A Learning Theory Approach to Non-Interactive Database Privacy

Avrim Blum, Carnegie Mellon University
Katrina Ligett, California Institute of Technology
Aaron Roth, University of Pennsylvania

In this paper we demonstrate that, ignoring computational constraints, it is possible to release synthetic databases that are useful for accurately answering large classes of queries while preserving differential privacy. Specifically, we give a mechanism that privately releases synthetic data useful for answering a class of queries over a *discrete* domain with error that grows as a function of the size of the smallest net approximately representing the answers to that class of queries. We show that this in particular implies a mechanism for counting queries that gives error guarantees that grow only with the VC-dimension of the class of queries, which itself grows at most logarithmically with the size of the query class.

We also show that it is not possible to release even simple classes of queries (such as intervals and their generalizations) over *continuous* domains with worst-case utility guarantees while preserving differential privacy. In response to this, we consider a relaxation of the utility guarantee and give a privacy preserving polynomial time algorithm that for any halfspace query will provide an answer that is accurate for some small perturbation of the query. This algorithm does not release synthetic data, but instead another data structure capable of representing an answer for each query. We also give an efficient algorithm for releasing synthetic data for the class of interval queries and axis-aligned rectangles of constant dimension over *discrete* domains.

1. INTRODUCTION

As large-scale collection of personal information becomes more commonplace, 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 often are not able to release their data in raw form. Moreover, it is far from clear whether or not these data curators can allow analysts access to the data in *any* form if they are to provide a rigorous measure of privacy to the individuals whose data is contained in the data sets. 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 [DN04; BDMN05; DMNS06] formalizes the notion of *differential privacy*. Informally, an algorithm satisfies differential privacy if modifying a single database element does not change the probability of any outcome of the privatization mechanism by more than some small amount (see Definition 2.1 for a formal definition). The definition is intended to capture the notion that "distributional information is not private": that it is acceptable to release information that is encoded in aggregate over the dataset, but not information that is encoded only in the single record of an individual. In other words, 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 they may later plausibly claim any other fake set of values, as changing one person's entries would produce nearly the same probability distribution over outputs.

Lower bounds of Dinur and Nissim [DN03] imply that one cannot hope to be able to usefully answer large numbers of arbitrary queries to arbitrarily low error. In this paper, motivated by learning theory, we propose the study of privacy-preserving mechanisms that are useful for answering all queries in a particular class (such as all conjunction queries or all halfspace queries), that is large but specified a-priori. In

particular, we focus on counting 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 are accurate up to error of magnitude at most α.

Building on the techniques of McSherry and Talwar and Kasiviswanathan et al. [MT07; KLN$^+$08], we show that for discretized domains, for any concept class that admits an α-net $\mathcal{N}_\alpha$, it is possible to privately release synthetic data that is useful for the class, with error that grows proportionally to the *logarithm* of the size of $\mathcal{N}_\alpha$. As a consequence, we show that it is possible to release data useful for a set of counting queries with error that grows proportionally to the VC-dimension of the class of queries. The algorithm is not in general computationally efficient. We are able to give a different algorithm that efficiently releases synthetic data for the class of interval queries (and more generally, axis-aligned rectangles in fixed dimension) that achieves guarantees in a similar range of parameters.

Unfortunately, we show that for non-discretized domains, under the above definition of usefulness, it is impossible to publish a differentially private database that is useful in the worst case 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 [AB99; Vap98; SS02]; 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.

1.1. Prior and Subsequent Work

1.1.1. Prior Work. Recent work on theoretical guarantees for data privacy was initiated by [DN03]. The notion of differential privacy, finally formalized by [DMNS06; Dwo06], 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 formalized by McSherry and Talwar [MT07].

Much of the initial work focused on *lower bounds*. Dinur and Nissim [DN03] showed that any mechanism that answers substantially more than a linear number of *subset-sum* queries with error $o(\sqrt{n})$ yields what they called *blatant non-privacy* – i.e. it allows an adversary to reconstruct all but a $o(1)$ fraction of the original database. They also show that releasing the answers to all subset sum queries with error $o(n)$ leads to blatant non-privacy. In this paper, we use a similar argument to show that the accuracy for mechanisms that restrict themselves to fixed classes of queries must depend on the VC-dimension of those classes. Dwork et al. [DMT07] extend this result to the case in which the private mechanism can answer a constant fraction of queries with arbitrary error, and show that still if the error on the remaining queries is $o(\sqrt{n})$, the result is blatant non-privacy. Dwork and Yekhanin [DY08] give further improvements. These results easily extend to the case of counting queries which we consider here.

Dwork et al. [DMNS06], in the paper that defined differential privacy, show that releasing the answers to k *low sensitivity* queries (a generalization of the counting queries we consider here) with noise drawn independently from the Laplace distribution with scale k/ϵ preserves ϵ-differential privacy. Unfortunately, the noise scales

linearly in the number of queries answered, and so this mechanism can only answer a sub-linear number of queries with non-trivial accuracy. Blum et al. [BDMN05] 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 access the database by asking counting-queries perturbed with independent noise from the Laplace distribution, and so can still only make a sublinear number of queries. In this paper, we give a mechanism for privately answering counting queries with noise that grows only logarithmically with the number of queries asked (or more generally with the VC-dimension of the query class). This improvement allows an analyst to answer an exponentially large number of queries with non-trivial error, rather than only linearly many.

Most similar to this paper is the work of Kasiviswanathan et al. [KLN$^+$08] and McSherry and Talwar [MT07]. Kasiviswanathan et al. study what can be learned privately when what is desired is that the hypothesis output by the learning algorithm satisfies differential privacy. They show that in a PAC learning model in which the learner has access to the private database, ignoring computational constraints, anything that is PAC learnable is also privately PAC learnable. We build upon the technique in their paper to show that in fact, it is possible to privately release a dataset that is simultaneously useful for any function in a concept class of polynomial VC-dimension. Kasiviswanathan et al. also study several restrictions on learning algorithms, show separation between these learning models, and give efficient algorithms for learning particular concept classes. Both our paper and [KLN$^+$08] rely on the exponential mechanism, which was introduced by McSherry and Talwar [MT07]

1.1.2. Subsequent Work. Since the original publication of this paper in STOC 2008 [BLR08] there has been a substantial amount of follow up work. A sequence of papers by Dwork et al. [DNR$^+$09; DRV10] give a non-interactive mechanism for releasing counting queries with accuracy that depends in a similar way to the mechanism presented in this paper on the total number of queries asked, but has a better dependence on the database size. This comes at the expense of relaxing the notion of ϵ-differential privacy to an approximate version called (ϵ, δ)-differential privacy. The mechanism of [DRV10] also extends to arbitrary low-sensitivity queries rather than only counting queries. This extension makes crucial use of the relaxation to (ϵ, δ)-privacy, as results such as those given in this paper cannot be extended to arbitrary low-sensitivity queries while satisfying ϵ-differential privacy as shown recently by De [De11].

Roth and Roughgarden [RR10] showed that bounds similar to those achieved in this paper can also be achieved in the *interactive* setting, in which queries are allowed to arrive online and must be answered before the next query is known. In many applications, this gives a large improvement in the accuracy of answers, because it allows the analyst to pay for those queries which were actually asked in the course of a computation (which may be only polynomially many), as opposed to all queries which might potentially be asked, as is necessary for a non-interactive mechanism. Hardt and Rothblum [HR10] gave an improved mechanism for the interactive setting based on the multiplicative weights framework which achieves bounds comparable to the improved bounds of [DRV10], also in the interactive setting. An offline version of this mechanism (constructed by pairing the online mechanism with an agnostic learner for the class of queries of interest) was given by [GHRU11; HLM12]. Gupta, Roth, and Ullman unified the online mechanisms of [RR10; HR10] into a generic framework (and improved their error bounds) by giving a generic reduction from online learning algorithms in the mistake bound model to private query release algorithms in the interactive setting [GRU11]. [GRU11] also give a new mechanism based on this reduction that achieves

improved error guarantees for the setting in which the database size is comparable to the size of the data universe.

There has also been significant subsequent attention paid to the specific problem of releasing the class of conjunctions (a special case of counting queries) with low error using algorithms with more efficient run-time than the one given in this paper. Gupta et al. [GHRU11] give an algorithm which runs in time polynomial in the size of the database, and releases the class of conjunctions to $O(1)$ *average* error while preserving differential privacy. Hardt, Rothblum, and Servedio [HRS11] give an algorithm which runs in time proportional d^k (for databases over a data universe $X = \{0,1\}^d$) and releases conjunctions of most k variables with worst-case error guarantees. Their algorithm improves over the Laplace mechanism (which also requires run-time d^k) because it only requires that the database size be proportional to $d^{\sqrt{k}}$ (The Laplace mechanism would require a database of size d^k). As a building block for this result, they also give a mechanism with run-time proportional to $d^{\sqrt{k}}$ which gives average-case error guarantees.

