Mach Learn (2014) 94:401–437
DOI 10.1007/s10994-013-5404-1

Bounds on the sample complexity for private learning and private data release

Amos Beimel · Hai Brenner ·
Shiva Prasad Kasiviswanathan · Kobbi Nissim

Received: 3 July 2011 / Accepted: 5 August 2013 / Published online: 18 September 2013
© The Author(s) 2013

Abstract Learning is a task that generalizes many of the analyses that are applied to collections of data, in particular, to collections of sensitive individual information. Hence, it is natural to ask what can be learned while preserving individual privacy. Kasiviswanathan et al. (in SIAM J. Comput., 40(3):793–826, 2011) initiated such a discussion. They formalized the notion of *private learning*, as a combination of PAC learning and differential privacy, and investigated what concept classes can be learned privately. Somewhat surprisingly, they showed that for finite, discrete domains (ignoring time complexity), every PAC learning task could be performed privately with polynomially many labeled examples; in many natural cases this could even be done in polynomial time.

While these results seem to equate non-private and private learning, there is still a significant gap: the sample complexity of (non-private) PAC learning is crisply characterized in terms of the VC-dimension of the concept class, whereas this relationship is lost in the constructions of private learners, which exhibit, generally, a higher sample complexity.

Looking into this gap, we examine several private learning tasks and give tight bounds on their sample complexity. In particular, we show strong separations between sample complexities of proper and improper private learners (such separation does not exist for non-private learners), and between sample complexities of efficient and inefficient proper private

Editor: Phil Long.

A. Beimel (✉) · K. Nissim
Dept. of Computer Science, Ben-Gurion University, Beer-Sheva, Israel
e-mail: beimel@cs.bgu.ac.il

K. Nissim
e-mail: kobbi@cs.bgu.ac.il

H. Brenner
Dept. of Mathematics, Ben-Gurion University, Beer-Sheva, Israel
e-mail: haib@bgu.ac.il

S.P. Kasiviswanathan
General Electric Research Center, San Ramon, CA, USA
e-mail: kasivisw@gmail.com

learners. Our results show that VC-dimension is not the right measure for characterizing the sample complexity of proper private learning.

We also examine the task of *private data release* (as initiated by Blum et al. in STOC, pp. 609–618, 2008), and give new lower bounds on the sample complexity. Our results show that the logarithmic dependence on size of the instance space is essential for private data release.

Keywords Differential privacy · PAC learning · Sample complexity · Private data release

1 Introduction

Consider a scenario in which a survey is conducted among a sample of random individuals and data mining techniques are applied to learn information on the entire population. If such information will disclose information on the individuals participating in the survey, then they will be reluctant to participate in the survey. To address this question, Kasiviswanathan et al. (2011) introduced the notion of *private learning*, where a private learner is required to output a hypothesis that gives accurate classification while protecting the privacy of the individual samples from which the hypothesis was obtained.

The definition of a private learner is a combination of two qualitatively different notions. One is that of probably approximately correct (PAC) learning (Valiant 1984), the other of differential privacy (Dwork et al. 2006). PAC learning, on one hand, is an average case requirement, which requires that the output of the learner on most samples is good. Differential privacy, on the other hand, is a *worst-case* requirement. It is a strong notion of privacy that provides meaningful guarantees in the presents of powerful attackers and is increasingly accepted as a standard for providing rigorous privacy. Recent research on privacy has shown, somewhat surprisingly, that it is possible to design differentially private variants of many analyses. Further discussions on differential privacy can be found in the surveys of Dwork (2009, 2011).

We next give more details on PAC learning and differential privacy. In *PAC learning*, a collection of samples (labeled examples) is generalized into a hypothesis. It is assumed that the examples are generated by sampling from some (unknown) distribution $\mathcal{D}$ and are labeled according to an (unknown) concept c taken from some concept class $\mathcal{C}$. The learned hypothesis h should predict with high accuracy the labeling of examples taken from the distribution $\mathcal{D}$, an *average-case* requirement. In *differential privacy* the output of a learner should not be significantly affected if a particular example is replaced with an arbitrary example. Concretely, differential privacy considers the collection of samples as a database, defines that two databases are neighbors if they differ in exactly one sample, and requires that for every two neighboring databases the output distribution of a private learner should be similar.

In this paper, we consider private learning of finite, discrete domains. Finite domains are natural as computers only store information with finite precision. The work of Kasiviswanathan et al. (2011) demonstrated that private learning in such domains is feasible—any concept class that is PAC learnable can be learned privately (but not necessarily efficiently), by a "private Occam's razor" algorithm, with sample complexity that is logarithmic in the size of the hypothesis class.[1] Furthermore, taking into account the earlier result

[1] Chaudhuri and Hsu (2011) prove that this is not true for continuous domains.

of Blum et al. (2005) (that all concept classes that can be efficiently learned in the *statistical queries* model can be learned privately and efficiently) and the efficient private parity learner of Kasiviswanathan et al. (2011), we get that most "natural" computational learning tasks can be performed privately and efficiently (i.e., with polynomial resources). This is important as learning problems generalize many of the computations performed by analysts over collections of sensitive data.

The results of Blum et al. (2005), Kasiviswanathan et al. (2011) show that private learning is feasible in an extremely broad sense, and hence, one can essentially equate learning and private learning. However, the costs of the private learners constructed in Blum et al. (2005), Kasiviswanathan et al. (2011) are generally higher than those of non-private ones by factors that depend not only on the privacy, accuracy, and confidence parameters of the private learner. In particular, the well-known relationship between the sample complexity of PAC learners and the VC-dimension of the concept class (ignoring computational efficiency) (Blumer et al. 1989) does not hold for the above constructions of private learners; the sample complexity of the algorithms of Blum et al. (2005), Kasiviswanathan et al. (2011) is proportional to the logarithm of the size of the concept class. Recall that the VC-dimension of a concept class is bounded by the logarithm of its size, and is significantly lower for many interesting concept classes, hence, there may exist learning tasks for which "very practical" non-private learner exists, but any private learner is "impractical" (with respect to the sample size required).

The focus of this work is on a fine-grain examination of the differences in complexity between private and non-private learning. The hope is that such an examination will eventually lead to an understanding of which complexity measure is relevant for the sample complexity of private learning, similar to the well-understood relationship between the VC-dimension and sample complexity of PAC learning. Such an examination is interesting also for other tasks, and a second task we examine is that of releasing a *sanitization* of a data set that simultaneously protects privacy of individual contributors and offers utility to the data analyst. See the discussion in Sect. 1.1.2.

1.1 Our contributions

We now give a brief account of our results. Throughout this rather informal discussion we will treat the accuracy, confidence, and privacy parameters as constants (a detailed analysis revealing the dependency on these parameters is presented in the technical sections). We use the term "efficient" for polynomial time computations.

Following standard computational learning terminology, we will call learners for a concept class $\mathcal{C}$ that only output hypotheses in $\mathcal{C}$ *proper*, and other learners *improper*. The original motivation in computational learning theory for this distinction is that there exist concept classes $\mathcal{C}$ for which proper learning is computationally intractable (Pitt and Valiant 1988), whereas it is tractable to learn $\mathcal{C}$ improperly (Valiant 1984). As we will see below, the distinction between proper and improper learning is useful also when discussing private learning, and for reasons other than making intractable learning tasks tractable. Our results on private learning are summarized in Table 1.

1.1.1 Proper and improper private learning

It is instructive to look into the construction of the private Occam's razor algorithm of Kasiviswanathan et al. (2011) and see why its sample complexity is proportional to the logarithm of the size of the hypothesis class used. The algorithm uses the exponential mechanism

Table 1 Our separation results (ignoring dependence on ϵ, α, β), where $\ell(d)$ is any function that grows as $\omega(\log d)$

Concept class	Sample complexity		
$\mathtt{POINT}_d$	Non-Private Learning (Proper or Improper)	Improper Private Learning	Proper Private Learning
	$\Theta(1)$ (Blumer et al. 1989; Ehrenfeucht et al. 1989)	$\Theta(1)$	$\Theta(d)$
$\widehat{\mathtt{POINT}}_d$	Non-Private Learning (Efficient or Inefficient)	Inefficient Proper Private Learning	Efficient Proper [a] Private Learning
	$\Theta(1)$ (Blumer et al. 1989; Ehrenfeucht et al. 1989)	$\Theta(\ell(d))$	$\Theta(d)$

[a] These bounds are for a slightly relaxed notion of proper learners as detailed in Sect. 6

of McSherry and Talwar (2007) to choose a hypothesis. The choice is probabilistic, where the probability mass that is assigned to each of the hypotheses decreases exponentially with the number of samples that are inconsistent with it. A union-bound argument is used in the claim that the construction actually yields a learner, and a sample size that is logarithmic in the size of the hypothesis class is needed for the argument to go through. The question is whether such sample size is required?

To address the above question, we consider a simple, but natural, class $\mathtt{POINT} = \{\mathtt{POINT}_d\}$ containing the concepts $c_j : \{0,1\}^d \to \{0,1\}$ where $c_j(x) = 1$ for $x = j$, and 0 otherwise. The VC-dimension of $\mathtt{POINT}_d$ is one, and hence, it can be learned (non-privately and efficiently, properly or improperly) with merely $O(1)$ samples.

In sharp contrast, (when used for properly learning $\mathtt{POINT}_d$) the above-mentioned private Occam's razor algorithm from Kasiviswanathan et al. (2011) requires $O(\log(|\mathtt{POINT}_d|)) = O(d)$ samples—obtaining the largest possible gap in sample complexity when compared to non-private learners! Our first result is a matching lower bound. We prove that any *proper* private learner for $\mathtt{POINT}_d$ must use $\Omega(d)$ samples, therefore, answering negatively the question (from Kasiviswanathan et al. (2011)) of whether proper private learners should exhibit sample complexity that is approximately the VC-dimension (or even a function of the VC-dimension) of the concept class.[2]

A natural way to improve the sample complexity is to use the private Occam's razor to improperly learn $\mathtt{POINT}_d$ with a smaller hypothesis class that is still expressive enough for $\mathtt{POINT}_d$, reducing the sample complexity to the logarithm of the smaller hypothesis class. We show that this indeed is possible, as there exists a hypothesis class of size $O(d)$ that can be used for learning $\mathtt{POINT}_d$ improperly, yielding an algorithm with sample complexity $O(\log d)$. Furthermore, this bound is tight, any hypothesis class for learning $\mathtt{POINT}_d$ must contain $\Omega(d)$ hypotheses. These bounds are interesting as they give a separation between proper and improper private learning—proper private learning of $\mathtt{POINT}_d$ requires $\Omega(d)$ samples, whereas $\mathtt{POINT}_d$ can be improperly privately learned using $O(\log d)$ samples. Note that such a combinatorial separation does not exist for non-private learning, as VC-dimension number of samples are *needed* and *sufficient* for both proper and improper non-private learners. Furthermore, the $\Omega(d)$ lower bound on the size of the hypothesis class maps a clear boundary to what can be achieved in terms of sample complexity using the private Occam's razor for $\mathtt{POINT}_d$. It might even suggest that *any* private learner for $\mathtt{POINT}_d$ should use $\Omega(\log d)$ samples.

[2] Our proof technique yields lower bounds not only on private learning $\mathtt{POINT}_d$ properly, but on private learning of any concept class $\mathcal{C}$ with various hypothesis classes that we call α-minimal for $\mathcal{C}$.

It turns out, however, that the intuition expressed in the last sentence is at fault. We construct an efficient improper private learner for $\texttt{POINT}_d$ that uses merely $O(1)$ samples, hence, establishing the strongest possible separation between proper and improper private learners. For the construction, we extrapolate on a technique from the efficient private parity learner of Kasiviswanathan et al. (2011). The construction of Kasiviswanathan et al. (2011) utilizes a natural non-private proper learner, and hence, results in a proper private learner. Due to the bounds mentioned above, we cannot use a proper learner for $\texttt{POINT}_d$, and hence, we construct an improper (rather unnatural) learner to base our construction upon. Our construction utilizes a double-exponential hypothesis class, and hence, is inefficient (even outputting a hypothesis requires super-polynomial time). We use a simple compression using pseudorandom functions (akin to Mishra and Sandler (2006)) to make the algorithm efficient.

The above two improper learning algorithms use "heavy" hypotheses, that is, the hypotheses are Boolean functions that return 1 on many inputs (in contrast to a point function that returns 1 on exactly one input). Informally, each such heavy hypothesis protects the privacy since it could have been returned on many different concepts. The main technical point in these algorithms is how to choose a heavy hypothesis with a small error. To complete the picture, we prove that using heavy hypotheses is unavoidable: Every private learning algorithm for $\texttt{POINT}_d$ that uses $o(d)$ samples must use heavy hypotheses.

Next we look into the concept class $\texttt{INTERVAL} = \{\texttt{INTERVAL}_d\}$, where for $T = 2^d$ we define $\texttt{INTERVAL}_d = \{c_1, \dots, c_{T+1}\}$ and, for $1 \leq j \leq T + 1$, the concept $c_j : \{1, \dots, T + 1\} \rightarrow \{0, 1\}$ is defined as follows: $c_j(x) = 1$ for $x < j$ and $c_j(x) = 0$ otherwise. As with $\texttt{POINT}_d$, it is easy to show that the sample complexity of any proper private learner for $\texttt{INTERVAL}_d$ is $\Omega(d)$. We give two results regarding the sample complexity of improper private learning of $\texttt{INTERVAL}_d$. The first result shows that if a sublinear (in d) sample complexity private learner exists for $\texttt{INTERVAL}_d$, then it must output, with high probability, a very "complex looking" hypothesis in the sense that the hypothesis must switch from zero to one (and vice-versa) exponentially many times, unlike any concept $c_j \in \texttt{INTERVAL}_d$ that switches only once from one to zero at j. The second result considers a generalization of the technique that yielded the $O(1)$ sample improper private learner for $\texttt{POINT}_d$, and shows that it alone would not yield a private learner for $\texttt{INTERVAL}_d$ with sublinear (in d) sample complexity.

We apply the above lower bound on the number of samples for proper private learning $\texttt{POINT}_d$ to show a separation in the sample complexity of efficient proper private learners (under a slightly relaxed definition of proper learning) and inefficient proper private learners. More concretely, assuming the existence of a pseudorandom generator with exponential stretch, we present a concept class $\widehat{\texttt{POINT}}_d$—a subset of $\texttt{POINT}_d$—such that every efficient private learner that learns $\widehat{\texttt{POINT}}_d$ using $\texttt{POINT}_d$ requires $\Omega(d)$ samples. In contrast, an inefficient proper private learner exists that uses only a super-logarithmic number of samples. This is the first example in private learning where requiring efficiency on top of privacy comes at a price of larger sample size.

1.1.2 The sample size of non-interactive sanitization mechanisms

Given a database containing a collection of individual information, a sanitization is a release of information that protects the privacy of the individual contributors while offering utility to the analyst using the database. The setting is non-interactive if once the sanitization is released, then the original database and the curator play no further role. Blum et al. (2008) presented a construction of such non-interactive sanitizers for count queries. Let $\mathcal{C}$ be a concept class consisting of efficiently computable predicates from a discretized domain X to $\{0, 1\}$. Given a collection D of data items taken from X, Blum et al. employ the exponential mechanism (McSherry and Talwar 2007) to (inefficiently) obtain another collection D' with data items from X such that D' maintains approximately correct count of $\sum_{d \in D} c(d)$ for all concepts $c \in \mathcal{C}$ provided that the size of D is $O(\log(|X|) \cdot \mathit{VCDIM}(\mathcal{C}))$. As D' is generated using the exponential mechanism, the differential privacy of D is protected. The database D' is referred to as a *synthetic* database as it contains data items drawn from the same universe (i.e., from X) as the original database D.

We provide a new lower bound for non-interactive sanitization mechanisms. We show that for $\texttt{POINT}_d$ every non-interactive sanitization mechanism that is useful[3] for $\texttt{POINT}_d$ requires a database of size $\Omega(d)$. This lower bound is tight as the sanitization mechanism of Blum et al. for $\texttt{POINT}_d$ uses a database of size $O(d \cdot \mathit{VCDIM}(\texttt{POINT}_d)) = O(d)$. Our lower bound holds even if the sanitized output is an arbitrary data structure, i.e., not necessarily a synthetic database.

A preliminary version of this paper appeared in the 7th Theory of Cryptography Conference (TCC), 2010. The TCC paper contained a proof sketch of the results presented in Sects. 3, 4.2, 6, and 7. The results presented in Sects. 4.1, 4.3, and 5 are new.

1.2 Related work

The notion of PAC learning was introduced by Valiant (1984). The notion of differential privacy was introduced by Dwork et al. (2006). Private learning was introduced in Kasiviswanathan et al. (2011). Beyond proving that (ignoring computation) every concept class with finite, discrete domain can be PAC learned privately (see Theorem 3.2 below), Kasiviswanathan et al. proved an equivalence between learning in the statistical queries model and private learning in the local communication model (a.k.a. *randomized response*). The general private data release mechanism we mentioned above was introduced in Blum et al. (2008) along with a specific construction for halfspace queries. Also as mentioned above, both Kasiviswanathan et al. (2011) and Blum et al. (2008) use the exponential mechanism of McSherry and Talwar (2007), a generic construction of differential private analyses, which (in general) does not yield efficient algorithms.

A recent work of Dwork et al. (2009) considered the complexity of non-interactive sanitization under two settings: (a) sanitized output is a synthetic database, and (b) sanitized output is some arbitrary data structure. For the task of sanitizing with a synthetic database they show a separation between efficient and inefficient sanitization mechanisms based on whether the size of the instance space and the size of the concept class is polynomial in a (security) parameter or not. For the task of sanitizing with an arbitrary data structure they show a tight connection between the complexity of sanitization and traitor tracing schemes used in cryptography. They leave the problem of separating efficient private and inefficient private learning open.

Following the preliminary version of our paper (Beimel et al. 2010), Chaudhuri and Hsu (2011) study the sample complexity for private learning infinite concept classes when the data is drawn from a continuous distribution. Using techniques very similar to ours, they show that, under these settings, there exists a simple concept class for which any proper learner that uses a finite number of examples and guarantees differential privacy, fails to satisfy accuracy guarantee for at least one unlabeled data distribution. This implies that

[3] Informally, a mechanism is useful for a concept class if for every input, the output of the mechanism maintains approximately correct counts for all concepts in the concept class.

the results of Kasiviswanathan et al. (2011) do not extend to infinite hypothesis classes on continuous data distributions.

Chaudhuri and Hsu (2011) also study learning algorithms that are only required to protect the privacy of the labels (and not necessary protect the privacy of the examples themselves). They prove upper bounds and lower bounds for this scenario. In particular, they prove a lower bound on the sample complexity using the doubling dimension of the disagreement metric of the hypothesis class with respect to the unlabeled data distribution. This result does not imply our results. For example, the class $\mathtt{POINT}_d$ can be properly learned using $O(1)$ samples while protecting the privacy of the labels, while we prove that $\Omega(d)$ samples are required to properly learn this class while protecting the privacy of the examples and the labels. It seems that label privacy may give enough protection in the restricted setting where the content of the underlying examples is publicly known. However, in many settings this information is highly sensitive. For example, in a database containing medical records we wish to protect the identity of the people in the sample (i.e., we do not want to disclose that they have been to a hospital).

