Journal of Machine Learning Research 20 (2019) 1-34 Submitted 8/18; Published 6/19

Simultaneous Private Learning of Multiple Concepts*

Mark Bun MBUN@BU.EDU
Department of Computer Science
Boston University

Kobbi Nissim KOBBI.NISSIM@GEORGETOWN.EDU
Department of Computer Science
Georgetown University

Uri Stemmer U@URI.CO.IL
Department of Computer Science
Ben-Gurion University

Editor: Mehryar Mohri

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* (Dwork et al., 2006b) 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 (Valiant, 1984). These tasks abstract many of the computations performed over sensitive information (Kasiviswanathan et al., 2011).

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

*. A preliminary version of this paper appeared in ITCS 2016.

$x \in X$ and $y_i = c_i(x)$. The goal of a multi-learner is to output k hypotheses $(h_1, \dots, h_k)$ 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 $(y_1, \dots, y_k)$). 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 a 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. (2005) and the notion of *private learning* was put forward and formally researched by Kasiviswanathan et al. (2011). 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 (Beimel et al., 2014; Chaudhuri and Hsu, 2011; Beimel et al., 2013a; Feldman and Xiao, 2014). 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 (Beimel et al., 2013a,b; Feldman and Xiao, 2014; Bun et al., 2015; Alon et al., 2018). 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 (Blum et al., 2013; Roth and Roughgarden, 2010; Hardt and Rothblum, 2010). For example, Blum et al. (2013) 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* (this returned database $\hat{D}$ is called a *sanitized* database, and the algorithm computing $\hat{D}$ is called a *sanitizer*). 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 (Beimel et al., 2014, 2013a,b; Bun et al., 2015). 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 $\mathrm{VC}(C)$ which is needed to learn non-privately. Building on techniques of Beimel et al. (2015), 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 (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 a pure differentially private agnostic k-learner for C with sample complexity $O(m + k \cdot \mathrm{VC}(C))$.*

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 $O(m + \sqrt{k} \cdot \mathrm{VC}(C))$.

The best known general-purpose sanitizers require sample complexity $m = O(\mathrm{VC}(C) \log |X|)$ for pure differential privacy (Blum et al., 2013) and $m = O(\log |C| \sqrt{\log |X|})$ for approximate differential privacy (Hardt and Rothblum, 2010). However, for specific concept classes (such as $\mathtt{POINT}_X$ and $\mathtt{THRESH}_X$), 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 functions 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 40 and 39).

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 (Feldman et al., 2009; Hardt and Talwar, 2010; Beimel et al., 2014) shows that any non-trivial concept class requires sample complexity $\Omega(k)$ to privately k-learn (Theorem 52). For approximate differential privacy, we use fingerprinting codes (Boneh and Shaw, 1998; Bun et al., 2014) to show that unlike points and parities, threshold functions require sample complexity $\tilde{\Omega}(k^{1/3})$ to PAC learn privately (Corollary 46). Moreover, any non-trivial concept class requires sample complexity $\tilde{\Omega}(\sqrt{k})$ to privately learn in the agnostic model (Theorem 47). 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. (2015) 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, $O(1)$ samples 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. Bun et al. (2015) and Alon et al. (2018) showed that the sample complexity of approximate-private learning can be asymptotically larger than that of non-private learning, but this separation is very mild. Specifically, they showed that learning one threshold function over a domain X requires sample complexity at least $\Omega(\log^* |X|)$ with approximate differential privacy, as opposed to $O(1)$ without privacy. An interesting open question is whether there are cases in which a stronger separation exists, or is it the case that the extra costs in the sample complexity of approximate private learning can always be reduced to $O(\log^* |X|)$. While we do not address this question directly, we exhibit a strong separation for multi-learning. In particular, learning k thresholds with approximate differential privacy requires $\tilde{\Omega}(k^{1/3})$ samples, while $O(1)$ samples suffices non-privately.

1.2. Related work

Differential privacy was defined in (Dwork et al., 2006b) and the relaxation to approximate differential privacy is from (Dwork et al., 2006a). Most related to our work is the work on private learning and its sample complexity (Blum et al., 2005; Kasiviswanathan et al., 2011; Chaudhuri and Hsu, 2011; Dwork et al., 2010; Beimel et al., 2013a,b, 2014, 2015, 2019; Kaplan et al., 2019; Feldman and Xiao, 2014; Bun et al., 2015; Alon et al., 2018) and the early work on sanitization (Blum et al., 2013). That many "natural" learning tasks can be performed privately was shown in the early work of Blum et al. (2005) and Kasiviswanathan et al. (2011). A characterization for the sample complexity of *pure-private* learners was given by Beimel et al. (2013a), 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 (Beimel et al., 2013a), Feldman and Xiao (2014) 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.