Range queries—which extend the class of constant-dimensional interval queries which we consider in this paper—have also subsequently received substantial attention [XWG10; HRMS10; LHR+10; LM11; LM12a; LM12b; MN12; HLM12].

There has also been progress in proving lower bounds. Dwork et al. [DNR+09] show that in general, the problem of releasing synthetic data giving non-trivial error for arbitrary classes of counting queries requires run-time that is linear in the size of the data universe and the size of the query class (modulo cryptographic assumptions). This in particular precludes improving the run-time of the general mechanism presented in this paper to be only polynomial in the size of the database. Ullman and Vadhan [UV11] extend this result to show that releasing synthetic data is hard even for the simple class of conjunctions of at most 2 variables. This striking result emphasizes that output representation is extremely important, because it is possible to release the answers to all of the (at most d^2) conjunctions of size 2 privately and efficiently using output representations other than synthetic data. Kasiviswanathan et al. [KRSU10] extend the lower bounds [DN03] from arbitrary subset-sum queries to hold also for an algorithm that only releases conjunctions. Hardt and Talwar showed how to prove lower bounds for differentially query release using packing arguments, and gave an optimal lower bound for a certain range of parameters [HT10]. De recently refined this style of argument and extended it to additional settings [De11]. Gupta et al. [GHRU11] showed that the class of queries that can be released by mechanisms that access the database using only *statistical queries* (which includes almost all mechanisms known to date, with the exception of the parity learning algorithm of [KLN+08]) is equal to the class of queries that can be agnostically learned using statistical queries. This rules out a mechanism even for releasing conjunctions to subconstant error which accesses the data using only a polynomial number of statistical queries.

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 $\tilde{O}(\text{VCDIM}(C)/\alpha^2)$ is sufficient so that with high probability, for *every* query $q \in C$, the proportion of examples in D satisfying q is

within $\pm\alpha$ of the true probability mass under $\mathcal{D}$ [AB99; Vap98].[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] Note that since the simple Laplace mechanism 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 3.10), since $|C| \geq 2^{\text{VCDIM(C)}}$.

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.

2. DEFINITIONS

We consider databases which are n-tuples from some abstract domain X: i.e. $D \in X^n$. We will also write $n = |D|$ for the size of the database. For clarity, we think of n as being publicly known (and, in particular, all databases have the same size n), but as we will discuss, this assumption can be removed. We think of X as the set of all possible data-records. For example, if data elements are represented as bit-strings of length d, then $X = \{0,1\}^d$ would be the boolean hypercube in d dimensions. Databases are not endowed with an ordering: they are simply multi-sets (they can contain multiple copies of the same element $x \in X$).

A database access mechanism is a randomized mapping $A : X^n \to R$, where R is some arbitrary range. We say that A outputs synthetic data if its output is itself a database, and if the intended evaluation of a query on the output is the obvious one: i.e. if $R = X^*$, and f is evaluated on $A(D) = D'$ by computing $f(D')$.

Our privacy solution concept will be the by now standard notion of differential privacy. Crucial to this definition will be the notion of *neighboring databases*. We say that two databases $D, D' \in X^n$ are *neighboring* if they differ in only a single data element: i.e. they are neighbors if their symmetric difference $|D \Delta D'| \leq 2$.

Definition 2.1 (*Differential Privacy [DMNS06]*). A database access mechanism $A : X^n \to R$ is ϵ-differentially private if for all neighboring pairs of databases $D, D' \in X^n$

[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.

and for all outcome events $S \subseteq R$, the following holds:

$$\Pr[A(D) \in S] \leq \exp(\epsilon) \Pr[A(D') \in S]$$

Remark 2.2. In the differential privacy literature, there are actually two related, though distinct, notions of differential privacy. In the notion we adopt above, the database size n is publicly known and two databases are neighboring if one can be derived from the other by *swapping* one database element for another. That is, in this notion, an individual deciding whether to submit either accurate or fake personal information is assured that an observer would not be able to tell the difference. In the other notion, n is itself private information, and two databases are neighboring if one can be derived from the other by *adding or removing* a single database element. That is, in the second notion, an individual deciding whether to submit any information at all is assured that an observer cannot tell the difference. The two notions are very similar: two databases that are neighboring in the public n regime are at distance at most 2 in the private n regime. Similarly, using standard techniques, in the private n regime, n can still be estimated accurately to within an additive factor of $O(1/\epsilon)$, which almost always allows simulation of the public n regime up to small loss. Here, we adopt the public n regime because it greatly simplifies our analysis and notation; nevertheless up to constants, all of our results can be adapted to the private n regime using standard techniques. We re-prove our main result (our release mechanism for counting queries) for the private n version of differential privacy in the appendix.

Definition 2.3. The *global sensitivity* of a query f is its maximum difference when evaluated on two neighboring databases:

$$GS_f^n = \max_{D,D' \in X^n : |D \Delta D'| = 2} |f(D) - f(D')|.$$

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

Definition 2.4. A (normalized) *counting query* Q_φ, defined in terms of a predicate $\varphi : X \to \{0,1\}$ is defined to be

$$Q_\varphi(D) = \frac{1}{|D|} \sum_{x \in D} \varphi(x).$$

It evaluates to the fraction of elements in the database that satisfy the predicate φ.

OBSERVATION 2.5. *For any predicate $\varphi : X \to \{0,1\}$, the corresponding counting query $Q_\varphi : X^* \to [0,1]$ has global sensitivity $GS^n_{Q_\varphi} \leq 1/n$*

Remark 2.6. Note that because we regard n as publicly known, the global sensitivity of a normalized counting query is well defined. We could equally well work with unnormalized counting queries, which have sensitivity 1 in both the public and private n regime, but this would result in more cumbersome notation later on.

We remark that everything in this paper easily extends to the case of more general *linear* queries, which can are defined analogously to counting queries, but involve real valued predicates $\varphi : X \to [0,1]$. For simplicity we restrict ourselves to counting queries in this paper, but see [Rot10] for the natural extension to linear queries.

A key measure of complexity that we will use for counting queries is VC-dimension. VC-dimension is strictly speaking a measure of complexity of classes of predicates, but we will associate the VC-dimension of classes of predicates with their corresponding class of counting queries.

Definition 2.7 (*Shattering*). A class of predicates P *shatters* a collection of points $S \subseteq X$ if for every $T \subseteq S$, there exists a $\varphi \in P$ such that $\{x \in S : \varphi(x) = 1\} = T$. That is, P shatters S if for every one of the $2^{|S|}$ subsets T of S, there is some predicate in P that labels exactly those elements as positive, and does not label any of the elements in $S \setminus T$ as positive.

Definition 2.8 (*VC-Dimension*). A collection of predicates P has VC-dimension d if there exists some set $S \subseteq X$ of cardinality $|S| = d$ such that P shatters S, and P does not shatter any set of cardinality $d + 1$. We denote this quantity by VC-DIM(P). We abuse notation and also write VC-DIM(C) where C is a class of counting queries, to denote the VC-dimension of the corresponding collection of predicates.

Dwork et al. [DMNS06] give a mechanism which can answer any single low-sensitivity query while preserving differential privacy:

Definition 2.9 (*Laplace mechanism*). The Laplace mechanism responds to a query Q by returning $Q(D) + Z$ where Z is a random variable drawn from the Laplace distribution: $Z \sim \text{Lap}(GS_Q^n/\epsilon)$.

The Laplace distribution with scale b, which we denote by Lap(b), has probability density function

$$f(x|b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right)$$

THEOREM 2.10 (DWORK ET AL. [DMNS06]). *The Laplace mechanism preserves ϵ-differential privacy.*

This mechanism answers queries interactively, but for a fixed privacy level, its accuracy guarantees degrade linearly in the number of queries that it answers. The following composition theorem is useful: it tells us that a mechanism which runs k ϵ-differentially private subroutines is $k\epsilon$-differentially private.

THEOREM 2.11 (DWORK ET AL. [DKM$^+$06]). *If mechanisms $M_1, \ldots, M_k$ are each ϵ-differentially private, then the mechanism M defined by the (string) composition of the k mechanisms: $M(D) = (M_1(D), \ldots, M_k(D))$ is $k\epsilon$-differentially private.*

We propose to construct database access mechanisms which produce one-shot (non-interactive) outputs that 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, as implied by the lower bounds of Dinur and Nissim [DN03], we cannot hope to be able to usefully answer arbitrary queries. We instead seek to release synthetic databases which are "useful" (defined below) for restricted classes of queries C.

