A Learning Theory Approach to Non-Interactive Database Privacy *

Avrim Blum Katrina Ligett † Aaron Roth

Computer Science Department
Carnegie Mellon University
{avrim,katrina,alroth}@cs.cmu.edu

ABSTRACT

We demonstrate that, ignoring computational constraints, it is possible to release privacy-preserving databases that are useful for all queries over a discretized domain from any given concept class with polynomial VC-dimension. We show a new lower bound for releasing databases that are useful for halfspace queries over a continuous domain. Despite this, we give a privacy-preserving polynomial time algorithm that releases information useful for all halfspace queries, for a slightly relaxed definition of usefulness. Inspired by learning theory, we introduce a new notion of data privacy, which we call *distributional privacy*, and show that it is strictly stronger than the prevailing privacy notion, differential privacy.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms

Algorithms, Security, Theory

Keywords

non-interactive database privacy, learning theory

1. INTRODUCTION

As large-scale collection of personal information becomes easier, the problem of database privacy is increasingly important. In many cases, we might hope to learn useful information from sensitive data (for example, we might learn a correlation between smoking and lung cancer from a collection of medical records). However, for legal, financial, or moral reasons, administrators of sensitive datasets might not want to release their data. If those with the expertise to learn from large datasets are not the same as those who administer the datasets, what is to be done? In order to study this problem theoretically, it is important to quantify what exactly we mean by "privacy."

A series of recent papers [11, 9, 16] formalizes the notion of *differential privacy*. A database privatization mechanism (which may be either interactive or non-interactive) satisfies differential privacy if the addition or removal of a single database element does not change the probability of any outcome of the privatization mechanism by more than some small amount. The definition is intended to capture the notion that "distributional information is not private"—we may reveal that smoking correlates to lung cancer, but not that any individual has lung cancer. Individuals may submit their personal information to the database secure in the knowledge that (almost) nothing can be discovered from the database with their information that could not have been discovered without their information.

There has been a series of lower bound results [6, 9, 10] that suggests that non-interactive databases (or interactive databases that can be queried a linear number of times) cannot accurately answer all queries, or an adversary will be able to reconstruct all but a $1 - o(1)$ fraction of the original database exactly (obviously a very strong violation of privacy). As a result, most recent work has focused on the design of interactive mechanisms that answer only a sublinear number of queries. However, since these mechanisms may only answer a sublinear number of queries *in total* (not per user), after which point they must be destroyed, this limits their practicality in situations where the number of queries that might be asked is comparable to or larger than the number of entries in the database.

In this paper, motivated by learning theory, we propose the study of privacy-preserving mechanisms that are useful for all queries in a particular class (such as all conjunctive queries or all halfspace queries). In particular, we focus on predicate queries of the form, "what fraction of the database entries satisfy predicate φ?" and say that a sanitized output is useful for a class C if the answers to all queries in C have changed by at most some $\pm\epsilon$. In doing so, we circumvent existing lower bounds for non-interactive databases that only hold for particular types of queries, such as subset sum queries [6, 9, 10]. Building on the techniques of Kasiviswanathan et al. [14], we show that for discretized domains, for any concept class with polynomial VC-dimension, it is possible to release differential-privacy-preserving databases that are simultaneously useful for all queries in the concept class. The algorithm may not be computationally efficient in general, though we do have a computationally efficient method for range queries over a finite interval with bounded precision.

*Supported in part by the National Science Foundation under grant CCF-0514922.

†Supported in part by an AT&T Labs Graduate Research Fellowship and an NSF Graduate Research Fellowship.

STOC'08, May 17–20, 2008, Victoria, British Columbia, Canada.

Unfortunately, we show that for non-discretized domains, under the above definition of usefulness, it is impossible to publish a differentially private non-interactive database that is useful for even quite simple classes such as interval queries. We next show how, under a natural relaxation of the usefulness criterion, one can release information that can be used to usefully answer (arbitrarily many) halfspace queries while satisfying privacy. In particular, instead of requiring that useful mechanisms answer each query approximately correctly, we allow our algorithm to produce an answer that is approximately correct *for some nearby query*. This relaxation is motivated by the notion of large-margin separators in learning theory [1, 19, 18]; in particular, queries with no data points close to the separating hyperplane must be answered accurately, and the allowable error more generally is a function of the fraction of points close to the hyperplane.

We also introduce a new concept, *distributional privacy*, which makes explicit the notion that when run on a database drawn from a distribution, privacy-preserving mechanisms should reveal only information about the underlying distribution, and nothing else. Given a distribution $\mathcal{D}$ over database points, a database privatization mechanism satisfies distributional privacy if with high probability, drawing an entirely new database from $\mathcal{D}$ does not change the probability of any outcome of the privatization mechanism by more than some small amount. We show that distributional privacy is a strictly stronger guarantee than differential privacy by showing that any mechanism that satisfies distributional privacy also satisfies differential privacy, but that there are some functions that can be answered accurately while satisfying differential privacy, and yet reveal information about the particular database (although not about any particular database element) that is not "distributional."

We also show, in Appendix A, a small separation between interactive and non-interactive privacy-preserving mechanisms for predicate queries.

1.1 Related Work

Recent work on theoretical guarantees for data privacy was initiated by [6]. The notion of differential privacy, developed in a series of papers [4, 7, 8, 9, 10, 11, 12, 16, 15, 3, 17], separates issues of privacy from issues of outside information by defining privacy as indistinguishability of neighboring databases. This captures the notion that (nearly) anything that can be learned if your data is included in the database can also be learned without your data. This notion of privacy ensures that users have very little incentive to withhold their information from the database. The connection between data privacy and incentive-compatibility was recently formalized by McSherry and Talwar [15].

Prior work on interactive mechanisms has implied a number of impossibility results for non-interactive mechanisms. Dinur and Nissim [6] show that if a database answers all subset sum with less than $o(\sqrt{n})$ noise, this would allow an adversary to reconstruct a $1 - o(1)$ fraction of the database. Dwork et al. [10] show that even if the privacy-preserving mechanism is allowed to answer a small constant fraction of queries arbitrarily, if the remaining queries still are perturbed with $o(\sqrt{n})$ noise, an adversary can still reconstruct the database.

Dwork et al. [9] define a notion called *global sensitivity* and show that releasing a database perturbed with noise proportional to the global sensitivity of the query functions can preserve privacy, with the caveat that such mechanisms can only answer a sublinear number of queries in total, and then no further information about the database can ever be released. Blum et al. [4] consider a model of learning and show that concept classes that are learnable in the statistical query (SQ) model are also learnable from a polynomially sized dataset accessed through an interactive differential-privacy-preserving mechanism. We note that such mechanisms still may only answer a fixed number of queries in total.