The problem of learning multiple concepts simultaneously (without privacy) has been considered before. Motivated by the problem of bridging computational learning and reasoning, Valiant (2006) 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:

	PAC learning and agnostic learning		
C	proper	improper	References
$\texttt{POINT}_X$	$k + \log\|C\|$	k	Thm. 31, Cor. 33
$\texttt{THRESH}_X$	$k + \log\|C\|$		Thm. 31
General	$\min\{k \log\|C\|, k\,\mathrm{VC}(C) + \log\|X\|\,\mathrm{VC}(C)\}$		Thm. 31
$\texttt{PAR}_d$ (uniform)	$k \log\|C\|$		Thm. 31

Lower bounds:

	PAC learning and agnostic learning		
C	proper	improper	References
$\texttt{POINT}_X$	$k + \log\|C\|$	k	Thm. 52, (Beimel et al., 2014)
$\texttt{THRESH}_X$	$k + \log\|C\|$		Thm. 52, (Beimel et al., 2014) (Feldman and Xiao, 2014)
$\texttt{PAR}_d$ (uniform)	$k \log\|C\|$		Thm. 55

Multi-learning with approximate differential privacy.

Upper bounds:

C	PAC learning (proper and improper)	Agnostic learning (proper and improper)
$\texttt{POINT}_X$	1 (Thm. 40)	$\sqrt{k}$ (Cor. 37)
$\texttt{THRESH}_X$	$2^{\log^*\|X\|} + \sqrt{k}$ (Cor. 38)	
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. 31)	
$\texttt{PAR}_d$ (uniform)	$\log\|C\|$ (Thm. 39)	$\sqrt{k}\log\|C\|$ (Thm. 31)

Lower bounds:

C	PAC learning (proper and improper)	Agnostic learning (proper and improper)	References
$\texttt{POINT}_X$	1	$\sqrt{k}$	Cor. 50
$\texttt{THRESH}_X$	$\log^*\|X\| + k^{1/3}$	$\log^*\|X\| + \sqrt{k}$	Cor. 46, Cor. 50, (Bun et al., 2015), (Alon et al., 2018)
$\texttt{PAR}_d$ (uniform)	$\log\|C\|$	$\sqrt{k} + \log\|C\|$	Cor. 50

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 (Population Error) *Let $\mathcal{P} \in \Delta(X \times \{0,1\})$ be a probability distribution over $X \times \{0,1\}$. The* population 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 population 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 3 (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 (Valiant, 1984) to multi-labeled databases (the standard PAC definition is obtained by setting $k = 1$):

Definition 4 (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 tuple $(c_1,\ldots,c_k)$ 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 5 (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 j^{th} 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 6 *Fix a concept class C over domain X. A set $\{x_1, \dots, x_d\} \in X$ is* shattered *by C if for every labeling $b \in \{0,1\}^d$, there exists $c \in C$ such that $b_1 = c(x_1), \dots, b_d = c(x_d)$. The* Vapnik-Chervonenkis (VC) dimension *of C, denoted $\mathrm{VC}(C)$, 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 population error of a hypothesis h and its empirical error (observed on a large enough sample) are similar.

Definition 7 (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 $\mathrm{error}_S(h) = \frac{1}{n}|\{i : h(x_i) \neq y_i\}|$.*

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 $\mathrm{error}_D(c,h) = \frac{1}{n}|\{i : h(x_i) \neq c(x_i)]$.*

Theorem 8 (VC-Dimension Generalization Bound, e.g. (Blumer et al., 1989)) *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 $n \geq \frac{64}{\alpha}(\mathrm{VC}(C)\ln(\frac{64}{\alpha}) + \ln(\frac{8}{\beta}))$ and the x_i are drawn i.i.d. from $\mathcal{D}$, it holds that*

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

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

Theorem 9 (VC-Dimension Agnostic Generalization Bound, e.g. (Anthony and Bartlett, 2009; Anthony and Shawe-Taylor, 1993)) *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 $n \geq \frac{64}{\alpha^2}(\mathrm{VC}(H)\ln(\frac{6}{\alpha}) + \ln(\frac{8}{\beta}))$ i.i.d. elements from $\mathcal{P}$, it holds that*

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

Using theorems 8 and 9, an upper bound of $O(\mathrm{VC}(C))$ 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 8 and 9 state that when an empirical learner is run on sufficiently many samples, it is also accurate with respect to a distribution on inputs.

Definition 10 (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 $(c_1, \dots, c_k)$ from C and database $S = ((x_i, c_1(x_i), \dots, c_k(x_i)))_{i=1}^n \in (X \times \{0,1\}^k)^n$, algorithm $\mathcal{A}$ outputs k hypotheses $(h_1, \dots, h_k)$ from H satisfying*

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

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

Definition 11 (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*

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

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

Theorem 12 *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* $\max\{n, \frac{32}{\alpha}(\mathrm{VC}(H\oplus C)\log(32/\alpha)+\log(8/\beta'))\}$ *samples (resp.* $\max\{n, \frac{64}{\alpha^2}(\mathrm{VC}(H)\log(6/\alpha) + \log(8k/\beta'))$ *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 8) states that if $m \ge \frac{32}{\alpha}(d\log(32/\alpha) + \log(8/\beta'))\}$, then

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

where $d = \mathrm{VC}(H \oplus C)$. 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 9) yields

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

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

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 13 *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 12 noting that $\mathrm{VC}(C \oplus C) = O(\mathrm{VC}(C))$.