Definition 2.12 (*Usefulness*). A database access mechanism A is (α, δ)-*useful* with respect to queries in class C if for every database $D \in X^n$, with probability at least $1 - \delta$, the output of the mechanism $\widehat{D} = A(D)$ satisfies:

$$\max_{Q \in C} |Q(\widehat{D}) - Q(D)| \leq \alpha$$

In this paper, we will derive (α, δ)-useful mechanisms from small α-nets:

Definition 2.13 (α-*net*). An α-net of databases with respect to a class of queries C is a set $N \subset X^*$ such that for all $D \in X^n$, there exists an element of the α-net $D' \in N$ such that:

$$\max_{Q \in C} |Q(D) - Q(D')| \leq \alpha$$

We write $N_\alpha(C)$ to denote an α-net of minimum cardinality among the set of all α-nets for C.

3. GENERAL RELEASE MECHANISM

In this section we present our general release mechanism. It is an instantiation of the *exponential mechanism* of McSherry and Talwar [MT07].

Given some arbitrary range $\mathcal{R}$, the exponential mechanism is defined with respect to some quality function $q : X^n \times \mathcal{R} \to \mathbb{R}$, which maps database/output pairs to quality scores. We should interpret this intuitively as a measure stating that fixing a database D, the user would prefer the mechanism to output some element of $\mathcal{R}$ with as high a quality score as possible.

Definition 3.1 (*The Exponential Mechanism [MT07]*). The exponential mechanism $M_E(D, q, \mathcal{R}, \epsilon)$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp(\frac{\epsilon q(D,r)}{2GS_q^n})$.

McSherry and Talwar showed that the exponential mechanism preserves differential privacy. It is important to note that the exponential mechanism can define a complex distribution over a large arbitrary domain, and so it may not be possible to implement the exponential mechanism efficiently when the range of the mechanism is super-polynomially large in the natural parameters of the problem. This will be the case with our instantiation of it.

THEOREM 3.2 ([MT07]). *The exponential mechanism preserves ϵ-differential privacy.*

ALGORITHM 1: NetMechanism(D, C, ϵ, α)

let $\mathcal{R} \leftarrow N_{\alpha/2}(C)$.
let $q : X^n \times \mathcal{R} \to \mathbb{R}$ be defined to be:

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

Sample And Output $D' \in \mathcal{R}$ with the exponential mechanism $M_E(D, q, \mathcal{R}, \epsilon)$

We first observe that the Algorithm 1, the Net mechanism, preserves ϵ-differential privacy.

PROPOSITION 3.3. *The Net mechanism is ϵ-differentially private.*

PROOF. The Net mechanism is simply an instantiation of the exponential mechanism. Therefore, privacy follows from Theorem 3.2. □

We may now analyze the usefulness of the Net mechanism. A similar analysis of the exponential mechanism appears in [MT07].

PROPOSITION 3.4. *For any class of queries C (not necessarily counting queries) the Net mechanism is (α, δ)-useful for any α such that:*

$$\alpha \geq \frac{4\Delta}{\epsilon} \log \frac{N_\alpha(C)}{\delta}$$

where $\Delta = \max_{Q \in C} GS_Q^n$.

PROOF. First observe that the sensitivity of the quality score $GS_q^n \leq \max_{Q \in C} GS_Q^n = \Delta$.

By the definition of an $\alpha/2$-net, we know that there exists some $D^* \in \mathcal{R}$ such that $q(D, D^*) \geq -\alpha/2$. By the definition of the exponential mechanism, this D^* is output with probability proportional to at least $\exp(\frac{-\epsilon\alpha}{4GS_q^n})$. Similarly, there are at most $|N_\alpha(C)|$ databases $D' \in \mathcal{R}$ such that $q(D, D') \leq -\alpha$ (simply because $\mathcal{R} = N_\alpha(C)$). Hence, by a union bound, the probability that the exponential mechanism outputs some D' with $q(D, D') \leq -\alpha$ is proportional to at most $|N_\alpha(C)| \exp(\frac{-\epsilon\alpha}{2GS_q^n})$. Therefore, if we denote by A the event that the Net mechanism outputs some D^* with $q(D, D^*) \geq -\alpha/2$, and denote by B the event that the Net mechanism outputs some D' with $q(D, D') \leq -\alpha$, we have:

$$\begin{aligned} \frac{\Pr[A]}{\Pr[B]} &\geq \frac{\exp(\frac{-\epsilon\alpha}{4\Delta})}{|N_\alpha(C)| \exp(\frac{-\epsilon\alpha}{2\Delta})} \\ &= \frac{\exp(\frac{\epsilon\alpha}{4\Delta})}{|N_\alpha(C)|} \end{aligned}$$

Note that if this ratio is at least $1/\delta$, then we will have proven that the Net mechanism is (α, δ) useful with respect to C. Solving for α, we find that this is condition is satisfied so long as

$$\alpha \geq \frac{4\Delta}{\epsilon} \log \frac{N_\alpha(C)}{\delta}$$

□

We have therefore reduced the problem of giving upper bounds on the usefulness of differentially private database access mechanisms to the problem of upper bounding the sensitivity of the queries in question, and the size of the smallest α-net for the set of queries in question. Recall that for *counting* queries Q on databases of size n, we always have $GS_Q^n \leq 1/n$. Therefore we have the immediate corollary:

COROLLARY 3.5. *For any class of counting queries C the Net mechanism is (α, δ)-useful for any α such that:*

$$\alpha \geq \frac{4}{\epsilon n} \log \frac{N_\alpha(C)}{\delta}$$

To complete the proof of utility for the Net mechanism for counting queries, it remains to prove upper bounds on the size of minimal α-nets for counting queries. We begin with a bound for finite classes of queries.

THEOREM 3.6. *For any finite class of counting queries C:*

$$|N_\alpha(C)| \leq |X|^{\frac{\log |C|}{\alpha^2}}$$

In order to prove this theorem, we will show that for any collection of counting queries C and for any database D, there is a "small" database D' of size $|D'| = \frac{\log |C|}{\alpha^2}$ that approximately encodes the answers to every query in C, up to error α. Crucially, this bound will be independent of $|D|$.

LEMMA 3.7. *For any $D \in X^n$ and for any finite collection of counting queries C, there exists a database D' of size*

$$|D'| = \frac{\log |C|}{\alpha^2}$$

such that:

$$\max_{Q \in C} |Q(D) - Q(D')| \leq \alpha$$

PROOF. Let $m = \frac{\log |C|}{\alpha^2}$ We will construct a database D' by taking m uniformly random samples from the elements of D. Specifically, for $i \in \{1, \ldots, m\}$ let X_i be a random variable taking value x_j with probability $|\{x \in D : x = x_j\}|/|D|$, and let D' be the database containing elements $X_1, \ldots, X_m$. Now fix any $Q_\varphi \in C$ and consider the quantity $Q_\varphi(D')$. We have: $Q_\varphi(D') = \frac{1}{m} \sum_{i=1}^{m} \varphi(X_i)$. We note that each term of the sum $\varphi(X_i)$ is a bounded random variable taking values $0 \leq \varphi(X_i) \leq 1$, and that the expectation of $Q_\varphi(D')$ is:

$$\mathrm{E}[Q(D')] = \frac{1}{m} \sum_{i=1}^{m} \mathrm{E}[\varphi(X_i)] = Q_\varphi(D)$$

Therefore, we can apply a standard Chernoff bound which gives:

$$\Pr\left[|Q_\varphi(D') - Q_\varphi(D)| > \alpha\right] \leq 2e^{-2m\alpha^2}$$

Taking a union bound over all of the counting queries $Q_\varphi \in C$, we get:

$$\Pr\left[\max_{Q_\varphi \in C} |Q_\varphi(D') - Q_\varphi(D)| > \alpha\right] \leq 2|C|e^{-2m\alpha^2}$$

Plugging in the chosen number of samples m makes the right hand side smaller than 1 (so long as $|C| > 2$), proving that there exists a database of size m satisfying the stated bound, which completes the proof of the lemma. □

Now we can complete the proof of Theorem 3.6.

PROOF OF THEOREM 3.6. By Lemma 3.7, we have that for any $D \in X^*$ there exists a database $D' \in X^*$ with $|D'| = \frac{\log |C|}{\alpha^2}$ such that $\max_{Q_\varphi \in C} |Q_\varphi(D) - Q_\varphi(D')| \leq \alpha$. Therefore, if we take $N = \{D' \in X^* : |D'| = \frac{\log |C|}{\alpha^2}\}$ to be the set of *every* database of size $\frac{\log |C|}{\alpha^2}$, we have an α-net for C. Since

$$|N| = |X|^{\frac{\log |C|}{\alpha^2}}$$

and by definition $|N_\alpha(C)| \leq |N|$, we have proven the theorem. □

When the cardinality of the concept class is untenably large, we can replace lemma 3.7 with the following lemma:

LEMMA 3.8 ([AB99; VAP98]). *For any $D \in X^*$ and for any collection of counting queries C, there exists a database D' of size*

$$|D'| = O(\textbf{\textit{VCDIM}}(C)\log(1/\alpha)/\alpha^2)$$

such that:

$$\max_{Q \in C} |Q(D) - Q(D')| \leq \alpha$$

This lemma straightforwardly gives an analogue of Theorem 3.6:

THEOREM 3.9. *For any class of counting queries C:*

$$|N_\alpha(C)| \leq |X|^{O(\textbf{\textit{VCDIM}}(C)\log(1/\alpha)/\alpha^2)}$$

Note that we always have $\text{VCDIM}(C) \leq \log |C|$ for finite classes of counting queries, and so (modulo constants and the $\log(1/\alpha)$ term) Theorem 3.9 is strictly stronger than Theorem 3.6.

