \ldots, n$. The codebook is built from a number of highly structured columns, where a "column of type i" consists of n bits where the first i bits are set to 1 and the last $n - i$ bits are set to 0. For $i = 1, \ldots, n - 1$, each column of type i is repeated a total of $k/(n-1)$ times, and the codebook W is obtained as a random permutation of these k columns. The security of the Boneh-Shaw code is a consequence of the secrecy of this random permutation. If a coalition of pirates is missing the codeword of user i, then it is unable to distinguish columns of type $i - 1$ from columns of type i. Hence, if a pirate codeword is too consistent with a user i's codeword in both the columns of type $i - 1$ and the columns of type i, a tracing algorithm can reasonably conclude that user i contributed to it. Boneh and Shaw showed that such a code is indeed secure for $k = \tilde{O}(n^3)$.

To see how this fingerprinting code gives a lower bound for multi-learning thresholds, consider thresholds over the data universe $X = \{1, \ldots, |X|\}$ for $|X| \geq n$. The key observation is that each column of the Boneh-Shaw codebook can be obtained as a labeling of the examples $1, \ldots, n$ by a threshold concept. Namely, a column of type i is the labeling of $1, \ldots, n$ by the concept c_i. Now suppose a coalition of users $T \subseteq [n]$ constructs a database S where each row is an example $i \in T$ together with the labels $w_{i1}, \ldots, w_{ik}$ coming from the codeword given to user i. Let $(h_1, \ldots, h_k)$ be the hypotheses produced by running a threshold multi-learner on the database. If every user has a bit b at index j of her codeword, then the hypothesis produced by the learner must also evaluate to b on most of the examples. Thus, the empirical averages of the hypotheses $(h_1, \ldots, h_k)$ on the examples can be used to obtain a pirate codeword satisfying the marking

assumption. The security of the fingerprinting code, i.e. the fact that this codeword can be traced back to a user $i \in T$, implies that the learner cannot be differentially private. Hence, n samples is insufficient for privately learning $k = \tilde{O}(n^3)$ threshold concepts, giving a sample complexity lower bound of $\tilde{\Omega}(k^{1/3})$.

The lower bounds in this section are stated for empirical learning, but extend to PAC learning by Theorem 2.24. We also remark that they hold against the relaxed privacy notion of *label privacy*, where differential privacy only needs to hold with respect to changing the labels of one example.

4.1 Fingerprinting Codes

An (n,k)-*fingerprinting code* consists of a pair of randomized algorithms (Gen, Trace). The parameter n is the number of users supported by the fingerprinting code, and k is the length of the code. The codebook generator Gen produces a *codebook* $W \in \{0,1\}^{n \times k}$. Each row $w_i \in \{0,1\}^k$ of W is the *codeword* of user i. For a subset $T \subseteq [n]$, we let W_T denote the set $\{w_i : i \in T\}$ of codewords belonging to users in T. The accusation algorithm Trace takes as input a pirate codeword w' and accuses some $i \in [n]$ (or $\perp$ if it fails to accuse any user).

We define the feasible set of pirate codewords for a coalition T and codebook W by

$$F(W_T) = \{w' \in \{0,1\}^k : \forall j = 1, \ldots, k \; \exists i \in S \text{ s.t. } w_{ij} = w'_j\}.$$

The basic marking assumption is that the pirate codeword $w' \in F(W_T)$. We say column j is b-*marked* if $w_{ij} = b$ for every $i \in [n]$.

Definition 4.1 (Fingerprinting Codes). For $n, k \in \mathbb{N}$ and $\xi \in (0, 1]$, a pair of algorithms (Gen, Trace) is an (n,k)-*fingerprinting code with security* ξ if Gen outputs a codebook $W \in \{0,1\}^{n \times k}$ and for every (possibly randomized) adversary $\mathcal{A}_{FP}$, and every coalition $T \subseteq [n]$, if we take $w' \leftarrow_{\text{R}} \mathcal{A}_{FP}(W_T)$, then the following properties hold.