Most similar to this paper is the recent work of Kasiviswanathan et al. [14]. Kasiviswanathan et al. study what can be learned privately when what is desired is that the hypothesis output by the learning algorithm satisfies differential privacy. They show that in a PAC learning model in which the learner has access to the private database, ignoring computational constraints, anything that is PAC learnable is also privately PAC learnable. We build upon the technique in their paper to show that in fact, it is possible to privately release a dataset that is simultaneously useful for any function in a concept class of polynomial VC-dimension. This resolves an open question posed by [14] about whether a VC-dimension analogue of Occam's razor holds in their private learning model. Kasiviswanathan et al. also study several restrictions on learning algorithms, show separation between these learning models, and give efficient algorithms for learning particular concept classes.

In this work, we study non-interactive database release mechanisms, which may be used to answer an unlimited number of queries. We circumvent the existing lower bounds by only guaranteeing usefulness for queries in restricted classes. Blum et al. [4] consider running machine learning algorithms on datasets that are accessed through interactive privacy-preserving mechanisms. In contrast, we show how to release data sets from which one can usefully learn the values of all functions in restricted concept classes.

1.2 Motivation from Learning Theory

From a machine learning perspective, one of the main *reasons* one would want to perform statistical analysis of a database in the first place is to gain information about the population from which that database was drawn. In particular, a fundamental result in learning theory is that if one views a database as a collection of random draws from some distribution $\mathcal{D}$, and one is interested in a particular class C of boolean predicates over examples, then a database D of size $\tilde{O}(\text{VCDIM}(C)/\epsilon^2)$ is sufficient so that with high probability, for *every* query $q \in C$, the proportion of examples in D satisfying q is within $\pm\epsilon$ of the true probability mass under $\mathcal{D}$ [1, 19].[1] Our main result can be viewed as asking how much larger does a database D have to be in order to do this in a privacy-preserving manner: that is, to allow one to (probabilistically) construct an output $\hat{D}$ that accurately approximates $\mathcal{D}$ with respect to all queries in C, and yet that reveals no extra information about database D.[2] In fact, our notion of distributional privacy (Section 7) is motivated by this view. Note that since *interactive* privacy mechanisms can handle arbitrary queries of this form so long as only $o(n)$ are requested, our objective is interesting only for classes C that contain $\Omega(n)$, or even exponentially in n many queries. We will indeed achieve this (Theorem 3.1), since $|C| \geq 2^{\text{VCDIM(C)}}$.

[1] Usually, this kind of uniform convergence is stated as empirical error approximating true error. In our setting, we have no notion of an "intrinsic label" of database elements. Rather, we imagine that different users may be interested in learning different things. For example, one user might want to learn a rule to predict feature x_d from features $x_1, \ldots, x_{d-1}$; another might want to use the first half of the features to predict a certain boolean function over the second half.

[2] Formally, we only care about $\hat{D}$ approximating D with respect to C, and want this to be true no matter how D was constructed. However, if D *was* a random sample from a distribution $\mathcal{D}$, then D will approximate $\mathcal{D}$ and therefore $\hat{D}$ will as well.

1.3 Organization

We present essential definitions in Section 2. In Section 3, we show that, ignoring computational constraints, one can release sanitized databases over discretized domains that are useful for *any* concept class with polynomial VC-dimension. We then, in Section 4, give an efficient algorithm for privately releasing a database useful for the class of interval queries. We next turn to the study of halfspace queries over $\mathbb{R}^d$ and show in Section 5 that, without relaxing the definition of usefulness, one cannot release a database that is privacy-preserving and useful for halfspace queries over a continuous domain. Relaxing our definition of usefulness, in Section 6, we give an algorithm that in polynomial time, creates a sanitized database that usefully and privately answers all halfspace queries. We present an alternative definition of privacy and discuss its relationship to differential privacy in Section 7. In Appendix A, we give a separation of interactive and non-interactive databases for predicate queries.

2. DEFINITIONS

For a database D, let A be a database access mechanism. For an interactive mechanism, we will say that $A(D, Q)$ induces a distribution over outputs for each query Q. For a non-interactive mechanism, we will say that $A(D)$ induces a distribution over outputs.

We say that an interactive database access mechanism A satisfies *α-differential privacy* if for all neighboring databases D_1 and D_2 (differing in only a single element), for all queries Q, and for all outputs x,

$$\Pr[A(D_1, Q) = x] \leq e^{\alpha} \Pr[A(D_2, Q) = x].$$

We say that a non-interactive database sanitization mechanism A satisfies *α-differential privacy* if for all neighboring databases D_1 and D_2, and for all sanitized outputs $\widehat{D}$,

$$\Pr[A(D_1) = \widehat{D}] \leq e^{\alpha} \Pr[A(D_2) = \widehat{D}].$$

In Section 7, we propose an alternate definition of privacy, distributional privacy, and show that it is strictly stronger than differential privacy. For simplicity, however, in the main body of the paper, we use the standard definition, differential privacy. All of these proofs can be adapted to the distributional privacy notion.

DEFINITION 2.1. *The* global sensitivity *of a query f is its maximum difference when evaluated on two neighboring databases:*

$$GS_f = \max_{D_1, D_2 : d(D_1, D_2) = 1} |f(D_1) - f(D_2)|.$$

In this paper, we consider the private release of information useful for classes of *predicate queries*.

DEFINITION 2.2. *A* predicate query *Q_φ for any predicate φ is defined to be*

$$Q_\varphi(D) = \frac{|\{x \in D : \varphi(x)\}|}{|D|}.$$

OBSERVATION 2.3. *For any predicate φ, $GS_{Q_\varphi} \leq 1/n$.*

Previous work shows how one can construct database access mechanisms that can answer any low-sensitivity query while preserving differential privacy:

DEFINITION 2.4. *Let the interactive mechanism PRIVATE$_\alpha(D, Q)$ respond to queries Q by returning $Q(D) + Z$ where Z is a random variable drawn from the Laplace distribution: $Z \sim Lap(GS_Q/\alpha)$.*

THEOREM 2.5 (DWORK ET AL. [9]). *PRIVATE$_\alpha(D, Q)$ preserves α-differential privacy.*

However, lower bounds of Dinur and Nissim [6] and Dwork et al. [9] imply that such mechanisms can only answer a sublinear number of queries on any database. Note that these mechanisms can only answer a sublinear number of queries *in total*, not per user.