Finally, we can instantiate Corollary 3.5 to give our main utility theorem for the Net mechanism.

THEOREM 3.10. *For any class of counting queries C, there exists constant c such that the Net mechanism is (α, δ)-useful for:*

$$\alpha \geq c \cdot \left(\frac{\textit{VCDIM}(C) \log |X| + \log 1/\delta}{\epsilon n} \right)^{1/3}.$$

PROOF. The instantiation guarantees the existence of constants c_1 and c_2 such that the Net mechanism gives (α, δ)-usefulness for an α such that:

$$\alpha \geq \frac{4}{\epsilon n} \left(\frac{c_1 \text{VCDIM}(C) \log(1/\alpha) \log |X|}{\alpha^2} + c_2 \log |X| + \log 1/\delta \right)$$

We assume that $\alpha \leq 1/2$ (i.e. that the error guaranteed by the theorem is nontrivial). In this case, we have $\alpha^2 / \log(1/\alpha) < 1$ and so it is only pessimistic to take:

$$\frac{\alpha^3}{\log(1/\alpha)} \geq \frac{4}{\epsilon n} \left(c_1 \text{VCDIM}(C) \log |X| + c_2 \log |X| + \log 1/\delta \right).$$

Moreover, we have $\alpha^3 \leq \alpha^3 / \log(1/\alpha)$, and so it is only further pessimistic to consider

$$\alpha^3 \geq \frac{4}{\epsilon n} \left(c_1 \text{VCDIM}(C) \log |X| + c_2 \log |X| + \log 1/\delta \right).$$

Solving for α yields

$$\left(\frac{4(c_1 \text{VCDIM}(C) \log |X| + c_2 \log |X| + \log 1/\delta)}{3\epsilon n} \right)^{1/3},$$

which yields the theorem. □

Theorem 3.10 shows that a database of size $\widetilde{O}(\frac{\log |X| \text{VCDIM}(C)}{\alpha^3 \epsilon})$ is sufficient in order to output a set of points that is α-useful for a concept class C, while simultaneously preserving ϵ-differential privacy. If we were to view our database as having been drawn from some distribution $\mathcal{D}$, this is only an extra $\widetilde{O}(\frac{\log |X|}{\alpha \epsilon})$ factor larger than what would be required to achieve α-usefulness with respect to $\mathcal{D}$, even without any privacy guarantee.

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.

3.1. The Necessity of a Dependence on VC-Dimension

We just gave an ϵ-differentially private mechanism that is (α, δ)-useful with respect to any set of counting queries C, when given a database of size $n \geq \widetilde{O}(\frac{\log |X| \text{VCDIM}(C)}{\alpha^3 \epsilon})$. In this section, we show that the dependence on the VC-dimension of the class C is tight.

The proof follows an argument similar to one used by Dinur and Nissim to show that no private mechanism can answer *all* counting queries to nontrivial accuracy [DN03]. Fix a class of counting queries C corresponding to a class of predicates P of VC-dimension d. Let $S \subset X$ denote a set of universe elements of size $|S| = d$ that are *shattered* by P, as guaranteed by the definition of VC-dimension. We will consider all

subsets $T \subset S$ of size $|T| = d/2$. Denote this set by $\mathcal{D}_S = \{T \subset S : |T| = d/2\}$ For each such $T \in \mathcal{D}_S$, let φ_T be the predicate such that:

$$\varphi_T(x) = \begin{cases} 1, & x \in T; \\ 0, & x \notin T. \end{cases}$$

as guaranteed by the definition of shattering, and let $Q_T = Q_{\varphi_T}$ be the corresponding counting query. Note that for $T \in \mathcal{D}_S$, we can treat Q_T as an element of C for the purpose of evaluation against databases $\subset S$, because there must exist some element of C that induces the same partition of S as Q_T does. In what follows, we restrict ourselves to databases consisting of elements of S, and so we adopt this convention.

We begin with a proof of "blatant non-privacy", like that shown by Dinur and Nissim [DN03].

LEMMA 3.11. *For any $0 < \delta < 1$, let M be an (α, δ)-useful mechanism for C. Given as input $M(T)$ where T is any database $T \in \mathcal{D}_S$, there is a procedure which with probability $1 - \delta$ reconstructs a database T' with $|T' \Delta T| \leq d\alpha$. $M(T)$ is not required to be synthetic data.*

PROOF. Write $D' = M(T)$. With probability at least $1 - \delta$, we have $\max_{Q \in C} |Q(T) - Q(D')| \leq \alpha$. Then with probability $1 - \delta$, the following reconstruction succeeds: return $T' = \text{argmax}_{T' \in \mathcal{D}_S} Q_{T'}(D')$. (That is, T' is the database in $\mathcal{D}_S$ that best matches $M(T)$.)

Note that the fraction of T reconstructed by T' is exactly $Q_{T'}(T) = Q_T(T')$. Thus,

$$\begin{aligned} Q_T(T') &= Q_{T'}(T) \\ &\geq Q_{T'}(D') - \alpha && \text{by } (\alpha, \delta)\text{-usefulness of } D' \\ &\geq Q_T(D') - \alpha && \text{by choice of } T' \text{ as best match for } D' \\ &\geq Q_T(T) - 2\alpha && \text{by } (\alpha, \delta)\text{-usefulness of } D' \\ &= 1 - 2\alpha, \end{aligned}$$

which completes the proof, since $|T| = |T'| = d/2$. □

We now explore the consequences this blatant non-privacy has for ϵ-differential privacy.

THEOREM 3.12. *For any class of counting queries C, for any $0 < \delta < 1$, if M is an ϵ-differentially private mechanism that is (α, δ) useful for C given databases of size $n \leq \frac{VCDIM(C)}{2}$, then $\alpha \geq \frac{1}{2(\exp(\epsilon)+1)}$.*

PROOF. Let $T \in \mathcal{D}_S$ be a set selected uniformly at random, $D' = M(T)$, and let T' be the set reconstructed from $D' = M(T)$ as in Lemma 3.11.

Select $x \in T$ uniformly at random, and $y \in S \setminus T$ uniformly at random. Let $\hat{T} = (T \setminus \{x\}) \cup \{y\}$ be the set obtained by swapping element x out and replacing it with y. Note that (x, y) are uniformly random over pairs of elements in S such that the first is in T and the second is not in T; similarly, (x, y) are uniformly random over pairs of elements in S such that the first is not in $\hat{T}$ and the second is in $\hat{T}$. Let $\hat{T}'$ be the set reconstructed from $D' = M(\hat{T})$.

Except with probability at most 2δ, we have the following properties of the reconstructions:

$$\begin{aligned}\Pr[x \in T' \text{ given input } T] &= \frac{|T| - (1/2)|T\Delta T'|}{|T|} \\ &\geq \frac{\frac{d}{2} - d\alpha}{\frac{d}{2}} \\ &= 1 - 2\alpha\end{aligned}$$

and

$$\begin{aligned}\Pr[x \in \hat{T}' \text{ given input } \hat{T}] &= \frac{(1/2)|\hat{T}\Delta\hat{T}'|}{|\hat{T}|} \\ &\leq \frac{d\alpha}{\frac{d}{2}} \\ &= 2\alpha\end{aligned}$$

Now recall that T and $\hat{T}$ are neighboring databases, with $|T\Delta\hat{T}| \leq 2$, and so by the fact that M is ϵ-differentially private, we also know:

$$\exp(\epsilon) \geq \frac{\Pr[x \in T' \text{ given input } T]}{\Pr[x \in \hat{T}' \text{ given input } \hat{T}]} \geq \frac{1 - 2\alpha}{2\alpha} = \frac{1}{2\alpha} - 1,$$

and so

$$\alpha \geq \frac{1}{2(\exp(\epsilon) + 1)},$$

as desired. □

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 databases that consist of distinct points, for clarity.

Consider a database D of n points in $\{1, \ldots, 2^d\}$ (in Corollary 5.2 we show some discretization is necessary). Given $a_1 \leq a_2$, both in $\{1, 2, \ldots, 2^d\}$, 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 $GS^n_{Q_{[a_1,a_2]}} = 1/n$, and we may answer interval queries while preserving ϵ-differential privacy by adding noise proportional to $\text{Lap}(1/(\epsilon n))$.