It is well known that for all concept classes $\mathcal{C}$, every learner for $\mathcal{C}$ requires $\Omega(\mathit{VCDIM}(\mathcal{C}))$ samples (Ehrenfeucht et al. 1989). This lower bound on the sample size also holds for private learning. Blum et al. (2013) show that this result extends to the setting of private data release. They show that for all concept classes $\mathcal{C}$, every non-interactive sanitization mechanism that is useful for $\mathcal{C}$ requires $\Omega(\mathit{VCDIM}(\mathcal{C}))$ samples (remember that the best upper bound is $O(\log(|X|) \cdot \mathit{VCDIM}(\mathcal{C})))$. We show in Sect. 7 that the lower bound of $\Omega(\mathit{VCDIM}(\mathcal{C}))$ is not tight—there exists a concept class $\mathcal{C}$ of constant VC-dimension such that every non-interactive sanitization mechanism that is useful for $\mathcal{C}$ requires a much larger sample size.

Tools for private learning (not in the PAC setting) were studied in a few papers; such tools include, for example, private logistic regression (Chaudhuri and Monteleoni 2008) and private empirical risk minimization (Chaudhuri et al. 2011; Kifer et al. 2012).

1.3 Questions for future exploration

The motivation of this work was to study the connection between non-private and private learning. We believe that the ideas developed in this work are a first step in developing a general theory of private learning. In particular, we believe that there is a combinatorial measure that characterizes private learning (for non-private learning such combinatorial measure exists—the VC dimension). Such characterization was given recently in Beimel et al. (2013).

In this paper, the ideas used for lower bounding sample size for proper private learning of points is also used to establish a lower bound on the sample size for sanitization of databases. Other connections between private learning and sanitization were explored in (Blum et al. 2008). The open question is there is a deeper connection between the models, i.e., does any bound for one task imply a similar bound for the other?

1.4 Organization

In Sect. 2, we define private learning. In Sect. 3, we prove lower bounds on proper private learning, and in Sect. 4, we describe efficient improper private learning algorithms for the $\mathtt{POINT}$ concept class. In Sect. 5, we discuss private learning of the $\mathtt{INTERVAL}$ concept class. In Sect. 6, we show a separation between efficient and inefficient proper private learning. Finally, in Sect. 7, we prove a lower bound for non-interactive sanitization.

2 Preliminaries

Notation We use $[n]$ to denote the set $\{1, 2, \dots, n\}$. The notation $O_\gamma(g(n))$ is a shorthand for $O(h(\gamma) \cdot g(n))$ for some non-negative function h. Similarly, the notation $\Omega_\gamma(g(n))$. We use $\operatorname{negl}(\cdot)$ to denote functions from $\mathbb{R}^+$ to $[0, 1]$ that decrease faster than any inverse polynomial.

2.1 Preliminaries from privacy

A database is a vector $D = (d_1, \dots, d_m)$ over a domain X, where each entry $d_i \in D$ represents information contributed by one individual. Databases D and D' are called *neighbors* if they differ in exactly one entry (i.e., the Hamming distance between D and D' is 1). An algorithm is private if neighboring databases induce nearby distributions on its outcomes. Formally:

Definition 2.1 (Differential Privacy (Dwork et al. 2006)) A randomized algorithm $\mathcal{A}$ is ϵ-differentially private if for all neighboring databases D, D', and for all sets $\mathcal{S}$ of outputs,

$$\Pr\bigl[\mathcal{A}(D) \in \mathcal{S}\bigr] \le \exp(\epsilon) \cdot \Pr\bigl[\mathcal{A}\bigl(D'\bigr) \in \mathcal{S}\bigr]. \tag{1}$$

The probability is taken over the random coins of $\mathcal{A}$.

An immediate consequence of (1) is that for any two databases D, D' (not necessarily neighbors) of size m, and for all sets $\mathcal{S}$ of outputs, $\Pr[\mathcal{A}(D) \in \mathcal{S}] \ge \exp(-\epsilon m) \cdot \Pr[\mathcal{A}(D') \in \mathcal{S}]$.

2.2 Preliminaries from learning theory

We consider Boolean classification problems. A concept $c : X \to \{0, 1\}$ is a function that labels *examples* taken from the domain X by either 0 or 1. The domain X is understood to be an ensemble $X = \{X_d\}_{d\in\mathbb{N}}$ (typically, $X_d = \{0, 1\}^d$) and a *concept class* $\mathcal{C}$ is an ensemble $\mathcal{C} = \{\mathcal{C}_d\}_{d\in\mathbb{N}}$ where $\mathcal{C}_d$ is a class of concepts mapping X_d to $\{0, 1\}$. In this paper X_d is always a finite, discrete set. A concept class comes implicitly with a way to represent concepts and $\operatorname{size}(c)$ is the size of the (smallest) representation of the concept c under the given representation scheme.

PAC learning algorithms are given examples sampled according to an unknown probability distribution $\mathcal{D}$ over X_d, and labeled according to an unknown *target* concept $c_d \in \mathcal{C}_d$. Define the error of a hypothesis $h : X_d \to \{0, 1\}$ as

$$\underset{\mathcal{D}}{\operatorname{error}}(c, h) = \Pr_{x\sim\mathcal{D}}\bigl[h(x) \ne c(x)\bigr].$$

Definition 2.2 (PAC Learning (Valiant 1984)) An algorithm $\mathcal{A}$ is an (α, β)-*PAC learner* of a concept class $\mathcal{C}_d$ over X_d using hypothesis class $\mathcal{H}_d$ and sample size n if for all concepts $c \in \mathcal{C}_d$, all distributions $\mathcal{D}$ on X_d, given an input $D = (d_1, \dots, d_n)$, where $d_i = (x_i, c(x_i))$ with x_i drawn i.i.d. from $\mathcal{D}$ for all $i \in [n]$, algorithm $\mathcal{A}$ outputs a hypothesis $h \in \mathcal{H}_d$ satisfying

$$\Pr\bigl[\underset{\mathcal{D}}{\operatorname{error}}(c, h) \le \alpha\bigr] \ge 1 - \beta.$$

The probability is taken over the randomness of the learner $\mathcal{A}$ and the sample points chosen according to $\mathcal{D}$.

An Algorithm $\mathcal{A}_1$, whose inputs are d, α, β, and a set of samples (labeled examples) D, is a *PAC learner* of a concept class $\mathcal{C} = \{\mathcal{C}_d\}_{d\in\mathbb{N}}$ over $X = \{X_d\}_{d\in\mathbb{N}}$ using hypothesis class $\mathcal{H} = \{\mathcal{H}_d\}_{d\in\mathbb{N}}$ if there exists a polynomial $p(\cdot,\cdot,\cdot,\cdot)$ such that for all $d \in \mathbb{N}$ and $0 < \alpha, \beta < 1$, the Algorithm $\mathcal{A}_1$ $(d, \alpha, \beta, \cdot)$ is an (α, β)-PAC learner of the concept class $\mathcal{C}_d$ over X_d using hypothesis class $\mathcal{H}_d$ and sample size $n = p(d, \text{size}(c), 1/\alpha, \log(1/\beta))$.[4] If $\mathcal{A}$ runs in time polynomial in d, size(c), $1/\alpha$, $\log(1/\beta)$, we say that it is an *efficient PAC learner*. Also the learner is called a *proper* PAC learner if $\mathcal{H} = \mathcal{C}$, otherwise it is called an *improper* PAC learner.

A concept class $\mathcal{C} = \{\mathcal{C}_d\}_{d\in\mathbb{N}}$ over $X = \{X_d\}_{d\in\mathbb{N}}$ is *PAC learnable* using hypothesis class $\mathcal{H} = \{\mathcal{H}_d\}_{d\in\mathbb{N}}$ if there exists a PAC learner $\mathcal{A}$ learning $\mathcal{C}$ over X using hypothesis class $\mathcal{H}$. If $\mathcal{A}$ is an efficient PAC learner, we say that $\mathcal{C}$ is efficiently PAC learnable.

It is well known that improper learning is more powerful than proper learning. For example, Pitt and Valiant (1988) show that unless **RP** = **NP**, k-term DNF are not efficiently learnable by k-term DNF, whereas it is possible to learn a k-term DNF efficiently using k-CNF (Valiant 1984). For more background on learning theory, see (Kearns and Vazirani 1994).

Definition 2.3 (VC-Dimension (Vapnik and Chervonenkis 1971)) Let $\mathcal{C} = \{\mathcal{C}_d\}$ be a class of concepts over $X = \{X_d\}$. We say that $\mathcal{C}_d$ shatters a point set $Y \subset X_d$ if $|\{c(Y) : c \in \mathcal{C}_d\}| = 2^{|Y|}$, i.e., the concepts in $\mathcal{C}_d$ when restricted to Y produce all the $2^{|Y|}$ possible assignments on Y. The VC-dimension of $\mathcal{C}_d$ ($\mathit{VCDIM}(\mathcal{C}_d)$) is defined as the size of a maximum point set that is shattered by $\mathcal{C}_d$, as a function of d.

Theorem 2.4 (Blumer et al. 1989) *Let $\mathcal{C}_d$ be a concept class over X_d. There exists an (α, β)-PAC learner that learns $\mathcal{C}_d$ using $\mathcal{C}_d$ using $O((\mathit{VCDIM}(\mathcal{C}_d) \cdot \log(\frac{1}{\alpha}) + \log(\frac{1}{\beta}))/\alpha)$ samples.*

2.3 Private learning

Definition 2.5 (Private PAC Learning (Kasiviswanathan et al. 2011)) Let d, α, β be as in Definition 2.2 and $\epsilon > 0$. A concept class $\mathcal{C}$ is privately PAC learnable using $\mathcal{H}$ if there exists a learning Algorithm $\mathcal{A}_1$ that takes inputs $\epsilon, d, \alpha, \beta, D$, returns a hypothesis $\mathcal{A}(\epsilon, d, \alpha, \beta, D)$, and satisfies

Sample efficiency. The number of samples (labeled examples) in D is polynomial in $1/\epsilon$, d, size(c), $1/\alpha$, and $\log(1/\beta)$;

Privacy. For all d and $\epsilon, \alpha, \beta > 0$, algorithm $\mathcal{A}(\epsilon, d, \alpha, \beta, \cdot)$ is ϵ-differentially private (as formulated in Definition 2.1);

Utility. For all $\epsilon > 0$, algorithm $\mathcal{A}(\epsilon, \cdot, \cdot, \cdot, \cdot)$ PAC learns $\mathcal{C}$ using $\mathcal{H}$ (as formulated in Definition 2.2).

An Algorithm $\mathcal{A}_1$ is an efficient private PAC learner if it runs in time polynomial in $1/\epsilon$, d, size(c), $1/\alpha$, $\log(1/\beta)$. Also the private learner is called *proper* if $\mathcal{H} = \mathcal{C}$, otherwise it is called *improper*.

[4] The definition of PAC learning usually only requires that the sample complexity is polynomial in $1/\beta$ (rather than $\log(1/\beta)$). However, these two requirements are equivalent (see, e.g., Kearns and Vazirani 1994, Sect. 4.2).

Remark 2.6 The privacy requirement in Definition 2.5 is a worst-case requirement. That is, Inequality (1) must hold for every pair of neighboring databases D, D' (even if these databases are not consistent with any concept in $\mathcal{C}$). In contrast, the utility requirement is an average-case requirement, where we only require the learner to succeed with high probability over the distribution of the databases. This qualitative difference between the utility and privacy of private learners is crucial. A wrong assumption on how samples are formed that leads to a meaningless outcome can usually be replaced with a better one with very little harm. No such amendment is possible once privacy is lost due to a wrong assumption. See Kasiviswanathan et al. (2011) for further discussion.

Note also that each entry d_i in a database D is a labeled example. That is, we protect the privacy of both the example and its label.

Observation 2.7 *The computational separation between proper and improper learning also holds when we add the privacy constraint. That is, unless* $\boldsymbol{RP} = \boldsymbol{NP}$, *no proper private learner can learn k-term DNF, whereas there exists an efficient improper private learner that can learn k-term DNF using a k-CNF. The efficient k-term DNF learner of* Valiant (1984) *uses statistical queries* (*SQ*) (Kearns 1998), *which can be simulated efficiently and privately as shown by* Blum et al. (2005), Kasiviswanathan et al. (2011).

More generally, such a gap can be shown for any concept class that cannot be properly PAC learned, but can be efficiently learned (*improperly*) *in the statistical queries model.*

2.4 Concentration bounds

Chernoff bounds give exponentially decreasing bounds on the tails of distributions. Specifically, let $X_1, \ldots, X_n$ be independent random variables where $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$ for some $0 < p < 1$. Clearly, $\mathbb{E}[\sum_i X_i] = pn$. Chernoff bounds show that the sum is concentrated around this expected value: For every $0 < \delta \leq 1$,

$$\begin{aligned}
&\Pr\left[\sum_i X_i \geq (1+\delta)\,\mathbb{E}\left[\sum_i X_i\right]\right] \leq \exp\left(-\mathbb{E}\left[\sum_i X_i\right]\delta^2/3\right),\\
&\Pr\left[\sum_i X_i \leq (1-\delta)\,\mathbb{E}\left[\sum_i X_i\right]\right] \leq \exp\left(-\mathbb{E}\left[\sum_i X_i\right]\delta^2/2\right),\\
&\Pr\left[\left|\sum_i X_i - \mathbb{E}\left[\sum_i X_i\right]\right| \geq \delta\right] \leq 2\cdot\exp\left(-2\delta^2/n\right).
\end{aligned} \tag{2}$$

The first two inequalities are known as the multiplicative Chernoff bounds (Chernoff 1952), and the last inequality is known as the Chernoff-Hoeffding bound (Hoeffding 1963).

3 Proper learning vs. proper private learning

We begin by recalling the upper bound on the sample (database) size for private learning from Kasiviswanathan et al. (2011). The bound in Kasiviswanathan et al. (2011) is for agnostic learning, and we restate it for (non-agnostic) PAC learning using the following notion of α-representation:

Definition 3.1 We say that a hypothesis class $\mathcal{H}_d$ α-represents a concept class $\mathcal{C}_d$ over the domain X_d if for every $c \in \mathcal{C}_d$ and every distribution $\mathcal{D}$ on X_d there exists a hypothesis $h \in \mathcal{H}_d$ such that $\mathrm{error}_{\mathcal{D}}(c, h) \leq \alpha$.

Theorem 3.2 (Kasiviswanathan et al. (2011), restated) *Assume that there is a hypothesis class $\mathcal{H}_d$ that $\alpha/2$-represents a concept class $\mathcal{C}_d$. Then, for every $0 < \beta < 1$, there exists a private PAC learner for $\mathcal{C}_d$ using $\mathcal{H}_d$ that uses $O((\log(|\mathcal{H}_d|) + \log(1/\beta))/(\epsilon\alpha))$ samples, where ϵ, α, and β are the parameters of the private learner. The learner might not be efficient.*

In other words, using Theorem 3.2 the number of samples that suffices for learning a concept class $\mathcal{C}_d$ is logarithmic in the size of the smallest hypothesis class that α-represents $\mathcal{C}_d$. For comparison, the number of samples required for learning $\mathcal{C}_d$ non-privately is characterized by the VC-dimension of $\mathcal{C}_d$ (by the lower bound of Ehrenfeucht et al. (1989) and the upper bound of Blumer et al. (1989)).

In the following, we will investigate private learning of the following simple concept class. Let $T = 2^d$ and $X_d = \{1, \dots, T\}$. Define the concept class $\mathtt{POINT}_d$ to be the set of points over $\{1, \dots, T\}$:

Definition 3.3 (Concept Class $\mathtt{POINT}_d$) For $j \in [T]$, define $c_j : [T] \to \{0, 1\}$ as $c_j(x) = 1$ if $x = j$, and $c_j(x) = 0$ otherwise. Furthermore, define $\mathtt{POINT}_d = \{c_j\}_{j \in [T]}$.

We note that we use the set $\{1, \dots, T\}$ for notational convenience only—when discussing the concept class $\mathtt{POINT}_d$ we never use the fact that the elements in T are integer numbers.

The class $\mathtt{POINT}_d$ trivially α-represents itself, and hence, we get using Theorem 3.2 that it is (properly) PAC learnable using $O((\log(|\mathtt{POINT}_d|) + \log(1/\beta))/(\epsilon\alpha)) = O((d + \log(1/\beta))/(\epsilon\alpha))$ samples. For completeness, we give an efficient implementation of this learner.

Lemma 3.4 *There is an efficient proper private PAC learner for* $\mathtt{POINT}_d$ *that uses* $O((d + \log(1/\beta))/\epsilon\alpha)$ *samples.*

Proof We adapt the learner of Kasiviswanathan et al. (2011). Let $\mathtt{POINT}_d = \{c_1, \dots, c_{2^d}\}$. The learner uses the exponential mechanism of McSherry and Talwar (2007). Let $D = ((x_1, y_1), \dots, (x_m, y_m))$ be a database of samples (the labels y_i's are assumed to be consistent with some concept in $\mathtt{POINT}_d$). Define for every $c_j \in \mathtt{POINT}_d$,

$$q(D, c_j) = -\left|\left\{i : y_i \neq c_j(x_i)\right\}\right|,$$

i.e., $q(D, c_j)$ is negative of the number of points in D misclassified by c_j. The private learner $\mathcal{A}$ is defined as follows: output hypothesis $c_j \in \mathtt{POINT}_d$ with probability proportional to $\exp(\epsilon \cdot q(D, c_j)/2)$. Since the exponential mechanism is ϵ-differentially private (McSherry and Talwar 2007), $\mathcal{A}$ is ϵ-differentially private. By Kasiviswanathan et al. (2011), if $m = O((d + \log(1/\beta))/(\epsilon\alpha))$, then $\mathcal{A}$ is also a proper PAC learner.

We now show that $\mathcal{A}$ can be implemented efficiently. Implementing the exponential mechanism requires computing $q(D, c_j)$ for $1 \leq j \leq 2^d$. However, $q(D, c_j)$ is same for all $j \notin \{x_1, \dots, x_m\}$ and can be computed in $O(m)$ time, that is, $q(D, c_j) = q_D$, where $q_D = -|\{i : y_i = 1\}|$. Also for any $j \in \{x_1, \dots, x_m\}$, the value of $q(D, c_j)$ can be computed in $O(m)$ time. Let

$$P = \left(\sum_{j \in \{x_1, \dots, x_m\}} \exp\bigl(\epsilon \cdot q(D, c_j)/2\bigr)\right) + \bigl(2^d - m\bigr)\exp(\epsilon \cdot q_D/2).$$

The Algorithm $\mathcal{A}_1$ can be efficiently implemented as the following sampling procedure:

1. For $j \in \{x_1, \ldots, x_m\}$, with probability $\exp(\epsilon \cdot q(D, c_j)/2)/P$, output c_j.
2. With probability $(2^d - m) \cdot \exp(\epsilon \cdot q_D/2)/P$, pick uniformly at random a hypothesis from $\texttt{POINT}_d \setminus \{c_{x_1}, \ldots, c_{x_m}\}$ and output it. □

3.1 Separation between proper learning and proper private learning

We now show that private learners may require many more samples than non-private ones. We prove that for any proper private earner for the concept class $\texttt{POINT}_d$ the required number of samples is at least logarithmic in the size of the concept class, matching Theorem 3.2, whereas there exists non-private proper learners for $\texttt{POINT}_d$ that use only a constant number of samples.