Completeness: $\Pr[w' \in F(W_T) \wedge \text{Trace}(w') = \perp] \leq \xi,$

Soundness: $\Pr[\text{Trace}(w') \in [n] \setminus T] \leq \xi,$

Each probability is taken over the coins of Gen, Trace, and $\mathcal{A}_{FP}$. The algorithms Gen and Trace may share a common state, which is hidden to ease notation.

4.2 Lower Bound for Improper PAC Learning

Our lower bounds for multi-learning follow from constructions of fingerprinting codes with additional structural properties.

Definition 4.2. Let C be a concept class over a domain X. An (n,k)-fingerprinting code (Gen, Trace) is *compatible with concept class C* if there exist $x_1, \ldots, x_n \in X$ such that for every codebook W in the support of Gen, there exist concepts $c_1, \ldots, c_k$ such that $w_{ij} = c_j(x_i)$ for every $i = 1, \ldots, n$ and $j = 1, \ldots, k$.

Theorem 4.3. *Suppose there exists an (n,k)-fingerprinting code compatible with a concept class C with security ξ. Let $\alpha \leq 1/3$, $\beta, \epsilon > 0$, and $\delta < \frac{1-\xi-\beta}{n} - e^\epsilon \xi$. Then every (improper) (α, β)-accurate and (ϵ, δ)-differentially private empirical k-learner for C requires sample complexity greater than n.*

The proof of Theorem 4.3 follows the ideas sketched above.

Proof. Let $(\mathrm{Gen}, \mathrm{Trace})$ be an (n, k)-fingerprinting code compatible with the concept class C, and let $x_1, \ldots, x_n \in X$ be its associated universe elements. Let $D = (x_1, \ldots, x_n)$ and let $\mathcal{A}$ be an (α, β)-accurate empirical k-learner for C with sample complexity n. We will use $\mathcal{A}$ to design an adversary $\mathcal{A}_{FP}$ against the fingerprinting code.

Let $T \subseteq [n]$ be a coalition of users, and consider a codebook $W \leftarrow_{\mathrm{R}} \mathrm{Gen}$. The adversary strategy $\mathcal{A}_{FP}(W_T)$ begins by constructing a labeled database $S = (S_i)_{i=1}^n$ by setting $S_i = (x_i, w_{i1}, \ldots, w_{ik})$ for each $i \in T$ and to a nonce row for $i \notin T$. It then runs $\mathcal{A}(S)$ obtaining hypotheses $(h_1, \ldots, h_k)$. Finally, it computes for each $j = 1, \ldots, k$ the averages

$$h_j(D) = \frac{1}{n} \sum_{i=1}^n h_j(x_i)$$

and produces a pirate word w' by setting each w'_j to the value of a_j rounded to 0 or 1.

Now consider the coalition $T = [n]$. Since the fingerprinting code is compatible with C, each column $(w_{1j}, \ldots, w_{nj}) = (c_j(x_1), \ldots, c_j(x_n))$ for some concept $c_j \in C$. Thus, if the hypotheses $(h_1, \ldots, h_k)$ are α-accurate for $(c_1, \ldots, c_k)$ on S, then $w' \in F(W_T) = F(W)$. Therefore, by the completeness property of the code and the (α, β)-accuracy of $\mathcal{A}$, we have

$$\Pr\left[\mathrm{Trace}(\mathcal{A}_{FP}(W)) \neq \perp\right] \geq 1 - \xi - \beta.$$

In particular, there exists an i^* for which

$$\Pr\left[\mathrm{Trace}(\mathcal{A}_{FP}(W)) = i^*\right] \geq \frac{1 - \xi - \beta}{n}.$$

