Private Query Release Assisted by Public Data

Raef Bassily [1] **Albert Cheu** [2] **Shay Moran** [3] **Aleksandar Nikolov** [4] **Jonathan Ullman** [2] **Zhiwei Steven Wu** [5]

Abstract

We study the problem of differentially private query release assisted by access to public data. In this problem, the goal is to answer a large class $\mathcal{H}$ of statistical queries with error no more than α using a combination of public and private samples. The algorithm is required to satisfy differential privacy only with respect to the private samples. We study the limits of this task in terms of the private and public sample complexities. Our upper and lower bounds on the private sample complexity have matching dependence on the dual VC-dimension of $\mathcal{H}$. For a large category of query classes, our bounds on the public sample complexity have matching dependence on α.

1. Introduction

The ability to answer statistical queries on a sensitive data set in a privacy-preserving way is one of the most fundamental primitives in private data analysis. In particular, this task has been at the center of the literature of differential privacy since its emergence (Dinur & Nissim, 2003; Dwork et al., 2006) and is also central to the upcoming US Census 2020 release (Dajani et al., 2017). In its basic form, the problem of differentially private query release can be described as follows. Given a class $\mathcal{H}$ of queries $h : \mathcal{X} \to \{\pm 1\}$[1] defined over some domain $\mathcal{X}$, and a data set $\vec{x}$ of i.i.d. samples from some unknown distribution $\mathbf{D}$ over $\mathcal{X}$, the goal is to construct an (ε, δ)-differentially private algorithm that, given $\mathcal{H}$ and $\vec{x}$, outputs a mapping $G : \mathcal{H} \to [-1, 1]$ such that for every $h \in \mathcal{H}$, $G(h)$ gives an accurate estimate for the true mean $\mathop{\mathbb{E}}_{x\sim\mathbf{D}}[h(x)]$ up to some error α. A central question in this problem is concerned with characterizing the *private sample complexity*, which is the least amount of private samples required to perform this task up to some error α. This question has been extensively studied in the literature on differential privacy (Dinur & Nissim, 2003; Hardt & Rothblum, 2010; Muthukrishnan & Nikolov, 2012; Blum et al., 2013; Bun et al., 2018; Steinke & Ullman, 2015). For general query classes, it was shown that the optimal bound on the private sample complexity in terms of $|\mathcal{X}|$, $|\mathcal{H}$, and the privacy parameters is attained by the Private Multiplicative Weights (PMW) algorithm due to Hardt and Rothblum (Hardt & Rothblum, 2010). This optimality was established by the lower bound due to Bun et al. (Bun et al., 2018). This result implied the impossibility of differentially private query release for certain classes of infinite size. Moreover, subsequent results by (Bun et al., 2015; Alon et al., 2019b) implied that this impossibility is true even for a simple class such as one-dimensional thresholds over $\mathbb{R}$. On the other hand, without any privacy constraints, the query release problem is equivalent to attaining uniform convergence over $\mathcal{H}$, and hence the sample complexity is given by d/α^2, where d is the VC-dimension of $\mathcal{H}$.

In practice, it is often feasible to collect some amount of "public" data that poses no privacy concerns. For example, in the language of consumer privacy, there is considerable amount of data collected from the so-called "opt-in" users, who voluntarily offer or sell their data to companies or organizations. Such data is deemed by its original owner to pose no threat to personal privacy. There are also a variety of other sources of public data that can be harnessed.

Motivated by the above observation, and by the limitations in the standard model of differentially private query release, in this work, we study a relaxed setting of this problem, which we call *Public-data-Assisted Private (PAP)* query release. In this setting, the query-release algorithm has access to two types of samples from the unknown distribution: private samples that contain personal and sensitive information (as in the standard setting), and public samples. The goal is to design algorithms that can exploit as little public data as possible to achieve non-trivial savings in sample complexity over standard DP query-release algorithms, while still providing strong privacy guarantees for the private dataset.

[1]Department of Computer Science & Engineering, The Ohio State University. [2]Khoury College of Computer Sciences, Northeastern University [3]Google AI Princeton [4]Department of Computer Science, University of Toronto [5]Department of Computer Science and Engineering, University of Minnesota. Correspondence to: Raef Bassily <rbbassily@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

[1]In this work, we focus on classes of binary functions (known in the literature of DP as *counting queries*).

1.1. Our results

In this work we study the private and public sample complexities of PAP query-release algorithms, and give upper and lower bounds on both.

1. **Upper bounds:** We give a construction of a PAP algorithm that solves the query release problem for any query class $\mathcal{H}$ using only $\approx d/\alpha$ public samples, and $\approx \sqrt{p}d^{3/2}/\alpha^2$ private samples, where d is the VC-dimnension of $\mathcal{H}$ and p is the dual VC-dimension of $\mathcal{H}$.

   Recall that d/α^2 samples are necessary even without privacy constraints, so our upper bound on the public sample complexity shows a nearly quadratic saving.

2. **Lower bound on private sample complexity:** We show that there is a query class $\mathcal{H}$ with VC-dimension $\log(p)$ and dual VC-dimension p such that any PAP algorithm either requires $\Omega(1/\alpha^2)$ public samples or requires $\tilde{\Omega}(\sqrt{p}/\alpha)$ total samples. Thus the $\sqrt{p}$ dependence above is unavoidable. For this class, $O(\log(p)/\alpha^2)$ public samples are enough to solve the problem with no private samples, and $\tilde{O}(\log(p)/\alpha^2 + \sqrt{p}/\alpha)$ private samples are enough to solve the problem with no public samples. Thus, public samples do not help, unless there are nearly enough public samples to solve the problem without private samples.

3. **Lower bound on public sample complexity:** We show that if the class $\mathcal{H}$ has infinite *Littlestone dimension*,[2] then any PAP query-release algorithm for $\mathcal{H}$ must have *public* sample complexity $\Omega(1/\alpha)$. The simplest example of such a class is the class of one-dimensional thresholds over $\mathbb{R}$. This class has VC-dimension 1, and therefore demonstrates that the upper bound above is nearly tight.

1.2. Techniques

Upper bounds: The first key step in our construction for the upper bound is to use the public data to construct a finite class $\widetilde{\mathcal{H}}$ that forms a "good approximation" of the original class $\mathcal{H}$. Such approximation is captured via the notion of an α-cover (Definition 1). The number of public examples that suffices to construct such approximation is about d/α (Alon et al., 2019a). Given this finite class $\widetilde{\mathcal{H}}$, we then reduce the original domain $\mathcal{X}$ to a finite set $\mathcal{X}_{\widetilde{\mathcal{H}}}$ of representative domain points, which is defined via an equivalence relation induced by $\mathcal{H}$ over $\mathcal{X}$ (Definition 2). Using Sauer's Lemma, we can show that the size of such a representative domain is at most $\left(\frac{e\,|\widetilde{\mathcal{H}}|}{p}\right)^p$, where p is the dual VC-dimension of $\mathcal{H}$. At this point, we can reduce our problem to DP query-release for the finite query class $\widetilde{\mathcal{H}}$ and the finite domain $\mathcal{X}_{\widetilde{\mathcal{H}}}$, which we can solve via the PMW algorithm (Hardt & Rothblum, 2010; Dwork & Roth, 2014).