To prove the lower bound, we show that a large collection of m-record databases $D_1, \ldots, D_N$ exists, with the property that every PAC learner has to output a different hypothesis for each of these databases (recall that in our context a database is a collection of labeled examples, supposedly drawn from some distribution and labeled consistently with some target concept). As any two databases D_a and D_b differ on at most m entries, differential privacy implies that a private learner must output on input D_a the hypothesis that is accurate for D_b (and not accurate for D_a) with probability at least $(1 - \beta) \cdot \exp(-\epsilon m)$. Since this holds for every pair of databases, unless m is large enough we get that the private learner's output on D_a is, with high probability, a hypothesis that is not accurate for D_a.

In Theorem 3.6, we prove a general lower bound on the sample complexity of private learning of a class $\mathcal{C}_d$ by a hypothesis classes $\mathcal{H}_d$ that is α-minimal for $\mathcal{C}_d$ as defined in Definition 3.5. In Corollary 3.8, we prove that Theorem 3.6 implies the claimed lower bound for proper private learning of $\texttt{POINT}_d$. In Lemma 3.9, we improve this lower bound for $\texttt{POINT}_d$ by a factor of $1/\alpha$.

Definition 3.5 If $\mathcal{H}_d$ α-represents $\mathcal{C}_d$, and every $\mathcal{H}'_d \subsetneq \mathcal{H}_d$ does not α-represent $\mathcal{C}_d$, then we say that $\mathcal{H}_d$ is *α-minimal* for $\mathcal{C}_d$.

Theorem 3.6 *Let $\mathcal{H}_d$ be an α-minimal representation for $\mathcal{C}_d$. Then, any private PAC learner that learns $\mathcal{C}_d$ using $\mathcal{H}_d$ requires $\Omega((\log(|\mathcal{H}_d|) + \log(1/\beta))/\epsilon)$ samples, where ϵ, α, and β are the parameters of the private learner.*

Proof Let $\mathcal{C}_d$ be a class of concepts over the domain X_d and let $\mathcal{H}_d$ be α-minimal for $\mathcal{C}_d$. Since for every $h \in \mathcal{H}_d$, the class $\mathcal{H}_d \setminus \{h\}$ does not α-represent $\mathcal{C}_d$, we get that there exists a concept $c_h \in \mathcal{C}_d$ and a distribution $\mathcal{D}_h$ on X_d such that on inputs drawn from $\mathcal{D}_h$ and labeled by c_h, every PAC learner (that learns $\mathcal{C}_d$ using $\mathcal{H}_d$) has to output h with probability at least $1 - \beta$.

Let $\mathcal{A}$ be a private learner that learns $\mathcal{C}_d$ using $\mathcal{H}_d$, and suppose $\mathcal{A}$ uses m samples. We next show that for every $h \in \mathcal{H}_d$ there exists a database $D_h \in X_d^m$ on which $\mathcal{A}$ has to output h with probability at least $1 - \beta$. To see that, note that if $\mathcal{A}$ is run on m examples chosen i.i.d. from the distribution $\mathcal{D}_h$ and labeled according to c_h, then $\mathcal{A}$ outputs h with probability at least $1 - \beta$ (where the probability is taken over the randomness of $\mathcal{A}$ and the sample points chosen according to $\mathcal{D}$). Hence, a collection of m labeled examples over which $\mathcal{A}$ outputs h with probability at least $1 - \beta$ exists, and D_h is set to contain these m samples.

Take $h, h' \in \mathcal{H}_d$ such that $h \neq h'$ and consider the two corresponding databases D_h and $D_{h'}$ with m entries each. Clearly, they differ in at most m entries, and hence, we get by the differential privacy of $\mathcal{A}$ that

$$\begin{aligned}\Pr\big[\mathcal{A}(D_h)=h'\big] &\geq \exp(-\epsilon m)\cdot \Pr\big[\mathcal{A}(D_{h'})=h'\big]\\ &\geq \exp(-\epsilon m)\cdot(1-\beta).\end{aligned}$$

Since the above inequality holds for every two databases corresponding to a pair of hypotheses in $\mathcal{H}$, we fix an arbitrary $h\in\mathcal{H}$ and get,

$$\begin{aligned}\Pr\big[\mathcal{A}(D_h)\neq h\big]=\Pr\big[\mathcal{A}(D_h)\in\mathcal{H}_d\setminus\{h\}\big]&=\sum_{h'\in\mathcal{H}_d\setminus\{h\}}\Pr\big[\mathcal{A}(D_h)=h'\big]\\ &\geq(|\mathcal{H}_d|-1)\cdot\exp(-\epsilon m)\cdot(1-\beta).\end{aligned}$$

On the other hand, we chose D_h such that $\Pr[\mathcal{A}(D_h)=h]\geq 1-\beta$, equivalently, $\Pr[\mathcal{A}(D_h)\neq h]\leq\beta$. Therefore, $(|\mathcal{H}_d|-1)\cdot\exp(-\epsilon m)\cdot(1-\beta)\leq\beta$. Solving the last inequality for m, we get $m=\Omega((\log(|\mathcal{H}_d|)+\log(1/\beta))/\epsilon)$ as required. □

Using Theorem 3.6, we now prove a lower bound on the number of samples needed for proper private learning concept class $\mathtt{POINT}_d$.

Proposition 3.7 $\mathtt{POINT}_d$ *is* α*-minimal for itself for every* $\alpha<1$.

Proof Clearly, $\mathtt{POINT}_d$ α-represents itself. To show minimality, consider a subset $\mathcal{H}'_d\subsetneq\mathtt{POINT}_d$, where $c_i\notin\mathcal{H}'_d$. Under the distribution $\mathcal{D}$ that chooses i with probability one, $\mathrm{error}_{\mathcal{D}}(c_i,c_j)=1$ for all $j\neq i$. Hence, $\mathcal{H}'_d$ does not α-represent $\mathtt{POINT}_d$. □

The VC-dimension of $\mathtt{POINT}_d$ is one.[5] It is well known that a standard (non-private) proper learner uses approximately VC-dimension number of samples to learn a concept class (Blumer et al. 1989). In contrast, we get that far more samples are needed for any proper private learner for $\mathtt{POINT}_d$. The following corollary follows directly from Theorem 3.6 and Proposition 3.7:

Corollary 3.8 *Every proper private PAC learner for* $\mathtt{POINT}_d$ *requires* $\Omega((d+\log(1/\beta))/\epsilon)$ *samples.*

We now show that the lower bound for $\mathtt{POINT}_d$ can be improved by a factor of $1/\alpha$, matching (up to constant factors) the upper bound in Theorem 3.2.

Lemma 3.9 *Every proper private PAC learner for* $\mathtt{POINT}_d$ *requires* $\Omega((d+\log(1/\beta))/(\epsilon\alpha))$ *samples.*

Proof Define the distributions $\mathcal{D}_i$ (where $2\leq i\leq T$) on X_d as follows: point 1 is picked with probability $1-\alpha$ and point i is picked with probability α. The support of $\mathcal{D}_i$ is on points 1 and i.

We say a database $D=(d_1,\ldots,d_m)$ where $d_j=(x_j,y_j)$ for all $j\in[m]$ is *good* for distribution $\mathcal{D}_i$ if at most $2\alpha m$ points from $x_1,\ldots,x_m$ equal i. Let D_i be a database

[5] Note that every singleton $\{j\}$ where $j\in[T]$ is shattered by $\mathtt{POINT}_d$ as $c_j(j)=1$ and $c_{j'}(j)=0$ for all $j'\neq j$. No set of two points $\{j,j'\}$ is shattered by $\mathtt{POINT}_d$ as $c_{j''}(j)=c_{j''}(j')=1$ for no $j''\in[T]$.

where $x_1, \dots, x_m$ are i.i.d. samples from $\mathcal{D}_i$ with $y_j = c_i(x_j)$ for all $j \in [m]$. By Chernoff bound, the probability that D_i is good for distribution $\mathcal{D}_i$ is at least $1 - \exp(-\alpha m/3)$. Let $\mathcal{A}$ be a proper private learner. On D_i, $\mathcal{A}$ has to output $h = c_i$ with probability at least $1 - \beta$ (otherwise, if $\mathcal{A}$ outputs some $h = c_j$, where $j \neq i$, then $\operatorname{error}_{\mathcal{D}_i}(c_i, h) = \operatorname{error}_{\mathcal{D}_i}(c_i, c_j) = \Pr_{x \sim \mathcal{D}_i}[c_i(x) \neq c_j(x)] > \alpha$, thus, violating the PAC learning condition for accuracy). Hence, the probability that either D_i is not good or $\mathcal{A}$ fails to return c_i on D_i is at most $\exp(-\alpha m/3) + \beta$. Therefore, with probability at least $1 - \beta - \exp(-\alpha m/3)$, the database D_i is good and $\mathcal{A}$ returns c_i on D_i. Thus, for every i there exists a database D_i that is good for $\mathcal{D}_i$ such that $\mathcal{A}$ returns c_i on D_i with probability at least $1 - \Gamma$, where $\Gamma = \beta + \exp(-\alpha m/3)$.

Fix such databases $D_2, \dots, D_T$. For every j, the databases D_2 and D_j differ in at most $4\alpha m$ entries (since each of them contains at most $2\alpha m$ entries that are not 1). Therefore, by the guarantees of differential privacy,

$$\Pr\bigl[\mathcal{A}(D_2) \in \{c_3, \dots, c_T\}\bigr] \geq (T-2)\exp(-4\epsilon\alpha m)(1-\Gamma) = \bigl(2^d - 2\bigr)\exp(-4\epsilon\alpha m)(1-\Gamma).$$

Algorithm $\mathcal{A}_1$ on input D_2 outputs c_2 with probability at least $1 - \Gamma$. Therefore,

$$\bigl(2^d - 2\bigr)\exp(-4\epsilon\alpha m)(1-\Gamma) \leq \Gamma.$$

Solving for m, we get the claimed bound. □

We conclude this section showing that every hypothesis class $\mathcal{H}$ that α-represents $\texttt{POINT}_d$ should have at least d hypotheses. Therefore, if we use Theorem 3.2 to learn $\texttt{POINT}_d$ we need $\Omega(\log d)$ samples.

Lemma 3.10 *Let $\alpha < 1/2$. $|\mathcal{H}| \geq d$ for every hypothesis class $\mathcal{H}$ that α-represents $\texttt{POINT}_d$.*

Proof Let $\mathcal{H}$ be a hypothesis class with $|\mathcal{H}| < d$. Consider a table whose $T = 2^d$ columns correspond to the possible 2^d inputs $1, \dots, T$, and whose $|\mathcal{H}|$ rows correspond to the hypotheses in $\mathcal{H}$. The (i, j)th entry in the table is 0 or 1 depending on whether the ith hypothesis gives 0 or 1 on input j. Since $|\mathcal{H}| < d = \log(T)$, at least two columns $j \neq j'$ are identical, that is, $h(j) = h(j')$ for every $h \in \mathcal{H}$. Consider the concept $c_j \in \texttt{POINT}_d$ (defined as $c_j(x) = 1$ if $x = j$, and 0 otherwise), and the distribution $\mathcal{D}$ with probability mass $1/2$ on both j and j'. We get that $\operatorname{error}_{\mathcal{D}}(c_j, h) \geq 1/2 > \alpha$ for all $h \in \mathcal{H}$ (since for any hypothesis $h(j) = h(j')$, the hypothesis either errs on j or on j'). Therefore, $\mathcal{H}$ does not α-represent $\texttt{POINT}_d$. □

4 Proper private learning vs. improper private learning

We now use $\texttt{POINT}_d$ to show a separation between proper and improper private PAC learning. One-way of achieving a smaller sample complexity is to use Theorem 3.2 to improperly learn $\texttt{POINT}_d$ with a hypothesis class $\mathcal{H}$ that α-represents $\texttt{POINT}_d$, but is of size smaller than $|\texttt{POINT}_d|$. By Lemma 3.10, we know that every such $\mathcal{H}$ should have at least d hypotheses.

In Sect. 4.1, we show that there does exist a $\mathcal{H}$ with $|\mathcal{H}| = O(d)$ that α-represents $\texttt{POINT}_d$. This immediately gives a separation—proper private learning $\texttt{POINT}_d$ requires

$\Omega_{\alpha,\beta,\epsilon}(d)$ samples, whereas $\mathtt{POINT}_d$ can be improperly privately learned using $O_{\alpha,\beta,\epsilon}(\log d)$ samples.[6]

We conclude that α-representing hypothesis classes can, hence, be a natural and powerful tool for constructing efficient private learners. One may even be tempted to think that no better learners exist, and furthermore, that the sample complexity of private learning is characterized by the size of the smallest hypothesis class that α-represents the concept class. Our second result, presented in Sect. 4.2, shows that this is not the case, and in fact, other techniques yield a much more efficient learner using only $O_{\alpha,\beta,\epsilon}(1)$ samples, and hence demonstrating the strongest possible separation between proper and improper private learners. The reader interested only in the stronger result may choose to skip directly to Sect. 4.2.

4.1 Improper private learning of $\mathtt{POINT}_d$ using $O_{\alpha,\beta,\epsilon}(\log d)$ samples

We next construct a private learner applying the construction of Theorem 3.2 to the class $\mathtt{POINT}_d$. For that we (randomly) construct a hypothesis class $\mathcal{H}_d$ that α-represents the concept class $\mathtt{POINT}_d$, where $|\mathcal{H}_d| = O_\alpha(d)$. Lemma 3.10 shows that this is optimal up to constant factors. In the rest of this section, a set $A \subseteq [T]$ represents the hypothesis h_A, where $h_A(i) = 1$ if $i \in A$ and $h_A(i) = 0$ otherwise.

To demonstrate the main idea of our construction, we begin with a construction of a hypothesis class $\mathcal{H}_d = \{A_1, \dots, A_k\}$ that α-represents $\mathtt{POINT}_d$, where $k = O(\sqrt{T}/\alpha) = O(\sqrt{2^d}/\alpha)$ (this should be compared to the size of $\mathtt{POINT}_d$ which is 2^d). Every $A_i \in \mathcal{H}_d$ is a subset of $\{1, \dots, T\}$, such that

(1) For every $j \in \{1, \dots, T\}$ there are more than $1/\alpha$ sets in $\mathcal{H}$ that contain j; and
(2) For every $1 \le i_1 < i_2 \le k$, $|A_{i_1} \cap A_{i_2}| \le 1$.

We next argue that the class $\mathcal{H}_d$ α-represents $\mathtt{POINT}_d$. For every concept $c_j \in \mathtt{POINT}_d$ there are hypotheses $A_1, \dots, A_p \in \mathcal{H}_d$ that contain j (where $p = \lfloor 1/\alpha \rfloor + 1$) and are otherwise disjoint (that is, the intersection between any two sets A_{i_1} and A_{i_2} is exactly j). Fix a distribution $\mathcal{D}$. For every A_i, $\mathrm{error}_{\mathcal{D}}(c_j, A_i) = \Pr_{\mathcal{D}}[A_i \setminus \{j\}]$. Since there are more than $1/\alpha$ such sets and the sets $A_i \setminus \{j\}$ are disjoint, there exists at least one set such that $\mathrm{error}_{\mathcal{D}}(c_j, A_i) \le \alpha$. Thus, $\mathcal{H}_d$ α-represents the concept class $\mathtt{POINT}_d$.

We want to show that there is a hypothesis class, whose size is $O(\sqrt{T}/\alpha)$, that satisfies the above two requirements. As an intermediate step, we show a construction of size $O(T)$. We consider a projective plane with T points and T lines (each line is a set of points) such that for any two points there is exactly one line containing them and for any two lines there is exactly one point contained in both of them. Such projective plane exists whenever $T = q^2 + q + 1$ for a prime power q (see, e.g., Hughes and Piper 1973). Furthermore, the number of lines passing through each point is $q + 1$. If we take the lines as the hypothesis class for $q \ge 1/\alpha$, then they satisfy the above requirements, thus, they α-represent $\mathtt{POINT}_d$. However, the number of hypotheses in the class is T and no progress was made.

We modify the above projective plane construction. We start with a projective plane with $2T$ points and choose a subset of the lines: We choose each line at random with probability $O(1/(\sqrt{T}\alpha))$. Since these lines are part of the projective plane, they satisfy the above requirement (2). It can be shown that with positive probability for at least half of the j's requirement (1) is satisfied and the number of chosen lines is $O(\sqrt{T}/\alpha)$. We choose such

[6] Remember, the notation $O_{\alpha,\beta,\epsilon}(g(n))$ is a shorthand for $O(h(\alpha, \beta, \epsilon) \cdot g(n))$ for some non-negative function h. Similarly, the notation $\Omega_{\alpha,\beta,\epsilon}(g(n))$.

lines, eliminate points that are contained in less than $1/\alpha$ chosen lines, and get the required construction with T points and $O(\sqrt{T}/\alpha)$ lines. The details of the last steps are omitted. We next show a much more efficient construction based on the above idea.

Lemma 4.1 *For every $\alpha < 1$, there is a hypothesis class $\mathcal{H}_d$ that α-represents $\texttt{POINT}_d$ such that $|\mathcal{H}_d| = O(d/\alpha^2)$.*

Proof We will show how to construct a hypothesis class $\mathcal{H}_d = \{S_1, \dots, S_k\}$, where every $S_i \in \mathcal{H}_d$ is a subset of $\{1, \dots, T\}$ and for every j

$$\begin{aligned}&\text{There are } p = \log T \cdot (1 + \lfloor 1/\alpha \rfloor) \text{ sets } A_1, \dots, A_p \text{ in } \mathcal{H}_d \text{ that contain } j \text{ such that}\\ &\text{for every } b \neq j, \text{ the point } b \text{ is contained in less than } \log T \text{ of the sets } A_1, \dots, A_p.\end{aligned} \tag{3}$$

First we show that $\mathcal{H}_d$ α-represents $\texttt{POINT}_d$. Fix a concept $c_j \in \texttt{POINT}_d$ and a distribution $\mathcal{D}$, and consider hypotheses $A_1, \dots, A_p$ in $\mathcal{H}_d$ that contain j. Since every point in these hypotheses is contained in less than $\log T$ sets,

$$\sum_{i=1}^{p} \Pr_{\mathcal{D}}\bigl[A_i \setminus \{j\}\bigr] < \log T \cdot \Pr_{\mathcal{D}}\Biggl[\bigcup_{i=1}^{p}\bigl(A_i \setminus \{j\}\bigr)\Biggr] \le \log T.$$

Thus, there exists at least one set A_i such that $\mathrm{error}_{\mathcal{D}}(c_j, A_i) = \Pr_{\mathcal{D}}[A_i \setminus \{j\}] \le \log T/p < \alpha$. This implies that $\mathcal{H}_d$ α-represents the concept class $\texttt{POINT}_d$.