We propose to construct database access mechanisms whose results can be released to the public, and so can necessarily be used to answer an arbitrarily large number of queries. We seek to do this while simultaneously preserving privacy. However, in order to circumvent the lower bounds of Dinur and Nissim [6] and Dwork et al. [9], we cannot hope to be able to usefully answer arbitrary queries. We instead seek to answer restricted classes of queries while preserving "usefulness," which we define as follows:

DEFINITION 2.6 (USEFULNESS DEFINITION 1). *A database mechanism A is (ϵ, δ)-*useful *for queries in class C if with probability $1 - \delta$, for every $Q \in C$ and every database D, for $\widehat{D} = A(D)$, $|Q(\widehat{D}) - Q(D)| \leq \epsilon$.*

3. GENERAL RELEASE MECHANISM

In this section we show that (ignoring computational considerations) it is possible to release a non-interactive database useful for any concept class with polynomial VC-dimension, while preserving α-differential privacy, given an initial database of polynomial size. Our use of the exponential mechanism is inspired by its use by Kasiviswanathan et al. [14].

THEOREM 3.1. *For any class of functions C, and any database $D \subset \{0,1\}^d$ such that*

$$|D| \geq O\left(\frac{dVCDIM(C)\log(1/\epsilon)}{\epsilon^3 \alpha} + \frac{\log(1/\delta)}{\alpha\epsilon}\right)$$

we can output an (ϵ, δ)-useful database $\widehat{D}$ that preserves α-differential privacy. Note that the algorithm is not necessarily efficient.

We give an (inefficient) algorithm that outputs a sanitized database $\widehat{D}$ of size $\widetilde{O}(\text{VCDIM}(C)/\epsilon^2)$. We note that the size of the output database is independent of the size of our initial database. This is sufficient for (ϵ, δ)-usefulness because the set of all databases of this size forms an ϵ-cover with respect to C of the set of all possible databases.

LEMMA 3.2 ([1, 19]). *Given any database D there exists a database $\widehat{D}$ of size $m = O(VCDIM(C)\log(1/\epsilon)/\epsilon^2)$ such that $\max_{h \in C} |h(D) - h(\widehat{D})| < \epsilon/2$.*

PROOF. This follows from standard sample complexity bounds. □

McSherry and Talwar [15] define the exponential mechanism as follows:

DEFINITION 3.3. *For any function $q : ((\{0,1\}^d)^n \times (\{0,1\}^d)^m) \to \mathbb{R}$ and input database D, the exponential mechanism outputs each database $\widehat{D}$ with probability proportional to $e^{q(D,\widehat{D})\alpha n/2}$.*

THEOREM 3.4 ([15]). *The exponential mechanism preserves $(\alpha n GS_q)$-differential privacy.*

PROOF OF THEOREM 3.1. We use the exponential mechanism and define our quality function q to be:

$$q(D, \widehat{D}) = -\max_{h \in C} \left| h(D) - h(\widehat{D}) \right|$$

Note that $GS_q = 1/n$. In order to show that this mechanism satisfies (ϵ, δ)-usefulness, we must show that it outputs some database $\widehat{D}$ with $q(D, \widehat{D}) \geq -\epsilon$ except with probability δ.

Any output database $\widehat{D}$ with $q(D, \widehat{D}) \leq -\epsilon$ will be output with probability at most proportional to $e^{-\alpha\epsilon n/2}$. There are at most 2^{dm} possible output databases, and so by a union bound, the probability that we output any database $\widehat{D}$ with $q(D, \widehat{D}) \leq -\epsilon$ is at most proportional to $2^{dm}e^{-\alpha\epsilon n/2}$.

Conversely, we know by Lemma 3.2 that there exists some $\widehat{D} \in (\{0,1\}^d)^m)$ such that $q(D, \widehat{D}) \geq -\epsilon/2$, and therefore that such a database is output with probability at least proportional to $e^{-\alpha\epsilon n/4}$.

Let A be the event that the exponential mechanism outputs some database $\widehat{D}$ such that $q(D, \widehat{D}) \geq -\epsilon/2$. Let B be the event that the exponential mechanism outputs some database $\widehat{D}$ such that $q(D, \widehat{D}) \leq -\epsilon$. We must show that $\Pr[A]/\Pr[B] \geq (1-\delta)/\delta$.

$$\begin{aligned}\frac{\Pr[A]}{\Pr[B]} &\geq \frac{e^{-\alpha\epsilon n/4}}{2^{dm}e^{-\alpha\epsilon n/2}} \\ &= \frac{e^{\alpha\epsilon n/4}}{2^{dm}}\end{aligned}$$

Setting this quantity to be at least $1/\delta > (1-\delta)/\delta$, we see that it is sufficient to take

$$\begin{aligned}n &\geq \frac{4}{\epsilon\alpha}\left(dm + \ln\frac{1}{\delta}\right) \\ &\geq O\left(\frac{d\text{VCDIM}(C)\log(1/\epsilon)}{\epsilon^3\alpha} + \frac{\log(1/\delta)}{\alpha\epsilon}\right).\end{aligned}$$

This result extends in a straightforward manner to the case of any discretized database domain, not just a boolean space. □

Theorem 3.1 shows that a database of size $\widetilde{O}(\frac{d\text{VCDIM}(C)}{\epsilon^3\alpha})$ is sufficient in order to output a set of points that is ϵ-useful for a concept class C, while simultaneously preserving α-differential privacy. If we were to view our database as having been drawn from some distribution $\mathcal{D}$, this is only an extra $\widetilde{O}(\frac{d}{\epsilon\alpha})$ factor larger than what would be required to achieve ϵ-usefulness with respect to $\mathcal{D}$, even without any privacy guarantee! In fact, as we will show in Theorem A.6, it is impossible to release a database that is $o(1/\sqrt{n})$-useful for the class of parity functions while preserving privacy, and so a dependence on ϵ of at least $\Omega(1/\epsilon^2)$ is necessary.

The results in this section only apply for discretized database domains, and may not be computationally efficient. We explore these two issues further in the remaining sections of the paper.

4. INTERVAL QUERIES

In this section we give an *efficient* algorithm for privately releasing a database useful for the class of interval queries over a discretized domain, given a database of size only polynomial in our privacy and usefulness parameters. We note that our algorithm is easily extended to the class of axis-aligned rectangles in d dimensional space for d a constant; we present the case of $d = 1$ for clarity.