Lower bound on private sample complexity: The proof of the lower bound is based on the *robust tracing attack* of Dwork et al. (Dwork et al., 2015). That work proves privacy lower bounds for the class of *decision stumps* over the domain $\{-1,1\}^p$, which contains queries of the form $q_i(\vec{x}) = x_i$. Specifically, they show that for any algorithm that takes at most $s \approx \sqrt{p}/\alpha$ samples, and releases the class of decision stumps with accuracy α, there is some attacker that can "detect" the presence of at least $t \approx 1/\alpha^2$ of the samples. Therefore, if the number of public samples is at most $t-1$, the attacker can detect the presence of one of the private samples, which means the algorithm cannot be differentially private with respect to the private samples.

Lower bound on public sample complexity: This lower bound is derived in two steps. First, we show that PAP query-release for a class $\mathcal{H}$ implies PAP learning (studied in (Beimel et al., 2013; Alon et al., 2019a)[3]) of the same class with the same amount of public data. This step follows from a straightforward generalization of an analogous result by (Beimel et al., 2013) in the standard DP model with no public data. Second, we invoke the lower bound of (Alon et al., 2019a) on the public sample complexity of PAP learning.

1.3. Other related work

To the best of our knowledge, our work is the first to formally study differentially private query release assisted with public data. There have been a few works that considered the analog of our problem in the supervised-learning setting. In particular, the notion of differentially private PAC learning assisted with public data was introduced by Beimel et al. in (Beimel et al., 2013), where it was called "semi-private learning." They gave a construction of a learning algorithm in this setting, and derived upper bounds on the private and public sample complexities. The paper (Alon et al., 2019a) revisited this problem and gave nearly optimal bounds on the private and public sample complexities in the agnostic PAC model. The work of (Alon et al., 2019a) emphasizes the notion of α-covers as a useful tool in the construction of such learning algorithms. Our PAP algorithm demonstrates the usefulness of this notion in the query-release setting.

In the learning setting, there are also several other works that considered similar relaxations, e.g., (Chaudhuri & Hsu, 2011; Beimel et al., 2013) who studied the notion of

[2] The Littlestone dimension is a combinatorial parameter that arises in online learning (Littlestone, 1988; Ben-David et al., 2009).

[3] These works use the term "semi-private" instead of PAP.

"label-private learning", where only the labels in the training set are considered private. Another line of work considers the setting in which the learning task can be reduced to privately answering classification queries (Hamm et al., 2016; Papernot et al., 2017; 2018; Bassily et al., 2018; Nandi & Bassily, 2019), where the goal is to construct a differentially private classification algorithm that predicts the labels of a sequence of public feature-vectors given a private training set as input.

2. Preliminaries

We use $\mathcal{X}$ to denote an arbitrary data universe, $\mathbf{D}$ to denote a distribution over $\mathcal{X}$, and $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ to denote a binary *hypothesis class*.

2.1. Tools from learning theory

The VC dimension of a binary hypothesis class $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ is denoted by $\mathsf{VC}(\mathcal{H})$.

We will use the following notion of coverings:

Definition 1 (α-cover for a hypothesis class)**.** A family of hypotheses $\widetilde{\mathcal{H}}$ is said to form an α-cover for a hypothesis class $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ with respect to a distribution $\mathbf{D}$ over $\mathcal{X}$ if for every $h \in \mathcal{H}$, there is $\tilde{h} \in \widetilde{\mathcal{H}}$ such that $\mathbb{P}_{x \sim \mathbf{D}}\left[h(x) \neq \tilde{h}(x)\right] \leq \alpha$.

Definition 2 (Representative domain (a.k.a. the dual class))**.** Let $\widetilde{\mathcal{H}} \subseteq \{\pm 1\}^{\mathcal{X}}$ be a hypothesis class. Define an equivalence relation on $\mathcal{X}$ by $x \simeq x'$ if and only if $(\forall h \in \widetilde{\mathcal{H}}) : h(x) = h(x')$. The representative domain induced by $\widetilde{\mathcal{H}}$ on $\mathcal{X}$, denoted by $\mathcal{X}_{\widetilde{\mathcal{H}}}$, is a complete set of distinct representatives from $\mathcal{X}$ for this equivalence relation.

For example, let $\widetilde{\mathcal{H}}$ be a class of M binary thresholds over $\mathbb{R}$ given by: $h_{t_j}(x) = +1$ iff $x \leq t_j$, $j \in [M]$, $-\infty < t_1 < t_2 < \ldots < t_M < \infty$. Then, a representative domain in this case is a set of $M+1$ distinct elements; one from each of the following intervals $(-\infty, t_1], (t_1, t_2], \ldots, (t_M, \infty)$. More generally, if $\widetilde{\mathcal{H}}$ is a class of halfspaces then a representative domain contains exactly one point in each cell of the hyperplane arrangement induced by $\widetilde{\mathcal{H}}$ (see Figure 1).

Note that when $\widetilde{\mathcal{H}}$ is finite then any representative domain for $\widetilde{\mathcal{H}}$ has size at most $2^{|\widetilde{\mathcal{H}}|}$, since the equivalence class of each $x \in \mathcal{X}$ is determined by the binary vector $(h(x))_{h \in \widetilde{\mathcal{H}}}$. Moreover, one can also make the following simple claim, which is a direct consequence of the Sauer-Shelah Lemma (Sauer, 1972) together with the fact that a representative domain $\mathcal{X}_{\widetilde{\mathcal{H}}}$ has a one-to-one correspondence with the *dual* class of $\widetilde{\mathcal{H}}$. We use $\mathsf{VC}^{\perp}(\mathcal{H})$ to denote the dual VC-dimension of a hypothesis class $\mathcal{H}$, namely, $\mathsf{VC}^{\perp}(\mathcal{H})$ is the VC-dimension of the dual class of $\mathcal{H}$.