For the agnostic case, consider the algorithm $\mathcal{A}$ that on input S outputs hypotheses $(h_1, \ldots, h_k)$ that minimize the quantities $\mathrm{error}_{S_j}(h_j)$. Applying the agnostic generalization bound (Anthony and Bartlett, 2009), 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 12. ■

It is known that even for $k = 1$, the sample complexities of PAC and agnostic learning are at least $\Omega(\mathrm{VC}(C)/\alpha)$ and $\Omega(\mathrm{VC}(C)/\alpha^2)$, 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.

$\mathtt{POINT}_X$: 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. $\mathtt{POINT}_X = \{c_x : x \in X\}$ where $c_x(y) = 1$ iff $y = x$. The VC-dimension of $\mathtt{POINT}_X$ is 1 for any X.

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

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

In this work, we focus our study of the concept class $\mathtt{PAR}_d$ 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 14 (PAC Learning $\mathtt{PAR}_d$ under Uniform) *Algorithm $\mathcal{A}$ is an (α, β)-PAC k-learner for $\mathtt{PAR}_d$ 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*

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

Definition 15 (Agnostically Learning $\mathtt{PAR}_d$ under Uniform) *Algorithm $\mathcal{A}$ is an (α, β)-PAC agnostic k-learner for $\mathtt{PAR}_d$ 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*

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

where $\mathcal{P}_j$ is the marginal distribution of $\mathcal{P}$ on the examples and the j^{th} 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., $|\{i : (x_i, y_{1,i}, \dots, y_{k,i}) \neq (x'_i, y'_{1,i}, \dots, y'_{k,i})\}| = 1$.

Definition 16 (Differential Privacy (Dwork et al., 2006b)) *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 17 (Composition of Differential Privacy (Dwork et al., 2006a; Dwork and Lei, 2009; Dwork et al., 2010)) *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 Tools

The most basic constructions of differentially private algorithms are via the Laplace Mechanism as follows.

Definition 18 (The Laplace Distribution) *A random variable has probability distribution* $\mathrm{Lap}(b)$ *if its probability density function is $f(x) = \frac{1}{2b}\exp(-\frac{|x|}{b})$, where $x \in \mathbb{R}$.*

Definition 19 (Sensitivity) *The* sensitivity *of a function $f : X^n \to \mathbb{R}^d$ is the smallest s such that for every neighboring $D, D' \in X^n$, we have $\|f(D) - f(D')\|_1 \leq s$. We use the term* sensitivity-s function *to mean a function of sensitivity $\leq s$.*

Theorem 20 (The Laplace Mechanism (Dwork et al., 2006b)) *Let $f : X^n \to \mathbb{R}^d$ be a function of sensitivity s. The mechanism A that on input $D \in X^n$ adds independently generated noise with distribution* $\mathrm{Lap}(\frac{s}{\epsilon})$ *to each of the d output terms of $f(D)$ preserves ϵ-differential privacy.*

We next describe the exponential mechanism of McSherry and Talwar (2007), which is an important building block in many differentially private constructions. 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 by sampling a random $f \in \mathcal{F}$ with probability $\propto \exp(\epsilon \cdot q(S, f)/2\Delta q)$, where Δq is the sensitivity of the quality function q, defined as the maximum over all $f \in \mathcal{F}$ of the sensitivity of the function $q(\cdot, f)$.

Proposition 21 (Properties of the Exponential Mechanism (McSherry and Talwar, 2007))

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 $|\mathcal{F}| \cdot \exp(-\epsilon tn/2)$.*

We next describe algorithm $\mathcal{A}_{\text{dist}}$ of Smith and Thakurta (2013). Our discussion follows the treatment of (Beimel et al., 2013b). As before, a quality function $q : X^* \times \mathcal{F} \to \mathbb{N}$ defines an optimization problem. Algorithm $\mathcal{A}_{\text{dist}}$ privately identifies the *exact* maximizer as long as it is sufficiently stable.