Consider a database D of n points in $[0,1]$ in which the entries are discretized to b bits of precision; our bounds will be polynomial in b (in Corollary 5.2 we show some discretization is necessary). Given $a_1 \leq a_2$, both in $[0,1]$, let I_{a_1,a_2} be the indicator function corresponding to the interval $[a_1, a_2]$. That is:

$$I_{a_1,a_2}(x) = \begin{cases} 1, & a_1 \leq x \leq a_2; \\ 0, & \text{otherwise.}\end{cases}$$

DEFINITION 4.1. *An interval query $Q_{[a_1,a_2]}$ is defined to be*

$$Q_{[a_1,a_2]}(D) = \sum_{x\in D}\frac{I_{a_1,a_2}(x)}{|D|}.$$

Note that $GS_{Q_{[a_1,a_2]}} = 1/n$, and we may answer interval queries while preserving α-differential privacy by adding noise proportional to $\text{Lap}(1/(\alpha n))$.

Given a database D, we will use α'-differential privacy preserving interval queries to perform a binary search on the interval $[0,1]$ and partition it into sub-intervals containing probability mass in the range $[\epsilon_1/2 - \epsilon_2, \epsilon_1/2 + \epsilon_2]$. Because of the discretization, the depth of this search is at most b. We will then output a dataset that has $(\epsilon_1/2)\cdot n$ points in each of these intervals. Because we have constructed this dataset using only a small number of privacy preserving queries, its release will also preserve privacy, and it will be (ϵ, δ)-useful for the class of interval queries with an appropriate choice of parameters. Finally, this simple mechanism is clearly computationally efficient.

THEOREM 4.2. *With $\alpha' = (\epsilon\alpha)/4b$, $\epsilon_1 = (\epsilon/2)$ and $\epsilon_2 = (\epsilon^2/8)$, the above mechanism preserves α-differential privacy while being (ϵ, δ)-useful for the class of interval queries given a database of size:*

$$|D| \geq O\left(\frac{b(\log b + \log(1/\epsilon\delta))}{\alpha\epsilon^3}\right)$$

PROOF. We first bound the number of privacy preserving queries our algorithm makes. It finally produces $2/\epsilon_1$ intervals. Since D is defined over a discretized space, we can identify each interval with the at most b queries on its path through the binary search procedure, and so we will make a total of at most $2b/\epsilon_1 = 4b/(\epsilon)$ α'-differential privacy preserving queries. Since the differential-privacy parameter composes, with $\alpha' = (\epsilon\alpha)/4b$, our algorithm indeed preserves α differential privacy.

Suppose that the binary search procedure indeed returns intervals each containing probability mass in the range $[\epsilon_1/2-\epsilon_2, \epsilon_1/2+\epsilon_2]$. Any query will intersect at most two of these intervals only partially. In the worst case, this introduces $\epsilon_1 = \epsilon/2$ error to the query ($\epsilon_1/2$ error from each interval that partially overlaps with the query). Since each query can only overlap at most $2/\epsilon_1$ intervals, and each interval contains a probability mass that deviates from the true probability mass in D by at most ϵ_2, this introduces an additional $2\epsilon_2/\epsilon_1 = \epsilon/2$ error, for a total error rate $\leq \epsilon$. Therefore, to complete the proof, we only need to bound the size of D necessary such that the probability that any of the $2b/\epsilon_1$ privacy preserving queries returns an answer that deviates from the true answer (in D) by more than ϵ_2 is less than δ. Let us call this event FAILURE. Since the event that any single query has error rate $\geq \epsilon_2$ is $\Pr[\text{Lap}(1/(\alpha' n)) \geq \epsilon_2] \leq e^{\alpha'\epsilon_2 n}$, this follows from a simple union bound:

$$\Pr[\text{FAILURE}] \leq \frac{2b}{\epsilon_1}e^{(-\epsilon\alpha/4b)\epsilon_2 n} \leq \delta.$$

Solving, we find

$$n \geq \frac{4b(\log 2b) + \log(1/\epsilon_1\delta)}{\alpha\epsilon\epsilon_2} = O\left(\frac{b(\log b + \log(1/\epsilon\delta))}{\alpha\epsilon^3}\right)$$

is sufficient. □

We note that although the class of intervals (and more generally, low dimensional axis-aligned rectangles) is a simple class of functions, it nevertheless contains exponentially (in b) many queries,

and so it is not feasible to simply ask all possible interval queries using an interactive mechanism.

While it is true that intervals (and low dimensional axis-aligned rectangles) have constant VC-dimension and polynomial ϵ-cover size, we can trivially extend the above results to the class of unions of t intervals by dividing ϵ by t and answering each interval separately. This class has VC-dimension $O(t)$ and exponentially large ϵ-cover size.

5. LOWER BOUNDS

Could we possibly modify the results of Sections 4 and 3 to hold for non-discretized databases? Suppose we could usefully answer an arbitrary number of queries in some simple concept class C representing interval queries on the real line (for example, "How many points are contained within the following interval?") while still preserving privacy. Then, for any database containing single-dimensional real valued points, we would be able to answer median queries with values that fall between the $1/2 - \delta, 1/2 + \delta$ percentile of database points by performing a binary search on D using A (where $\delta = \delta(\epsilon)$ is some small constant depending on the usefulness parameter ϵ). However, answering such queries is impossible while guaranteeing differential privacy. Unfortunately, this would seem to rule out usefully answering queries in simple concept classes such as halfspaces and axis-aligned rectangles, that are generalizations of intervals.

THEOREM 5.1. *No mechanism A can answer median queries M with outputs that fall between the $1/2 - k, 1/2 + k$ percentile with positive probability on any real valued database D, while still preserving α-differential privacy, for $k < 1/2$ and any α.*

PROOF. Consider real valued databases containing elements in the interval $[0, 1]$. Let $D_0 = (0, \ldots, 0)$ be the database containing n points with value 0. Then we must have $\Pr[A(D_0, M) = 0] > 0$. Since $[0, 1]$ is a continuous interval, there must be some value $v \in [0, 1]$ such that $\Pr[A(D_0, M) = v] = 0$. Let $D_n = (v, \ldots, v)$ be the database containing n points with value v. We must have $\Pr[A(D_n, M) = v] > 0$. For $1 < i < n$, let $D_i = (\underbrace{0, \ldots, 0}_{n-i}, \underbrace{v, \ldots, v}_{i})$. Then we must have for some i, $\Pr[A(D_i, M) = v] = 0$ but $\Pr[A(D_{i+1}, M) = v] > 0$. But since D_i and D_{i+1} differ only in a single element, this violates differential privacy. □

COROLLARY 5.2. *No mechanism can be (ϵ, δ)-useful for the class of interval queries, nor for any class C that generalizes interval queries to higher dimensions (for example, halfspaces, axis-aligned rectangles, or spheres), while preserving α-differential privacy, for any $\epsilon = o(n)$ and any α.*