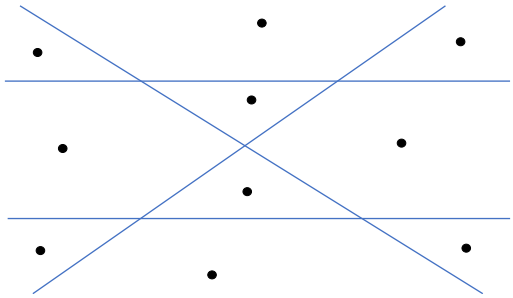

Figure 1. A representative domain for a finite class of halfspaces

Claim 3. *Let $\widetilde{\mathcal{H}}$ be a finite class of binary functions defined over a domain $\mathcal{X}$. Then, the size of a representative domain $\mathcal{X}_{\widetilde{\mathcal{H}}}$ satisfies:* $|\mathcal{X}_{\widetilde{\mathcal{H}}}| \leq \left(\frac{e\,|\widetilde{\mathcal{H}}|}{\mathsf{VC}^{\perp}(\widetilde{\mathcal{H}})}\right)^{\mathsf{VC}^{\perp}(\widetilde{\mathcal{H}})}$.

The following useful fact gives a worst-case upper bound on the dual VC-dimension in terms of the VC-dimension.

Fact 4 ((Assouad, 1983))**.** *Let $\mathcal{H}$ be a binary hypothesis class. The dual VC-dimension of $\mathcal{H}$ satisfies:* $\mathsf{VC}^{\perp}(\mathcal{H}) < 2^{\mathsf{VC}(\mathcal{H})+1}$.

Notation: In Section 3, we will use the following notation. Let $\widetilde{\mathcal{H}}$ be a hypothesis class defined over a domain $\mathcal{X}$. For any $x \in \mathcal{X}$, we define $\mathcal{X}_{\widetilde{\mathcal{H}}}(x)$ as the representative $s \in \mathcal{X}_{\widetilde{\mathcal{H}}}$ such that $x \simeq s$, where $\simeq$ is the equivalence relation described in Definition 2. Note that by definition this $s \in \mathcal{X}_{\widetilde{\mathcal{H}}}$ is unique. Moreover, for any n and any $\vec{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$, we define $\mathcal{X}_{\widetilde{\mathcal{H}}}(\vec{x}) \triangleq \left(\mathcal{X}_{\widetilde{\mathcal{H}}}(x_1), \ldots, \mathcal{X}_{\widetilde{\mathcal{H}}}(x_n)\right)$.

Definition 5 (Query Release)**.** Given a distribution $\mathbf{D}$ over $\mathcal{X}$ and a binary hypothesis class $\mathcal{H}$, a *query release* data structure $G \in [-1,1]^{\mathcal{H}}$ (equivalently, $G : \mathcal{H} \to [-1,1]$) estimates the expected label $\mathbb{E}_{x \sim \mathbf{D}}[h(x)]$ for all $h \in \mathcal{H}$. The worst-case error is defined as

$$\mathrm{Err}_{\mathbf{D},\mathcal{H}}(G) \triangleq \sup_{h \in \mathcal{H}} |\, G(h) - \mathop{\mathbb{E}}_{x \sim \mathbf{D}}[h(x)]\,|.$$

2.2. Tools from Differential Privacy

Two datasets $\vec{x}, \vec{x}\,' \in \mathcal{X}^n$ are neighboring when they differ on one element.

Definition 6 (Differential Privacy)**.** A randomized algorithm $A : \mathcal{X}^n \to \mathcal{Z}$ is (ε, δ)-differentially private if for all neighboring $\vec{x}, \vec{x}\,'$ and all $Z \subseteq \mathcal{Z}$

$$\mathbb{P}[A(\vec{x}) \in Z] \leq e^{\varepsilon}\mathbb{P}[A(\vec{x}\,') \in Z] + \delta$$

Private Multiplicative Weights (PMW): In our construction in Section 3, we will use, as a black box, a well-studied algorithm in differential privacy known as Private

Multiplicative-Weights (Hardt & Rothblum, 2010). We will use a special case of the offline version of the PMW algorithm. Namely, the input query class $\widetilde{\mathcal{H}}$ is finite, and PMW runs over all the queries in the input class (in any order) to perform its updates, and finally outputs a query release data structure $\widetilde{G} \in [-1,1]^{\widetilde{\mathcal{H}}}$. When the input private data set $\vec{s}$ is drawn i.i.d. from some unknown distribution $\widetilde{\mathbf{D}}$, the accuracy goal is to have a data structure $\widetilde{G}$ such that $\mathrm{Err}_{\widetilde{\mathbf{D}},\widetilde{\mathcal{H}}}(\widetilde{G})$ is small. The outline (inputs and output of the PMW algorithm) is described in Algorithm 1.

Algorithm 1 An outline for the Private Multiplicative Weights Algorithm (PMW)

Input: Private data set $\vec{s} \in \widetilde{\mathcal{X}}^n$ (where $\widetilde{\mathcal{X}}$ is a finite domain); A finite query (hypothesis) class $\widetilde{\mathcal{H}} \subseteq \{\pm 1\}^{\widetilde{\mathcal{X}}}$; accuracy parameters α, β; privacy parameters ε, δ.

Output: A data structure $\widetilde{G} : \widetilde{\mathcal{H}} \to [-1,1]$.

The following lemma is an immediate corollary of the accuracy guarantee of the PMW algorithm (Hardt & Rothblum, 2010; Dwork & Roth, 2014). In particular, it follows from combining the empirical accuracy of PMW with a standard uniform-convergence argument.

Lemma 7 (Corollary of Theorem 4.15 in (Dwork & Roth, 2014))**.** *For any $0 < \varepsilon, \delta < 1$, the PMW algorithm is (ε, δ)-differentially private. Let $\widetilde{\mathbf{D}}$ be any distribution over $\widetilde{\mathcal{X}}$. For any $0 < \alpha, \beta < 1$, given an input data set $\vec{s} \sim \widetilde{\mathbf{D}}^n$ such that*

$$n \geq \frac{200}{\varepsilon\,\alpha^2} \cdot \sqrt{\log\left(|\widetilde{\mathcal{X}}|\right)\,\log(2/\delta)} \cdot \left(\log\left(|\widetilde{\mathcal{H}}|\right) + \log\left(\frac{128\,\log\left(|\widetilde{\mathcal{X}}|\right)}{\alpha^2\beta}\right)\right),$$

then, with probability at least $1-\beta$, PMW outputs a data structure $\widetilde{G}$ satisfying $Err_{\widetilde{\mathbf{D}},\widetilde{\mathcal{H}}}(\widetilde{G}) \leq \alpha$.

2.3. Our Model: Differentially private query release assisted by public data

In this paper, we study an extension of the problem of differentially private query release (Dwork & Roth, 2014) where the input data to the algorithm comes in two types: private and public. More precisely, the problem we study can be defined as follows. Let $\mathbf{D}$ be any distribution over a data domain $\mathcal{X}$. Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be a set of queries (here, we assume that $\mathcal{H}$ is a binary hypothesis class). We consider a family of algorithms whose generic description (namely, inputs and outputs) is given by Algorithm 2.