Algorithm 1 $\mathcal{A}_{\text{dist}}$

```
Input: Privacy parameters $\epsilon, \delta$, 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 22 (Properties of $\mathcal{A}_{\text{dist}}$ (Smith and Thakurta, 2013))

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

2.5. Differentially Private Sanitization

Data sanitization is a fundamental task in differential privacy. 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 23 (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 et al. (2013) showed that any concept class C admits a differentially private sanitizer with sample complexity $O(\mathrm{VC}(C)\log|X|)$:

Theorem 24 (Blum et al. (2013)) *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 (2010) can sometimes achieve lower sample complexity (roughly $O(\log|C|\sqrt{\log|X|})$).

Theorem 25 (Hardt and Rothblum (2010)) *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. For example, Bun et al. (2015) gave a sanitizer for threshold functions with sample complexity roughly $2^{\log^*|X|}$ (improving on work of Beimel et al. (2013b)).

Proposition 26 (Bun et al. (2015)) *There exists an (α,β)-accurate and (ϵ,δ)-differentially private sanitizer for $\mathtt{THRESH}_X$ with sample complexity*

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

2.6. Private learners and multi-learners

Generalizing on the concept of private learners (Kasiviswanathan et al., 2011), 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 27 (Kasiviswanathan et al. (2011)) *Let C be a concept class, and $\alpha,\beta,\epsilon>0$. There exists an (α,β,ϵ)-private agnostic proper learner for C with sample complexity*

$$O\left((\log|C| + \log 1/\beta)(1/(\epsilon\alpha) + 1/\alpha^2)\right).$$

A number of works (Beimel et al., 2014, 2013a,b; Feldman and Xiao, 2014; Bun et al., 2015) have established upper and lower bounds for learning the specific concept classes $\texttt{POINT}_X$ and $\texttt{THRESH}_X$. In the case of pure differential privacy, $\texttt{POINT}_X$ requires $\Theta(\log|X|)$ samples to learn properly (Beimel et al., 2014), but can be learned improperly with $O(1)$ samples. On the other hand, the class of threshold functions $\texttt{THRESH}_X$ require $\Omega(\log|X|)$ samples to learn, even improperly (Feldman and Xiao, 2014). In the case of approximate differential privacy, $\texttt{POINT}_x$ and $\texttt{THRESH}_x$ can be learned properly with sample complexities $O(1)$ (Beimel et al., 2013b) and $\tilde{O}(2^{\log^*|X|})$ (Bun et al., 2015), respectively. Moreover, learning threshold functions requires sample complexity $\Omega(\log^*|X|)$ (Bun et al., 2015; Alon et al., 2018).

2.7. Private PAC learning vs. Empirical Learning

We saw by Theorem 12 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 (Bun et al., 2015), 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 28 *Let $\epsilon \leq 1$. Suppose $\mathcal{A}$ is an (ϵ,δ)-differentially private (α,β)-accurate (agnostic) PAC k-learner for a concept class $\mathcal{C}$ with sample complexity n. Then there is an (ϵ,δ)-differentially private (α,β)-accurate (agnostic) empirical k-learner $\tilde{\mathcal{A}}$ for $\mathcal{C}$ 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 (Bun et al., 2015). 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 $S = ((x_i, y_{1,i}, \dots, y_{k,i}))_{i=1}^m \in (X \times \{0,1\}^k)^m$. 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 $(h_1,\dots,h_k)$ 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 (Kasiviswanathan et al., 2011; Bun et al., 2015), which shows that the sampling procedure does not hurt privacy.

Lemma 29 *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*

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

■

3. Differentially Private Sanitization for Point Functions

We begin by presenting a new sanitizer for the class of point functions, which will be useful for our constructions in the following sections. Our point sanitizer improves and simplifies a result of Beimel et al. (2013b), and exhibits an essentially optimal sample complexity.

Proposition 30 *There exists an (α, β)-accurate and (ϵ, δ)-differentially private sanitizer for $\texttt{POINT}_X$ 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 produce, 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 2 Query release for $\texttt{POINT}_X$

```
Input: Privacy parameters $(\epsilon, \delta)$, 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 2 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 $\texttt{POINT}_x(D) \leq \alpha/4 + 1/n$, 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 $\texttt{POINT}_x(D) > \alpha/4$, we conclude that the mechanism is α-accurate for all queries with probability at least $1-\beta$.

■

4. Upper Bounds on the Sample Complexity of Private Multi-Learners

4.1. Generic Construction

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

Theorem 31 *Let C be a finite concept class, and let $k \geq 1$. There exists a proper agnostic (α,β,ϵ)-private PAC k-learner for C with sample complexity*

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

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 17 to argue the overall privacy guarantee of the resulting learner, we get the following observation.

Observation 32 *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 32 in order to obtain a private k-learner. For example, Beimel et al. (2014, 2013a) gave an improper agnostic $(\alpha, \beta, \epsilon)$-PAC learner for $\texttt{POINT}_X$ with sample complexity $O_\alpha(\frac{1}{\epsilon}\log\frac{1}{\beta})$. Using Observation 32 yields the following corollary.

Corollary 33 *There exists an improper agnostic $(\alpha, \beta, \epsilon)$-PAC k-learner for $\texttt{POINT}_X$ with sample complexity $O_{\alpha,\beta,\epsilon}(k \log k)$.*

For a general concept class C, we can use Observation 32 with the generic construction of Theorem 27, stating that for every concept class C there exists a private agnostic proper learner $\mathcal{A}$ that uses $O(\log |C|)$ labeled examples.

Corollary 34 *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 1 *There exists a proper agnostic $(\alpha, \beta, \epsilon)$-PAC k-learner for $\texttt{PAR}_d$ with sample complexity $O_{\alpha,\beta,\epsilon}(kd + k \log k)$.*

As we will see in Section 6, the bounds of Corollary 33 and Example 1 on the sample complexity of k-learning $\texttt{POINT}_X$ and $\texttt{PAR}_d$ are tight (up to logarithmic factors). That is, with pure-differential privacy, the direct sum gives (roughly) optimal bounds for improperly learning $\texttt{POINT}_X$, and for (properly or improperly) learning $\texttt{PAR}_d$. This is not the case for learning $\texttt{THRESH}_X$ or for *properly* learning learning $\texttt{POINT}_X$.

In order to avoid the factor $k \log |C|$ (or $\sqrt{k} \log |C|$) in Corollary 34, we now show how an idea used in (Beimel et al., 2015) (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 $\texttt{THRESH}_X$ and for properly learning learning $\texttt{POINT}_X$ 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. (2015) 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 a minimal subset of C such that for every $c \in C$, there exists $f^* \in H$ such that $f^*(x) = c(x)$ for every $x \in \hat{D}$. Then in particular, for every j, there exists $f_j^* \in H$ that agrees with c_j on $\hat{D}$, i.e., there 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 $\text{error}_D(h_j, c_j)$.