PROOF. Consider any real valued database containing elements in the interval $[0, 1]$. If A is (ϵ, δ)-useful for interval queries and preserves differential privacy, then we can construct a mechanism A' that can answer median queries with outputs that fall between the $1/2 - k, 1/2 + k$ percentile with positive probability while preserving differential privacy. By Theorem 5.1, this is impossible. A' simply computes $\widehat{D} = A(D)$, and performs binary search on $\widehat{D}$ to find some interval $[0, a]$ that contains $n/2 \pm \epsilon$ points. Privacy is preserved since we only access D through A, which by assumption preserves differential privacy. With positive probability, all interval queries on $\widehat{D}$ are correct to within $\pm\epsilon$, and so the binary search can proceed. Since $\epsilon = o(n)$, the result follows. □

We may get around the impossibility result of Corollary 5.2 by relaxing our definitions. One approach is to discretize the database domain, as we do in Sections 3 and 4. Another approach, which we take in Section 6, is to relax our definition of usefulness:

DEFINITION 5.3 (USEFULNESS DEFINITION 2). *A database mechanism A is $(\epsilon, \delta, \gamma)$-*useful *for queries in class C according to some metric d if with probability $1 - \delta$, for every $Q \in C$ and every database D, $|Q(A(D)) - Q'(D)| \leq \epsilon$ for some $Q' \in C$ such that $d(Q, Q') \leq \gamma$.*

6. ANSWERING HALFSPACE QUERIES

Here, we consider databases that contain n elements in $\mathbb{R}^d$. In this section, we show how to efficiently release information useful (according to definition 5.3) for the class of halfspace queries for any constant $\gamma > 0$. Throughout this section, we assume without loss of generality that the database points are scaled into the unit sphere. Additionally, when we project the points into a lower-dimensional space, we rescale them to the unit sphere. A halfspace query specifies a hyperplane in $\mathbb{R}^d$ and asks how many points fall above it:

DEFINITION 6.1. *Given a database $D \subset \mathbb{R}^d$ and unit length $y \in \mathbb{R}^d$, a halfspace query H_y is*

$$H_y(D) = \frac{|\{x \in D : \sum_{i=1}^{d} x_i \cdot y_i \geq 0\}|}{|D|}.$$

The assumption that halfspaces pass through the origin is without loss of generality since we can view translated halfspaces as passing through the origin in a space of dimension $d + 1$.

In this section, we give an algorithm that is $(\epsilon, \delta, \gamma)$-useful for the class of halfspace queries. For a point x we will write $\hat{x}$ for the normalization $x/||x||_2$. We define the distance between a point x and a halfspace H_y by $d(x, H_y) = |\hat{x} \cdot y|$. For convenience, we define the distance between two halfspaces H_{y_1} and H_{y_2} to be the sin of the angle between y_1 and y_2; by a slight abuse of notation, we will denote this by $d(y_1, y_2)$. In particular, for a point x and two halfpaces H_{y_1} and H_{y_2}, $d(x, H_{y_1}) \leq d(x, H_{y_2}) + d(y_1, y_2)$. If $d(y_1, y_2) \leq \gamma$ we say that H_{y_1} and H_{y_2} are γ-*close*. Given a halfspace H_{y_1}, our goal is to output a value v such that $|v - H_{y_2}(D)| < \epsilon$ for some H_{y_2} that is γ-close to H_{y_1}. Equivalently, we may arbitrarily count or not count any point $x \in D$ such that $d(x, H_{y_1}) \leq \gamma$. We note that γ is similar to the notion of margin in machine learning, and that even if H_{y_1} and H_{y_2} are γ-close, this does not imply that $H_{y_1}(D)$ and $H_{y_2}(D)$ are close, unless most of the data points are outside a γ margin of H_{y_1} and H_{y_2}.

We circumvent the halfspace-lower bound of Corollary 5.2 by considering a class of *discretized* halfspaces:

DEFINITION 6.2. *A halfspace query H_y is* b-discretized *if for each $i \in [d]$, y_i can be specified with b-bits. Let C_b be the set of all b-discretized halfspaces in $\mathbb{R}^d$.*

We first summarize the algorithm, with the parameters to be specified later. Our use of random projections is similar to that in the work of Indyk and Motwani [13] on approximate nearest neighbor queries.

Our algorithm performs m random projections $P_1, \ldots, P_m$ of the data onto $\mathbb{R}^k$. A random projection of n points from $\mathbb{R}^d$ to $\mathbb{R}^k$ is defined as follows:

DEFINITION 6.3. *A random projection P_i from $\mathbb{R}^d$ to $\mathbb{R}^k$ is defined by a $d \times k$ random matrix M_i with entries chosen independently and uniformly at random from $\{-1, 1\}$. We write the projection of point $x \in \mathbb{R}^d$ as $P_i(x) = (1/\sqrt{k})x \cdot M_i$. We write the projection of a database $D \in (\mathbb{R}^d)^n$ as $P_i(D) = \{P_i(x) : x \in D\}$.*

For each projected database $P_i(D)$ we ask $O(1/\gamma^{k-1})$ privacy-preserving canonical halfspace queries. To answer a halfspace query H_y, for each projection P_i, we consider $H_{P_i(y)}$ and associate with it the answer of the closest canonical halfspace in that projection. Finally, we return the median value of these queries over all m projections.

THEOREM 6.4 (JOHNSON-LINDENSTRAUSS [5, 2]). *Consider a random projection P of a point x and a halfspace H_y onto a random k-dimensional subspace. Then*

$$\Pr[|d(x, H_y) - d(P(x), H_{P(y)})| \geq \gamma/4] \leq 2e^{-((\gamma/16)^2-(\gamma/16)^3)k/4}.$$

That is, projecting x and H_y significantly changes the distance between the point and the halfspace with only a small probability.

We choose k such that the probability that projecting a point and a halfspace changes their distance by more than $\gamma/4$ is at most $\epsilon_1/4$. Solving, this yields

$$k \geq \frac{4\ln(8/\epsilon_1)}{(\gamma/16)^2 - (\gamma/16)^3}.$$

Given a halfspace H_y and a point x, we say that a projection P makes a mistake relative to x and H_y if $d(x, H_y) \geq \gamma/4$, but $\text{sign}(x \cdot y) \neq \text{sign}(P(x) \cdot P(y))$. We have chosen k such that the expected fraction of mistakes relative to any halfspace H_y in any projection P is at most $\epsilon_1/4$. By Markov's inequality, therefore, the probability that a projection makes more than $\epsilon_1 n$ mistakes relative to a particular halfspace is at most $1/4$.