Given the query class $\mathcal{H}$, a private data set $\vec{x} \sim \mathbf{D}^n$ (i.e., a data set whose elements belong to n private users), and a public data set $\vec{w} \in \mathcal{X}^m$ (i.e., a data set whose elements belong to m users with no privacy constraint), the algorithm outputs a query release data structure $G : \mathcal{H} \to [-1,1]$. Such an algorithm is required to be (ε, δ)-differentially private but only with respect to the private data set. We call such an algorithm *Public-data-Assisted Private (PAP) query-release* algorithm. The accuracy/utility of the algorithm is determined by the worst-case estimation error incurred by its output data structure G on any query (hypothesis) $h \in \mathcal{H}$.

Algorithm 2 An outline for a generic Public-data-Assisted Private (PAP) algorithm for query release

Input: Private data set $\vec{x} \in \mathcal{X}^n$; public data set $\vec{w} \in \mathcal{X}^m$; A query (hypothesis) class $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$; accuracy and confidence parameters α, β; privacy parameters ε, δ.

Output: A data structure $G : \mathcal{H} \to [-1,1]$.

Definition 8 ($(\alpha, \beta, \varepsilon, \delta)$ PAP query-release algorithm)**.** Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be a query class. Let $\mathcal{A} : \mathcal{X}^* \to [-1,1]^{\mathcal{H}}$ be a randomized algorithm in the family outlined in Algorithm 2. We say that $\mathcal{A}$ is $(\alpha, \beta, \varepsilon, \delta)$ Public-data-Assisted Private (PAP) query-release algorithm for $\mathcal{H}$ with private sample size n and public sample size m if the following conditions hold:

1. For every distribution $\mathbf{D}$ over $\mathcal{X}$, given data sets $\vec{x} \sim \mathbf{D}^n$ and $\vec{w} \sim \mathbf{D}^m$ as inputs to $\mathcal{A}$, with probability at least $1-\beta$ (over the choice of $\vec{x}, \vec{w}$, and the random coins of $\mathcal{A}$), $\mathcal{A}$ outputs a function (data structure) $\mathcal{A}(\vec{x}, \vec{w}) = G \in [-1,1]^{\mathcal{H}}$ satisfying $\mathrm{Err}_{\mathbf{D},\mathcal{H}}(G) \leq \alpha$.
2. For all $\vec{w} \in \mathcal{X}^m$, $\mathcal{A}(\cdot, \vec{w})$ is (ε, δ)-differentially private.

Remark 9. *In our description in Algorithm 2, the algorithm is required to output a data structure $G : \mathcal{H} \to [-1,1]$ and not necessarily a "synthetic" data set $\vec{v} \in \mathcal{X}^{n'}$ for some number n' as in what is referred to as "private proper sanitizers" in (Beimel et al., 2013). In that special case, obviously the output data set can be used to define a data structure G'; namely, for any $h \in \mathcal{H}$, $G'(h) \triangleq \frac{1}{n'}\sum_{i\in[n']} h(v_i)$. Moreover, in the general case, ignoring computational complexity, the output data structure can also be used to construct a data set as pointed out in Remark 2.18 of (Beimel et al., 2013) . In particular, given a data structure G whose error $\leq \alpha$, then it suffices find a data set $\vec{v} \in \mathcal{X}^{n'}$, where $n' > \mathsf{VC}(\mathcal{H})/\alpha^2$, such that $|\frac{1}{n'}\sum_{i\in[n']} h(v_i)) - G(h)| \leq 2\alpha$ for all $h \in \mathcal{H}$, and hence the accuracy requirement would follow by the triangle inequality. Also, we know that this data set must exist. This is because by a standard uniform-convergence*

argument, a data set $\vec{s} \sim \mathbf{D}^{n'}$ *will, with a non-zero probability, satisfy* $|\frac{1}{n'}\sum_{i\in[n']} h(s_i)) - \mathop{\mathbb{E}}_{x\sim\mathbf{D}}[h](x)| \leq \alpha$ *for all* $h \in \mathcal{H}$*, and hence, by the triangle inequality,* $|\frac{1}{n'}\sum_{i\in[n']} h(s_i)) - G(h)| \leq 2\alpha$ *for all* $h \in \mathcal{H}$*.*

3. A PAP Query-Release Algorithm for Classes of Finite VC-Dimension

We now describe a construction of a public-data-assisted private query release algorithm that works for any class with a finite VC-dimension.

Our construction is given by Algorithm 3. The key idea of the construction is to use the public data to create a finite α-cover $\widetilde{\mathcal{H}}$ for the input query class $\mathcal{H}$ (see Definition 1), then, run the PMW algorithm on the finite cover and the representative domain $\mathcal{X}_{\widetilde{\mathcal{H}}}$ given by the dual of $\widetilde{\mathcal{H}}$ (see Definition 2).

Theorem 10 (Upper Bound)**.** $\mathcal{A}_{\mathrm{PAP}}$ *(Algorithm 3) is an* $(\alpha, \beta, \varepsilon, \delta)$ *public-data-assisted private query-release algorithm for* $\mathcal{H}$ *whose private and public sample complexities satisfy:*

$$n = O\left(\frac{(d\,\log(1/\alpha) + \log(1/\beta))^{3/2}\sqrt{p\,\log(1/\delta)}}{\varepsilon\,\alpha^2}\right),$$

$$m = O\left(\frac{d\,\log(1/\alpha) + \log(1/\beta)}{\alpha}\right),$$

where $d = \mathsf{VC}(\mathcal{H})$ *and* $p = \mathsf{VC}^{\perp}(\mathcal{H})$*.*

Remark: Using Fact 4, we can further bound the private sample complexity for general query classes as

$$n = O\left(\frac{(d\,\log(1/\alpha) + \log(1/\beta))^{3/2}\sqrt{2^d\,\log(1/\delta)}}{\varepsilon\,\alpha^2}\right).$$

In the proof of Theorem 10, we use the following lemma from (Alon et al., 2019a).