We now give the algorithm. Algorithm 2 repeatedly performs a binary search to partition the unit interval into regions that have approximately an α' fraction of the point mass in them. It then releases a database that has exactly an α'-fraction of the

ALGORITHM 2: ReleaseIntervals(D, α, ϵ)

let $\alpha' \leftarrow \alpha/6$, MaxIntervals $\leftarrow \lceil 4/3\alpha' \rceil$, $\epsilon' \leftarrow \epsilon/(d \cdot \text{MaxIntervals})$.
let Bounds be an array of length MaxIntervals
let $i \leftarrow 1$, Bounds$[0] \leftarrow 1$
while *Bounds*$[i-1] < 2^d$ **do**
  $a \leftarrow$ Bounds$[i-1]$, $b \leftarrow (2^d - a + 1)/2$, increment $\leftarrow (2^d - a + 1)/4$
  while *increment* ≥ 1 **do**
    let $\hat{v} \leftarrow Q_{[a,b]}(D) + \text{Lap}(1/(\epsilon' n))$
    if $\hat{v} > \alpha'$ **then let** $b \leftarrow b -$ increment
    else let $b \leftarrow b +$ increment
    let increment $\leftarrow$ increment$/2$
  let Bounds$[i] \leftarrow b$, $i \leftarrow i + 1$
Output D', a database that has $\alpha' m$ points in each interval [Bounds$[j-1]$, Bounds$[j]$] for each $j \in [i]$, for any $m > \frac{1}{\alpha'}$.

point mass in each of the intervals that it has discovered. There are at most $\approx 1/\alpha'$ such intervals, and each binary search terminates after at most d rounds (because the interval consists of at most 2^d points). Therefore, the algorithm requires only $\approx d/\alpha'$ accesses to the database, and each one is performed in a privacy preserving manner using noise from the Laplace mechanism. The privacy of the mechanism then follows immediately:

THEOREM 4.2. *ReleaseIntervals is ϵ-differentially private.*

PROOF. The algorithm runs a binary search at most $\lceil 4/3\alpha' \rceil$ times. Each time, the search halts after d queries to the database using the Laplace mechanism. Each query is ϵ'-differentially private (the sensitivity of an interval query is $1/n$ since it is a counting query). Privacy then follows from the definition of ϵ' and the fact that the composition of k differentially private mechanisms is $k\epsilon$ differentially private. □

THEOREM 4.3. *ReleaseIntervals is (α, δ)-useful for databases of size:*

$$n \geq \frac{288d}{\epsilon\alpha^3} \cdot \log\left(\frac{8d}{\delta\alpha}\right)$$

PROOF. By a union bound and the definition of the Laplace distribution, if the database size n satisfies the hypothesis of the theorem, then except with probability at most δ, none of the $(4/3)d/\alpha'$ draws from the Laplace distribution have magnitude greater than α'^2. That is, we have

$$\begin{aligned} \max |\hat{v} - Q_{[a,b]}(D)| &\leq \frac{\log(\frac{8d}{\alpha\delta})}{\epsilon' n} \\ &\leq \frac{8d \log(\frac{8d}{\alpha\delta})}{\epsilon\alpha n} \\ &\leq \alpha'^2 \end{aligned}$$

except with probability δ. Conditioned on this event occurring, for each interval [Bounds$[j-1]$,Bounds$[j]$] for $j \in [i]$, $f_{\text{Bounds}[j-1],\text{Bounds}[j]}(D) \in [\alpha' - \alpha'^2, \alpha' + \alpha'^2]$. In the synthetic database D' released, each such interval contains exactly an α' fraction of the database elements. We can now analyze the error incurred on any query when evaluated on the synthetic database instead of on the real database. Any interval [Bounds$[j-1]$,Bounds$[j]$] $\subset [a, b]$ will contribute error at most α' to the total, and any interval [Bounds$[j-1]$,Bounds$[j]$] $\not\subset [a, b]$ that also intersects with $[a, b]$ contributes

error at most $(\alpha' + \alpha'^2)$ to the total. Note that there are at most 2 intervals of this second type. Therefore, on any query $Q_{[a,b]}$ we have:

$$\begin{aligned}|Q_{[a,b]}(D') - Q_{[a,b]}(D)| &\leq \sum_{j:[\text{Bounds}[j-1],\text{Bounds}[j]]\cap[a,b]\neq\emptyset} |Q_{[\text{Bounds}[j-1],\text{Bounds}[j]]}(D) - Q_{[\text{Bounds}[j-1],\text{Bounds}[j]]}(D')| \\ &\leq \frac{4}{3\alpha'}\alpha'^2 + 2(\alpha' + \alpha'^2) \\ &\leq 6\alpha' \\ &= \alpha\end{aligned}$$

□

We note that although the class of intervals is simple, we are able to answer 2^{2d} queries over a universe of size 2^d, while needing a database of size only poly(d) and needing running time only poly(d).

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 $50-\delta, 50+\delta$ percentile of database points by performing a binary search on D using A (where $\delta = \delta(\alpha)$ is some small constant depending on the usefulness parameter α). 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.

We say that a mechanism answers a median query M usefully if it outputs a real value r such that r falls between the $50-\delta$ and $50+\delta$ percentiles of points in database D for some $\delta < 50$.

THEOREM 5.1. *No mechanism A can answer median queries M with outputs that fall between the $50-\delta, 50+\delta$ percentile with positive probability on any real valued database D, while still preserving ϵ-differential privacy, for $\delta < 50$ and any ϵ.*

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. Suppose A can answer median queries usefully. Then we must have $\Pr[A(D_0, M) = 0] > 0$ since every point in D_0 is 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 operating on continuous valued datasets can be (α,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 ϵ-differential privacy, for any $\alpha, \delta < 1/2$ and any $\epsilon \geq 0$.*

PROOF. Consider any real valued database containing elements in the interval $[0,1]$. If A is (α,δ)-useful for interval queries and preserves differential privacy, then

we can construct a mechanism A' that can answer median queries usefully while preserving differential privacy. By Theorem 5.1, this is impossible. A' simply computes $\widehat{D} = A(D)$, and performs binary search over queries on $\widehat{D}$ to find some interval $[0, a]$ that contains $n/2 \pm \alpha n$ points. Privacy is preserved since we only access D through A, which by assumption preserves ϵ-differential privacy. With positive probability, all interval queries on $\widehat{D}$ are correct to within $\pm\alpha$, and so the binary search can proceed. Since $\alpha < 1/2$, the result follows. □

Remark 5.3. We note that we could have replaced a "continuous" universe in our argument with a finely discretized universe. In this case, we would get a lower bound in which the accuracy would depend on the discretization parameter.

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.

6. ANSWERING HALFSPACE QUERIES

In this section, we give a non-interactive mechanism for releasing the answers to "large-margin halfspace" queries, defined over databases consisting of n unit vectors in $\mathbb{R}^d$. The mechanism we give here will be different from the other mechanisms we have given in two respects. First, although it is a non-interactive mechanism, it will not output synthetic data, but instead another data structure representing the answers to its queries. Second, it will not offer a utility guarantee for all halfspace queries, but only those that have "large margin" with respect to the private database. Large margin, which we define below, is a property that a halfspace has with respect to a particular database. Note that by our impossibility result in the previous section, we know that without a relaxation of our utility goal, no private useful mechanism is possible.

Definition 6.1 (*Halfspace Queries*). For a unit vector $y \in \mathbb{R}^d$, the *halfspace* query $f_y : \mathbb{R}^d \to \{0, 1\}$ is defined to be:

$$f_y(x) = \begin{cases} 1, & \text{If } \langle x, y\rangle > 0; \\ 0, & \text{Otherwise.} \end{cases}$$

With respect to a database, a halfspace can have a certain *margin* γ:

Definition 6.2 (*Margin*). A halfspace query f_y has margin γ with respect to a database $D \in (\mathbb{R}^d)^n$ if for all $x \in D$: $|\langle x, y\rangle| \geq \gamma$.

Before we present the algorithm, we will introduce a useful fact about random projections, called the Johnson-Lindenstrauss lemma. It states, roughly, that the norm of a vector is accurately preserved with high probability when the vector is projected into a lower dimensional space with a random linear projection.

THEOREM 6.3 (THE JOHNSON-LINDENSTRAUSS LEMMA [DG99; ACH03; BBV06]). *For $d > 0$ an integer and any $0 < \varsigma, \tau < 1/2$, let A be a $T \times d$ random matrix with $\pm 1/\sqrt{T}$ random entries, for $T \geq 20\varsigma^{-2}\log(1/\tau)$. Then for any $x \in \mathbb{R}^d$:*

$$\Pr_A[|||Ax||_2^2 - ||x||_2^2| \geq \varsigma||x||_2^2] \leq \tau$$

For our purposes, the relevant fact will be that norm preserving projections also preserve pairwise inner products with high probability. The following corollary is well known.