The probability δ_1 that more than $m/2$ projections make more than $\epsilon_1 n$ mistakes relative to *any* discretized halfspace is at most $2^{bd}e^{-m/12}$ by a Chernoff bound and a union bound. Solving for m, this gives

$$m \geq 12\left(\ln\left(\frac{1}{\delta_1}\right) + \ln(2)bd\right).$$

For each projection P_i, we select a $(3/4)\gamma$-net of halfspaces N_i, such that for every vector $y_1 \in \mathbb{R}^k$ corresponding to halfspace H_{y_1}, there exists a halfspace $H_{y_2} \in N_i$ such that $d(y_1, y_2) \leq (3/4)\gamma$. We note that $|N_i| = O(1/\gamma^{k-1})$. For each projection P_i and for each $H_y \in N_i$, we record the value of

$$v_y^i = \text{PRIVATE}_{\alpha/(m|N_i|)}(P_i(D), H_y).$$

We note that since we make $m|N_i|$ queries in total, these queries preserve α-differential privacy.

Taking a union bound over the halfspaces in each N_i, we find that the probability δ_2 that any of the v_y^i differ from $H_y(P_i(D))$ by more than ϵ_2 is at most $m \cdot O(1/\gamma)^{k-1}e^{-(\epsilon_2 n\alpha)/(mO(1/\gamma^{k-1}))}$. Solving for n, we find that

$$n \geq \frac{\log(1/\delta_2) + \log m + (k-1)\log(1/\gamma) + mO(1/\gamma)^{k-1}}{\epsilon_2\alpha}$$

$$= O\left(\frac{1}{\epsilon_2\alpha}\left(\log(1/\delta_2) + \log\log 1/\delta_1 + \log bd + \log(1/\epsilon_1)\right.\right.$$

$$\left.\left. + (\log 1/\delta_1 + bd)(1/\epsilon_1)^{(4\log(1/\gamma))/((\gamma/16)^2-(\gamma/16)^3)}\right)\right).$$

To respond to a query H_y, for each projection P_i we first compute

$$H_{y_i'} = \operatorname*{argmin}_{H_{y_i'} \in N_i} d(P(y), y_i').$$

We recall that by construction, $d(P(y), y_i') \leq (3/4)\gamma$. We then return the median value from the set $\{v_{y_i'}^i : i \in [m]\}$.

THEOREM 6.5. *The above algorithm is $(\epsilon, \gamma, \delta)$-useful while maintaining α-differential privacy for a database of size poly$(\log(1/\delta), 1/\epsilon, 1/\alpha, b, d)$ and running in time poly$(\log(1/\delta), 1/\epsilon, 1/\alpha, b, d)$, for constant γ.*

PROOF. Above, we set the value of m such that for any halfspace query H_y, with probability at most δ_1, no more than an ϵ_1 fraction of the points have the property that they are outside of a γ margin of H_y but yet their projections are within a $(3/4)\gamma$ margin of $H_{P_i(y)}$, where i is the index of the median projection. Therefore, answering a query $H_{y'}$, where y' is $(3/4)\gamma$-close to $P_i(y)$, only introduces ϵ_1 error. Moreover, we have chosen n such that except with probability δ_2, the privacy-preserving queries introduce no more than an additional ϵ_2 error. The theorem follows by setting $\epsilon_1 = \epsilon_2 = \epsilon/2$ and $\delta_1 = \delta_2 = \delta/2$, and setting n, m, and k as above. □

7. DISTRIBUTIONAL PRIVACY

We say that an interactive database mechanism A satisfies (α, β)-*distributional privacy* if for any distribution over database elements $\mathcal{D}$, with probability $1 - \beta$, two databases D_1 and D_2 consisting of n elements drawn *without replacement* from $\mathcal{D}$, for any query Q and output x satisfies

$$\Pr[A(D_1, Q) = x] \leq e^\alpha \Pr[A(D_2, Q) = x].$$

Similarly, for non-interactive mechanisms, a mechanism A satisfies (α, β)-*distributional privacy* if for any distribution over database elements D, with probability $1 - \beta$, two databases D_1 and D_2 consisting of n elements drawn *without replacement* from $\mathcal{D}$, and for all sanitized outputs $\widehat{D}$,

$$\Pr[A(D_1) = \widehat{D}] \leq e^\alpha \Pr[A(D_2) = \widehat{D}].$$

For example, suppose that a collection of hospitals in a region each treats a random sample of patients with disease X. Distributional privacy means that a hospital can release its data anonymously, without necessarily revealing which hospital the data came from. Actually, our main motivation is that this definition is particularly natural from the perspective of learning theory: given a sample of points drawn from some distribution $\mathcal{D}$, one would like to reveal no more information about the sample than is inherent in $\mathcal{D}$ itself.

We will typically think of β as being exponentially small, whereas α must be $\Omega(1/n)$ for A to be useful.

7.1 Relationship Between Definitions

It is not a priori clear whether either differential privacy or distributional privacy is a stronger notion than the other, or if the two are equivalent, or distinct. On the one hand, differential privacy only provides a guarantee when D_1 and D_2 differ in a single element,[3] whereas distributional privacy can provide a guarantee for two databases D_1 and D_2 that differ in all of their elements. On the other hand, distributional privacy makes the strong assumption that the elements in D_1 and D_2 are drawn from some distribution $\mathcal{D}$, and allows for privacy violations with some exponentially small probability β (necessarily: with some small probability, two databases drawn from the same distribution might nevertheless be completely different). However, as we show, distributional privacy is a strictly stronger guarantee than differential privacy. For clarity, we prove this for interactive mechanisms only, but the results hold for non-interactive mechanisms as well, and the proofs require little modification.

[3]We get $t\alpha$-differential privacy for D_1 and D_2 that differ in t elements.