Lemma 11 (Lemma 3.3 in (Alon et al., 2019a))**.** *Let* $\vec{w} \sim \mathbf{D}^m$*, where* $m = O\left(\frac{d\,\log(1/\alpha)+\log(1/\beta)}{\alpha}\right)$*. Then, with probability at least* $1 - \beta/2$*, the family* $\widetilde{\mathcal{H}}$ *constructed in Step 3 of Algorithm 3 is an* $\alpha/4$*-cover for* $\mathcal{H}$ *w.r.t.* $\mathbf{D}$*. In particular, for every* $h \in \mathcal{H}$*, we have* $\mathop{\mathbb{P}}_{x\sim\mathbf{D}}\left[h(x) \neq \tilde{h}(x)\right] \leq \alpha/4$*, where* $\tilde{h} = \mathrm{Project}_{\widetilde{\mathcal{H}},\vec{w}}(h)$ *(see Algorithm 3 for the definition of* $\mathrm{Project}_{\widetilde{\mathcal{H}},\vec{w}}$*).*

Proof of Theorem 10. First, note that for any realization of the public data set $\vec{w}$, $\mathcal{A}_{\mathrm{PAP}}$ is (ε, δ)-differentially private w.r.t. the private data set. Indeed, the private data set $\vec{x}$ is only used to construct $\vec{s} = \mathcal{X}_{\widetilde{\mathcal{H}}}(\vec{x})$, which is the input data set to the PMW algorithm. The output of PMW is

Algorithm 3 $\mathcal{A}_{\mathrm{PAP}}$ Public-data-assisted Private Query-Release Algorithm

Input: Private data set $\vec{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$; public data set $\vec{w} = (w_1, \ldots, w_m) \in \mathcal{X}^m$; A query class $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$; accuracy and confidence parameters α, β; privacy parameters ε, δ.

Output: A data structure $G : \mathcal{H} \to [-1, 1]$.

`/* Use public data to construct` α`-cover for` $\mathcal{H}$ `*/`

Let $T = \{\hat{w}_1, \ldots, \hat{w}_{\hat{m}}\}$ be the set of points in $\mathcal{X}$ appearing at least once in $\vec{w}$.

Let $\Pi_{\mathcal{H}}(T) = \{(h(\hat{w}_1), \ldots, h(\hat{w}_{\hat{m}})) : h \in \mathcal{H}\}$.

Initialize $\widetilde{\mathcal{H}} = \emptyset$.

For *each* $\vec{y} = (y_1, \ldots, y_{\hat{m}}) \in \Pi_{\mathcal{H}}(T)$

  Add to $\widetilde{\mathcal{H}}$ one arbitrary $h \in \mathcal{H}$ that satisfies $h(\hat{w}_j) = y_j$ for every $j = 1, \ldots, \hat{m}$.

Let $\mathcal{X}_{\widetilde{\mathcal{H}}}$ be a representative domain induced by $\widetilde{\mathcal{H}}$ on $\mathcal{X}$ (as in Definition 2).

`/* replace each point in the private data set` $\vec{x}$ `with its representative in` $\mathcal{X}_{\widetilde{\mathcal{H}}}$ `*/`

$\vec{s} \leftarrow \mathcal{X}_{\widetilde{\mathcal{H}}}(\vec{x})$

`/* Run PMW algorithm over the data-set of representatives` $\vec{s} \in \mathcal{X}^n_{\widetilde{\mathcal{H}}}$ `and` $\widetilde{\mathcal{H}}$ `*/`

$\widetilde{G} \leftarrow \mathrm{PMW}\left(\vec{s},\, \widetilde{\mathcal{H}},\, \alpha/2,\, \beta/2,\, \varepsilon,\, \delta\right)$.

Return $G = G\left(\vec{w},\, \widetilde{\mathcal{H}},\, \widetilde{G},\, \cdot\right)$ `/* see code below */`

//

`/* Construct a function` $G : \mathcal{H} \to [-1, 1]$ `as follows: */`

Function $G = G\left(\vec{w},\, \widetilde{\mathcal{H}},\, \widetilde{G},\, \cdot\right)$

Input: A query (hypothesis) $h \in \mathcal{H}$.

Output: An estimate $r \in [-1, 1]$.

$\tilde{h} \leftarrow \mathrm{Project}_{\widetilde{\mathcal{H}},\vec{w}}(h)$,

where $\mathrm{Project}_{\widetilde{\mathcal{H}},\vec{w}}(h)$ denotes the unique $\tilde{h} \in \widetilde{\mathcal{H}}$ s.t. $\left(\tilde{h}(w_1), \ldots, \tilde{h}(w_m)\right) = \left(h(w_1), \ldots, h(w_m)\right)$

$r \leftarrow \widetilde{G}(\tilde{h})$

Return r

then used to construct the output data structure G. Moreover, for any pair of neighboring data sets $\vec{x}, \vec{x}'$, the pair $\mathcal{X}_{\widetilde{\mathcal{H}}}(\vec{x})$, $\mathcal{X}_{\widetilde{\mathcal{H}}}(\vec{x}')$ cannot differ in more than one element. Hence, (ε, δ)-differential privacy of our construction follows from (ε, δ)-differential privacy of the PMW algorithm together with the fact that differential privacy is closed under post-processing.

Next, we prove the accuracy guarantee of our construction. By Lemma 11, it follows that with probability at least $1-\beta/2$, we have $\sup_{h\in\mathcal{H}} \left| \mathop{\mathbb{E}}_{x\sim\mathbf{D}}[h(x)] - \mathop{\mathbb{E}}_{x\sim\mathbf{D}}\left[\tilde{h}(x)\right] \right| \le \alpha/2$, where $\tilde{h} = \mathrm{Project}_{\widetilde{\mathcal{H}},\vec{w}}(h)$. Hence, it suffices to show that with probability at least $1-\beta/2$, $\mathrm{Err}_{\mathbf{D},\widetilde{\mathcal{H}}}(\widetilde{G}) \le \alpha/2$ (recall that $\widetilde{G}$ is the output data structure of the PMW algorithm). Note that by Sauer's lemma, we know $|\widetilde{\mathcal{H}}| \le \left(\frac{em}{d}\right)^d$, where $d = \mathsf{VC}(\mathcal{H})$. From the setting of m in the theorem statement, we hence have

$$\begin{aligned}\log\left(|\widetilde{\mathcal{H}}|\right) &= O\left(d\log(1/\alpha) + d\left(\log\left(1+\frac{\log(1/\beta)}{d}\right)\right)\right)\\ &= O\left(d\,\log(1/\alpha) + \log(1/\beta)\right),\end{aligned}$$