Lemma 35 *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 35 follows from the following lemma.

Lemma 36 *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 36 with the generic sanitizer of Theorem 24 or Theorem 25 results in Lemma 35.

Algorithm 3 *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 $(\epsilon, \delta)$-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 $\epsilon'$ to choose and return a hypothesis $h_j \in H$ with (approximately) minimal error on the examples in $S$ w.r.t. their $j^{\text{th}}$ label.
```

Proof [Proof of Lemma 36] The proof is via the construction of *GenericLearner* (algorithm 3). Note that *GenericLearner* only accesses S via a sanitizer (on Step 2) and using the exponential mechanism (on Step 5). Composition theorem 17 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 j^{th} 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 $|\text{error}_D(f,h) - \text{error}_{\widetilde{D}}(f,h)| \le \frac{2\alpha}{5}$.

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

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

We first argue that when these three events happen algorithm *GenericLearner* returns good hypotheses. Fix $1 \le j \le k$, and let $c_j^* = \text{argmin}_{f \in C}\{\text{error}_{\mathcal{P}_j}(f)\}$. 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. $\text{error}_{\mathcal{P}_j}(h_j) \le \alpha + \Delta$.

For every $(y_1, \dots, y_{|B|}) \in \Pi_C(B)$, algorithm *GenericLearner* adds to H a hypothesis f s.t. $\forall 1 \le \ell \le |B|,\ f(b_\ell) = y_\ell$. 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 $\text{error}_D(h_j^*, c_j^*) \le \frac{2\alpha}{5}$. Using the triangle inequality (and event E_2) we get that this h_j^* satisfies $\text{error}_{S|_j}(h_j^*) \le \text{error}_D(h_j^*, c_j^*) + \text{error}_{S|_j}(c_j^*) \le \frac{3\alpha}{5} + \Delta$. Thus, event E_3 ensures that algorithm *GenericLearner* chooses (using the exponential mechanism) a hypothesis $h_j \in H$ s.t. $\text{error}_{S|_j}(h_j) \le \frac{4\alpha}{5} + \Delta$. Event E_2 ensures, therefore, that this h_j satisfies $\text{error}_{\mathcal{P}_j}(h_j) \le \alpha + \Delta$. 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 population error. Specifically, by setting $n \ge O(\frac{1}{\alpha^2}\,\text{VC}(C)\log(\frac{k}{\alpha\beta}))$, Theorem 9 ensures that Event E_2 occurs with probability at least $(1 - \frac{2}{5}\beta)$.

Assuming that $n \ge m$ (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

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

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 21). Note that $\log|H| \le |B| \le |\widetilde{D}| = O\left(\frac{\text{VC}(C)}{\alpha^2}\log(\frac{1}{\alpha})\right)$. Therefore, for $n \ge O\left(\frac{\text{VC}(C)}{\alpha^3\epsilon'}\log(\frac{1}{\alpha}) + \frac{1}{\alpha\epsilon'}\log(\frac{k}{\beta})\right)$, Event E_3 occurs with probability at least $(1 - \frac{\beta}{5})$.

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

Theorem 31 now follows by combining Lemma 35 and Corollary 34.

For certain concept classes, there are sanitizers with substantially lower sample complexity than the generic sanitizers. Combining Lemma 35 with Proposition 30, we obtain:

Corollary 37 *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 35 with Proposition 26, we obtain:

Corollary 38 *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.

4.2. 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.

4.2.1. Learning Parities under the Uniform Distribution

Theorem 39 *For every k, d there exists an $(\alpha{=}0, \beta, \epsilon, \delta)$-PAC (non-agnostic) k-learner for* $\texttt{PAR}_d$ *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 $\texttt{PAR}_d$ under the uniform distribution is $\Omega(d)$. Hence the sample complexity of privately k-learning $\texttt{PAR}_d$ (non-agnostically) under the uniform distribution is of the same order as that of non-private k-learning.

For the intuition behind Theorem 39, let $c_1, \ldots, c_k$ denote the k target concepts, and consider the quality function $q(D, (h_1, \ldots, h_k)) = \max_{1 \le j \le k}\{\text{error}_D(h_j, c_j)\}$. 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. (2007). We divide the input sample into a small number

Algorithm 4 *ParityLearner*

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

1. Split $S$ into $m = O(\frac{1}{\epsilon}\log(\frac{1}{\beta\delta}))$ disjoint samples $S_1, \dots, S_m$ of size $O(d)$ each. Initiate
   $Y$ as the empty multiset.
2. For every $1 \le t \le m$:
   (a) For every $1 \le j \le k$ try to use Gaussian elimination to identify a parity
       function $y_j$ that agrees with the labels of the $j^{\text{th}}$ 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 $\mathcal{A}_{\text{dist}}$ with privacy parameters $\epsilon, \delta$ to choose and return a vector of $k$
   parity functions $(h_1, \dots, h_k) \in (\texttt{PAR}_d)^k$ with a large number of appearances in $Y$.
```

of subsamples, use Gaussian elimination to (non-privately) identify a candidate hypothesis vector on each subsample, and then select from these candidates privately.