On the other hand, by the soundness property of the code,

$$\Pr\left[\mathrm{Trace}(\mathcal{A}_{FP}(W_{-i^*})) = i^*\right] \leq \xi.$$

Thus, $\mathcal{A}$ cannot be (ϵ, δ)-differentially private whenever

$$\frac{1 - \xi - \beta}{n} > e^\epsilon \cdot \xi + \delta.$$

$\square$

Remark 4.4. *If we additionally assume that there exists an element $x_0 \in X$ with $c_1(x_0) = c_2(x_0) = \cdots = c_k(x_0)$, then we can use a "padding" argument to obtain a stronger lower bound of $n/3\alpha$. More specifically, suppose $c_1(x_0) = \cdots = c_k(x_0) = 0$. We pad the database S constructed above with $(1/3\alpha - 1)n$ copies of the junk row $(x_0, 0, \ldots, 0)$. Now if a hypothesis h is α-accurate for a 0-marked column, it's empirical average will be at most α. On the other hand, an α-accurate hypothesis for a 1-marked column will have empirical average at least 2α. Since there is a gap between these two quantities, a pirate algorithm can still turn an accurate vector of k hypotheses into a feasible codeword.*

As observed earlier, the (n, k)-Boneh-Shaw code is compatible with the concept class THRESH_X for any $|X| \geq n$. Thus, instantiating Theorem 4.3 (and Remark 4.4) with the Boneh-Shaw code yields a lower bound for k-learning thresholds.

Lemma 4.5 ([12]). *Let X be a totally ordered domain with $|X| \geq n$ for some $n \in \mathbb{N}$. Then there exists an (n, k)-fingerprinting code compatible with the concept class THRESH_X with security ξ as long as $k \geq 2n^3 \log(2n/\xi)$.*

Corollary 4.6. *Every improper (α, β)-accurate and $(\epsilon = O(1), \delta = o(1/n))$-differentially private empirical k-learner for THRESH_X requires sample complexity $\min\{|X|, \tilde{\Omega}(k^{1/3}/\alpha)\}$.*

Discussion. Compatibility with a concept class is an interesting measure of the complexity of a fingerprinting code which warrants further attention. Peikert, shelat, and Smith [29] showed that structural constraints (related to compatibility) on a fingerprinting code give a lower bound on its length beyond the general lower bound of $k = \tilde{\Omega}(n^2)$ for arbitrary fingerprinting codes. In particular, they showed that the length $k = \tilde{O}(n^3)$ of the Boneh-Shaw code is essentially tight for the "multiplicity paradigm", where a codebook is a random permutation of a fixed set of columns, each repeated the same number of times. We take this as evidence that our $\tilde{\Omega}(k^{1/3})$ lower bound for $\texttt{THRESH}_X$ cannot be improved via compatible fingerprinting codes. However, closing the gap between our lower bound and the upper bound of roughly $\sqrt{k}$ remains an intriguing open question.

A natural avenue for obtaining stronger $\text{poly}(k)$ lower bounds for private k-learning is to identify compatible fingerprinting codes with shorter length. Tardos [33] showed the existence of an (n, k)-fingerprinting code of optimal length $k = \tilde{O}(n^2)$ (see Proposition 4.9). The construction of his code differs significantly from multiplicity paradigm: for each column j of the Tardos code, a bias $p_j \in (0, 1)$ is sampled from a fixed distribution, and then each bit of the column is sampled i.i.d. with bias p_j. Hence, the columns of the Tardos code are supported on all bit vectors in $\{0, 1\}^n$. This means that for a concept class C to be compatible with the (n, k)-Tardos code, it must be the case that $\text{VC}(C) \geq n$. Thus, the lower bound one obtains against k-learning C only matches the lower bound for PAC learning C (without privacy). It would be very interesting to construct a fingerprinting code of optimal length $k = \tilde{O}(n^2)$ with substantially fewer than 2^n column types (and hence compatible with a concept class of VC-dimension smaller than n).

4.3 Lower Bound for Agnostic Learning

In the agnostic learning model, a learner has to perform well even when the columns of a multi-labeled database do not correspond to any concept. This allows us to apply the argument of Theorem 4.3 without the constraint of compatibility. The result is that *any* fingerprinting code, in particular one with optimal length, gives an agnostic learning lower bound for any non-trivial concept class.