We next show how to construct $\mathcal{H}_d$. Let $k = 8ep^2/\log T$ (that is, $k = O(\log T/\alpha^2)$). We choose k random subsets of $\{1, \dots, 2T\}$ of size $4pT/k$. We will show that a point j satisfies (3) with probability at least $3/4$. We assume $d \ge 16$ (and hence, $p \ge 16$ and $T \ge 16$).

Fix j. The expected number of sets that contain j is $k \cdot (4pT/k)/(2T) = 2p$, thus, by Chebyshev inequality, the probability that less than p sets contain j is less than $2/p \le 1/8$. We call this event BAD_1.

Let j be such that there are at least p sets that contain j and let $A_1, \dots, A_p$ be p of them. Notice that $A_1 \setminus \{j\}, \dots, A_p \setminus \{j\}$ are random subsets of $\{1, \dots, 2T\} \setminus \{j\}$ of size $(4pT/k) - 1$. Now fix $b \neq j$. The probability that a random subset of $\{1, \dots, 2T\} \setminus \{j\}$ of size $(4pT/k) - 1$ contains b is $(4pT/k - 1)/(2T - 1) < 2p/k$. For $\log T$ random sets of size $(4pT/k) - 1$, the probability that all of them contain b is less than $(2p/k)^{\log T}$. Thus, the probability that there is a $b \in \{1, \dots, 2T\}$, where $b \neq j$, and $\log T$ sets among $A_1, \dots, A_p$ such that these $\log T$ sets contains b is less than

$$\begin{aligned} 2T \cdot \binom{p}{\log T}(2p/k)^{\log T} &\le 2T \cdot (\mathrm{e}p/\log T)^{\log T}(2p/k)^{\log T} \quad \bigl(\text{where } \mathrm{e} = \exp(1)\bigr)\\ &= 2T \cdot \bigl(2\mathrm{e}p^2/(k \log T)\bigr)^{\log T}. \end{aligned}$$

By the choice of k, $2\mathrm{e}p^2/(k \log T) = 1/4$, thus, the above probability is at most $2T \cdot (1/4)^{\log T} = 2/T \le 1/8$. We call this event BAD_2.

To conclude, the probability that j does not satisfy (3) is the probability that either BAD_1 or BAD_2 happens which is at most $1/4$. Therefore, the expected number of j's that do not satisfy (3) is less than $T/2$. By Markov inequality, the probability that more than T points j do not satisfy (3) is less than $1/2$. We take $k = O(\log T/\alpha^2)$ subsets of $\{1, \dots, 2T\}$, denoted $S_1, \dots, S_k$, such that at least T points j satisfy (3). By the probabilistic argument above, such sets exist. Let V be a set of size T of the points that satisfy (3), and define

$\mathcal{H}_d = \{S_1 \cap V, \dots, S_k \cap V\}$. Finally, by a simple renaming, we can assume that $\mathcal{H}_d$ contains subsets of $\{1, \dots, T\}$ as required. □

From Lemma 4.1 and Theorem 3.2 we get:

Theorem 4.2 *There exists an improper private PAC learner for* $\texttt{POINT}_d$ *that uses* $O((\log d + \log \frac{1}{\alpha} + \log \frac{1}{\beta})/\epsilon\alpha)$ *samples, where* ϵ, α, *and* β *are the parameters of the private learner.*

There is a difference between the use of improper learning in Theorem 4.2 and typical use of improper learning in non-private settings. Typically, a non-private learner uses a hypothesis class that is *larger* than the size of concept class. This larger class enables learning in polynomial time. We get an improved sample complexity by learning using a hypothesis class whose size is *smaller* than the concept class.

4.2 Improper private learning of $\texttt{POINT}_d$ using $O_{\alpha,\beta,\epsilon}(1)$ samples

We now show a stronger separation result, namely, that $\texttt{POINT}_d$ can be privately (and efficiently) learned by an improper learner using just $O_{\alpha,\beta,\epsilon}(1)$ samples. We begin by presenting a non-private improper PAC learner $\mathcal{A}_1$ for $\texttt{POINT}_d$ that succeeds with only constant probability. Roughly, $\mathcal{A}_1$ applies a simple proper learner for $\texttt{POINT}_d$, and then modifies its outcome by adding random "noise". We then use sampling to convert $\mathcal{A}_1$ into a private learner $\mathcal{A}_2$; like $\mathcal{A}_1$ the probability that $\mathcal{A}_2$ succeeds in learning $\texttt{POINT}_d$ is only a constant. Later we amplify the success probability of $\mathcal{A}_2$ to get a private PAC learner. Both $\mathcal{A}_1$ and $\mathcal{A}_2$ are inefficient as they output hypotheses with exponential description length. However, using a pseudorandom function it is possible to compress the outputs of $\mathcal{A}_1$ and $\mathcal{A}_2$, and achieve a private learning algorithms whose running time is efficient. This is explained in Sect. 4.2.1.

Algorithm $\mathcal{A}_2$ described below is $\epsilon^\star$-differentially private, where $\epsilon^\star = \ln(4)$ is a fixed constant. To construct an ϵ-differentially private algorithm for every ϵ, we describe a transformation in Lemma 4.4 that takes a bigger sample and replaces some samples with $\star$ and executes $\mathcal{A}_2$ on the resulting sample. Therefore, we assume that some of the sample points given to $\mathcal{A}_1$ and $\mathcal{A}_2$ are $\star$.

Algorithm $\mathcal{A}_1$ Given a sample $z_1, \dots, z_m$, where every z_i is either a labeled example (x_i, y_i) or $\star$, Algorithm $\mathcal{A}_1$ performs the following:

1. If $z_1, \dots, z_m$ is not consistent with any concept in $\texttt{POINT}_d$, return $\perp$ (this happens only if for two indices $i, j \in [m]$ such that $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$ either (1) $x_i \neq x_j$ and $y_i = y_j = 1$ or (2) $x_i = x_j$ and $y_i \neq y_j$).
2. If $y_i = 0$ for all $i \in [m]$ such that $z_i \neq \star$, then let $c = \mathbf{0}$ (the all zero hypothesis); otherwise, let c be the (unique) hypothesis from $\texttt{POINT}_d$ that is consistent with the labeled examples in the sample.
3. Modify c at random to get a hypothesis h: for each $x \in [T]$ independently let $h(x) = 1 - c(x)$ with probability $\alpha/8$ and, otherwise let $h(x) = c(x)$. Return h.

We next argue that if the sample $z_1, \dots, z_m$ contains at least $2\ln(4)/\alpha$ examples $z_i = (x_i, y_i)$ such that each x_i is drawn i.i.d. according to a distribution $\mathcal{D}$ on $[T]$, and the examples are labeled consistently according to some $c_j \in \texttt{POINT}_d$, then $\Pr[\text{error}_{\mathcal{D}}(c_j, c) \geq$

$\alpha/2] \le 1/4$. If the examples are labeled consistently according to some $c_j \ne \mathbf{0}$, then $c \ne c_j$ only if $(j,1)$ is not in the sample and in this case $c = \mathbf{0}$. If $\Pr_{x\sim\mathcal{D}}[x = j] < \alpha/2$ and $(j,1)$ is not in the sample, then $c = \mathbf{0}$ and $\mathrm{error}_{\mathcal{D}}(c_j, \mathbf{0}) < \alpha/2$. Otherwise $\Pr_{x\sim\mathcal{D}}[x = j] \ge \alpha/2$; thus, the probability that all examples of the form (x_i, y_i) are not $(j,1)$ is at most $((1-\alpha/2)^{2/\alpha})^{\ln(4)} \le 1/4$ (as there are at least $2\ln(4)/\alpha$ such examples).

To see that $\mathcal{A}_1$ PAC learns $\mathtt{POINT}_d$ (with confidence at least $1/2$) note that,

$$\mathbb{E}_h\Big[\underset{\mathcal{D}}{\mathrm{error}}(c,h)\Big] = \mathbb{E}_h\,\mathbb{E}_{x\sim\mathcal{D}}\big[\big|h(x)-c(x)\big|\big] = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_h\big[\big|h(x)-c(x)\big|\big] = \frac{\alpha}{8},$$

and hence, using Markov's inequality,

$$\Pr_h\Big[\underset{\mathcal{D}}{\mathrm{error}}(c,h) \ge \alpha/2\Big] \le 1/4.$$

Combining this with $\Pr[\mathrm{error}_{\mathcal{D}}(c_j, c) \ge \alpha/2] \le 1/4$ and $\mathrm{error}_{\mathcal{D}}(c_j, h) \le \mathrm{error}_{\mathcal{D}}(c_j, c) + \mathrm{error}_{\mathcal{D}}(c, h)$, implies that $\Pr[\mathrm{error}_{\mathcal{D}}(c_j, h) \ge \alpha] \le 1/2$.

Algorithm $\mathcal{A}_2$ We now modify the learner $\mathcal{A}_1$ to get a private learner $\mathcal{A}_2$ (a similar idea was used in Kasiviswanathan et al. (2011) for learning parity functions). Given a sample $z_1, \dots, z_{m'}$, where every z_i is either a labeled example (x_i, y_i) or $\star$, Algorithm $\mathcal{A}_2$ performs the following:

1. With probability $\alpha/8$, return $\perp$.
2. Construct a set $S \subseteq [m']$ by picking each element of $[m']$ with probability $p = \alpha/4$.
3. Run the non-private learner $\mathcal{A}_1$ on the examples indexed by S.

Claim 4.3 *Let* $\alpha < 1/2$, $\epsilon^\star = \ln(4)$, *and* $\beta^\star = 3/4$. *Algorithm* $\mathcal{A}_2$ *is an* $\epsilon^\star$*-differentially private* $(\alpha, \beta^\star)$*-PAC learner for the class* $\mathtt{POINT}_d$ *provided that it is given a sample which contains at least* $32\ln(4)/\alpha^2$ *labeled examples* (*i.e.*, $m' \ge 32\ln(4)/\alpha^2$).

Proof We first show that $\mathcal{A}_2$ PAC learns $\mathtt{POINT}_d$ with confidence at least $\beta^\star = 3/4$. Let S be the set chosen by $\mathcal{A}_2$. The expected number of samples is at least $p \cdot (32\ln(4))/\alpha^2 = 8\ln(4)/\alpha$. By Chernoff bound, the probability that the sample indexed by S contains less than $2\ln(4)/\alpha$ (in fact, $4\ln(4)/\alpha$) samples is less than $\exp(-\ln(4)/\alpha) < 1/16$ (since $\mathcal{A}_2$ gets at least $32\ln(4)/\alpha^2$ labeled examples and $\alpha < 1/2$). Algorithm $\mathcal{A}_2$ can err only when either $\mathcal{A}_1$ does not get $2\ln(4)/\alpha$ labeled examples, or when $\mathcal{A}_1$ errs, or when $\mathcal{A}_2$ returns $\perp$ in Step (1). Therefore, we get that $\mathcal{A}_2$ PAC learns $\mathtt{POINT}_d$ with accuracy parameter $\alpha' = \alpha$ and confidence parameter $\beta' = 1/16 + 1/2 + \alpha/8 \le 3/4$.

We next show that $\mathcal{A}_2$ is $\epsilon^\star$-differentially private. Let D, D' be two neighboring databases, and assume that they differ on the ith entry. Recall that after sampling S, one of them can be consistent with some c_j, while the other might not be consistent. First let us analyze the probability of $\mathcal{A}_2$ outputting $\perp$:

$$\begin{aligned}
\frac{\Pr[\mathcal{A}_2(D) = \perp]}{\Pr[\mathcal{A}_2(D') = \perp]} &= \frac{p\cdot\Pr[\mathcal{A}_2(D) = \perp \mid i \in S] + (1-p)\cdot\Pr[\mathcal{A}_2(D) = \perp \mid i \notin S]}{p\cdot\Pr[\mathcal{A}_2(D') = \perp \mid i \in S] + (1-p)\cdot\Pr[\mathcal{A}_2(D') = \perp \mid i \notin S]} \\
&\le \frac{p\cdot 1 + (1-p)\cdot\Pr[\mathcal{A}_2(D) = \perp \mid i \notin S]}{p\cdot 0 + (1-p)\cdot\Pr[\mathcal{A}_2(D') = \perp \mid i \notin S]} \\
&= \frac{p}{(1-p)\cdot\Pr[\mathcal{A}_2(D') = \perp \mid i \notin S]} + 1 \le \frac{8p}{\alpha(1-p)} + 1,
\end{aligned}$$

where the last equality follows by noting that if $i \notin S$ then $\mathcal{A}_2$ is equally likely to output $\perp$ on D and D', and the last inequality follows as $\perp$ is returned with probability $\alpha/8$ in Step (1) of Algorithm $\mathcal{A}_2$.

For the more interesting case, where $\mathcal{A}_2$ outputs a hypothesis h, we get:

$$\begin{aligned}\frac{\Pr[\mathcal{A}_2(D)=h]}{\Pr[\mathcal{A}_2(D')=h]} &= \frac{p\cdot\Pr[\mathcal{A}_2(D)=h \mid i\in S]+(1-p)\cdot\Pr[\mathcal{A}_2(D)=h \mid i\notin S]}{p\cdot\Pr[\mathcal{A}_2(D')=h \mid i\in S]+(1-p)\cdot\Pr[\mathcal{A}_2(D')=h \mid i\notin S]}\\ &\le \frac{p\cdot\Pr[\mathcal{A}_2(D)=h \mid i\in S]+(1-p)\cdot\Pr[\mathcal{A}_2(D)=h \mid i\notin S]}{p\cdot 0+(1-p)\cdot\Pr[\mathcal{A}_2(D')=h \mid i\notin S]}\\ &= \frac{p}{1-p}\cdot\frac{\Pr[\mathcal{A}_2(D)=h \mid i\in S]}{\Pr[\mathcal{A}_2(D)=h \mid i\notin S]}+1,\end{aligned}$$

where the last equality uses the fact that if $i \notin S$ then $\mathcal{A}_2$ is equally likely to output h on D and D'. If in D the ith row is $\star$, then $\Pr[\mathcal{A}_2(D)=h \mid i\in S]=\Pr[\mathcal{A}_2(D)=h \mid i\notin S]=\Pr[\mathcal{A}_2(D')=h \mid i\notin S]$, and the above ratio is bounded by $p/(1-p)+1=1/(1-\alpha/4)<4/3<e^{\epsilon^\star}$.

To complete the proof, we need to bound the ratio of $\Pr[\mathcal{A}_2(D)=h \mid i\in S]$ to $\Pr[\mathcal{A}_2(D)=h \mid i\notin S]$ when $z_i=(x_i,y_i)$.

$$\begin{aligned}&\frac{\Pr[\mathcal{A}_2(D)=h \mid i\in S]}{\Pr[\mathcal{A}_2(D)=h \mid i\notin S]}\\ &\quad= \frac{\sum_{R\subseteq[m']\setminus\{i\}}\Pr[\mathcal{A}_2(D)=h \mid S=R\cup\{i\}]\cdot\Pr[\mathcal{A}_2 \text{ selects } R \text{ from } [m']\setminus\{i\}]}{\sum_{R\subseteq[m']\setminus\{i\}}\Pr[\mathcal{A}_2(D)=h \mid S=R]\cdot\Pr[\mathcal{A}_2 \text{ selects } R \text{ from } [m']\setminus\{i\}]}\\ &\quad\le \max_{R\subseteq[m']\setminus\{i\}}\frac{\Pr[\mathcal{A}_2(D)=h \mid S=R\cup\{i\}]}{\Pr[\mathcal{A}_2(D)=h \mid S=R]}. \qquad (4)\end{aligned}$$

In the max in (4), we only need to consider sets R such that the sample labeled by the elements in R is consistent, that is, $\Pr[\mathcal{A}_2(D)=h \mid S=R]>0$. Now having or not having access to (x_i,y_i) can only affect the choice of $h(x_i)$, and since $\mathcal{A}_1$ flips the output with probability $\alpha/8$, we get

$$\max_{R\subseteq[m']\setminus\{i\}}\frac{\Pr[\mathcal{A}_2(D)=h \mid S=R\cup\{i\}]}{\Pr[\mathcal{A}_2(D)=h\ |S=R]}\le\frac{1-\alpha/8}{\alpha/8}\le\frac{8}{\alpha}.$$

Putting everything together, we get

$$\frac{\Pr[\mathcal{A}_2(D)=h]}{\Pr[\mathcal{A}_2(D')=h]}\le\frac{8p}{\alpha(1-p)}+1=\frac{8}{(4-\alpha)}+1<3+1=e^{\epsilon^\star}. \qquad \square$$

Algorithm $\mathcal{A}_2$ is $\epsilon^\star$-differentially private for some fixed $\epsilon^\star$. We reduce $\epsilon^\star$ to any desired ϵ using the following lemma (implicit in Kasiviswanathan et al. (2011)). In this lemma, we assume that the learning algorithm can handle "undefined entries", i.e., entries of the form $\star$.[7]

[7] These $\star$ entries cannot be simply removed as the question if two databases are neighbors depends on the locations of the $\star$'s.