Proof [Proof of Theorem 39] 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 $\mathcal{A}_{\text{dist}}$ on Y preserves privacy (applying *ParityLearner* on neighboring inputs amounts to executing $\mathcal{A}_{\text{dist}}$ on neighboring inputs).

Now fix k target concepts $c_1, \dots, c_k \in \texttt{PAR}_d$ 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 $c_1, \dots, c_k$. Observe that (for every $1 \le t \le m$) we have that S_t contains i.i.d. elements from U_d labeled by $c_1, \dots, c_k$. We use D_t to denote the unlabeled portion of S_t. Standard arguments in learning theory (cf. Theorem 8) state that for $|S_t| \ge \Omega(d)$,

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

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

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

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

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

So, for every $1 \le t \le m$, 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 j^{th} target concept itself (with empirical error 0). In such a case, step 2a (Gaussian elimination) identifies exactly the vector of k target concepts $(c_1, \dots, c_k)$. Since $m \geq O(\log(\frac{1}{\beta}))$, the Chernoff bound ensures that except with probability $\beta/2$, the vector $(c_1, \dots, c_k)$ is identified in at least 3/4 of the iterations of step 2. Assuming that this is the case, the vector $(c_1, \dots, c_k)$ 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 $n \geq O(\frac{d}{\epsilon}\log(\frac{1}{\beta\delta}))$. ■

4.2.2. Learning Points

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

Theorem 40 *For every domain X and every $k \in \mathbb{N}$ there exists an $(\alpha, \beta, \epsilon, \delta)$-PAC (non-agnostic) k-learner for $\texttt{POINT}_X$ 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 17, algorithm *PointLearner* is (ϵ, δ)-differentially private.

Let μ be a distribution over X, and let $c_1, \dots, c_k \in \texttt{POINT}_X$ be the fixed target concepts. Consider the execution of *PointLearner* on a database $S = (x_i, y_{i,1}, \dots, y_{i,k})_{i=1}^n$ sampled from μ and labeled by $c_1, \dots, c_k$. 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 $|\frac{1}{m}\#_{\hat{D}}(x) - \frac{1}{n}\#_S(x)| \leq \alpha/30$.

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 $x \in G$.

Algorithm 5 *PointLearner*

```
Input: Privacy parameters $\epsilon, \delta$, 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 $S$.
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 30).
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 $j^{\text{th}}$ 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 $j^{\text{th}}$ entry, and define $h_j : X \to \{0,1\}$ as $h_j(y) = 1$ iff $y = x$.
6. Return $h_1, \ldots, h_k$.
```

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 j^{th} 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 j^{th} 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 j^{th} 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 j^{th} 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 30.

Theorem 8 (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 12 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 $1 - \beta/4$, 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 $(c_1(x), \dots, c_k(x)) \triangleq \vec{c}(x)$. 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 $1 - \beta$. ■

5. Approximate Privacy Lower Bounds from Fingerprinting Codes

In this section, we show how fingerprinting codes can be used to obtain poly(k) lower bounds against privately learning k concepts, even for very simple concept classes. Fingerprinting codes were introduced by Boneh and Shaw (1998) to address the problem of watermarking digital content. The connection between fingerprinting codes and differential privacy lower bounds was established by Bun et al. (2014) in the context of private query release, and has since been extended to a number of other differentially private analyses (Bassily et al., 2014; Dwork et al., 2014; Steinke and Ullman, 2015; Bun et al., 2015).

A (fully-collusion-resistant) fingerprinting code is a scheme for distributing codewords $w_1, \dots, w_n$ 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 $i = 1, \dots, n$. 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 $i = 1, \dots, n - 1$, 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 $X = \{1, \dots, |X|\}$ 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 $1, \dots, n$ by a threshold concept. Namely, a column of type i is the labeling of $1, \dots, n$ 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 $w_{i1}, \dots, w_{ik}$ coming from the codeword given to user i. Let $(h_1, \dots, h_k)$ 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 $(h_1, \dots, h_k)$ 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 28. 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.

5.1. 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

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

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 41 (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 $w' \leftarrow_R \mathcal{A}_{FP}(W_T)$, 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.*

5.2. Lower Bound for Improper PAC Learning

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

Definition 42 *Let C be a concept class over a domain X. An (n,k)-fingerprinting code* $(\mathrm{Gen},\mathrm{Trace})$ *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 43 *Suppose there exists an (n,k)-fingerprinting code compatible with a concept class C with security ξ. Let $\alpha \le 1/3$, $\beta,\epsilon > 0$, 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 43 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

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

In particular, there exists an i^* for which

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

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

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

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

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

■

Remark 44 *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 $\texttt{THRESH}_X$ for any $|X| \geq n$. Thus, instantiating Theorem 43 (and Remark 44) with the Boneh-Shaw code yields a lower bound for k-learning thresholds.

Lemma 45 (Boneh and Shaw (1998)) *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 $\texttt{THRESH}_X$ with security ξ as long as $k \geq 2n^3 \log(2n/\xi)$.*

Corollary 46 *Every improper (α, β)-accurate and $(\epsilon = O(1), \delta = o(1/n))$-differentially private empirical k-learner for $\texttt{THRESH}_X$ 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 et al. (2003) 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 poly(k) lower bounds for private k-learning is to identify compatible fingerprinting codes with shorter length. Tardos (2008) showed the existence of an (n, k)-fingerprinting code of optimal length $k = \tilde{O}(n^2)$ (see Proposition 49). 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 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).