Theorem 4.7. *Suppose there exists an (n, k)-fingerprinting code with security ξ. Let C be a concept class with at least two distinct concepts. Let $\alpha \leq 1/3$, $\beta, \epsilon > 0$, and $\delta < \frac{1 - \xi - \beta}{n} - e^\epsilon \xi$. Then every (improper) agnostic (α, β)-accurate and (ϵ, δ)-differentially private empirical k-learner for C requires sample complexity greater than n.*

Proof. The proof follows in much the same way as that of Theorem 4.3. Let $(\text{Gen}, \text{Trace})$ be an (n, k)-fingerprinting code, and let $x \in X$ be such that there exist $c_0, c_1 \in C$ with $c_0(x) = 0$ and $c_1(x) = 1$. Let $\mathcal{A}$ be an agnostic (α, β)-accurate empirical k-learner for C with sample complexity n. Define a the fingerprinting code adversary $\mathcal{A}_{FP}$ just as in Theorem 4.3. Namely, $\mathcal{A}_{FP}$ constructs examples of the form $(x, w_{i1}, \ldots, w_{ij})$ with the available rows of the fingerprinting code, runs $\mathcal{A}$ on the result, and returns the rounded empirical averages of the k resulting hypotheses.

To show that $\mathcal{A}$ cannot be (ϵ, δ)-differentially private, it suffices to show that if $\mathcal{A}$ produces accurate hypotheses $h_1, \ldots, h_k$, then the pirate codeword produced by $\mathcal{A}_{FP}$ is feasible. To see this, suppose $h_1, \ldots, h_k$ are accurate, i.e.

$$\max_{1 \leq j \leq k} \left(\text{error}_{S|_j}(h_j) - \min_{c \in C} \left(\text{error}_{S|_j}(c) \right) \right) \leq \alpha.$$

Let column j of the codebook W be 0-marked, i.e. $w_{ij} = 0$ for all $i \in [n]$. Recall that $c_0(x) = 0$, and hence $\text{error}_{S|_j}(c_0) = 0$. Therefore, since hypothesis h_j is α-accurate, we have $\text{error}_{S|_j}(h_j) \leq \alpha$. This implies that bit w'_j of the pirate codeword is 0. An identical argument shows that the bits of the pirate codeword in the 1-marked columns are also 1. Thus, if $\mathcal{A}$ produces accurate hypotheses, the pirate codeword produced by $\mathcal{A}_{FP}$ is feasible. The rest of the argument in the proof of Theorem 4.3 completes the proof. $\qquad\square$

Remark 4.8. *Just as in Remark 4.4, a padding argument shows how to obtain a lower bound of $n/3\alpha$ under some additional assumptions on C, e.g. if the distinct concepts also share a common point x' with $c_0(x') = c_1(x')$.*

Proposition 4.9 ([33])**.** *For $n \in \mathbb{N}$ and $\xi \in (0, 1)$, there exists an (n, k)-fingerprinting code with security ξ as long as $k = O(n^2 \log(n/\xi))$.*

Corollary 4.10. *Every improper agnostic (α, β)-accurate and $(\epsilon = O(1), \delta = o(1/n))$-differentially private empirical k-learner for $\texttt{POINT}_X, \texttt{THRESH}_X, \texttt{PAR}_d$ requires sample complexity $\min\{|X|, \tilde{\Omega}(k^{1/2})\}$.*

The same proof yields a lower bound for agnostically learning parities under the uniform distribution.

Proposition 4.11. *Suppose there exists an (n, k)-fingerprinting code with security ξ. Let $\alpha \leq 1/6, \beta > 0$ and $d = \log n$. Then every (improper) agnostic $(\alpha, \beta, \epsilon = O(1), \delta = o(1/n))$-PAC k-learner for $\texttt{PAR}_d$ requires sample complexity $\Omega(n)$.*