Lemma 4.4 *Let $\mathcal{A}$ be an $\epsilon^\star$-differentially private algorithm. Construct an algorithm $\mathcal{B}$ that on input a database $D=(d_1,\dots,d_n)$ constructs a new database D_s whose ith entry is d_i with probability $f(\epsilon,\epsilon^\star)=(\exp(\epsilon)-1)/(\exp(\epsilon^\star)+\exp(\epsilon)-\exp(\epsilon-\epsilon^\star)-1)$ and $\star$ otherwise, and then runs $\mathcal{A}$ on D_s. Then, $\mathcal{B}$ is ϵ-differentially private.*

Proof Let D, D' be neighboring databases, and assume they differ on the ith entry. Let $S\subseteq[n]$ denote the indices of the random set of entries that are not changed to $\star$. Let $q=f(\epsilon,\epsilon^\star)$. Since D and D' differ in just the ith entry, for any outcome t, $\Pr[\mathcal{A}(D_s)=t|i\notin S]=\Pr[\mathcal{A}(D'_s)=t|i\notin S]$. Thus,

$$\begin{aligned}
&\frac{\Pr[\mathcal{B}(D)=t]}{\Pr[\mathcal{B}(D')=t]}\\
&\quad=\frac{q\cdot\Pr[\mathcal{A}(D_s)=t|i\in S]+(1-q)\cdot\Pr[\mathcal{A}(D_s)=t|i\notin S]}{q\cdot\Pr[\mathcal{A}(D'_s)=t|i\in S]+(1-q)\cdot\Pr[\mathcal{A}(D_s)=t|i\notin S]}\\
&\quad=\frac{\sum_{R\subseteq[n]\setminus\{i\}}\Pr[S\setminus\{i\}=R]\cdot(q\cdot\Pr[\mathcal{A}(D_s)=t|S=R\cup\{i\}]+(1-q)\cdot\Pr[\mathcal{A}(D_s)=t|S=R])}{\sum_{R\subseteq[n]\setminus\{i\}}\Pr[S\setminus\{i\}=R]\cdot(q\cdot\Pr[\mathcal{A}(D'_s)=t|S=R\cup\{i\}]+(1-q)\cdot\Pr[\mathcal{A}(D_s)=t|S=R])}\\
&\quad\le\max_{R\subseteq[n]\setminus\{i\}}\frac{q\cdot\Pr[\mathcal{A}(D_s)=t|S=R\cup\{i\}]+(1-q)\cdot\Pr[\mathcal{A}(D_s)=t|S=R]}{q\cdot\Pr[\mathcal{A}(D'_s)=t|S=R\cup\{i\}]+(1-q)\cdot\Pr[\mathcal{A}(D_s)=t|S=R]}\\
&\quad\le\max_{R\subseteq[n]\setminus\{i\}}\frac{q\cdot\exp(\epsilon^\star)\cdot\Pr[\mathcal{A}(D_s)=t|S=R]+(1-q)\cdot\Pr[\mathcal{A}(D_s)=t|S=R]}{q\cdot\exp(-\epsilon^\star)\cdot\Pr[\mathcal{A}(D_s)=t|S=R]+(1-q)\cdot\Pr[\mathcal{A}(D_s)=t|S=R]}\\
&\quad=\frac{1+q\cdot(\exp(\epsilon^\star)-1)}{1-q\cdot(1-\exp(-\epsilon^\star))}=\exp(\epsilon).
\end{aligned}$$

The last inequality follows because by the guarantees of differential privacy

$$\Pr\bigl[\mathcal{A}(D_s)=t|S=R\cup\{i\}\bigr]\le\exp\bigl(\epsilon^\star\bigr)\cdot\Pr\bigl[\mathcal{A}(D_s)=t|S=R\cup\emptyset\bigr],$$

and

$$\begin{aligned}
\Pr\bigl[\mathcal{A}\bigl(D'_s\bigr)=t|S=R\cup\{i\}\bigr]&\ge\exp\bigl(-\epsilon^\star\bigr)\cdot\Pr\bigl[\mathcal{A}\bigl(D'_s\bigr)=t|S=R\cup\emptyset\bigr]\\
&=\exp\bigl(-\epsilon^\star\bigr)\cdot\Pr\bigl[\mathcal{A}(D_s)=t|S=R\cup\emptyset\bigr]\quad\bigl(\text{as } R\subseteq[n]\setminus\{i\}\bigr).
\end{aligned}$$

Therefore, $\mathcal{B}$ is an ϵ-differentially private algorithm. □

Claim 4.5 *Let $\alpha<1/2$, $0<\beta\le 1$ and $0<\epsilon<1$. There exists an ϵ-differentially private (α,β)-PAC learner for the class $\mathtt{POINT}_d$ which uses a sample of size* poly$(1/\epsilon, 1/\alpha, \log(1/\beta))$.

Proof We first apply the transformation described in Lemma 4.4 on Algorithm $\mathcal{A}_2$. Call the resulting Algorithm $\mathcal{A}_3$. In this case $\epsilon^\star=\ln(4)$ and

$$f\bigl(\epsilon,\epsilon^\star\bigr)=\frac{\exp(\epsilon)-1}{\exp(\epsilon^\star)+\exp(\epsilon)-\exp(\epsilon-\epsilon^\star)-1}>\epsilon/6$$

for $\epsilon<1$ (since $\exp(\epsilon)-1\ge\epsilon$). By Chernoff bound, if we take a sample of size $384\ln(4)/(\epsilon\alpha^2)$ and choose each example with probability at least $\epsilon/6$, then with probability at least $1-\exp(-32\ln(4))$ the resulting sample size is at least $32\ln(4)/\alpha^2$. Now if given $32\ln(4)/\alpha^2$ samples, $\mathcal{A}_2$ returns a hypothesis with error at most α with probability at least $1/4$. Therefore, the total probability that $\mathcal{A}_2$ returns a hypothesis with error greater than α is at most $\exp(-32\ln(4)) + 3/4$ (the first term comes from $\mathcal{A}_2$ not getting enough samples and the second term comes from $\mathcal{A}_2$ returning a hypothesis with error greater than α even after getting enough samples). Thus, the algorithm resulting from the transformation described in Lemma 4.4 returns a hypothesis with error at most α with probability at least $1 - (\exp(-32\ln(4)) + 3/4) > 1/5$ (i.e., confidence parameter of the above learner is $4/5$).

We next privately boost the confidence parameter of the learner from $4/5$ to any value $\beta > 0$ similar to Kasiviswanathan et al. (2011). We execute $N = \log_{5/4}(5/\beta)$ times algorithm $\mathcal{A}_3$ with accuracy $\alpha/8$ and disjoint samples; we get N hypotheses $\mathrm{Hyp} = \{h_1, \ldots, h_N\}$. With probability at least $1 - (4/5)^N = 1 - \beta/5$ at least one of the hypotheses has error less than $\alpha/8$. We need to privately choose such a hypothesis. To achieve this goal we take a fresh sample of size $m = 24\ln(3/\beta^2)/(\epsilon\alpha)$, compute the mistake of each hypothesis on this sample, and use the exponential mechanism of McSherry and Talwar (2007) to choose the hypothesis. Specifically, let m_i be the number of errors that hypothesis h_i has on the sample; return the hypothesis h_i with probability

$$\frac{\exp(-\epsilon m_i/2)}{\sum_{j=1}^{N}\exp(-\epsilon m_j/2)}.$$

Changing one example can reduce m_i by at most 1 and increase m_j by at most one for every $i \neq j$ (thus, increasing $\sum_{j=1}^{N}\exp(-\epsilon m_j/2)$ by at most $\exp(-\epsilon/2)$); therefore the selection of the hypothesis is ϵ-differentially private.

We next argue that with probability at least $1 - \beta$ the selected hypothesis h_i has error at most α. With probability at least $1 - \beta/5$, at least one of the hypotheses from Hyp has error less than $\alpha/8$; by Chernoff bound with probability at least $1 - \beta^2/3$ this hypothesis has empirical error[8] at most $\alpha/4$. Let us call $\mathcal{E}_1$ the event that there exists a hypothesis with error less than $\alpha/8$ and empirical error less than $\alpha/4$ in Hyp. Event $\mathcal{E}_1$ happens with probability at least $(1 - \beta/5)(1 - \beta^2/3) > 1 - (\beta/5 + \beta^2/3)$.

On the other hand, the probability that a hypothesis h_j that has error greater than α has empirical error $\leq \alpha/2$ is less than $\beta^2/3$. By the union bound, the probability that there is such hypothesis in Hyp is at most $\beta/3$ (since $N \leq 1/\beta$ for $\beta \leq 0.01$). Let us call $\mathcal{E}_2$ the event that all hypotheses in Hyp with error greater than α have empirical error greater than $\alpha/2$. Event $\mathcal{E}_2$ happens with probability at least $1 - \beta/3$.

Conditioned on $\mathcal{E}_1$, the probability that a hypothesis with empirical error $\geq \alpha/2$ is selected by the exponential mechanism is at most

$$\frac{\exp(-\epsilon\alpha m/4)}{\sum_{j=1}^{N}\exp(-\epsilon m_j/2)} \leq \frac{\exp(-\epsilon\alpha m/4)}{\exp(-\epsilon\alpha m/8)} = \exp(-\epsilon\alpha m/8).$$

The first inequality holds because conditioned on $\mathcal{E}_1$ there exists a hypothesis (say, h_ℓ) in Hyp with empirical error less than $\alpha/4$. Therefore, $m_\ell \leq (\alpha/4)m$, and

$$\sum_{j=1}^{N}\exp(-\epsilon m_j/2) \geq \exp(-\epsilon m_\ell/2) \geq \exp(-\epsilon\alpha m/8).$$

[8] Given an input $D = (d_1, \ldots, d_m)$ where each $d_i = (x_i, c(x_i))$ is a labeled example, the empirical error of h is $\frac{1}{m}|\{i : h(x_i) \neq c(x_i)\}|$.

Since $m = 24\ln(3/\beta)/(\epsilon\alpha)$, the value of $\exp(-\epsilon\alpha m/8)$ is at most $\beta^3/27$. Therefore, conditioned on $\mathcal{E}_1$ and $\mathcal{E}_2$, the probability that a specific hypothesis with error greater than α is selected by the exponential mechanism is at most $\beta^3/27$, and by the union bound, the probability that a hypothesis with error greater than α is selected by the exponential mechanism is at most $N \cdot \beta^3/27 \leq \beta^2/27$. By removing all the conditioning, we get that the selected hypothesis has error greater than α with probability at most $\beta/5 + \beta^2/3 + \beta/3 + \beta^2/27 \leq \beta$. □

4.2.1 Making the learner efficient

The outcome of $\mathcal{A}_1$ (hence, $\mathcal{A}_2$) is a hypothesis whose description is exponentially long (since it contains a list of the indices where the output was flipped). We now complete our construction by compressing this description using a pseudorandom function. The running time of the resulting algorithm is polynomial and the hypothesis it returns has a short description.

We use a slightly non-standard definition of (non-uniform) pseudorandom functions from binary strings of size d to bits; these pseudorandom functions can be easily constructed given standard pseudorandom functions (which in turn can be constructed under standard assumptions (Goldreich 2001)). Roughly speaking, a collection of functions is pseudorandom if it cannot be distinguished from truly random functions. We start by defining the random functions in our definition.

Definition 4.6 Define $H_d^q : \{0,1\}^d \to \{0,1\}$ as a random variable, where each value $H_d^q(x)$ for $x \in \{0,1\}^d$ is selected i.i.d. to be 1 with probability q and 0 otherwise.

We consider a (non-uniform) polynomial-time distinguishing algorithm (represented by a circuit) C_d that can query a function in polynomially many points. Any such algorithm should not be able to distinguish if the answers of the function are random or are answered according to a random function from the pseudorandom family. Formally,

Definition 4.7 Let $F = \{F_d\}_{d\in\mathbb{N}}$ be a function ensemble, where for every d, F_d is a set of functions from $\{0,1\}^d$ to $\{0,1\}$. We say that the function ensemble F is q-biased pseudorandom if for every family of polynomial-size circuits with oracle access $\{C_d\}_{d\in\mathbb{N}}$, every polynomial $p(\cdot)$, and all sufficiently large d's,

$$\left|\Pr\left[C_d^f\left(1^d\right)=1\right] - \Pr\left[C_d^{H_d^q}\left(1^d\right)=1\right]\right| < \frac{1}{p(d)}. \tag{5}$$

In the above inequality, the first probability is taken over the random choice of f with uniform distribution from F_d, and the second probability is taken over the random variable H_d^q.

For convenience, for $d \in \mathbb{N}$, we consider F_d as a set of functions from $\{1,\dots,T\}$ to $\{0,1\}$, where $T = 2^d$. We set $q = \alpha/4$ in the above definition. Using an $\alpha/4$-biased pseudorandom function ensemble F (such functions can be constructed from standard pseudorandom functions (Goldreich 2001)), we change Step (3) of Algorithm $\mathcal{A}_1$ as follows:

(3)′ If $c = \mathbf{0}$, let h be a random function from F_d. Otherwise (i.e., $c = c_j$ for some $j \in [T]$), let h be a random function from F_d subject to $h(j) = 1$. Return h.

Call the resulting modified Algorithm $\mathcal{A}_4$. We next show that $\mathcal{A}_4$ is a PAC learner. Note that there exists a negligible function negl such that for large enough d,

$$\left|\Pr\left[h(x)=1 | h(j)=1\right] - \alpha/4\right| \leq \text{negl}(d)$$

for every $x \in \{1, \dots, T\}$ (as otherwise, we get a non-uniform distinguisher for the ensemble F). Thus,

$$\begin{aligned}\mathbb{E}_{h\in F_d}\big[\underset{\mathcal{D}}{\operatorname{error}}(c,h)\big] &= \mathbb{E}_{h\in F_d}\,\mathbb{E}_{x\sim\mathcal{D}}\big[\big|h(x)-c(x)\big|\big]\\ &\le \mathbb{E}_{h\in F_d}\,\mathbb{E}_{x\sim\mathcal{D}}\big[h(x)\big] = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_{h\in F_d}\big[h(x)\big] \le \frac{\alpha}{4} + \operatorname{negl}(d).\end{aligned}$$

The first inequality follows as for all $x \in [T]$, $h(x) \ge c(x)$ by our restriction on the choice of h. Thus, by the same arguments as for $\mathcal{A}_1$, Algorithm $\mathcal{A}_4$ is a PAC learner.

We next modify Algorithm $\mathcal{A}_2$ by executing the learner $\mathcal{A}_4$ instead of the learner $\mathcal{A}_1$. Call the resulting modified Algorithm $\mathcal{A}_5$. To see that Algorithm $\mathcal{A}_5$ preserves differential privacy it suffices to give a bound on (4). By comparing the case where $S = R$ with $S = R \cup \{i\}$, we get that the probability for a hypothesis h can increase only if $c = \mathbf{0}$ when $S = R$, and $c = c_i$ when $S = R \cup \{i\}$. Therefore,

$$\max_{R\subseteq[m']\setminus\{i\}} \frac{\Pr[\mathcal{A}_5(D)=h \mid S = R\cup\{i\}]}{\Pr[\mathcal{A}_5(D)=h \mid S=R]} \le \frac{1}{(\alpha/4)-\operatorname{negl}(d)} \le \frac{1}{(\alpha/8)} = \frac{8}{\alpha}.$$

Applying the same steps as in the proof of Claim 4.5, we get the following result.

Theorem 4.8 *There exists an efficient improper private PAC learner for* $\mathtt{POINT}_d$ *that uses* $O_{\alpha,\beta,\epsilon}(1)$ *samples, where* ϵ, α, *and* β *are the parameters of the private learner.*

Lemma 3.9 and Theorem 4.8 give the following separation:

Theorem 4.9 *Every proper private PAC learner for* $\mathtt{POINT}_d$ *requires* $\Omega((d+\log(1/\beta))/(\epsilon\alpha))$ *samples, whereas there exists an efficient improper private PAC learner that can learn* $\mathtt{POINT}_d$ *using* $O_{\alpha,\beta,\epsilon}(1)$ *samples. Here,* ϵ, α, *and* β *are the parameters of the private learners.*

4.3 Restrictions on the hypothesis class of private learners with low sample complexity

We conclude this section by showing that every (improper) private learner for $\mathtt{POINT}_d$ using $o(d)$ samples must return hypotheses that evaluate to one on many points (in contrast, every hypothesis in $\mathtt{POINT}_d$ returns the value one on just one input). This explains why our algorithms for $\mathtt{POINT}_d$ that use $o(d)$ samples return "complex" hypotheses.

Definition 4.10 (weight) The *weight* of a hypothesis h is the number of points for which it returns the value one, i.e., $|\{i : h(i) = 1\}|$.

Theorem 4.11 *There exists no private PAC learner for* $\mathtt{POINT}_d$ *with sample complexity* $o_{\alpha,\beta,\epsilon}(d)$ *that for every distribution returns, with probability at least half, hypotheses with weight* $2^{o_{\alpha,\beta,\epsilon}(d)}$ (*where the probability is taken over the randomness of the learner and the sample points chosen according to the distribution*). *Here,* ϵ, α, *and* β *are the parameters of the private learner.*

Proof In the proof assume the contrary, i.e., there exists a private learner that for every distribution returns hypotheses with weight $2^{o_{\alpha,\beta,\epsilon}(d)}$ with probability at least half. We prove that, under this assumption, there is a proper private learning algorithm for $\texttt{POINT}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$, in contradiction with Lemma 3.9.

Let $c_t \in \texttt{POINT}_d$ be the target concept. Assume for contradiction that there exists an ϵ-differentially private (α, β)-PAC learner $\mathcal{A}'$ for $\texttt{POINT}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$ that for every distribution returns, with probability at least $1/2$, hypotheses of weight less than z, for $z = 2^{o_{\alpha,\beta,\epsilon}(d)}$ (where the probability is taken over the randomness of $\mathcal{A}'$ and the sample points chosen according to the distribution).

Let $\mathcal{D}$ denote the underlying sample distribution. Construct a proper learner $\mathcal{A}$ (for $\texttt{POINT}_d$) which on input $\epsilon, d, \alpha, \beta$ does the following:

1. Let $k = \ln(\beta/2)/\ln(3/4)$.
2. Invoke k times the algorithm $\mathcal{A}'$ with parameters $\epsilon, d, \alpha/2, \beta' = 1/4$, each time on a fresh $\log z$ sized i.i.d. sample drawn from $\mathcal{D}$ and labeled by c_t. Let $h_1, \dots, h_{k'}$ (where $k' \leq k$) be the hypotheses returned in these executions with weight less than z.
3. If $k' = 0$ halt with failure, otherwise set $\mathcal{H}_d = \{c_j : h_i(j) = 1 \text{ for some } i \in [k']\}$.
4. Invoke the proper private learner of Lemma 3.4 with parameters $\epsilon, \alpha, \beta/2$ and hypothesis class $\mathcal{H}_d$ on a fresh $\ell = O((\log(|\mathcal{H}_d|) + \log(1/\beta))/(\epsilon\alpha))$ sized i.i.d. sample drawn from $\mathcal{D}$ and labeled by c_t. Output the hypothesis returned by the learner.

Note that $\ell = O((\log(|\mathcal{H}_d|) + \log(1/\beta))/(\epsilon\alpha)) = o_{\alpha,\beta,\epsilon}(d)$, and that the sample complexity of $\mathcal{A}$ is $k \log z + \ell = o_{\alpha,\beta,\epsilon}(d)$. Furthermore, $\mathcal{A}$ always returns a hypothesis in $\texttt{POINT}_d$ (note that $\mathcal{H}_d \subset \texttt{POINT}_d$). Hence, if $\mathcal{A}$ is a private learner for $\texttt{POINT}_d$, we get a contradiction to Lemma 3.9.

Note that $\mathcal{A}$ is ϵ-differentially private (follows since $\mathcal{A}'$ is ϵ-differentially private and in Step (4), we invoke the ϵ-differentially private algorithm from Lemma 3.4 on a fresh sample).

To conclude the proof we show that $\mathcal{A}$ is indeed a learner for $\texttt{POINT}_d$. Note that for each of the hypotheses h_i returned by $\mathcal{A}'$ in Step (2), we have that

$$\text{Condition 1: } \Pr\Big[\underset{\mathcal{D}}{\text{error}}(c_t, h_i) \leq \alpha/2\Big] \geq 1 - \beta' = \frac{3}{4},$$

and

$$\text{Condition 2: } \Pr[h_i \text{ has weight less than } z] \geq \frac{1}{2},$$

where the probability is taken over the randomness of $\mathcal{A}'$ and the sample points chosen according to $\mathcal{D}$. We get that h_i satisfies both the above conditions with probability at least $1/4$, and the probability that none of the hypotheses $\mathcal{A}'$ outputs satisfy both these conditions is at most $(3/4)^k = \beta/2$.