Moreover, by Claim 3, we have

$$\begin{aligned}\log\left(|\,\mathcal{X}_{\widetilde{\mathcal{H}}}\,|\right) &= O\left(\mathsf{VC}^{\perp}(\widetilde{\mathcal{H}})\left(d\,\log(1/\alpha)+\log(1/\beta)\right)\right)\\ &\le O\left(p\left(d\,\log(1/\alpha)+\log(1/\beta)\right)\right),\end{aligned}$$

where $p = \mathsf{VC}^{\perp}(\mathcal{H})$. Thus, given the setting of n in the theorem statement, Lemma 7 implies that with probability at least $1-\beta/2$, our instantiation of the PMW algorithm yields $\widetilde{G}$ that satisfies

$$\begin{aligned}\alpha/2 &\ge \sup_{\tilde{h}\in\widetilde{\mathcal{H}}}\left|\widetilde{G}(\tilde{h}) - \mathop{\mathbb{E}}_{x\sim\mathbf{D}}\left[\tilde{h}\left(\mathcal{X}_{\widetilde{\mathcal{H}}}(x)\right)\right]\right|\\ &= \sup_{\tilde{h}\in\widetilde{\mathcal{H}}}\left|\widetilde{G}(\tilde{h}) - \mathop{\mathbb{E}}_{x\sim\mathbf{D}}\left[\tilde{h}(x)\right]\right|\\ &= \mathrm{Err}_{\mathbf{D},\widetilde{\mathcal{H}}}\left(\widetilde{G}\right),\end{aligned}$$

which completes the proof. □

4. A Lower Bound for Releasing Decision Stumps

In this section we give an example of a hypothesis class—*decision stumps* on $\{\pm1\}^p$—where additional public data "doesn't help" for private query release. This concept class can be released using either $\tilde{O}(\log(p)/\alpha^2 + \sqrt{p}/\alpha)$ private samples and no public samples, or using $O(\log(p)/\alpha^2)$ public samples and no private samples, but we show that every PAP query-release algorithm requires either $\tilde{\Omega}(\sqrt{p}/\alpha)$ private samples or $\Omega(1/\alpha^2)$ public samples. That is, making some samples public does not reduce the overall sample complexity until the number of the public samples is nearly enough to solve the problem on its own.

The class of decision stumps on $\{\pm1\}^p$ has dual-VC-dimension p, but VC-dimension just $\log p$, so this lower bound implies that the polynomial dependence on the dual-VC-dimension in Theorem 10 cannot be improved—there are classes with dual-VC-dimension p that require either $\tilde{\Omega}(\sqrt{p}/\alpha)$ private samples or $\Omega(1/\alpha^2)$ public samples.

Definition 12 (Binary Decision Stumps)**.** For any $p \in \mathbb{N}$, let $\mathcal{S}_p$ be a hypothesis class of hypotheses $h : \{\pm1\}^p \to \{\pm1\}$ consisting of all hypotheses of the form $h_i(x) = x_i$ for $i \in [p]$.

Theorem 13 (Lower Bound for Releasing Decision Stumps)**.** *Fix any $p \in \mathbb{N}$ and $\alpha > 0$. Suppose $\mathcal{A}$ is a PAP algorithm that takes n private samples and m public samples, satisfies $(1, 1/4n)$-differential privacy, and is (α,α)-accurate for the class of decision stumps $\mathcal{S}_p$. Then either $n = \tilde{\Omega}(\frac{\sqrt{p}}{\alpha})$ or $m = \Omega(1/\alpha^2)$.*

Thus, if $m = o(1/\alpha^2)$, then the number of private samples must scale proportionally to $\sqrt{p}$ as in our upper bound in Theorem 10.

The main ingredient in the proof is a result of Dwork et al. (Dwork et al., 2015). Informally, what this theorem says is that for any algorithm that releases accurate answers to the class of decision stumps using too small of a dataset, there is an attacker who can identify a large number of that algorithm's samples.

Theorem 14 (Special Case of Theorem 17 of (Dwork et al., 2015))**.** *For every $p \in \mathbb{N}$ and $\alpha > 0$, there exists a number $r = \tilde{\Omega}(\frac{\sqrt{p}}{\alpha})$ and a number $t = \Omega(\frac{1}{\alpha^2})$ such that the following holds: For every query-release algorithm $\mathcal{A}$ with total sample size $s \in [t, r+t]$ that is (α,α)-accurate for the class $\mathcal{S}_p$ of decision stumps on $\{\pm1\}^p$, there exists a distribution $\widetilde{\mathbf{D}}$ over $\{\pm1\}^p$ and an attacker $\mathcal{T}$ who takes as input the vector of answers $q \in [-1,1]^p$ and an example $y \in \{\pm1\}^p$ and outputs either* IN *or* OUT *such that*

$$\mathop{\mathbb{P}}_{\substack{z_1,\ldots,z_s,y\sim\widetilde{\mathbf{D}}\\ q\sim\mathcal{A}(z)}}\left[\mathcal{T}(q,y) = \mathrm{OUT}\right] \ge 1 - \tfrac{1}{(r+t)^2},\ \textit{and}$$

$$\mathop{\mathbb{P}}_{\substack{z_1,\ldots,z_s\sim\widetilde{\mathbf{D}}\\ q\sim\mathcal{A}(z)}}\left[|\{i\in[s] : \mathcal{T}(q,z_i) = \mathrm{IN}\}| \ge \tfrac{t}{2}\right] \ge 1 - \tfrac{1}{(r+t)^2}.$$

Proof of Theorem 13. Fix $p, \alpha > 0$ and let r and t be the values specified in Theorem 14. Suppose that $\mathcal{A}$ is a PAP algorithm that is (α,α)-accurate for the class $\mathcal{S}_p$ with n private samples and m public samples. We will show that either $n > r$ or $m \ge t/2$. First, note that the accuracy condition of $\mathcal{A}$ implies that we must have $n + m > t$ by the standard lower bound on the sample complexity of query release even without any privacy constraints. Thus,

to prove the theorem statement, it suffices to show that if $m \leq \frac{t}{2} - 1,\ t < n + m \leq r + t$, and $\mathcal{A}$ is (α, α)-accurate, then $\mathcal{A}$ cannot satisfy $(1, 1/4n)$-differential privacy w.r.t. its private samples. Indeed, this would imply that either $m \geq t/2$ or $n > t/2 + r > r$.

Let $\widetilde{\mathbf{D}}$ be the distribution promised by Theorem 14. Let $\vec{x} = (x_1, \ldots, x_n) \sim \widetilde{\mathbf{D}}^n$ be a set of n private samples and $\vec{w} = (w_1, \ldots, w_m) \sim \widetilde{\mathbf{D}}^m$ be a set of public samples, and let $\vec{z} = (x_1, \ldots, x_n, w_1, \ldots, w_m)$ be the combined set of samples. Let $q \sim \mathcal{A}(\vec{z})$. By Theorem 14,

$$\begin{aligned}
&\mathbb{P}\left[|\{i \in [n+m] : \mathcal{T}(q, z_i) = \text{IN}\}| \geq m + 1\right] \\
&\geq \mathbb{P}\left[|\{i \in [n+m] : \mathcal{T}(q, z_i) = \text{IN}\}| \geq t/2\right] \\
&\geq 1 - 1/(r+t)^2 \\
&\geq 1 - 1/(n+m)^2,
\end{aligned}$$

where the first inequality follows from the assumption that $m \leq \frac{t}{2} - 1$, and the last inequality follows from the assumption that $n + m \leq r + t$.