COROLLARY 6.4 (THE JOHNSON-LINDENSTRAUSS LEMMA FOR INNER PRODUCTS). *For $d > 0$ an integer and any $0 < \varsigma, \tau < 1/2$, let A be a $T \times d$ random matrix with*

$\pm 1/\sqrt{T}$ *random entries, for* $T \geq 20\varsigma^{-2}\log(1/\tau)$. *Then for any* $x \in \mathbb{R}^d$*:*

$$\Pr_A[|\langle (Ax),(Ay)\rangle - \langle x,y\rangle| \geq \frac{\varsigma}{2}(||x||_2^2 + ||y||_2^2)] \leq 2\tau$$

PROOF. Consider the two vectors $u = x+y$ and $v = x-y$. We apply Theorem 6.3 to u and v. By a union bound, except with probability 2τ we have: $|||A(x+y)||_2^2 - ||x+y||_2^2| \leq \varsigma||x+y||_2^2$ and $|||A(x-y)||_2^2 - ||x-y||_2^2| \leq \varsigma||x-y||_2^2$. Therefore:

$$\begin{aligned}
\langle (Ax),(Ay)\rangle &= \frac{1}{4}\left(\langle A(x+y),A(x+y)\rangle - \langle A(x-y),A(x-y)\rangle\right) \\
&= \frac{1}{4}\left(||A(x+y)||_2^2 - ||A(x-y)||_2^2\right) \\
&\leq \frac{1}{4}\left((1+\varsigma)||x+y||_2^2 - (1-\varsigma)||x-y||_2^2\right) \\
&= \langle x,y\rangle + \frac{\varsigma}{2}\left(||x||_2^2 + ||y||_2^2\right)
\end{aligned}$$

An identical calculation shows that $\langle (Ax),(Ay)\rangle \geq \langle x,y\rangle - \frac{\varsigma}{2}\left(||x||_2^2 + ||y||_2^2\right)$, which completes the proof. □

Instead of outputting synthetic data, our algorithm outputs a data structure based on a collection of random projections. The ReleaseHalfspaces algorithm selects m projection matrices $A_1 \ldots A_m$ to project the original database into a low dimensional space $\mathbb{R}^T$, as well as a collection of 'canonical' halfspaces U_T in T dimensions. ReleaseHalfspaces then computes these canonical halfspace queries on each projection of the original data, and releases noisy versions of the answers, along with $\{A_i\}$ and U_T.

More formally, the output of ReleaseHalfspaces is a Projected Halfspace Data Structure:

Definition 6.5 (*Projected Halfspace Data Structure*). A T dimensional projected halfspace data structure of size m, $D_H = \{\{A_i\}, U, \{v_{i,j}\}\}$ consists of three parts:

(1) m matrices $A_1, \ldots, A_m \in \mathbb{R}^{T\times d}$ mapping vectors from $\mathbb{R}^d$ to vectors in $\mathbb{R}^T$.
(2) A collection of T-dimensional unit vectors $U_T \subset \mathbb{R}^T$.
(3) For each $i \in [m]$ and $j \in U$, a real number $v_{i,j} \in \mathbb{R}$.

A projected halfspace data structure D_H can be used to evaluate a halfspace query f_y as follows. To denote the evaluation of a halfspace query on a projected halfspace data structure, we write $\text{Eval}(f_y, D_H)$. When the meaning is clear from context, we abuse notation and simply write $f_y(D_H)$ to denote this evaluation:
Eval(f_y, D_H):

(1) Compute y' by rounding the components of y to the nearest multiple of $\gamma/(8\sqrt{d})$ and projecting the resulting vector onto the nearest point on the d-dimensional unit ball.
(2) For $i \in [m]$, compute the projection $\hat{y'}_i \in \mathbb{R}^T$ as: $\hat{y'}_i = A_i \cdot y'$.
(3) For each $i \in [m]$ compute $u_{j(i)} = \text{argmin}_{u_j \in U_T} ||\hat{y'}_i - u_j||_2$
(4) Output $\frac{1}{m}\sum_{i=1}^m v_{i,j(i)}$

Definition 6.6. A γ-net for unit vectors in $\mathbb{R}^d$ is a set of points $U_d \subset \mathbb{R}^d$ such that for all $x \in \mathbb{R}^d$ with $||x||_2 = 1$:

$$\min_{y' \in U_d} ||x - y'||_2 \leq \gamma$$

The collection U_T of canonical halfspaces is selected to form a $\gamma/4$-net such that for every $y \in \mathbb{R}^T$ with $||y||_2 = 1$, there is some $u \in U_T$ such that $||y - u||_2 \leq \gamma/4$. The size of U_T will be exponential in T, but we choose T to be only a constant (in n and d—unfortunately not in γ and α), so that maintaining such a set is feasible. Each $v_{i,j}$ will represent the approximate answer to the query f_{u_j} on a projection of the private database by A_i. The Johnson-Lindenstrauss lemma will guarantee that not many points with large margin are shifted across the target halfspace by any particular projection, and the average of the approximate answers across all m projections will with high probability be accurate for every halfspace.

First we bound the size of the needed net U_T for halfspaces.

CLAIM 6.7. *There is a γ-net U_T for unit vectors in $\mathbb{R}^T$ of size $|U_T| \leq \left(\frac{2\sqrt{T}}{\gamma}\right)^T$, such that all elements of the γ-net are unit vectors.*

PROOF. First, construct U' by taking the space of all T-dimensional unit vectors and discretizing each coordinate to the nearest multiple of $\gamma/(2\sqrt{T})$ (the coordinates will remain between 0 and 1), and then transform it into U by projecting each point in U' to its nearest point on the unit ball. There are $\left(\frac{2\sqrt{T}}{\gamma}\right)^T$ such vectors.

For any unit $x \in \mathbb{R}^T$, let $y = \text{argmin}_{y \in U} ||x - y||_2$, $w = \text{argmin}_{w \in U'} ||x - w||_2$, and $z = \text{argmin}_{z \in U} ||w - z||_2$. Note that z is simply the projection of w onto the unit ball, and $||w - z||_2 \leq \sqrt{\sum_{i=1}^T (\gamma/2\sqrt{T})^2} = \gamma/2$. Similarly, $||w - x||_2 \leq \sqrt{\sum_{i=1}^T (\gamma/2\sqrt{T})^2} = \gamma/2$. Then by the triangle inequality, we have:

$$\begin{aligned} ||x - y||_2 &\leq ||x - z||_2 \\ &\leq ||x - w||_2 + ||w - z||_2 \\ &\leq \gamma. \end{aligned}$$

□

We can now present our algorithm.

ALGORITHM 3: ReleaseHalfspaces($D, d, \gamma, \alpha, \epsilon$)

let:

$$\varsigma \leftarrow \frac{\gamma}{4} \quad \tau \leftarrow \frac{\alpha}{8} \quad T \leftarrow \lceil 20\varsigma^{-2} \log(1/\tau) \rceil \quad m \leftarrow \frac{2}{\alpha^2}\left(d \log(8\sqrt{d}/\gamma) + \log(6/\beta)\right)$$

let U_T be a $\gamma/4$-net for unit vectors in $\mathbb{R}^T$.
for $i = 1$ *to* m **do**
  let $A_i \in \{-1/\sqrt{T}, 1/\sqrt{T}\}^{T \times d}$ be a uniformly random matrix for each $i \in [m]$.
  let $\hat{D}_i \subset \mathbb{R}^T$ be $\hat{D}_i = \{A_i x : x \in D\}$, followed by normalization to unit length of each point.
  for *each* $x_j \in U$ **do let** $p_{i,j} \leftarrow \text{Lap}\left(\frac{m|U|}{\epsilon n}\right)$, $v_{i,j} \leftarrow f_{x_j}(\hat{D}_i) + p_{i,j}$
Release $D_H = (\{A_i\}, U, U', \{v_{i,j}\})$.

THEOREM 6.8. *ReleaseHalfspaces preserves ϵ-differential privacy.*

PROOF. Privacy follows from the fact that the composition of k ϵ-differentially private mechanisms is $k\epsilon$-differentially private. The algorithm makes $m|U_T|$ calls to the Laplace mechanism, and each call preserves $\epsilon/(m|U_T|)$-differential privacy (since each query has sensitivity $1/n$). □

THEOREM 6.9. *Consider a database D of unit vectors in $\mathbb{R}^d$ with:*

$$n \geq \frac{m(8\sqrt{T}/\gamma)^T}{\epsilon} \log\left(\frac{2m(8\sqrt{T}/\gamma)^T}{\beta}\right)$$

for $m = \frac{2}{\alpha^2}\left(d\log(8\sqrt{d}/\gamma) + \log(6/\beta)\right)$. Then except with probability at most β, $D_H = $ ReleaseHalfSpaces$(D, d, \gamma, \alpha, \epsilon)$ is such that for each unit vector $y \in \mathbb{R}^d$ with margin γ with respect to D: $|f_y(D) - f_y(D_H)| \leq \alpha$. The running time of the algorithm and the bound on the size of D are both polynomial for $\gamma, \alpha \in \Omega(1)$.