THEOREM 7.1. *If A satisfies (α, β)-distributional privacy for any $\beta = o(1/n^2)$, then A satisfies α-differential privacy.*

PROOF. Consider any database D_1 drawn from domain R, and any neighboring database D_2 that differs from D_1 in only a single element $x \in R$. Let $\mathcal{D}$ be the uniform distribution over the set of $n+1$ elements $D_1 \cup \{x\}$. If we draw two databases D_1', D_2' from $\mathcal{D}$, then with probability $2/n^2$ we have $\{D_1', D_2'\} = \{D_1, D_2\}$, and so if $\beta = o(1/n^2)$, we have with certainty that for all outputs $\widehat{D}$ and for all queries Q,

$$\Pr[A(D_1, Q) = \widehat{D}] \leq e^{\alpha} \Pr[A(D_2, Q) = \widehat{D}].$$

Therefore, A satisfies α-differential privacy. □

DEFINITION 7.2. *Define the* mirrored mod m function *as follows:*

$$F_m(x) = \begin{cases} x \mod m, & \text{if } x \mod 2m < m; \\ -x - 1 \mod m, & \text{otherwise.} \end{cases}$$

For a database $D \subset \{0,1\}^n$, define the query

$$Q_m(D) = \frac{F_m(\sum_{i=0}^{n-1} D[i])}{|D|}.$$

Note that the global sensitivity of any query Q_m satisfies $GS_{Q_m} \leq 1/n$. Therefore, the mechanism A that answers queries Q_n by $A(D, Q_m) = Q_m(D) + Z$ where Z is drawn from $\text{Lap}(1/(\alpha n)) = \text{Lap}(GS_{Q_m}/\alpha)$ satisfies α-differential privacy, which follows from the results of Dwork et al. [9].

THEOREM 7.3. *There exist mechanisms A with α-differential privacy, but without (α, β)-distributional privacy for any $\alpha < 1$, $\beta = o(1)$ (that is, for any meaningful values of α, β).*

PROOF. Consider databases with elements drawn from $\mathcal{D} = \{0,1\}^n$ and the query $Q_{2/\alpha}$. As observed above, a mechanism A such that $A(D, Q_i) = Q_i(D) + Z$ for $Z \sim \text{Lap}(1/(\alpha n))$ has α-differential privacy for any i. Note however that with constant probability, two databases D_1, D_2 drawn from $\mathcal{D}$ have $|Q_{2/\alpha}(D_1) - Q_{2/\alpha}(D_2)| \geq 1/(\alpha n)$. Therefore, for any output x, we have that with constant probability,

$$\begin{aligned} \frac{\Pr[A(D_1, Q_{2/\alpha}) = x]}{\Pr[A(D_2, Q_{2/\alpha}) = x]} &= e^{-\alpha |Q_{2/\alpha}(D_1) - Q_{2/\alpha}(D_2)|} \\ &= e^{-\alpha n (\frac{1}{\alpha n})} \\ &= \frac{1}{e}. \quad \square \end{aligned}$$

Although there are simpler functions for which preserving distributional privacy requires more added noise than preserving differential privacy, the mirrored-mod function above is an example of a function for which it is possible to preserve differential privacy usefully, but yet impossible to reveal any useful information while preserving distributional privacy.

We note that in order for distributional privacy to imply differential privacy, it is important that in the definition of distributional privacy, database elements are drawn from some distribution $\mathcal{D}$ *without replacement*. Otherwise, for any non-trivial distribution, there is some database D_* that is drawn with probability at most $1/2^n$, and we may modify any distributional-privacy preserving mechanism A such that for every query Q, $A(D_*, Q) = D_*$, and for any $D_i \neq D_*$, $A(D_i, Q)$ behaves as before. Since this new behavior occurs with probability $\leq \beta$ over draws from D for $\beta = O(1/2^n)$, A still preserves distributional privacy, but no longer preserves differential privacy (which requires that the privacy guarantee hold for *every* pair of neighboring databases).

8. CONCLUSIONS AND OPEN PROBLEMS

In this work, we view the problem of database privacy through the lens of learning theory. This suggests both a new definition of privacy, distributional privacy (which we show is strictly stronger than differential privacy), and the idea that we can study usefulness relative to particular classes of functions. Restricting our notion of usefulness to particular classes of functions allows us to circumvent the lower bounds of [6, 9, 10] which show that non-interactive privacy preserving database access mechanisms can not in general be as useful as interactive mechanisms. In fact, we are able to show that it is possible to release privacy-preserving databases that are useful for all queries over a discretized domain in a concept class with polynomial VC-dimension. We show that this discretization is necessary by proving that it is impossible to privately release a database that is useful for halfspace queries without relaxing our definition of usefulness, but we demonstrate an algorithm that does so efficiently under a small relaxation of this definition.

This work demonstrates that the existing very strong lower bounds for useful, privacy-preserving, non-interactive mechanisms are not insurmountable, but can be circumvented by a number of reasonable relaxations to the standard definitions. However, our paper leaves a number of important questions open. Prime among them is the question of *efficient* private data release—we have shown that information theoretically it is possible to release a database that is useful for any concept class with polynomial VC-dimension (under our original, strong definition of usefulness) while preserving differential privacy, but we know how to do this *efficiently* only for the simplest classes of functions. Is it possible to *efficiently* privately and usefully release a database for every concept class with polynomial VC-Dimension? Is it possible for the class of conjunctions? For the class of parity functions?

One approach to efficient database release is to efficiently sample from the distribution defined by the exponential mechanism in Theorem 3.1. In order to do so, it might be necessary to relax our quality function, since even computing the quality function on a particular input/output database pair is as hard as agnostically learning, over arbitrary distributions, the concept class for which we want to guarantee usefulness. (To see this, consider labeling the points in the input database as positive examples and those in the output database as negative.) Additionally, we note that the ability to agnostically learn a concept class is not by itself enough to efficiently *sample* from the desired distribution; one approach to sampling is to design a random Markov process that converges quickly to the desired stationary distribution.

9. ACKNOWLEDGMENTS

We thank David Abraham, Cynthia Dwork, Shiva Kasiviswanathan, Adam Meyerson, Sofya Raskhodnikova, Amit Sahai, and Adam Smith for many useful discussions. We thank Ryan O'Donnell for the insight that led to the proof of Theorem A.6.

10. REFERENCES

[1] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[2] M. Balcan, A. Blum, and S. Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65(1):79–94, 2006.

[3] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART*

Symposium on Principles of Database Systems, pages 273–282, 2007.

[4] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 128–138, 2005.

[5] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss Lemma. *International Computer Science Institute, Technical Report*, pages 99–006, 1999.

[6] I. Dinur and K. Nissim. Revealing information while preserving privacy. *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.

[7] C. Dwork. Differential privacy. *Proc. ICALP*, 2006.

[8] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our Data, Ourselves: Privacy via Distributed Noise Generation. *Proceedings of Advances in CryptologyEurocrypt 2006*, pages 486–503, 2006.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[10] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of LP decoding. *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*, pages 85–94, 2007.

[11] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. *Proc. CRYPTO*, pages 528–544, 2004.

[12] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS '03: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 211–222, New York, NY, USA, 2003. ACM.

[13] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. *Proceedings of the thirtieth annual ACM Symposium on Theory of Computing*, pages 604–613, 1998.

[14] S. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? http://arxiv.org/abs/0803.0924v1.