That is, with high probability, the attacker identifies at least $m + 1$ samples in the dataset. Let $\mathbf{1}\left(\mathcal{T}(q, z_i) = \text{IN}\right)$ be the indicator of the event $\mathcal{T}(q, z_i) = \text{IN}$. Therefore, we have

$$\begin{aligned}
&\sum_{i=1}^{n} \mathbb{P}\left[\mathcal{T}(q, x_i) = \text{IN}\right] + \sum_{i=1}^{m} \mathbb{P}\left[\mathcal{T}(q, w_i) = \text{IN}\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{n} \mathbf{1}\left(\mathcal{T}(q, x_i) = \text{IN}\right) + \sum_{i=1}^{m} \mathbf{1}\left(\mathcal{T}(q, w_i) = \text{IN}\right)\right] \\
&\geq (m+1) \cdot \mathbb{P}\left[|\{i \in [n+m] : \mathcal{T}(q, z_i) = \text{IN}\}| \geq m + 1\right] \\
&\geq (m+1)\left(1 - 1/\left(n+m\right)^2\right),
\end{aligned}$$

where the second step follows from Markov's inequality. Since

$$\sum_{i=1}^{m} \mathbb{P}\left[\mathcal{T}(q, w_i) = \text{IN}\right] \leq m$$

we can conclude that

$$\begin{aligned}
\sum_{i=1}^{n} \mathbb{P}\left[\mathcal{T}(q, x_i) = \text{IN}\right] &\geq (m+1)(1 - 1/(n+m)^2) - m \\
&= 1 - (m-1)/(n+m)^2 \\
&\geq 1 - 1/(n+m) \\
&\geq 1 - 1/n \geq 1/2
\end{aligned}$$

Therefore, there must exist a private sample i^* such that

$$\mathbb{P}_{\substack{x, w \sim \widetilde{\mathbf{D}} \\ q \sim \mathcal{A}(z)}} \left[\mathcal{T}(q, x_{i^*}) = \text{IN}\right] \geq 1/2n$$

Now, consider the dataset $\vec{z}_{\sim i^*}$ where we replace x_{i^*} in $\vec{z}$ with an independent sample $y \sim \widetilde{\mathbf{D}}$ but the rest of the samples in $\vec{z}_{\sim i^*}$ is the same as in $\vec{z}$. In this experiment x_{i^*} is now an independent sample from $\widetilde{\mathbf{D}}$, so Theorem 14 states that

$$\mathbb{P}_{\substack{\vec{x}, \vec{w}, y \sim \widetilde{\mathbf{D}} \\ q \sim \mathcal{A}(\vec{z}_{\sim i^*})}} \left[\mathcal{T}(q, x_{i^*}) = \text{IN}\right] \leq 1/(n+m)^2 \leq 1/n^2$$

However, note that the joint distribution $(\vec{z}, \vec{z}_{\sim i^*})$ is a distribution over pairs of datasets that differ on at most one private sample. Thus, we have shown that $\mathcal{A}$ cannot satisfy $(\varepsilon, 1/4n)$-differential privacy for its private samples unless

$$\frac{1}{2n} \leq e^{\varepsilon} \cdot \frac{1}{n^2} + \frac{1}{4n} \quad \Longrightarrow \quad \ln(n/4) \leq \varepsilon$$

The upshot is that $\mathcal{A}$ cannot be $(1, 1/4n)$-differentially private for $n \geq 11$. □

5. A Lower Bound on Public Sample Complexity

The goal of this section is to show a general lower bound on the public sample complexity of PAP query release. Our lower bound holds for classes with infinite Littlestone dimension. The Littlestone dimension is a combinatorial parameter of hypothesis classes that characterizes mistake and regret bounds in Online Learning (Littlestone, 1988; Ben-David et al., 2009). There are many examples of classes that have finite VC-dimension, but infinite Littlestone dimension. The simplest example is the class of threshold functions over $\mathbb{R}$ whose VC-dimension is 1, but has infinite Littlestone dimension. In (Alon et al., 2019b), it was shown that if a class has infinite Littlestone dimension, then it is not privately learnable.

Our lower bound is formally stated in the following theorem.

Theorem 15 (Lower bound on public sample complexity)**.** *Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be any query class that has infinite Littlestone dimension. Any PAP query-release algorithm for $\mathcal{H}$ must have public sample complexity $m = \Omega(1/\alpha)$, where α is the desired accuracy.*

We stress that the above lower bound on the public sample complexity holds regardless of the number of private samples, which can be arbitrarily large.

In the light of our upper bound in Section 3, our lower bound on the public sample complexity exhibits a tight dependence on the accuracy parameter α. That is, one cannot hope to attain public sample complexity that is $o(1/\alpha)$.

In the proof of the above theorem, we will refer to the following notion of private PAC learning with access to public data that was defined in (Alon et al., 2019a). For completeness, we restate this definition here.

Definition 16 ($(\alpha, \beta, \varepsilon, \delta)$ PAP Learner)**.** Let $\mathcal{H} \subset \{\pm 1\}^{\mathcal{X}}$ be a hypothesis class. A randomized algorithm $\mathcal{A}$ is

$(\alpha, \beta, \varepsilon, \delta)$ PAP learner for $\mathcal{H}$ with private sample size n and public sample size m if the following conditions hold:

1. For every distribution $\mathbf{D}$ over $\mathcal{Z} = \mathcal{X} \times \{\pm 1\}$, given data sets $\vec{x} \sim \mathbf{D}^n$ and $\vec{w} \sim \mathbf{D}^m$ as inputs to $\mathcal{A}$, with probability at least $1 - \beta$ (over the choice of $\vec{x}$, $\vec{w}$, and the random coins of $\mathcal{A}$), $\mathcal{A}$ outputs a hypothesis $\mathcal{A}(\vec{x}, \vec{w}) = \hat{h} \in \{\pm 1\}^{\mathcal{X}}$ satisfying

$$\mathrm{err}_{\mathbf{D}}\left(\hat{h}\right) \leq \inf_{h \in \mathcal{H}} \mathrm{err}_{\mathbf{D}}(h) + \alpha,$$

   where, for any hypothesis $h \in \{\pm 1\}^{\mathcal{X}}$, $\mathrm{err}_{\mathbf{D}}(h) \triangleq \mathbb{P}_{(x,y)\sim\mathbf{D}}[h(x) \neq y]$.