Proof sketch. By Lemma 2.24, it is enough to rule out a private empirical learner for a database whose n examples are the distinct binary strings in $\{0, 1\}^d$. To do so, we follow the proof of Theorem 4.7, highlighting the changes that need to be made. First, we let c_0 be the all-zeroes concept, and let c_1 be an arbitrary other parity function. Second, $\mathcal{A}_{FP}$ instead constructs examples of the form $(x_i, w_{i1}, \dots, w_{ik})$ where x_i is the ith binary string. Finally, when converting the hypotheses $(h_1, \dots, h_k)$ into a feasible codeword, we instead set w'_j to 0 if $h_j(D) \leq \alpha$, and set w'_j to 1 if $h_j(D) \geq \frac{1}{2} - \alpha$. This works because, while $\text{error}_{S|_j}(c_0) = 0$ with respect to 0-marked columns, any concept (and in particular, c_1) has error $\frac{1}{2}$ with respect to 1-marked columns. $\square$

5 Examples where Direct Sum is Optimal

In this section we show several examples for cases where the direct sum is (roughly) optimal. As we saw in Section 4, with (ϵ, δ)-differential privacy, every non-trivial *agnostic* k-learner requires sample complexity $\Omega(\sqrt{k})$. We can prove a similar result for ϵ-private learners, that holds even for non-agnostic learners:

Theorem 5.1. *Let C be any non-trivial concept class over a domain X (i.e., $|C| \geq 2$). Every proper or improper $(\alpha, \beta = \frac{1}{2}, \epsilon)$-private PAC k-learner for C requires sample complexity $\Omega(k/\epsilon)$.*

In [5, 6, 7], Beimel et al. presented an agnostic proper learner for $\texttt{POINT}_X$ with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(1)$ under (ϵ, δ)-privacy, and an agnostic improper learner for $\texttt{POINT}_X$ with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(1)$ under ϵ-privacy. Hence, using Observation 3.2 (direct sum) with their results yields an $(\alpha, \beta, \epsilon, \delta)$-PAC agnostic proper k-learner for $\texttt{POINT}_X$ with sample complexity $\tilde{O}_{\alpha,\beta,\epsilon,\delta}(\sqrt{k})$, and an $(\alpha, \beta, \epsilon)$-PAC agnostic improper k-learner for $\texttt{POINT}_X$ with sample complexity $\tilde{O}_{\alpha,\beta,\epsilon}(k)$. As supported by our lower bounds (Corollary 4.10 and Theorem 5.1), those learners have roughly optimal sample complexity (ignoring the dependency in $\alpha, \beta, \epsilon, \delta$ and logarithmic factors in k).

Proof of Theorem 5.1. The proof is based on a packing argument [25, 5]. Let $x \in X$ and $f, g \in C$ be s.t. $f(x) \neq g(x)$. Let μ denote the constant distribution over X giving probability 1 to the point x. Note that $\text{error}_\mu(f, g) = 1$. Moreover, observe that for every concept h, if $\text{error}_\mu(h, f) < 1$ then $h(x) = f(x)$, and similarly with h, g.

Let $\mathcal{A}$ be an $(\alpha, \beta, \epsilon)$-private PAC k-learner for C with sample complexity n. For every choice of k target functions $(c_1, \dots, c_k) = \vec{c} \in \{f, g\}^k$, let $S_{\vec{c}}$ denote the k-labeled database containing n copies of the point x, each of which is labeled by $c_1, \dots, c_k$. Without loss of generality, we can assume that on such

databases $\mathcal{A}$ returns hypotheses in $\{f, g\}$ (since under μ we can replace an arbitrarily chosen hypothesis h with f if $f(x) = h(x)$ or with g if $g(x) = h(x)$). Therefore, by the utility properties of $\mathcal{A}$, for every $\vec{c} = (c_1, \ldots, c_k) \in \{f, g\}^k$ we have that $\Pr_{\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) = (c_1, \ldots, c_k)] \geq \frac{1}{2}$. By changing the database $S_{\vec{c}}$ to $S_{\vec{c}'}$ one row at a time while applying the differential privacy constraint, we see that

$$\Pr_{\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) = (c'_1, \ldots, c'_k)] \geq \frac{1}{2} e^{-\epsilon n}.$$

Since the above inequality holds for every two databases $S_{\vec{c}}$ and $S_{\vec{c}'}$, we get

$$\frac{1}{2} \geq \Pr_{\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) \neq (c_1, \ldots, c_k)] \geq (2^k - 1)\frac{1}{2} e^{-\epsilon n}.$$

Solving for n, this yields $n = \Omega(k/\epsilon)$. $\qquad\square$

Remark 5.2. *The above proof could easily be strengthened to show that $n = \Omega(\frac{k}{\alpha\epsilon})$, provided that C contains two concepts f, g s.t. $\exists x, y \in X$ for which $f(x) \neq g(x)$ and $f(y) = g(y)$.*

The following lemma shows that the sample complexities of properly and improperly learning parities under the uniform distribution are the same. Thus, for showing lower bounds, it is without loss of generality to consider proper learners.