We henceforth assume that a hypothesis, h_i, returned by $\mathcal{A}'$ in Step (2) is of weight less than z and $\text{error}_{\mathcal{D}}(c_t, h_i) \leq \alpha/2$. We claim that in this case $\mathcal{H}_d$ contains a hypothesis $c_j \in \mathcal{H}_d$ for which $\text{error}_{\mathcal{D}}(c_t, c_j) \leq \alpha/2$, as if $h_i(t) = 1$ then we can set $j = t$, and otherwise, j can be any point such that $h_i(j) = 1$, as

$$\begin{aligned}\underset{\mathcal{D}}{\text{error}}(c_t, c_j) &= \Pr_{x\sim\mathcal{D}}[x = t] + \Pr_{x\sim\mathcal{D}}[x = j] \leq \Pr_{x\sim\mathcal{D}}[x = t] + \Pr_{x\sim\mathcal{D}}\big[h_i(x) = 1\big] \\ &= \underset{\mathcal{D}}{\text{error}}(c_t, h_i) \leq \alpha/2.\end{aligned}$$

In other words, $\mathcal{H}_d$ $\alpha/2$-represents $\{c_t\}$.

To conclude the proof, we observe that having $\mathcal{H}_d$ $\alpha/2$-represent $\{c_t\}$ suffices for the proof of Theorem 3.2, and hence, the hypothesis (in Step (4)) returned by the learner of Theorem 3.2 is with probability at least $1-\beta/2$ within error α from c_t.

To summarize, we get that $\mathcal{A}$ is a proper private learner for $\texttt{POINT}_d$ under distribution $\mathcal{D}$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$. Since this holds for every $\mathcal{D}$ this leads to a contradiction to Lemma 3.9 (the lemma shows that there exists a distribution for which there is no proper private learner for $\texttt{POINT}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$). □

5 Private learning of intervals (partial results)

In this section, we examine $\texttt{INTERVAL}_d$, a concept class that like $\texttt{POINT}_d$ is very natural and simple and has VC-dimension 1. By Theorem 3.6, any proper private learner for $\texttt{INTERVAL}_d$ requires $\Omega_{\alpha,\beta,\epsilon}(d)$ samples (as $\texttt{INTERVAL}_d$ is α-minimal for itself), and we ask whether stronger separation results than we showed for $\texttt{POINT}_d$ can be proved for$\texttt{INTERVAL}_d$. Specifically, we ask if we can prove a lower bound of $\omega_{\alpha,\beta,\epsilon}(1)$ for any private learner for$\texttt{INTERVAL}_d$ (i.e., also for improper private learners).

We give partial results towards answering this question. In Sect. 5.1, we show that if there exists an $O_{\alpha,\beta,\epsilon}(1)$ sample sized improper private learner for $\texttt{INTERVAL}_d$, then it must use hypotheses that are very unlike intervals, and in fact must *switch* exponentially many times between zero and one (this is similar to the result presented for $\texttt{POINT}_d$ in Sect. 4.3). Then, in Sect. 5.2, we take a deeper look into improper private learning of $\texttt{INTERVAL}_d$, and prove that the technique from Sect. 4.2 that yielded the efficient private learner for $\texttt{POINT}_d$ with sample complexity $O_{\alpha,\beta,\epsilon}(1)$ cannot yield an algorithm for $\texttt{INTERVAL}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$. In other words, the technique of adding independent noise from Sect. 4.2, even with exponentially many switch points, does not yield a learner for $\texttt{INTERVAL}_d$ with $o_{\alpha,\beta,\epsilon}(d)$ sample complexity.

Before proving the above results, let us first formally define $\texttt{INTERVAL}_d$ and establish a sample complexity lower bound for proper private learning this concept class.

Definition 5.1 The concept class $\texttt{INTERVAL}_d$ is $\{c_j : j \in \{1,\dots,T+1\}\}$ where $T=2^d$ and the concept $c_j : [T] \to \{0,1\}$ maps all $x<j$ to 1 and all $x \geq j$ to 0.

Unlike the concept class $\texttt{POINT}_d$, the values of elements of X_d are significant in the sense that the geometric relation of which point is to the left of the other is meaningful. Note that the cardinality of $\texttt{INTERVAL}_d$ is 2^d+1, and that it is α-minimal for itself (for all $\alpha<1/2$), and hence, we can use Theorem 3.6 and get a lower bound on the sample complexity of proper private learners for $\texttt{INTERVAL}_d$.

Lemma 5.2 *Every proper private PAC learner for* $\texttt{INTERVAL}_d$ *requires* $\Omega((d+(1/\beta))/\epsilon)$ *samples.*

5.1 Restrictions on the hypothesis class of private learners with low sample complexity

We give an insight on the structure of the hypothesis class of an improper private learner for $\texttt{INTERVAL}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$. We show that if such a learner for $\texttt{INTERVAL}_d$ exists, then it must return, with high probability, a hypothesis that switches frequently between zero and one. Therefore, the hypothesis outputted by the learner has a very different structure compared to the concepts in $\texttt{INTERVAL}_d$, which switch exactly once from 1 to 0. This result resembles Theorem 4.11, where we proved a similar structural statement for private learning POINT class.

Definition 5.3 (Switching Point) We say that j is a *switching point* in hypothesis h if $h(j) \neq h(j-1)$. If $h(j-1)=1$ we say that j is a *decreasing* switching point. Otherwise, we say the switching point is *increasing*. The points 1 and $T+1$ are also referred to as switching points. The point 1 is a increasing switching point if $h(1)=1$ and decreasing otherwise. The point $T+1$ is a increasing switching point if $h(T)=0$ and decreasing otherwise.

We next prove that every private learner with sample complexity $o_{\alpha,\beta,\epsilon}(d)$ returns with high probability a hypothesis with an exponential number of switching points. We prove this using a method similar to the proof of the previous theorem. We assume that a learner exists which returns with constant probability a hypothesis with too little switching points. We then show that a proper private learner can be reconstructed from this hypothesis. For the reconstruction, we use a simplified version of the exponential mechanism of McSherry and Talwar (2007). Existence of a proper private learner for the class $\mathtt{INTERVAL}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$ leads to a contradiction to Lemma 5.2.

Theorem 5.4 *There exists no private PAC learner for* $\mathtt{INTERVAL}_d$ *with sample complexity* $o_{\alpha,\beta,\epsilon}(d)$ *that for every distribution returns, with probability at least half, hypotheses with* $2^{o_{\alpha,\beta,\epsilon}(d)}$ *switching points (where the probability is taken over the randomness of the learner and the sample points chosen according to the distribution). Here,* ϵ, α, *and* β *are the parameters of the private learner.*

Proof Let $\mathcal{D}$ denote the underlying sample distribution. Every concept $c \in \mathtt{INTERVAL}_d$ consists of exactly one decreasing switching point. Discovering this point is discovering the accurate concept. Assume first that the target concept is c_t for some $1 \leq t \leq T+1$ and we have a hypothesis h such that $\mathrm{error}_{\mathcal{D}}(c_t, h) \leq \alpha$. Let j and k be two consecutive switching points in h such that $j \leq t \leq k$.[9] Assume first that the switching point j is decreasing (and, thus, k is increasing). Note that $c_j(x) = c_t(x) = 1$ for every $x < j$ and $c_j(x) = c_t(x) = 0$ for every $x \geq t$. Therefore, c_j is a hypothesis which only errs on $\{j, \dots, t-1\}$. Also $c_j(x) = h(x) = 0$ for every $x \in \{j, \dots, t-1\}$.

Therefore, we can refer to c_j as a concept which is reconstructed from h (it is chosen from h's switching points) and which fixes all of h's errors in $\{1, \dots, j-1\} \cup \{t, \dots, T\}$. On the other hand, h errs on every point in $\{j, \dots, t-1\}$, so c_j does not introduce new errors to h. We get that

$$\underset{\mathcal{D}}{\mathrm{error}}(c_t, c_j) \leq \underset{\mathcal{D}}{\mathrm{error}}(c_t, h) \leq \alpha.$$

Similarly, if j is an increasing switching point, then k is decreasing, then c_k is such that

$$\underset{\mathcal{D}}{\mathrm{error}}(c_t, c_k) \leq \underset{\mathcal{D}}{\mathrm{error}}(c_t, h) \leq \alpha.$$

Define

$$\mathtt{SWITCH}(h) = \{c_j : j \text{ is a switching point in } h\}.$$

[9] The switching points j and k exist as points 1 and $T+1$ are always switching points.

Note that $\mathtt{SWITCH}(h) \neq \emptyset$ by construction. By our discussion above, if h is such that $\mathrm{error}_{\mathcal{D}}(c_t, h) \leq \alpha$ then so is the case for at least one concept in $\mathtt{SWITCH}(h)$. Clearly, $|\mathtt{SWITCH}(h)|$ is bounded by the number of switching points in h.

Remark 5.5 Note that if the empirical error of h on some sample database D is less than α, then using same arguments as above there exists a concept in $\mathtt{SWITCH}(h)$ whose empirical error on D is also less than α.

As in Kasiviswanathan et al. (2011), we use the exponential mechanism in order to choose a hypothesis out of $\mathtt{SWITCH}(h)$ (we used the same mechanism in the proof of Claim 4.5).

We now have enough tools for the proof. Assume that $\mathcal{A}'$ is an ϵ-differentially private (α, β)-PAC learner for the class $\mathtt{INTERVAL}_d$ with a sample complexity $o_{\alpha,\beta,\epsilon}(d)$ that on every distribution returns, with probability at least $1/2$, hypotheses with at most $z = z(\alpha, \beta, \epsilon, d) = 2^{o_{\alpha,\beta,\epsilon}(d)}$ switching points. Let $s = 8\ln(\frac{12}{\beta})/(\alpha^2) + 8\ln(\frac{(6-\beta)z}{\beta})/(\alpha\epsilon) + K(\frac{1}{\alpha}\log\frac{1}{\beta} + \frac{1}{\alpha}\log\frac{1}{\alpha})$ for some constant K to be set below.

Construct a proper private learner $\mathcal{A}$ as follows:

1. Let $\alpha' = \frac{\alpha}{4}$; $\beta' = \frac{\beta}{6}$.
2. For i in $\{1, \ldots, \log\frac{1}{\beta'}\}$:
   (a) Draw $o_{\alpha,\beta,\epsilon}(d)$ new samples from $\mathcal{D}$ and label it by c_t. Let D' denote these labeled examples.
   (b) Apply $\mathcal{A}'$ with parameters $\epsilon, \alpha', \beta'$ on D'. Let h_i be the returned hypothesis.
3. Let $\hat{h}$ denote the first hypothesis in $\{h_1, \ldots, h_{\log(1/\beta')}\}$ such that $|\mathtt{SWITCH}(h_i)| \leq z$. If no such $\hat{h}$ exists, return "FAIL".
4. Draw s additional samples according to $\mathcal{D}$ and label it by c_t. Let D_s denote these labeled examples.
5. Choose a concept c out of $\mathtt{SWITCH}(\hat{h})$ using the exponential mechanism on D_s with parameter ϵ and return it.

We now show that $\mathcal{A}$ is a *proper* private (α, β)-PAC learner with sample complexity $o_{\alpha,\beta,\epsilon}(d)$. This is a contradiction to Lemma 5.2.

First, note that according to the assumption, Step (2a) is given enough samples. Also according to the assumption, for every i we have that $\Pr[|\mathtt{SWITCH}(h_i)| \geq z] \leq 1/2$. Therefore, Step (3) fails with probability at most $(1/2)^{\log(1/\beta')} = \beta'$. Since the chosen hypothesis $\hat{h}$ is a uniformly distributed hypothesis conditioned on $|\mathtt{SWITCH}(\hat{h})| \leq z$ (an event with probability at least half), the probability that $\mathrm{error}_{\mathcal{D}}(c_t, \hat{h}) \geq \alpha'$ is at most $2\beta' + \beta' = 3\beta'$ ($2\beta'$ comes from the Step (2b) and β' from Step (3)).

In our next analysis, we assume that $\mathrm{error}_{\mathcal{D}}(c_t, \hat{h}) < \alpha'$. Denote by $\widehat{\mathrm{error}}_{D_s}(h')$ the *empirical* error of a hypothesis h' on the samples D_s, and let $Q = \widehat{\mathrm{error}}_{D_s}(\hat{h})$. Clearly, $\mathbb{E}_{D_s}[Q] = \mathrm{error}_{\mathcal{D}}(c_t, \hat{h}) \leq \alpha'$, where the expectation is over the drawing of the samples D_s in Step (4). We can bound Q with high probability using Chernoff-Hoeffding bound (Inequality (2)) and get

$$\Pr\big[\big|Q - \mathbb{E}_{D_s}[Q]\big| \geq \alpha'\big] \leq 2\exp\big(-2s\alpha'^2\big).$$

Since $s > 8\ln(\frac{12}{\beta})/(\alpha^2) = \ln(\frac{2}{\beta'})/(2\alpha'^2)$, we have

$$\Pr\big[\big|Q - \mathbb{E}_{D_s}[Q]\big| \geq \alpha'\big] \leq \beta'.$$

Since $\mathbb{E}_{D_s}[Q] \le \alpha'$, we now have $\Pr[Q \ge 2\alpha'] \le \beta'$. For the analysis of the last step we assume that indeed

$$\widehat{\mathrm{error}}_{D_s}(\hat{h}) \le 2\alpha'.$$

Next, we analyze the complexity and accuracy of the exponential mechanism step. Let

$$\mathtt{good}(D_s, \hat{h}) = \left\{c_j \in \mathtt{SWITCH}(\hat{h}) : \widehat{\mathrm{error}}_{D_s}(c_j) \le 3\alpha'\right\}.$$

That is, $\mathtt{good}(D_s, \hat{h})$ contains the concepts in $\mathtt{SWITCH}(\hat{h})$ that are inconsistent with less than $3\alpha' s$ samples, i.e., concepts such that $m_{c_j} \le 3\alpha' s$. Let $\mathtt{bad}(D_s, \hat{h})$ be all the other concepts in $\mathtt{SWITCH}(\hat{h})$. Let $\mathcal{E}_{\mathtt{good}}$ (resp. $\mathcal{E}_{\mathtt{bad}}$) be the event that a concept in $\mathtt{good}(D_s, \hat{h})$ (resp. $\mathtt{bad}(D_s, \hat{h})$) is chosen by the exponential mechanism in Step (5). Remember, we assumed $\widehat{\mathrm{error}}_{D_s}(\hat{h}) \le 2\alpha'$. Also remember that if $\widehat{\mathrm{error}}_{D_s}(\hat{h}) \le 2\alpha'$, then, according to observations mentioned in Remark 5.5 there is at least one concept $c^\star \in \mathtt{SWITCH}(\hat{h})$ whose empirical error is also bounded by $2\alpha'$ (therefore, $c^\star \in \mathtt{good}(D_s, \hat{h})$). So in Step (5),

$$\begin{aligned}
\frac{\Pr[\mathcal{E}_{\mathtt{good}}]}{\Pr[\mathcal{E}_{\mathtt{bad}}]} &= \frac{\sum_{c_j \in \mathtt{good}(D_s,\hat{h})} \exp(-\epsilon \cdot m_{c_j}/2)}{\sum_{c_j \in \mathtt{bad}(D_s,\hat{h})} \exp(-\epsilon \cdot m_{c_j}/2)} \\
&\ge \frac{\exp(-\epsilon \cdot m_{c^\star}/2)}{\sum_{c_j \in \mathtt{bad}(D_s,\hat{h})} \exp(-\epsilon \cdot m_{c_j}/2)} \ge \frac{\exp(-\alpha' s \epsilon)}{\sum_{c_j \in \mathtt{bad}(D_s,\hat{h})} \exp(-3\alpha' s \epsilon/2)} \\
&\ge \frac{\exp(-\alpha' s \epsilon)}{|\mathtt{SWITCH}(\hat{h})| \cdot \exp(-3\alpha' s \epsilon/2)} = \frac{\exp(\alpha' s \epsilon/2)}{|\mathtt{SWITCH}(\hat{h})|} \\
&\ge \frac{\exp(\alpha' s \epsilon/2)}{z}.
\end{aligned}$$

Since $s > 8\ln(\frac{(6-\beta)z}{\beta})/(\alpha\epsilon) = 2\ln(\frac{(1-\beta')z}{\beta'})/(\alpha'\epsilon)$, we get that

$$\frac{\Pr[\mathcal{E}_{\mathtt{good}}]}{1 - \Pr[\mathcal{E}_{\mathtt{good}}]} = \frac{\Pr[\mathcal{E}_{\mathtt{good}}]}{\Pr[\mathcal{E}_{\mathtt{bad}}]} \ge \frac{1-\beta'}{\beta'}$$

and, thus, $\Pr[\mathcal{E}_{\mathtt{good}}] \ge 1 - \beta'$. Therefore, if $\hat{h}$ satisfies $\widehat{\mathrm{error}}_{D_s}(\hat{h}) \le 2\alpha'$ and it has less than z switching points, then Step (5) returns with probability at least $1 - \beta'$ a concept $c \in \mathtt{INTERVAL}_d$ such that $\widehat{\mathrm{error}}_{D_s}(c) \le 3\alpha'$. For our last analysis, we assume that indeed a concept with empirical error bounded by $3\alpha'$ was chosen in Step (5).

Finally, we show that c, the concept returned by $\mathcal{A}$, has indeed $\mathrm{error}_{\mathcal{D}}(c, c_t) \le \alpha$ with high probability. As the VC-dimension of $\mathtt{INTERVAL}_d$ is 1, by Blumer et al. (1989), there exists a constant ℓ such that whenever more than $\ell(\frac{1}{\alpha'}\log\frac{1}{\beta'} + \frac{1}{\alpha'}\log\frac{1}{\alpha'})$ samples are drawn from some distribution $\mathcal{D}$, then $\Pr[|\mathrm{error}_{\mathcal{D}}(c_t, c) - \widehat{\mathrm{error}}_{D_s}(c)| \ge \alpha'] \le \beta'$. Remember that $s > K(\frac{1}{\alpha}\log\frac{1}{\beta} + \frac{1}{\alpha}\log\frac{1}{\alpha})$ for some constant K (depending on ℓ). As we assumed $\widehat{\mathrm{error}}_{D_s}(c) \le 3\alpha'$, we finally have that $\mathrm{error}_{\mathcal{D}}(c_t, c) \le 4\alpha' = \alpha$ with probability at least $1 - \beta'$.

Next we analyze the confidence parameter of $\mathcal{A}$. We now list the bad events. As said before, the probability of $\mathrm{error}_{\mathcal{D}}(c_t, \hat{h}) \ge \alpha'$ at the end of Step (3) is bounded by $3\beta'$. After this $\hat{h}$ is chosen in Step (3), its empirical error on the samples D_s is too high with probability bounded by β'. The exponential mechanism fails to return a concept c with low empirical error on D_s with probability bounded by β'. Finally, if the exponential mechanism successfully returned a concept with low empirical error, then the misclassification error of c is too high with probability bounded by β'. Using the union bound, we get that the probability of any of the above bad events happening is bounded by $6\beta'$. Therefore,

$$\Pr\Big[\underset{\mathcal{D}}{\operatorname{error}}(c_t, c) \geq \alpha\Big] \leq 6\beta' = \beta.$$

We now calculate the sample complexity. Note that samples are drawn in Step (4) and many times in Step (2a). As we assumed the sample complexity of $\mathcal{A}'$ is $o_{\alpha,\beta,\epsilon}(d)$ and it is executed $\log(1/\beta')$ times, we get that the total sample complexity of this step is $o_{\alpha,\beta,\epsilon}(d)$. (Remember that α' and β' are of the same order as α and β.) Also note that since $z = 2^{o_{\alpha,\beta,\epsilon}(d)}$, the sample complexity of Step (4) is $s = o_{\alpha,\beta,\epsilon}(d)$. Therefore, the sample complexity of $\mathcal{A}$ is $\log(1/\beta') \cdot o_{\alpha,\beta,\epsilon}(d) + s = o_{\alpha,\beta,\epsilon}(d)$.