PROOF. The high-level idea of the proof is to argue that the algorithm consists of a sequence of weakenings or approximations of the true halfspace queries, and that with high probability all of these approximations are good.

The initial rounding step in the evaluation of a halfspace query against a projected halfspace data structure serves to discretize the set of halfspaces, to allow us to apply a union bound later in the proof. Essentially, we implicitly introduce a $\gamma/4$-net U_d on $\mathbb{R}^d$. Consider any $y \in \mathbb{R}^d$ such that f_y has margin γ with respect to D, and let $y' = \text{argmin}_{y' \in U_d} ||y - y'||_2$. Note that $f_{y'}$ has margin at least $\frac{3}{4}\gamma$ with respect to D and thus $f_y(D) = f_{y'}(D)$. Thus, in the remainder of the proof, we will consider halfspace queries corresponding to the elements of U_d. If our algorithm can maintain accuracy for these halfspaces, it will also maintain accuracy for all large margin halfspaces.

We first argue that with high probability, the value of a halfspace query $y' \in U_d$ on a point $x \in D$ is not changed substantially by projecting both x and y' into T-dimensional space before evaluating it. By Corollary 6.4, for each $i \in [m]$ and each $x \in D$, given the values of ς, τ, and T used in Algorithm 3,

$$\Pr_{A_i}\left[|\langle (A_i x), (A_i y')\rangle - \langle x, y'\rangle| \geq \frac{\varsigma}{2}(||x||_2^2 + ||y'||_2^2)\right] =$$
$$\Pr_{A_i}\left[|\langle (A_i x), (A_i y')\rangle - \langle x, y'\rangle| \geq \frac{\gamma}{8}(1+1)\right] =$$
$$\Pr_{A_i}\left[|\langle (A_i x), (A_i y')\rangle - \langle x, y'\rangle| \geq \frac{\gamma}{4}\right] \leq \alpha/4$$

By linearity of expectation, the expected number of points in D moved by more than $\gamma/4$ with respect to some y' in a given projection A_i is at most $\alpha n/4$. Recall that each y' has margin at least $\frac{3}{4}$ with respect to D, and so the expected number of points in D such that the projected halfspace and the original halfspace y' agree and the evaluation of the query in the projected space still has substantial margin ($\gamma/2$) is

$$E\left[\left|\left\{x \in D : (f_y(x) = f_{A_i y}(A_i x)) \wedge \left(|\langle A_i x, A_i y\rangle| \geq \frac{1}{2}\gamma\right)\right\}\right|\right] \geq n\left(1 - \frac{\alpha}{4}\right).$$

Next, we see that a projected halfspace query is always well approximated by the resulting closest net point in U_T, with respect to its answer on any unit vector $\hat{x} \in \mathbb{R}^T$. Recall U_T is a $\gamma/4$-net for unit vectors in $\mathbb{R}^T$. Consider the net point closest to the projection of y' under A_i, $u_{i,y'} = \text{argmin}_{u \in U} ||u - A_i y'||_2$. By the property of the net, $||u_{i,y'} - A_i y'||_2 \leq \gamma/4$, so

$$\begin{aligned}|\langle A_i y', \hat{x}\rangle| &= |\langle u_{i,y'}, \hat{x}\rangle + \langle A_i y' - u_{i,y'}, \hat{x}\rangle| \\ &\leq |\langle u_{i,y'}, \hat{x}\rangle| + ||A_i y' - u_{i,y'}||_2 ||x||_2 \\ &\leq \langle u_{i,y'}, \hat{x}\rangle + \gamma/4.\end{aligned}$$

We can combine these facts to see that the number of points on which the evaluation of the nearest canonical vector in the low-dimensional space agrees with the original

vector y' is:

$$E\left[\left|\left\{x \in D : \left(f_{y'}(x) = f_{u_{i,y'}}(A_i x)\right) \wedge \left(|\langle A_i x, u_{i,y'}\rangle| \geq \frac{1}{4}\gamma\right)\right\}\right|\right] \geq n\left(1 - \frac{\alpha}{4}\right).$$

In other words, $f_{y'}(D) - \alpha/4 \leq E[f_{u_{i,y'}}(\hat{D}_i)] \leq f_{y'}(D) + \alpha/4$.

There are three possible reasons the projected halfspace data structure might not provide accurate answers for all large-margin halfspaces, and we bound the probability of each failure mode by $\beta/3$:

(1) *There is a halfspace query y' such that the average value (over the m projections) of its canonical halfspace query on the projections of the database is far from the true value of query y' on the true database D.*
Note that for each i, $f_{u_{i,y'}}(\hat{D}_i)$ is an independent random variable taking values in the bounded range $[0,1]$, and so we are able to apply a Chernoff bound. For each y':

$$\Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} f_{u_{i,y'}}(\hat{D}_i) - E[f_{u_{y'}}(\hat{D})]\right| \geq \frac{\alpha}{2}\right] \leq 2\exp\left(-\frac{m\alpha^2}{2}\right)$$

Taking a union bound over all $(8\sqrt{d}/\gamma)^d$ vectors $y' \in U_d$ in the implicit high-dimensional net, plugging in our chosen value for the number of samples m, and recalling our bound on $E[f_{u_{y'}}(\hat{D})]$ we find that:

$$\Pr\left[\max_{y' \in U_d}\left|\frac{1}{m}\sum_{i=1}^{m} f_{u_{i,y'}}(\hat{D}_i) - f_{y'}(D)\right| \geq \frac{3\alpha}{4}\right] \leq \frac{\beta}{3}$$

(2) *Any one of the $|p_{i,j}|$ is very large.*
The algorithm makes $m|U_T|$ draws from the distribution $\text{Lap}\left(\frac{m|U_T|}{\epsilon n}\right)$ during its run, assigning these draws to values $p_{i,j}$. Except with probability at most $\beta/3$, we have for all i, j:

$$|p_{i,j}| \leq \frac{m|U_T|}{\epsilon n}\log\left(\frac{2m|U_T|}{\beta}\right) \leq 1,$$

plugging in the value of n from the theorem statement.

(3) *Even though all of the $|p_{i,j}|$ are less than 1, there exists a sequence $j(1), \ldots, j(m)$ that could be summed as a result of computing $f_y(D_H) = \frac{1}{m}\sum_{i=1}^{m} v_{i,j(i)}$, such that the average contribution of the noise is very large.*
Conditioning on $|p_{i,j}| \leq 1$ for all i, j and applying another Chernoff bound, we find that for any sequence of indices $j(i)$:

$$\Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} p_{i,j(i)}\right| \geq \alpha/4\right] \leq 2\exp\left(-\frac{m\alpha^2}{8}\right)$$

Again taking a union bound over $y' \in U_d$ and plugging in our value of the number of samples m, we find that:

$$\Pr\left[\max_{j(1),\ldots,j(m)}\left|\frac{1}{m}\sum_{i=1}^{m} p_{i,j(i)}\right| \geq \alpha/4\right] \leq \frac{\beta}{3}$$

Assuming we are in the $1-\beta$ probability situation when none of the three failure modes occur, we have for any y with margin γ with respect to D, for the corresponding

y':

$$\begin{aligned} f_{y'}(D_H) &= \frac{1}{m}\sum_{i=1}^{m} v_{i,j(i)} \\ &= \frac{1}{m}\left(\sum_{i=1}^{m} f_{u_{i,y'}}(\hat{D}_i) + \sum_{i=1}^{m} p_{i,j(i)}\right) \\ &\leq \frac{1}{m}\left(\sum_{i=1}^{m} f_{u_{i,y'}}(\hat{D}_i)\right) + \alpha/4 \\ &\leq f_{y'}(D) + \alpha \\ &= f_y(D) + \alpha, \end{aligned}$$

which completes the proof. □

7. CONCLUSIONS AND OPEN PROBLEMS

In this paper we have shown a very general information theoretic result: that small nets are sufficient to certify the existence of accurate, differentially private mechanisms for a class of queries. For counting queries, this allows algorithms which can accurately answer queries from a class C given a database that is only *logarithmic* in the size of C, or linear its VC-dimension. We then also gave an efficient algorithm for releasing the class of interval queries on a discrete interval, and for releasing large-margin halfspace queries in the unit sphere.

The main question left open by our work is the design of algorithms which achieve utility guarantees comparable to our Net mechanism, but have running time only polynomial in n, the size of the input database. This question is extremely interesting even for very specific classes of queries. Is there such a mechanism for the class of conjunctions? For the class of parity queries?

8. ACKNOWLEDGMENTS

We thank David Abraham, Cynthia Dwork, Shiva Kasiviswanathan, Adam Meyerson, Ryan O'Donnell, Sofya Raskhodnikova, Amit Sahai, and Adam Smith for many useful discussions.