Lemma 5.3. *Let $\alpha < 1/4$. Let $\mathcal{A}$ be a (possibly improper) $(\alpha, \beta, \epsilon, \delta)$-PAC k-learner for PAR_d under the uniform distribution with sample complexity n. Then there exists a proper $(\alpha' = 0, \beta, \epsilon, \delta)$-PAC k-learner $\mathcal{A}'$ for PAR_d (under the uniform distribution) with sample complexity n.*

Proof. The algorithm $\mathcal{A}'$ runs $\mathcal{A}$ and "rounds" each hypothesis h_j produced to the nearest parity function. That is, it outputs $(h'_1, \ldots, h'_k)$ where h'_j is a parity function that minimizes $\Pr_{x \sim U_d}[h'_j(x) \neq h_j(x)]$. Since this is just post-processing of the differentially private algorithm $\mathcal{A}$, the proper learner $\mathcal{A}$ remains (ϵ, δ)-differentially private.

Now suppose $(h_1, \ldots, h_k)$ is α-accurate for parity functions $(c_1, \ldots, c_k)$ on the uniform distribution. Then for each j,

$$\Pr_{x \sim U_d}[h'_j(x) \neq c_j(x)] \leq \Pr_{x \sim U_d}[h'_j(x) \neq h_j(x)] + \Pr_{x \sim U_d}[h_j(x) \neq c_j(x)]$$
$$\leq 2 \Pr_{x \sim U_d}[h_j(x) \neq c_j(x)]$$
$$\leq 2\alpha.$$

Hence, $\mathrm{error}_{U_d}(h'_j, c_j) < 1/2$. Since the error of any parity function from c_j (other than c_j itself) is exactly $1/2$ under the uniform distribution, we conclude that $(h'_1, \ldots, h'_k)$ is in fact 0-accurate for $(c_1, \ldots, c_k)$. $\quad\square$

Theorem 5.4. *Let $\alpha < \frac{1}{4}$. Every $(\alpha, \beta = \frac{1}{2}, \epsilon)$-PAC k-learner for PAR_d (under the uniform distribution) requires sample complexity $\Omega(kd/\epsilon)$.*

As we saw in Example 3.5, applying direct sum for k-learning parities results in a proper agnostic $(\alpha, \beta, \epsilon)$-PAC k-learner for PAR_d with sample complexity $O_{\alpha, \beta, \epsilon}(kd + k \log k)$. As stated by Theorem 5.4, this is the best possible (ignoring logarithmic factors and the dependency in α, β, ϵ).

Proof of Theorem 5.4. The proof is based on a packing argument [25, 5]. Let $\mathcal{A}$ be an $(\alpha, \beta, \epsilon)$-PAC k-learner for PAR_d with sample complexity n. By Lemma 5.3, we may assume $\mathcal{A}$ is proper and learns the hidden concepts exactly.

For every choice of k parity functions $(c_1, \ldots, c_k) = \vec{c} \in (\text{PAR}_d)^k$, let $S_{\vec{c}}$ denote a random k-labeled database containing n i.i.d. elements from U_d, each labeled by $(c_1, \ldots, c_k)$. By the utility properties of $\mathcal{A}$ we have that $\Pr_{U_d, \mathcal{A}}[\mathcal{A}(S_{\vec{c}}) = \vec{c}] \geq \frac{1}{2}$. In particular, for every $\vec{c} \in (\text{PAR}_d)^k$ there exists a database $D_{\vec{c}}$ labeled by $\vec{c}$ s.t. $\Pr_{\mathcal{A}}[\mathcal{A}(S_{\vec{c}}) = \vec{c}] \geq \frac{1}{2}$. By changing the database $D_{\vec{c}}$ to $D_{\vec{c}'}$ one row at a time while applying the differential privacy constraint, we see that

$$\Pr_{\mathcal{A}}[\mathcal{A}(D_{\vec{c}}) = \vec{c}'] \geq \frac{1}{2} e^{-\epsilon n}.$$