5.3. 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 43 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 47 *Suppose there exists an (n,k)-fingerprinting code with security ξ. Let C be a concept class with at least two distinct concepts. Let $\alpha \le 1/3$, $\beta, \epsilon > 0$, 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 43. Let $(\mathrm{Gen}, \mathrm{Trace})$ 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 43. Namely, $\mathcal{A}_{FP}$ constructs examples of the form $(x, w_{i1}, \dots, w_{ij})$ 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 $h_1, \dots, h_k$, then the pirate codeword produced by $\mathcal{A}_{FP}$ is feasible. To see this, suppose $h_1, \dots, h_k$ are accurate, i.e.

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

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 $\mathrm{error}_{S|_j}(c_0) = 0$. Therefore, since hypothesis h_j is α-accurate, we have $\mathrm{error}_{S|_j}(h_j) \le \alpha$. 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 43 completes the proof. ■

Remark 48 *Just as in Remark 44, 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 49 (Tardos (2008)) *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 50 *Every improper agnostic (α,β)-accurate and $(\epsilon = O(1), \delta = o(1/n))$-differentially private empirical k-learner for $\mathtt{POINT}_X, \mathtt{THRESH}_X, \mathtt{PAR}_d$ 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 51 *Suppose there exists an (n,k)-fingerprinting code with security ξ. Let $\alpha \le 1/6, \beta > 0$ and $d = \log n$. Then every (improper) agnostic $(\alpha, \beta, \epsilon = O(1), \delta = o(1/n))$-PAC k-learner for $\mathtt{PAR}_d$ requires sample complexity $\Omega(n)$.*

Proof [Proof sketch] By Lemma 28, 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 47, 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 $(x_i, w_{i1}, \ldots, w_{ik})$ where x_i is the ith binary string. Finally, when converting the hypotheses $(h_1, \ldots, h_k)$ 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 $\mathrm{error}_{S|_j}(c_0) = 0$ with respect to 0-marked columns, any concept (and in particular, c_1) has error $\frac{1}{2}$ with respect to 1-marked columns. ■

6. 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 5, 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 52 *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 $\Omega(k/\epsilon)$.*

Beimel et al. (2014, 2013a,b) presented an agnostic proper learner for $\mathtt{POINT}_X$ with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(1)$ under (ϵ, δ)-privacy, and an agnostic improper learner for $\mathtt{POINT}_X$ with sample complexity $O_{\alpha,\beta,\epsilon,\delta}(1)$ under ϵ-privacy. Hence, using Observation 32 (direct sum) with their results yields an $(\alpha, \beta, \epsilon, \delta)$-PAC agnostic proper k-learner for $\mathtt{POINT}_X$ with sample complexity $\tilde{O}_{\alpha,\beta,\epsilon,\delta}(\sqrt{k})$, and an $(\alpha, \beta, \epsilon)$-PAC agnostic improper k-learner for $\mathtt{POINT}_X$ with sample complexity $\tilde{O}_{\alpha,\beta,\epsilon}(k)$. As supported by our lower bounds (Corollary 50 and Theorem 52), those learners have roughly optimal sample complexity (ignoring the dependency in $\alpha, \beta, \epsilon, \delta$ and logarithmic factors in k).

Proof [Proof of Theorem 52] The proof is based on a packing argument (Hardt and Talwar, 2010; Beimel et al., 2014). 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 $\mathrm{error}_\mu(f, g) = 1$. Moreover, observe that for every concept h, if $\mathrm{error}_\mu(h, f) < 1$ 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 $(c_1, \ldots, c_k) = \vec{c} \in \{f, g\}^k$, let $S_{\vec{c}}$ denote the k-labeled database containing n copies of the point x, each of which is labeled by $c_1, \ldots, c_k$. 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

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

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

Remark 53 *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 54 *Let $\alpha < 1/4$. Let $\mathcal{A}$ be a (possibly improper) $(\alpha, \beta, \epsilon, \delta)$-PAC k-learner for $\mathtt{PAR}_d$ under the uniform distribution with sample complexity n. Then there exists a proper $(\alpha' = 0, \beta, \epsilon, \delta)$-PAC k-learner $\mathcal{A}'$ for $\mathtt{PAR}_d$ (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 $(h'_1, \dots, h'_k)$ 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 $(h_1, \dots, h_k)$ is α-accurate for parity functions $(c_1, \dots, c_k)$ on the uniform distribution. Then for each j,

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

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 $(h'_1, \dots, h'_k)$ is in fact 0-accurate for $(c_1, \dots, c_k)$. ■

Theorem 55 *Let $\alpha < \frac{1}{4}$. Every $(\alpha, \beta{=}\frac{1}{2}, \epsilon)$-PAC k-learner for $\mathtt{PAR}_d$ (under the uniform distribution) requires sample complexity $\Omega(kd/\epsilon)$.*

As we saw in Example 1, applying direct sum for k-learning parities results in a proper agnostic $(\alpha, \beta, \epsilon)$-PAC k-learner for $\mathtt{PAR}_d$ with sample complexity $O_{\alpha,\beta,\epsilon}(kd + k \log k)$. As stated by Theorem 55, this is the best possible (ignoring logarithmic factors and the dependency in α, β, ϵ).