[15] F. McSherry and K. Talwar. Mechanism Design via Differential Privacy. *Proceedings of the 48th Annual Symposium on Foundations of Computer Science*, 2007.

[16] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*, pages 75–84, 2007.

[17] V. Rastogi, D. Suciu, and S. Hong. The Boundary Between Privacy and Utility in Data Publishing. *VLDB*, 2007.

[18] A. J. Smola and B. Schölkopf. *Learning with Kernels*. MIT Press, 2002.

[19] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons Inc., 1998.

APPENDIX

A. PARITY: A SMALL SEPARATION

Dwork et al. [9] provide a separation between interactive and non-interactive differential-privacy preserving mechanisms for a class of queries that are not predicate queries. They also provide a separation between interactive and non-interactive "randomized-response" mechanisms for parity queries (defined below), which are predicate queries. "Randomized-response" mechanisms are a class of non-interactive mechanisms that independently perturb each point in D and release the independently perturbed points. Here, we provide a small separation between interactive mechanisms and arbitrary non-interactive mechanisms that output datasets useful for parity queries. We prove this separation for mechanisms that preserve differential privacy—our separation therefore also holds for *distributional*-privacy preserving mechanisms.

DEFINITION A.1. *Given a database D containing n points in $\{-1,1\}^d$, and for $S \subseteq \{1,\ldots,d\}$, a* parity query *is given by*

$$PQ_S = \frac{\left|\{x \in D : \prod_{i\in S} x_i = 1\}\right|}{|D|}.$$

We show that for any non-interactive mechanism A that preserves α-differential privacy for $\alpha = \Omega(1/\mathrm{poly}(n))$ and outputs a database $\widehat{D} = A(D)$, there exists some $S \subseteq \{1,\ldots,d\}$ such that $|PQ_S(\widehat{D}) - PQ_S(D)| = \Omega(1/\sqrt{n})$. This provides a separation, since for any S, $GS_{PQ_S} = 1/n$, and so for for any S, with high probability, the interactive mechanism $A(D,Q)$ of [9] satisfies $|A(D,PQ_S) - PQ_S(D)| = o(1/\sqrt{n})$ while satisfying α-differential privacy. This also shows that our bound from Theorem 3.1 cannot be improved to have a $o(1/\epsilon^2)$ dependence on ϵ.

We begin with the claim that given some database D consisting of n distinct points in $\{-1,1\}^d$, any non-interactive α-differential privacy preserving mechanism that outputs a sanitized database must with high probability output a database $\widehat{D}$ that differs from D on at least half of its points.

CLAIM A.2. *If the non-interactive mechanism A preserves α-differential privacy for $\alpha = \Omega(1/\mathrm{poly}(n))$, $\Pr[|\widehat{D} \cap D| \geq n/2] < 1/2$.*

We next present a few facts from discrete Fourier analysis.

PROPERTY A.3. *For any function $h : \{-1,1\} \to \mathbb{R}$, we may express h as a linear combination of parity functions: $h(x) = \sum_{S\subseteq\{1,\ldots,d\}} \hat{h}(S)\chi_S(x)$, where $\chi_S(x) = \prod_{i\in S} x_i$. Moreover, the coefficients $\hat{h}(S)$ take values*

$$\hat{h}(S) = \frac{1}{2^d} \sum_{x\in\{-1,1\}^d} g(x)\chi_S(x).$$

PROPERTY A.4 (PARSEVAL'S IDENTITY). *For any function $h : \{-1,1\} \to \mathbb{R}$,*

$$\frac{1}{2^d} \sum_{x\in\{-1,1\}^d} h(x)^2 = \sum_{S\subseteq\{1,\ldots,d\}} \hat{h}(S)^2.$$

LEMMA A.5. *For $D_1, D_2 \in (\{-1,1\}^d)^n$, if $|D_1 \cap D_2| \leq n/2$, then there exists $S \in \{1,\ldots,d\}$ such that $|PQ_S(D_1) - PQ_S(D_2)| = \Omega(1/\sqrt{n})$*[4].

PROOF. Let $f(x) : \{-1,1\}^d \to \{0,1\}$ be the indicator function of D_1: $f(x) = 1 \Leftrightarrow x \in D_1$. Similarly, let $g(x) : \{-1,1\}^d \to \{0,1\}$ be the indicator function of D_2. By our hypothesis,

$$\sum_{x\in\{-1,1\}^n} |f(x) - g(x)| \geq n/2.$$

[4]Note that we are implicitly assuming that $d = \Omega(\log n)$

Therefore,

$$\begin{aligned} n/2 &\leq \sum_{x\in\{-1,1\}^d} |f(x)-g(x)| \\ &= \sum_{x\in\{-1,1\}^d} (f(x)-g(x))^2 \\ &= 2^d \sum_{S\subseteq\{1,\ldots,d\}} (\hat{f}(S)-\hat{g}(S))^2, \end{aligned}$$

where the first equality follows from the fact that f and g have range $\{0,1\}$, and the second follows from Parseval's identity and the linearity of Fourier coefficients. Therefore, there exists some $S \subseteq \{1,\ldots,d\}$ such that $(\hat{f}(S)-\hat{g}(S))^2 \geq n/2^{2d+1}$, and so $|\hat{f}(S)-\hat{g}(S)| \geq \sqrt{n}/(2^d\sqrt{2})$). We also have

$$\begin{aligned} &\hat{f}(S)-\hat{g}(S) \\ &\quad= \frac{1}{2^d}\sum_{x\in\{-1,1\}^d} f(x)\chi_S(x) - \frac{1}{2^d}\sum_{x\in\{-1,1\}^d} g(x)\chi_S(x) \\ &\quad= \frac{1}{2^d}\sum_{x\in D_1} \chi_S(x) - \frac{1}{2^d}\sum_{x\in D_2} \chi_S(x) \\ &\quad= \frac{n}{2^{d-1}}(PQ_S(D_1)-PQ_S(D_2)). \end{aligned}$$

Therefore, $|(PQ_S(D_1)-PQ_S(D_2)| \geq \Omega(1/\sqrt{n})$, which completes the proof. □

Combining the Claim A.2 and Lemma A.5, we get our result:

THEOREM A.6. *For any non-interactive mechanism A that outputs a database $\hat{D}_1 = A(D_1)$ and preserves α-differential privacy for $\alpha = \Omega(1/\text{poly}(n))$, with probability $> 1/2$ there exists some $S \subseteq \{1,\ldots,d\}$ such that $|PQ_S(D_1) - PQ_S(\hat{D}_1)| = \Omega(1/\sqrt{n})$.*