Since the above inequality holds for every two databases $D_{\vec{c}}$ and $D_{\vec{c}'}$, we get

$$\frac{1}{2} \geq \Pr_{\mathcal{A}}[\mathcal{A}(D_{\vec{c}}) \neq \vec{c}] \geq (|\text{PAR}_d|^k - 1) \frac{1}{2} e^{-\epsilon n}.$$

Solving for n, this yields $n = \Omega(kd/\epsilon)$. $\qquad\square$

References

[1] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[2] M. Anthony and J. Shawe-Taylor. A result of Vapnik with applications. *Discrete Applied Mathematics*, 47(3):207–217, 1993.

[3] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473, 2014.

[4] A. Beimel, H. Brenner, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94(3):401–437, 2014.

[5] A. Beimel, S. P. Kasiviswanathan, and K. Nissim. Bounds on the sample complexity for private learning and private data release. In *TCC*, pages 437–454, 2010.

[6] A. Beimel, K. Nissim, and U. Stemmer. Characterizing the sample complexity of private learners. In *ITCS*, pages 97–110, 2013.

[7] A. Beimel, K. Nissim, and U. Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *APPROX-RANDOM*, pages 363–378, 2013.

[8] A. Beimel, K. Nissim, and U. Stemmer. Learning privately with labeled and unlabeled examples. In *SODA*, pages 461–477, 2015.

[9] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In *PODS*, pages 128–138, 2005.

[10] A. Blum, K. Ligett, and A. Roth. A learning theory approach to noninteractive database privacy. *J. ACM*, 60(2):12, 2013.

[11] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, Oct. 1989.

[12] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, 1998.

[13] M. Bun, K. Nissim, U. Stemmer, and S. Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 634–649, Berkeley, CA, USA, October 18-20, 2015.

[14] M. Bun, J. Ullman, and S. P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC*, pages 1–10, 2014.

[15] K. Chaudhuri and D. Hsu. Sample complexity bounds for differentially private learning. In S. M. Kakade and U. von Luxburg, editors, *COLT*, volume 19 of *JMLR Proceedings*, pages 155–186. JMLR.org, 2011.

[16] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[17] C. Dwork and J. Lei. Differential privacy and robust statistics. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 371–380, New York, NY, USA, 2009. ACM.

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

[19] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.

[20] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang. Analyze gauss: Optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 11–20, New York, NY, USA, 2014. ACM.

[21] D. Feldman, A. Fiat, H. Kaplan, and K. Nissim. Private coresets. In M. Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 361–370. ACM, 2009.

[22] V. Feldman and D. Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *COLT*, pages 1000–1019, 2014.

[23] A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. In *STOC*, pages 803–812, 2011.

[24] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70, 2010.

[25] M. Hardt and K. Talwar. On the geometry of differential privacy. In *STOC*, pages 705–714, 2010.

[26] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011.

[27] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.

[28] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 75–84, 2007.

[29] C. Peikert, abhi shelat, and A. Smith. Lower bounds for collusion-secure fingerprinting. In *SODA*, pages 472–479, 2003.

[30] A. Roth and T. Roughgarden. Interactive privacy via the median mechanism. In *STOC*, pages 765–774, 2010.

[31] A. Smith and A. Thakurta. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *COLT 2013*, pages 819–850, 2013.

[32] T. Steinke and J. Ullman. Between pure and approximate differential privacy. In *TPDP 2015*, 2015.

[33] G. Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), 2008.

[34] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, Nov. 1984.

[35] L. G. Valiant. Knowledge infusion. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1546–1551. AAAI Press, 2006.