Proof [Proof of Theorem 55] The proof is based on a packing argument (Hardt and Talwar, 2010; Beimel et al., 2014). Let $\mathcal{A}$ be an $(\alpha, \beta, \epsilon)$-PAC k-learner for $\mathtt{PAR}_d$ with sample complexity n. By Lemma 54, we may assume $\mathcal{A}$ is proper and learns the hidden concepts exactly.

For every choice of k parity functions $(c_1, \ldots, c_k) = \vec{c} \in (\mathtt{PAR}_d)^k$, 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 $\vec{c} \in (\mathtt{PAR}_d)^k$ 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

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

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

Acknowledgments

This work was done while Mark Bun was at the John A. Paulson School of Engineering & Applied Sciences, Harvard University and visiting Yale University, supported by an NDSEG fellowship and NSF grant CNS-1237235. Kobbi Nissim and Uri Stemmer were supported by NSF grant no. 1565387. TWC: Large: Collaborative: Computing Over Distributed Sensitive Data.

References

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. *CoRR*, abs/1806.00949, 2018. URL http://arxiv.org/abs/1806.00949.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations.* Cambridge University Press, New York, NY, USA, 1st edition, 2009. ISBN 052111862X, 9780521118620.

Martin Anthony and John Shawe-Taylor. A result of Vapnik with applications. *Discrete Applied Mathematics*, 47(3):207–217, 1993.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, pages 464–473, 2014. doi: 10.1109/FOCS.2014.56. URL http://dx.doi.org/10.1109/FOCS.2014.56.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *ITCS*, pages 97–110, 2013a.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *APPROX-RANDOM*, pages 363–378, 2013b.

Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94(3):401–437, 2014. doi: 10.1007/s10994-013-5404-1. URL http://dx.doi.org/10.1007/s10994-013-5404-1.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Learning privately with labeled and unlabeled examples. In *SODA*, pages 461–477, 2015. doi: 10.1137/1.9781611973730.32. URL http://dx.doi.org/10.1137/1.9781611973730.32.

Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. *CoRR*, abs/1902.10731, 2019. URL http://arxiv.org/abs/1902.10731.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *PODS*, pages 128–138, 2005.

Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *J. ACM*, 60(2):12, 2013. doi: 10.1145/2450142.2450148. URL http://doi.acm.org/10.1145/2450142.2450148.

Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, October 1989. ISSN 0004-5411. doi: 10.1145/76359.76371. URL http://doi.acm.org/10.1145/76359.76371.

Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, 1998.

Mark Bun, Jonathan Ullman, and Salil P. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC*, pages 1–10, 2014. doi: 10.1145/2591796.2591877. URL http://doi.acm.org/10.1145/2591796.2591877.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 634–649, Berkeley, CA, USA, October 18-20, 2015.

Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In Sham M. Kakade and Ulrike von Luxburg, editors, *COLT*, volume 19 of *JMLR Proceedings*, pages 155–186. JMLR.org, 2011.

Cynthia Dwork and Jing 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. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536466. URL http://doi.acm.org/10.1145/1536414.1536466.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006b.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.

Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li 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. ISBN 978-1-4503-2710-7. doi: 10.1145/2591796.2591883. URL `http://doi.acm.org/10.1145/2591796.2591883`.

Dan Feldman, Amos Fiat, Haim Kaplan, and Kobbi Nissim. Private coresets. In Michael 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. ISBN 978-1-60558-506-2. doi: 10.1145/1536414.1536465. URL `http://doi.acm.org/10.1145/1536414.1536465`.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *COLT*, pages 1000–1019, 2014. URL `http://jmlr.org/proceedings/papers/v35/feldman14b.html`.

Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70, 2010.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *STOC*, pages 705–714, 2010.

Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Differentially private learning of geometric concepts. *CoRR*, abs/1902.05017, 2019. URL `http://arxiv.org/abs/1902.05017`.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011. doi: 10.1137/090756090. URL `http://dx.doi.org/10.1137/090756090`.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-3010-9. doi: 10.1109/FOCS.2007.41. URL `http://dx.doi.org/10.1109/FOCS.2007.41`.

Kobbi Nissim, Sofya Raskhodnikova, and Adam 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.

Chris Peikert, abhi shelat, and Adam Smith. Lower bounds for collusion-secure fingerprinting. In *SODA*, pages 472–479, 2003. URL `http://dl.acm.org/citation.cfm?id=644108.644187`.

Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *STOC*, pages 765–774, 2010.

Adam Smith and Abhradeep Thakurta. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *COLT 2013*, pages 819–850, 2013. URL `http://jmlr.org/proceedings/papers/v30/Guha13.html`.

Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. In *TPDP 2015*, 2015.

Gábor Tardos. Optimal probabilistic fingerprint codes. *J. ACM*, 55(2), 2008.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL `http://doi.acm.org/10.1145/1968.1972`.

Leslie 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. URL `http://www.aaai.org/Library/AAAI/2006/aaai06-247.php`.