2. For all $\vec{w} \in \mathcal{Z}^m$, $\mathcal{A}(\cdot, \vec{w})$ is (ε, δ)-differentially private.

We say that $\mathcal{A}$ is proper PAP learner if $\mathcal{A}(\vec{x}, \vec{w}) \in \mathcal{H}$ with probability 1.

Proof. We prove the above theorem in two simple steps: we show that PAP query-release implies PAP learning, then invoke a lower bound on PAP learning classes with infinite Littlestone dimension. Both steps are formalized in the lemmas below.

Lemma 17 (General version of Theorem 5.5 in (Beimel et al., 2013))**.** *Let $\mathcal{H} \subseteq \{\pm 1\}^{\mathcal{X}}$ be any class of binary functions. If there exists an $(\alpha, \beta, \varepsilon, \delta)$ PAP query-release algorithm for $\mathcal{H}$ with private sample complexity n and public sample complexity m, then there exists an $(O(\alpha), O(\beta), O(\varepsilon), O(\delta))$ PAP learner for $\mathcal{H}$ with private sample complexity $n' = O(n \log(1/\alpha\beta)/\alpha^2\beta)$, and public sample complexity m.*

Lemma 18 (Theorem 4.1 in (Alon et al., 2019a))**.** *Let $\mathcal{H}$ be any class with an infinite Littlestone dimension (e.g., the class of thresholds over $\mathbb{R}$). Then, any PAP learner for $\mathcal{H}$ must have public sample complexity $m = \Omega(1/\alpha)$, where α is the excess error.*

Given these two lemmas, the proof is straightforward. To elaborate, note that Lemma 17 shows that for any class $\mathcal{H}$, a PAP query-release algorithm for $\mathcal{H}$ with public sample complexity m implies the existence of a PAP learner for $\mathcal{H}$ with the *same* public sample complexity (and essentially the same accuracy and privacy parameters). Hence, by Lemma 18, if $\mathcal{H}$ has infinite Littlestone dimension, then such public sample complexity must satisfy $m = \Omega(1/\alpha)$. This proves our theorem.

Although the proof of Lemma 17 is almost straightforward given Theorem 5.5 in (Beimel et al., 2013), we will elaborate on a couple of minor details. First, note that even though the reduction in (Beimel et al., 2013) involves pure differentially private algorithms, the same construction in their reduction would also work for the case of (ε, δ)-differential privacy with minor and obvious changes in the privacy analysis. Second, we note that the reduction in (Beimel et al., 2013) is for "proper sanitizers," which are query-release algorithms that are restricted to output a data set from the input domain rather than any data structure that maps $\mathcal{H}$ to $[-1, 1]$. As discussed in Remark 9, ignoring computational complexity, any PAP query-release algorithm satisfying Definition 8 can be transformed into a PAP query-release algorithm that outputs a data set from the input domain and has the same accuracy (up to a constant factor). Now, given these minor details and since any PAP algorithm can obviously be viewed as a differentially private algorithm operating on the private data set (by "hardwiring" the public data set into the algorithm), Lemma 17 follows by invoking the reduction in (Beimel et al., 2013). □

Acknowledgments

We thank the reviewers for their helpful comments. RB's research is supported by NSF Awards AF-1908281, SHF-1907715, Google Faculty Research Award, and OSU faculty start-up support. AC and JU were supported by NSF grants CCF-1718088, CCF-1750640, CNS-1816028, and CNS-1916020. AN is supported by an Ontario ERA, and an NSERC Discovery Grant RGPIN-2016-06333. ZSW is supported by a Google Faculty Research Award, a J.P. Morgan Faculty Award, and a Mozilla research grant. Part of this work was done while RB, AC, AN, JU, and ZSW were visiting the Simons Institute for the Theory of Computing.

References

Alon, N., Bassily, R., and Moran, S. Limits of private learning with access to public data. *arXiv:1910.11519 [cs.LG] (appeared at NeurIPS 2019)*, 2019a.

Alon, N., Livni, R., Malliaris, M., and Moran, S. Private pac learning implies finite littlestone dimension. *STOC 2019, pp. 852-860 (arXiv preprint arXiv:1806.00949)*, 2019b.

Assouad, P. Densité et dimension. In *Annales de l'Institut Fourier*, volume 33, pp. 233–282, 1983.

Bassily, R., Thakurta, A. G., and Thakkar, O. D. Model-agnostic private learning. In *Advances in Neural Information Processing Systems*, pp. 7102–7112, 2018.

Beimel, A., Nissim, K., and Stemmer, U. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 363–378. Springer, 2013.

Ben-David, S., Pál, D., and Shalev-Shwartz, S. Agnostic online learning. In *COLT*, volume 3, pp. 1, 2009.

Blum, A., Ligett, K., and Roth, A. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):1–25, 2013.

Bun, M., Nissim, K., Stemmer, U., and Vadhan, S. P. Differentially private release and learning of threshold functions. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pp. 634–649, 2015.

Bun, M., Ullman, J., and Vadhan, S. Fingerprinting codes and the price of approximate differential privacy. *SIAM Journal on Computing*, 47(5):1888–1938, 2018.

Chaudhuri, K. and Hsu, D. Sample complexity bounds for differentially private learning. In *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 155–186, 2011.

Dajani, A. N., Lauger, A. D., Singer, P. E., Kifer, D., Reiter, J. P., Machanavajjhala, A., Garfinkel, S. L., Dahl, S. A., Graham, M., Karwa, V., Kim, H., Lelerc, P., Schmutte, I. M., Sexton, W. N., Vilhuber, L., and Abowd, J. M. The modernization of statistical disclosure limitation at the U.S. census bureau, 2017. Presented at the September 2017 meeting of the Census Scientific Advisory Committee.

Dinur, I. and Nissim, K. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 202–210, 2003.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Dwork, C., Smith, A., Steinke, T., Ullman, J., and Vadhan, S. Robust traceability from trace amounts. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 650–669. IEEE, 2015.

Hamm, J., Cao, Y., and Belkin, M. Learning privately from multiparty data. In *International Conference on Machine Learning*, pp. 555–563, 2016.

Hardt, M. and Rothblum, G. N. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 61–70. IEEE, 2010.

Littlestone, N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

Muthukrishnan, S. and Nikolov, A. Optimal private halfspace counting via discrepancy. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 1285–1292, 2012.

Nandi, A. and Bassily, R. Privately answering classification queries in the agnostic pac model. *arXiv preprint arXiv:1907.13553. To appear in ALT 2020*, 2019.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *stat*, 1050, 2017.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

Sauer, N. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

Steinke, T. and Ullman, J. Between pure and approximate differential privacy. *arXiv preprint arXiv:1501.06095*, 2015.