Finally, note that we assumed $\mathcal{A}'$ maintains ϵ-differential privacy. Also the exponential mechanism maintains ϵ-differential privacy. Since any execution of the inner algorithms is on different independently drawn samples of the whole sample set, the learner $\mathcal{A}$ maintains ϵ-differential privacy.

Combining all the above statements we have that if there is an ϵ-differentially private $(\alpha/4, \beta)$-PAC learner for $\texttt{INTERVAL}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$ that for every distribution returns, with probability at least half, a hypotheses with $2^{\Omega_{\alpha,\beta,\epsilon}(d)}$ switching points, then there is a proper ϵ-differentially private (α, β)-PAC learner for $\texttt{INTERVAL}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$. This contradicts Lemma 5.2. □

5.2 Impossibility of private independent noise learners with low sample complexity

We next show that the ideas used to construct in Sect. 4.2 a private learner for $\texttt{POINT}_d$ with sample complexity $O_{\alpha,\beta,\epsilon}(1)$ cannot be used for $\texttt{INTERVAL}_d$. We begin by formalizing a class of *independent noise learners* that generalizes the construction in Sect. 4.2. We note that independent noise learners are allowed to output hypotheses whose description is exponential in d (recall that this issue was resolved for $\texttt{POINT}_d$ by using compression with pseudorandom functions).

Definition 5.6 (Private Independent Noise Learner) A private independent noise learner for a concept class $\mathcal{C}_d$ over X_d using sample size m' and parameters $\alpha', \beta', \epsilon$ is a pair of algorithms $(\mathcal{A}^{\text{outer}}, \mathcal{A}^{\text{inner}})$, called the *outer* and *inner* learners respectively, that for all concepts $c \in \mathcal{C}_d$, all distributions $\mathcal{D}$ on X_d, given an input $D = (d_1, \ldots, d_{m'})$, where $d_i = (x_i, c(x_i))$ with x_i drawn i.i.d. from $\mathcal{D}$ for all $i \in [m']$, does the following:

1. The outer learner $\mathcal{A}^{\text{outer}}$ is a private PAC learner (as defined in Definition 2.5) for $\mathcal{C}_d$ using the class of all $2^{|X_d|}$ functions $X_d \to \{0, 1\}$. Furthermore, $\mathcal{A}^{\text{outer}}(\epsilon, d, \alpha', \beta', D)$ is restricted to execute as follows:
   (a) Select parameters $\alpha^\star \leq \alpha'$, $\beta^\star \leq \beta'$, and a noise rate μ as a (deterministic) function of $\epsilon, \alpha', \beta'$.
   (b) Run $\mathcal{A}^{\text{inner}}(d, \alpha^\star, \beta^\star, D)$. Denote the output hypothesis $c^\star$.
   (c) If $c^\star \notin \mathcal{C}_d$ then output "fail" and halt. Otherwise, produce a hypothesis h by addition of noise to all entries of $c^\star$ independently, i.e., for all $x \in X_d$ set $h(x) = 1 - c^\star(x)$ with probability μ, and $h(x) = c^\star(x)$ otherwise.
2. The inner learner $\mathcal{A}^{\text{inner}}$ outputs with probability at least $1 - \beta^\star$ (over the randomness of $\mathcal{A}^{\text{inner}}$ and the sampling of D according to $\mathcal{D}$) a hypothesis $c^\star \in \mathcal{C}_d$ such that $\operatorname{error}_{\mathcal{D}}(c^\star, c) \leq \alpha^\star$.

Example 5.7 We show that Algorithm $\mathcal{A}_2$, described in Sect. 4.2, is a private independent noise learner for $\mathtt{POINT}_d$. In order to do this, we describe Algorithm $\mathcal{A}_2$ in a different way than the description in Sect. 4.2.[10] The outer learner is the learner defined in Definition 5.6 selecting parameters $\alpha^\star = \alpha'/2$, $\beta' = 3/4$, $\beta^\star = 1/2$, and a noise rate $\mu = \alpha'/8$. The inner learner does the following:

1. Set $\alpha = \alpha'$.
2. Get a sample $(x_1, y_1), \ldots, (x_{m'}, y_{m'})$, where x_i's are chosen according to $\mathcal{D}$ and $m' = 32\ln(4)/\alpha^2$.
3. With probability $\alpha/8$, return $\perp$.
4. Construct a set $S \subseteq [m']$ by picking each element of $[m']$ with probability $\alpha/4$.
5. If $((x_i, y_i))_{i\in S}$ is not consistent with any concept in $\mathtt{POINT}_d$, return $\perp$.
6. If $y_i = 0$ for all $i \in S$, then let $c = \mathbf{0}$ (the all zero hypothesis); otherwise, let c be the (unique) hypothesis from $\mathtt{POINT}_d$ that is consistent with the labeled example $((x_i, y_i))_{i\in S}$.

As analyzed in Sect. 4.2, Algorithm $\mathcal{A}_2$ is $\ln(4)$-differentially private. It is also (α', β')-PAC learner. To construct an algorithm that is ϵ-differentially private for smaller values of ϵ, we use a transformation described in Lemma 4.4. It can be seen that the resulting algorithm is also a private independent noise learner.

Furthermore, in the above description of $\mathcal{A}_2$, the confidence parameter is $\beta' = 3/4$. In Sect. 4.2, we boosted the confidence parameter by using the exponential mechanism. The resulting learning algorithm is *not* a private independent noise learner. However, for any constant β', we can modify $\mathcal{A}_2$ such that the resulting algorithm has confidence β' and is a private independent noise learner; however, the sample complexity of the resulting algorithm is not polynomial in $\log(1/\beta')$.

We next show that there is no private independent noise learner for $\mathtt{INTERVAL}_d$ using only $o_{\alpha,\beta,\epsilon}(d)$ samples. We will show that in this case, we can essentially recover the outcome of the inner learner (with probability at least $1-\beta$ a hypothesis in $\mathtt{INTERVAL}_d$) from the outcome of the outer learner. It follows then that the existence of a private independent noise learner for $\mathtt{INTERVAL}_d$ that uses $o_{\alpha,\beta,\epsilon}(d)$ samples implies a proper private learner for $\mathtt{INTERVAL}_d$ that uses $o_{\alpha,\beta,\epsilon}(d)$ samples, in contradiction with Lemma 5.2.

Theorem 5.8 *There is no private independent noise learner for* $\mathtt{INTERVAL}_d$ *for* $\beta' < 1/4$ *and* $\alpha' < \beta'/100$ *that learns using* $m' = o_{\alpha',\beta',\epsilon}(d)$ *samples.*

Proof Assume towards a contradiction that a private independent noise learner $(\mathcal{A}^{\text{outer}}, \mathcal{A}^{\text{inner}})$ exists for $\mathtt{INTERVAL}_d$. Let $\mathcal{D}$ denote the underlying sample distribution and $c_t \in \mathtt{INTERVAL}_d$ denote the target concept. Consider an execution of $\mathcal{A}^{\text{outer}}$ when invoked with parameters α', β' where $\beta' < 1/2$ (we will further restrict α', β' below). We first show a simple bound on the noise rate $\mu = \mu(\alpha', \beta')$ selected by $\mathcal{A}^{\text{outer}}$. Denote by $\alpha^\star \le \alpha'$, $\beta^\star \le \beta'$ the parameters that $\mathcal{A}^{\text{outer}}$ selects for the inner learner. Denote by $c^\star$ the concept returned by $\mathcal{A}^{\text{inner}}$ and by h the concept returned by $\mathcal{A}^{\text{outer}}$ (or $\perp$ if $\mathcal{A}^{\text{outer}}$ halts without an output).

Note that by the definition of a private independent noise learner, $\mathcal{A}^{\text{inner}}$ outputs $c^\star \in \mathtt{INTERVAL}_d$ satisfying $\text{error}_{\mathcal{D}}(c_t, c^\star) \le \alpha^\star$ with probability at least $1-\beta^\star$. Similarly, since $\mathcal{A}^{\text{outer}}$ is a learner, we get that $\mathcal{A}^{\text{outer}}$ outputs h satisfying $\text{error}_{\mathcal{D}}(c_t, h) \le \alpha'$ with probability

[10] For simplicity of the description, we ignore the fact that some of the sample points can be $\star$.

at least $1-\beta'$. In both cases, the probability is taken over the randomness in the execution of the learner (for $\mathcal{A}^{\text{outer}}$ this includes the randomness of $\mathcal{A}^{\text{inner}}$) and the sample points chosen according to $\mathcal{D}$. We, hence, define the event

$$\mathcal{E}: \begin{array}{l} \mathcal{A}^{\text{inner}} \text{ outputs } c^\star \in \texttt{INTERVAL}_d \text{ satisfying } \text{error}_{\mathcal{D}}(c_t, c^\star) \le \alpha^\star; \text{ and} \\ \mathcal{A}^{\text{outer}} \text{ outputs } h \text{ satisfying } \text{error}_{\mathcal{D}}(c_t, h) \le \alpha' \end{array}$$

and conclude that $\Pr[\mathcal{E}] \ge 1-\beta'-\beta^\star > 0$.

In the following, we bound $\mathbb{E}_h[\text{error}_{\mathcal{D}}(c_t,h)] \triangleq \mathbb{E}_h\,\mathbb{E}_{x\sim\mathcal{D}}[|h(x)-c_t(x)|]$, assuming $\mathcal{E}$. This will yield an upper bound on μ.

$$\begin{aligned} \mathbb{E}_h\Big[\underset{\mathcal{D}}{\text{error}}(c_t,h)\;|\mathcal{E}\Big] &= \mathbb{E}_h\,\mathbb{E}_{x\sim\mathcal{D}}\Big[\big|h(x)-c_t(x)\big|\;|\mathcal{E}\Big] \\ &\ge \mathbb{E}_h\Big[\mathbb{E}_{x\sim\mathcal{D}}\Big[\big|h(x)-c^\star(x)\big|\;|\mathcal{E}\Big] - \mathbb{E}_{x\sim\mathcal{D}}\Big[\big|c_t(x)-c^\star(x)\big|\;|\mathcal{E}\Big]\Big] && (6) \\ &\ge \mathbb{E}_h\,\mathbb{E}_{x\sim\mathcal{D}}\Big[\big|h(x)-c^\star(x)\big|\;|\mathcal{E}\Big] - \alpha^\star && (7) \\ &= \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_h\Big[\big|h(x)-c^\star(x)\big|\;|\mathcal{E}\Big] - \alpha^\star = \mu - \alpha^\star. && (8) \end{aligned}$$

Inequality (6) follows from the triangle inequality, i.e., $|h(x)-c^\star(x)| \le |h(x)-c_t(x)| + |c_t(x)-c^\star(x)|$, and Inequality (7) follows from $\text{error}_{\mathcal{D}}(c_t,c^\star) \le \alpha^\star$. On the other hand, by the definition of $\mathcal{E}$

$$\mathbb{E}_h\Big[\underset{\mathcal{D}}{\text{error}}(c_t,h)\;|\mathcal{E}\Big] < \alpha'. \qquad (9)$$

Noting that the setting of μ is deterministic (and, hence, the setting of μ does not depend on whether the event $\mathcal{E}$ holds), we get from Inequalities (8) and (9) that $\alpha' \ge \mu - \alpha^\star$, and hence, $\mu \le 2\alpha'$. It follows that by choosing α' to be small enough, we restrict μ to be small.

We now show how to reconstruct $c^\star$ from h. The *reconstruction algorithm* is as follows:

1. For every $t \in \{1,\dots,T+1\}$ define $\texttt{mismatch}(t,h) = |\{x < t : h(x) = 0\}| + |\{x \ge t : h(x) = 1\}|$.
2. Find ℓ for which $\texttt{mismatch}(\ell,h)$ is the lowest and return c_ℓ.
3. If no such unique point exists, return "FAIL".

We now bound the probability that $c_\ell \ne c^\star$. We call a point x for which noise was added by $\mathcal{A}^{\text{outer}}$ (i.e., $h(x) \ne c^\star(x)$) *dirty*, otherwise we call x *clean*. Let j be such that $c_j = c^\star$. Then, $\texttt{mismatch}(j,h)$ is the number of dirty points. The reconstruction algorithm fails to return $c^\star$ if and only if there is some point k such that $\texttt{mismatch}(k,h) \le \texttt{mismatch}(j,h)$. In this case, we say that k is *bad*. We show that for small enough μ, such a bad point exists only with constant probability. In the following, we assume that $k > j$ (the case $k < j$ is symmetric). First note that c_j and c_k disagree agree only on points in $\{j,\dots,k-1\}$ (i.e., $\texttt{mismatch}(j,h)$ and $\texttt{mismatch}(k,h)$ have the same contribution from points not between j and k). Now every dirty point in $\{j,\dots,k-1\}$ contributes 1 to $\texttt{mismatch}(j,h)$ and nothing to $\texttt{mismatch}(k,h)$, and similarly each clean point between $\{j,\dots,k-1\}$ contributes 1 to $\texttt{mismatch}(k,h)$ and nothing to $\texttt{mismatch}(j,h)$. Since we assumed that $\texttt{mismatch}(k,h) \le \texttt{mismatch}(j,h)$, it should be the case that at least half the entries in $\{j,\dots,k-1\}$ are dirty.

We consider the case where there is a bad point bigger than j (the case where it is smaller than j is handled analogously). Let $k > j$ be the *smallest* bad point which is bigger than j, that is, k is the smallest such that the number of dirty points in $\{j,\dots,k-1\}$ is at least the number of clean points. Hence, $k = j+1$ if and only if j is a dirty point; if $k > j+1$

then for all $j < \ell < k$ the number of clean entries in $\{j, \dots, \ell - 1\}$ exceeds the number of dirty points (otherwise ℓ is a bad point smaller than k). From the above arguments it follows that the number of clean points in $\{j, \dots, k - 1\}$ equals the number of dirty points in $\{j, \dots, k - 1\}$.

Let $\texttt{noise}_j$ be a sequence starting from j which indicates which entries in $c^\star$ were flipped by $\mathcal{A}^{\text{outer}}$, i.e., every dirty point bigger than j is marked by 1 in $\texttt{noise}_j$, and every clean point is marked by 0. According to the above analysis, we get that there exists a bad point $k > j$ only if

- $\texttt{noise}_j$ begins with 1 (this if the case when $k = j + 1$), or
- $\texttt{noise}_j$ begins with some Dyck word, where a Dyck word is a balanced string of "parentheses" in the sense that it consists of n zeros and n ones, and in every prefix the number of ones does not exceed the number of zeros (this is the case when $k > j + 1$).

The probability of $\texttt{noise}_j$ to begin with 1 is μ. The probability of $\texttt{noise}_j$ to start with a specific Dyck word of length $2n$ is $\mu^n(1 - \mu)^n$. The number of Dyck words of length $2n$ is the n^{th} Catalan number, $C_n = \frac{1}{n+1}\binom{2n}{n}$, and we get that the probability of a bad $k > j$ is bounded by

$$\mu + \sum_{n=1}^{\infty} C_n \cdot \mu^n (1 - \mu)^n.$$

Note that this is a loose bound because as every Dyck word is a prefix of longer Dyck words, and so we over count many possibilities of bad noise. Using the Stirling approximation, $C_n \approx \frac{4^n}{n^{3/2}\sqrt{\pi}} \leq \frac{4^n}{n\sqrt{\pi}}$ for every $n \geq 1$. Therefore, the probability of failure to reconstruct c_j from h due a bad $k > j$ is bounded by

$$\begin{aligned}\mu + \sum_{n=1}^{\infty} C_n \cdot \mu^n (1 - \mu)^n &\leq \mu + \sum_{n=1}^{\infty} C_n \cdot \mu^n \\ &\leq \mu + \sum_{n=1}^{\infty} \frac{(4\mu)^n}{n\sqrt{\pi}} = \mu + \frac{1}{\sqrt{\pi}} \sum_{n=1}^{\infty} \frac{(4\mu)^n}{n} \\ &= \mu + \frac{1}{\sqrt{\pi}} \big(- \ln (1 - 4\mu) \big).\end{aligned}$$

The last equality follows from the Taylor series of $\ln(x)$. As $(- \ln (1 - 4\mu)) < 5\mu$ for every $\mu \leq 0.09$, the probability of failure to reconstruct $c^\star$ out of h due to a bad $k > j$ is bounded by $\mu + \frac{1}{\sqrt{\pi}} \cdot 5\mu < 4\mu$. Due to symmetry, the probability of failing because of a bad $k < j$ is also bounded by 4μ. Thus, for small enough values of μ, the probability of failure to reconstruct $\mathcal{A}^{\text{inner}}$'s original output $c^\star$ (i.e., the probability that $c_\ell \neq c^\star$) from h is bounded by 8μ.

To conclude the proof, we construct $\mathcal{A}$, a proper private learner for $\texttt{INTERVAL}_d$, using $\mathcal{A}^{\text{outer}}$. Learner $\mathcal{A}$ executes as follows:

1. Let $\beta' = \frac{\beta}{4}$ and $\alpha' = \frac{\min(\alpha, \beta)}{100}$.
2. Apply $\mathcal{A}^{\text{outer}}$ with parameters $\epsilon, d, \alpha', \beta'$ to improperly learn $\texttt{INTERVAL}_d$ using $o_{\alpha',\beta',\epsilon}(d)$ samples. Let h be the output of $\mathcal{A}^{\text{outer}}$. If $\mathcal{A}^{\text{outer}}$ fails then halt.
3. Reconstruct a concept $c_\ell \in \texttt{INTERVAL}_d$ out of the noisy hypothesis h (as described in the reconstruction algorithm above) and return it.

Note that the sample complexity of $\mathcal{A}$ is $o_{\alpha',\beta',\epsilon}(d) = o_{\alpha,\beta,\epsilon}(d)$. Also note that the reconstruction step does not access D, but only the output of $\mathcal{A}^{\text{outer}}$. As $\mathcal{A}^{\text{outer}}$ is ϵ-differentially private, so is $\mathcal{A}$. Finally, note that the probability that $\mathcal{A}$ fails to output $c_\ell \in \texttt{INTERVAL}_d$ such that $\text{error}_{\mathcal{D}}(c_\ell, c) \le \alpha$ is bounded by the probability that the reconstruction algorithm fails, (i.e., $c_\ell \ne c^\star$) and the probability that $\mathcal{A}^{\text{inner}}$ fails to output $c^\star \in \texttt{INTERVAL}_d$ such that $\text{error}_{\mathcal{D}}(c^\star, c) \le \alpha^\star \le \alpha' \le \alpha$. Remember that $\mu \le 2\alpha'$. Since $2\alpha' \le 0.02$ (for $\alpha \le 1$) this implies that $\mu \le 0.02$ and the above condition $\mu \le 0.09$ is satisfied, and hence,

$$\Pr\Big[\underset{\mathcal{D}}{\text{error}}(c_\ell, c_t) \ge \alpha\Big] \le \beta^\star + 8\mu \le \beta' + 8 \cdot 2\alpha' \le \frac{\beta}{4} + 16 \cdot \frac{\beta}{100} \le \beta.$$

Note that $\beta^\star \le \beta'$ from the definition of private independent noise learner. Thus, the algorithm $\mathcal{A}$ returns a concept $c_\ell = c^\star \in \texttt{INTERVAL}_d$ such that $\Pr[\text{error}_{\mathcal{D}}(c_\ell, c_t) \ge \alpha] \le \beta$, and so it is a proper ϵ-differentially private (α, β)-PAC learner for $\texttt{INTERVAL}_d$ with sample complexity $o_{\alpha,\beta,\epsilon}(d)$, in contradiction to Lemma 5.2. □

6 Separation between efficient and inefficient proper private PAC learning

In this section, we use the sample size lower bound for proper private learning $\texttt{POINT}_d$ (Corollary 3.8) to obtain a separation between the sample complexities of efficient and inefficient proper private PAC learning. In the case of efficient proper private learning, we use a slightly relaxed notion of proper learning for reasons explained below.

In our separation we use pseudorandom generators, which we now define. Let U_r represent a uniformly random string from $\{0,1\}^r$. Let $\ell(d) : \mathbb{N} \to \mathbb{N}$ be a function and $G = \{G_d\}_{d \in \mathbb{N}}$ be a deterministic algorithm such that on input from $\{0,1\}^{\ell(d)}$ it returns an output from $\{0,1\}^d$. Informally, we say that G is pseudorandom generator if on $\ell(d)$ truly random bits it outputs d bits that are indistinguishable from d random bits. Formally, for every probabilistic polynomial time algorithm $\mathcal{B}$ there exists a negligible function $\text{negl}(d)$ (i.e., a function that is asymptotically smaller than $1/d^c$ for all $c > 0$) such that

$$\big|\Pr\big[\mathcal{B}\big(G_d(U_{\ell(d)})\big) = 1\big] - \Pr\big[\mathcal{B}(U_d) = 1\big]\big| \le \text{negl}(d). \tag{10}$$

Pseudorandom generators G with $\ell(d) = \omega(\log d)$ exist under various strong hardness assumptions (Goldreich 2001). The difference $d - \ell(d)$ is defined as the *stretch* of the pseudorandom generator. Let $\texttt{POINT}_d = \{c_1, \ldots, c_{2^d}\}$. To an efficient (polynomially bounded) private learner, the concept $c_{G_d(U_{\ell(d)})}$ would appear as a uniformly random concept picked from $\texttt{POINT}_d$. Define concept class

$$\widehat{\texttt{POINT}}_d = \big\{c_{G_d(r)} \,|\, r \in \{0,1\}^{\ell(d)}\big\}.$$

First, we show that, assuming G is a pseudorandom generator, there exists no efficient proper learner for $\widehat{\texttt{POINT}}_d$ (note that this statement holds even without the privacy constraint). Assume $\mathcal{A}_p$ is an efficient proper learner for $\widehat{\texttt{POINT}}_d$. We use $\mathcal{A}_p$ to construct a *distinguisher* for the pseudorandom generator as follows: Given $j \in \{1, \ldots, 2^d\}$, we construct the database D with m entries $(j, 1)$. If $\mathcal{A}_p(D) = c_j$, then the distinguisher returns 1, otherwise it returns 0.

(1) If j is in the image of G_d, then by the utility guarantee of the proper learner, $\mathcal{A}_p$ has to return c_j on D with probability at least $1 - \beta$. Thus, the distinguisher returns 1 with probability at least $1 - \beta$ when j is chosen from $G_d(U_{\ell(d)})$.

(2) If j is not in the image of G_d, then the database D is not labeled consistently by any concept in $\widehat{\texttt{POINT}}_d$. Consider any such j, a proper learner that returns a hypothesis from $\widehat{\texttt{POINT}}_d$ implies a distinguisher that never returns 1 (i.e., always returns 0). Therefore, the probability that the distinguisher returns 1 when $j = U_d$ is at most the probability that j is in the image of G_d, which is at most $\ell(d)/2^d = \mathrm{negl}(d)$.

To summarize, assuming $\mathcal{A}_p$ is an efficient proper learner for $\widehat{\texttt{POINT}}_d$, the distinguisher will return 1 with probability at least $1-\beta$ when $j = G_d(U_{\ell(d)})$, and with probability at most $\mathrm{negl}(d)$ when $j = U_d$, in contradiction to (10). We conclude that no efficient proper learner exists for $\widehat{\texttt{POINT}}_d$ and, therefore, we relax in the following our notion of proper private learners for $\widehat{\texttt{POINT}}$ to allow outputting hypothesis from $\texttt{POINT}$. We show that under this liberal relaxation, *efficient* proper learning of $\widehat{\texttt{POINT}}_d$ with sample complexity $o(d)$ is not possible. However, we show that *inefficient* proper private learning of $\widehat{\texttt{POINT}}_d$ with sample complexity $o(d)$ is possible under the strict definition of proper learning.

Sample complexity of efficiently private learning $\widehat{\texttt{POINT}}_d$ *using* $\texttt{POINT}_d$ Consider an efficient private learner $\mathcal{A}_{\mathrm{eff}}$ that learns $\widehat{\texttt{POINT}}_d$ using $\texttt{POINT}_d$ and has sample complexity m. We now show that either a distinguisher exists for the pseudorandom generator G_d or $m = \Omega_{\beta,\epsilon}(d)$. Assume $\beta < 1/4$.

We use $\mathcal{A}_{\mathrm{eff}}$ to construct a distinguisher for the pseudorandom generator as follows: Given $j \in \{1, \dots, 2^d\}$, we construct the database D with m entries $(j, 1)$. If $\mathcal{A}_{\mathrm{eff}}(D) = c_j$, then the distinguisher returns 1, otherwise it returns 0.

If for at least a 3/4th fraction of the values $j \in [2^d]$, algorithm $\mathcal{A}_{\mathrm{eff}}$, when applied to a database with m entries $(j, 1)$, does not return c_j with probability at least $3/4$, then the distinguisher succeeds in breaking the pseudorandom generator. This is because if the above statement is not true then the distinguisher returns 1 with probability at most $3/4$ when $j = U_d$, and the distinguisher will return 1 with probability at least $1-\beta > 3/4$ when $j = G_d(U_{\ell(d)})$.[11]

However, arguments similar as in the proof of Theorem 3.6 show that it is not possible to have a learner that on 3/4th fraction of the values $j \in [2^d]$, when applied to a database with $m = o((d + \log(1/\beta))/\epsilon)$ entries $(j, 1)$, returns c_j with probability at least $3/4$. This means that either we have a distinguisher for the pseudorandom generator or the sample complexity of $\mathcal{A}_{\mathrm{eff}}$ is at least $\Omega_{\beta,\epsilon}(d)$. So, assuming the existence of a pseudorandom generator, we get that there exists no efficient private learner that learns $\widehat{\texttt{POINT}}_d$ using $\texttt{POINT}_d$ and has $o((d + \log(1/\beta))/\epsilon)$ sample complexity.[12]

Sample complexity of inefficient proper private learners for $\widehat{\texttt{POINT}}_d$ If the learner is not polynomially bounded, then it can use the algorithm from Theorem 3.2 to privately learn $\widehat{\texttt{POINT}}_d$. Since $|\widehat{\texttt{POINT}}_d| = 2^{\ell(d)}$, the private learner from Theorem 3.2 uses $O((\ell(d) + \log(1/\beta))/(\epsilon\alpha))$ samples.

We get the following separation between efficient and inefficient proper private learning:

Theorem 6.1 *Let $\ell(d)$ be any function that grows as $\omega(\log d)$. Assuming the existence of a pseudorandom generator $G_d : \{0,1\}^{\ell(d)} \to \{0,1\}^d$, there exists no efficient proper*

[11] If j is in the image of G_d, then the analysis is same as (1) above. By utility guarantees, $\mathcal{A}_{\mathrm{eff}}$ has to return c_j on D with probability at least $1-\beta$. Thus, the distinguisher returns 1 with probability at least $1-\beta$ when j chosen from $G_d(U_{\ell(d)})$.

[12] An almost matching upper bound of $O((d + \log(1/\beta))/\epsilon\alpha)$ on the sample complexity for efficiently private learning $\widehat{\texttt{POINT}}_d$ using $\texttt{POINT}_d$ can be obtained as in Lemma 3.4.

PAC learner for $\widehat{\texttt{POINT}}_d$ *and every efficient (polynomial-time) private PAC learner that learns* $\widehat{\texttt{POINT}}_d$ *using* $\texttt{POINT}_d$ *requires* $\Omega((d+\log(1/\beta))/\epsilon)$ *samples, whereas there exists an inefficient proper private PAC learner that can learn* $\widehat{\texttt{POINT}}_d$ *using* $O((\ell(d)+\log(1/\beta))/(\epsilon\alpha))$ *samples.*

Remark 6.2 In the non-private setting, there exists an efficient proper learner that can learn $\widehat{\texttt{POINT}}_d$ using $\texttt{POINT}_d$ with $O((\log(1/\alpha)+\log(1/\beta))/\alpha)$ samples (as $\mathit{VCDIM}(\widehat{\texttt{POINT}}_d)=1$). In the non-private setting, we also know that even inefficient learners require $\Omega(\log(1/\beta)/\alpha)$ samples (Ehrenfeucht et al. 1989; Kearns and Vazirani 1994). Therefore, for $\widehat{\texttt{POINT}}_d$ the sample complexity difference that we observe in Theorem 6.1 does not exist without the privacy constraint.

7 Lower bounds for non-interactive sanitization

We now prove a lower bound on the database size (or sample size) needed to privately release an output that is useful for all concepts in a concept class. We start by recalling a definition and a result of Blum et al. (2008).

Let $X=\{X_d\}_{d\in\mathbb{N}}$ be some discretized domain and consider a class of predicates $\mathcal{C}$ over X. A database D contains points taken from X_d. A predicate query Q_c for $c: X_d\to\{0,1\}$ in $\mathcal{C}$ is defined as

$$Q_c(D)=\frac{|\{d_i\in D : c(d_i)=1\}|}{|D|}.$$

A sanitizer (or data release mechanism) is a differentially private algorithm $\mathcal{A}$ that gets as input a database D and outputs another database $\widehat{D}$ with entries taken from X_d. An algorithm $\mathcal{A}$ is (α,β)-useful for predicates in the class $\mathcal{C}$ if for every database D with probability at least $1-\beta$ the algorithm $\mathcal{A}(D)$ returns a database $\widehat{D}$ such that for every $c\in C$,

$$\left|Q_c(D)-Q_c(\widehat{D})\right|<\alpha.$$

Theorem 7.1 (Blum et al. 2008) *For any class of predicates* $\mathcal{C}$, *and any database* $D\in X_d^m$, *such that*

$$m\geq O\left(\frac{\log(|X_d|)\cdot \mathit{VCDIM}(\mathcal{C})\log(1/\alpha)}{\alpha^3\epsilon}+\frac{\log(1/\beta)}{\epsilon\alpha}\right),$$

there exists an (α,β)*-useful mechanism* $\mathcal{A}$ *that preserves* ϵ*-differential privacy. The algorithm might not be efficient.*

We show that the dependency on $\log(|X_d|)$ in Theorem 7.1 is essential: there exists a class of predicates $\mathcal{C}$ with VC-dimension $O(1)$ that requires $|D|=\Omega_{\alpha,\beta,\epsilon}(\log(|X_d|))$. For our lower bound, the sanitized output $\widehat{D}$ could be any arbitrary data structure (not necessarily a synthetic database). Remember that a synthetic database contains data drawn from the same domain as the original database and Theorem 7.1 outputs a synthetic database. For simplicity, however, here we focus on the case where the output is a synthetic database. The proof of this lower bound uses ideas from Sect. 3.1.

Theorem 7.2 *Every* ϵ*-differentially private non-interactive mechanism that is* (α,β)*-useful for* $\texttt{POINT}_d$ *requires an input database of size* $\Omega((d+\log(1/\beta))/(\epsilon\alpha))$.

Proof Let $T = 2^d$ and $X_d = [T]$ be the domain. Consider the class $\mathtt{POINT}_d$. For every $i \in [T]$, construct a database $D_i \in X_d^m$ by setting $(1-3\alpha)m$ entries as 1 and the remaining $3\alpha m$ entries as i (for $i = 1$ all entries of D_1 are 1). For $i \in [T] \setminus \{1\}$, we say that a database $\widehat{D}$ is α-useful for D_i if $2\alpha < Q_{c_i}(\widehat{D}) < 4\alpha$ and $1 - 4\alpha < Q_{c_1}(\widehat{D}) < 1 - 2\alpha$. We say that $\widehat{D}$ is α-useful for D_1 if $1 - \alpha < Q_{c_1}(\widehat{D}) \leq 1$. It follows that for $i \neq j$, if $\widehat{D}$ is α-useful for D_i then it is not α-useful for D_j.

Let $\widehat{\mathbb{D}}_i$ be the set of all databases that are α-useful for D_i. Note that for all $i \neq 1$, databases D_1 and D_i differ on $3\alpha m$ entries, and by our previous observation, $\widehat{\mathbb{D}}_1 \cap \widehat{\mathbb{D}}_i = \emptyset$. Let $\mathcal{A}$ be an (α, β)-useful private release mechanism for $\mathtt{POINT}_d$. For all i, on input D_i mechanism $\mathcal{A}$ should pick an output from $\widehat{\mathbb{D}}_i$ with probability at least $1 - \beta$. We get by the differential privacy of $\mathcal{A}$ that

$$\Pr\bigl[\mathcal{A}(D_1) \in \widehat{\mathbb{D}}_i\bigr] \geq \exp(-3\epsilon\alpha m) \Pr\bigl[\mathcal{A}(D_i) \in \widehat{\mathbb{D}}_i\bigr] \geq \exp(-3\epsilon\alpha m) \cdot (1-\beta).$$

Hence,

$$\begin{aligned}\Pr\bigl[\mathcal{A}(D_1) \notin \widehat{\mathbb{D}}_1\bigr] &\geq \Pr\Bigl[\mathcal{A}(D_1) \in \bigcup_{i\neq 1} \widehat{\mathbb{D}}_i\Bigr] \\ &= \sum_{i \neq 1} \Pr\bigl[\mathcal{A}(D_1) \in \widehat{\mathbb{D}}_i\bigr] \quad \text{(sets } \widehat{\mathbb{D}}_i \text{ are disjoint)} \\ &\geq (T-1)\exp(-3\epsilon\alpha m) \cdot (1-\beta).\end{aligned}$$

On the other hand, since $\mathcal{A}$ is (α, β)-useful, $\Pr[\mathcal{A}(D_1) \notin \widehat{\mathbb{D}}_1] < \beta$, and hence, we get that $m = \Omega((d + \log(1/\beta))/(\epsilon\alpha))$. □

Acknowledgements We thank Benny Applebaum, Eyal Kushilevitz, and Adam Smith for helpful initial discussions.

Amos Beimel's research was partly supported by the Israel Science Foundation (grant No. 938/09) and by the Frankel Center for Computer Science at Ben-Gurion University. Shiva Prasad Kasiviswanathan thanks Los Alamos National Laboratory and IBM T.J. Watson Research Center for supporting him while this research was performed. Hai Brenner and Kobbi Nissim's research was supported by the Israel Science Foundation (grant No. 860/06).

References

Beimel, A., Kasiviswanathan, S. P., & Nissim, K. (2010). Bounds on the sample complexity for private learning and private data release. In D. Micciancio (Ed.), *LNCS: Vol. 5978. TCC* (pp. 437–454). Berlin: Springer.

Beimel, A., Nissim, K., & Stemmer, U. (2013). Characterizing the sample complexity of private learners. In *ITCS* (pp. 97–110).

Blum, A., Dwork, C., McSherry, F., & Nissim, K. (2005). Practical privacy: the SuLQ framework. In *PODS* (pp. 128–138). New York: ACM.

Blum, A., Ligett, K., & Roth, A. (2008). A learning theory approach to non-interactive database privacy. In *STOC* (pp. 609–618). New York: ACM.

Blum, A., Ligett, K., & Roth, A. (2013). A learning theory approach to non-interactive database privacy. *Journal of the ACM*, *60*(2), 12.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, *36*(4), 929–965.

Chaudhuri, K., & Hsu, D. (2011). Sample complexity bounds for differentially private learning. *Journal of Machine Learning Research*, *19*, 155–186.

Chaudhuri, K., & Monteleoni, C. (2008). Privacy-preserving logistic regression. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *NIPS*, Cambridge: MIT Press.

Chaudhuri, K., Monteleoni, C., & Sarwate, A. D. (2011). Differentially private empirical risk minimization. *Journal of Machine Learning Research*, *12*, 1069–1109.

Chernoff, H. (1952). A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, *23*, 493–507.

Dwork, C. (2009). The differential privacy frontier. In O. Reingold (Ed.), *LNCS: Vol. 5444*. *TCC* (pp. 496–502). Berlin: Springer.

Dwork, C. (2011). A firm foundation for private data analysis. *Communications of the ACM*, *54*(1), 86–95.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In S. Halevi & T. Rabin (Eds.), *LNCS: Vol. 3876*. *TCC* (pp. 265–284). Berlin: Springer.

Dwork, C., Naor, M., Reingold, O., Rothblum, G., & Vadhan, S. (2009). On the complexity of differentially private data release. In *STOC* (pp. 381–390). New York: ACM.

Ehrenfeucht, A., Haussler, D., Kearns, M. J., & Valiant, L. G. (1989). A general lower bound on the number of examples needed for learning. *Information and Computation*, *82*(3), 247–261.

Goldreich, O. (2001). *Foundations of cryptography, volume basic tools*. Cambridge: Cambridge University Press.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, *58*(301), 13–30.

Hughes, D. R., & Piper, F. C. (1973). *Projective planes* (Vol. 6). Berlin: Springer.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., & Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, *40*(3), 793–826.

Kearns, M. J. (1998). Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, *45*(6), 983–1006. Preliminary version in proceedings of STOC'93.

Kearns, M. J., & Vazirani, U. V. (1994). *An introduction to computational learning theory*. Cambridge: MIT Press.

Kifer, D., Smith, A. D., & Thakurta, A. (2012). Private convex optimization for empirical risk minimization with applications to high-dimensional regression. *Journal of Machine Learning Research*, *23*, 25.

McSherry, F., & Talwar, K. (2007). Mechanism design via differential privacy. In *FOCS* (pp. 94–103). New York: IEEE Press.

Mishra, N., & Sandler, M. (2006). Privacy via pseudorandom sketches. In *PODS* (pp. 143–152). New York: ACM.

Pitt, L., & Valiant, L. G. (1988). Computational limitations on learning from examples. *Journal of the ACM*, *35*(4), 965–984.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 1134–1142.

Vapnik, V. N., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, *16*, 264.