REFERENCES

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

D. Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4):671–687, 2003.

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

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

A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 609–618. ACM, 2008.

A. De. Lower bounds in differential privacy. *Arxiv preprint arXiv:1107.2183*, 2011.

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

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: privacy via distributed noise generation. In *Advances in cryptology—EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Comput. Sci.*, pages 486–503. Springer, Berlin, 2006.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Theory of Cryptography Conference TCC*, volume 3876 of *Lecture Notes in Computer Science*, page 265. Springer, 2006.

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.

I. Dinur and K. Nissim. Revealing information while preserving privacy. In *22nd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 202–210, 2003.

C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Proceedings of CRYPTO*, Lecture Notes in Computer Science, pages 528–544. Springer, 2004.

C. Dwork, M. Naor, O. Reingold, G.N. Rothblum, and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st annual ACM Symposium on the Theory of computing*, pages 381–390. ACM New York, NY, USA, 2009.

C. Dwork, G.N. Rothblum, and S. Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annua IEEEl Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

C. Dwork. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, volume 4052 of *LECTURE NOTES IN COMPUTER SCIENCE*, page 1. Springer, 2006.

C. Dwork and S. Yekhanin. New efficient attacks on statistical disclosure control mechanisms. *Advances in Cryptology–CRYPTO 2008*, pages 469–480, 2008.

A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately Releasing Conjunctions and the Statistical Query Barrier. In *Proceedings of the 43rd annual ACM Symposium on the Theory of Computing*. ACM New York, NY, USA, 2011.

A. Gupta, A. Roth, and J. Ullman. Iterative constructions and private data release. *Arxiv preprint arXiv:1107.3731*, 2011.

M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. *Arxiv preprint arXiv:1012.4763*, 2012.

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

Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially-private queries through consistency. In *VLDB*, 2010.

M. Hardt, G.N. Rothblum, and R.A. Servedio. Private data release via learning thresholds. *Arxiv preprint arXiv:1107.2444*, 2011.

M. Hardt and K. Talwar. On the Geometry of Differential Privacy. In *The 42nd ACM Symposium on the Theory of Computing, 2010. STOC'10*, 2010.

S.P. Kasiviswanathan, H.K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What Can We Learn Privately? In *IEEE 49th Annual IEEE Symposium on Foundations of Computer Science, 2008. FOCS'08*, pages 531–540, 2008.

Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proc. 42nd STOC*. ACM, 2010.

C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor. Optimizing linear counting queries under differential privacy. In *PODS*, 2010.

Chao Li and Gerome Miklau. Efficient batch query answering under differential privacy. *CoRR*, abs/1103.1367, 2011.

Chao Li and Gerome Miklau. An adaptive mechanism for accurate query answering under differential privacy. *to appear, PVLDB*, 2012.

Chao Li and Gerome Miklau. Measuring the achievable error of query sets under differential privacy. *CoRR*, abs/1202.3399v2, 2012.

S. Muthukrishnan and Aleksandar Nikolov. Optimal private halfspace counting via discrepancy. In *STOC*, pages 1285–1292, 2012.

F. McSherry and K. Talwar. Mechanism Design via Differential Privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103, 2007.

A. Roth. Differential Privacy and the Fat Shattering Dimension of Linear Queries. In *Proceedings of the fourteenth annual workshop on randomization and computation (RANDOM 2010)*, 2010.

A. Roth and T. Roughgarden. Interactive Privacy via the Median Mechanism. In *The 42nd ACM Symposium on the Theory of Computing, 2010. STOC'10*, 2010.

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

Jonathan Ullman and Salil P. Vadhan. PCPs and the hardness of generating private synthetic data. In *TCC*, pages 400–416, 2011.

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

X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering*, pages 1200–1214, 2010.

A. RELEASING COUNTING QUERIES WHILE KEEPING n PRIVATE

In this section, we briefly sketch the adaptation of our mechanism for releasing data useful for counting queries to the private-n version of differential privacy. The technique for converting between private n and public n versions of differential privacy is standard.

For the private n version of differential privacy, we think of datasets $D \in X^*$, which can be multisets of any cardinality. Two datasets $D, D' \in X^*$ are now said to be neighboring if one can be obtained from the other by adding or removing a single element: i.e. D and D' are *neighbors* if $|D \triangle D'| \leq 1$. Private n differential privacy is identical to public n differential privacy, except that it uses this slightly modified definition of neighbors.

Definition A.1 (*Differential Privacy (private n version)*). A database access mechanism $A : X^* \to R$ is ϵ-differentially private if for all neighboring pairs of databases $D, D' \in X^*$ and for all outcome events $S \subseteq R$, the following holds:

$$\Pr[A(D) \in S] \leq \exp(\epsilon) \Pr[A(D') \in S]$$

When the size of the database n is private, it is more natural to discuss *unnormalized* counting queries.

Definition A.2. An unnormalized *counting query* Q_φ, defined in terms of a predicate $\varphi : X \to \{0, 1\}$ is defined to be

$$Q_\varphi(D) = \sum_{x \in D} \varphi(x).$$

It evaluates to the number of elements in the database that satisfy the predicate φ.

Note that the global sensitivity of an unnormalized counting query is 1, independent of the size of the database n. This allows us to apply techniques such as the Laplace mechanism and the Exponential mechanism without knowledge of n. Recall that instantiated for counting queries, the Net mechanism outputs a database of smaller cardinality than the private database D. When we were working with normalized queries, this did not matter: queries were evaluated at the same scale on all databases. When we are working with un-normalized queries, we must rescale the answers computed on a small database if we wish to interpret them as approximating their value on a larger database. Towards this end, suppose $D' \in X^*$ is a database of size $|D'| = m$. For fixed n', we write: $D'_{m,n'}$ to denote that answers computed on database D' should be rescaled to the range $[0, n']$. That is, given a counting query Q_φ, define:

$$Q_\varphi(D'_{m,n'}) = \frac{n'}{m} \sum_{x \in D'} \varphi(x)$$

Note that because $n = |D|$ must remain private, we will use a private estimate n' of n, rather than n itself when defining the rescaling factor for our output.

We can now give the private n version of the Net mechanism, adapted to counting queries.

THEOREM A.3. *PrivateNRelease preserves ϵ-differential privacy in the private-n model.*

ALGORITHM 4: PrivateNRelease(D, C, ϵ, α)

let $\hat{n} = |D| + \text{Lap}(2/\epsilon)$.
let $m \leftarrow \log|C|/\alpha^2$
let $\mathcal{R} \leftarrow \{D' \in X^* : |D'| = m\}$
let $q : X^n \times \mathcal{R} \to \mathbb{R}$ be defined to be:

$$q(D, D') = -\max_{Q \in C} \left|Q(D) - Q(D'_{m,\hat{n}})\right|$$

Sample $D' \in \mathcal{R}$ with the exponential mechanism $M_E(D, q, \mathcal{R}, \epsilon/2)$
Output $D'_{m,\hat{n}}$.

PROOF. We access the database only twice: once using the Laplace mechanism of [DMNS06], which is $\epsilon/2$-differentially private, and once using the exponential mechanism of [MT07], which is $\epsilon/2$-differentially private. Therefore, the mechanism in total is ϵ-differentially private by the privacy composition theorem of [DKM$^+$06]. □

THEOREM A.4. *With probability $1 - \delta$, the private n release mechanism outputs a database $D'_{m,\hat{n}}$ such that for all $Q \in C$: $|Q(D'_{m,\hat{n}}) - Q(D)| \leq \alpha n$ whenever:*

$$\alpha \geq \left(\frac{8 \log|C| \log|X|}{\epsilon n} + \frac{4}{\epsilon} \ln\left(\frac{2}{\delta}\right)\right)^{1/3}$$

where a and b are absolute constants.

PROOF. The proof is largely the same as for the public n version of the Net mechanism. Let $n = |D|$. First, by the properties of the Laplace distribution, we have that with probability $1 - \delta/2$, $|\hat{n} - n| \leq \frac{2\ln(2/\delta)}{\epsilon}$. For the rest of the argument, we condition on this event occurring. We also have by Lemma 3.7 that for all D, there exists a database $D' \in \mathcal{R}$ such that $|\frac{f(D')}{m} - \frac{f(D)}{n}| \leq \alpha$ (recall that our queries are now unnormalized). In other words:

$$\left|\frac{n}{m} f(D') - f(D)\right| \leq \alpha n$$

Combining these two facts, we have:

$$\left|\frac{\hat{n}}{m} f(D') - f(D)\right| \leq \alpha n + \frac{4 \ln 2/\delta}{\epsilon}$$

In other words, we have that $\mathcal{R}$ is an $\alpha'/2 \equiv \left(\alpha n + \frac{4\ln 2/\delta}{\epsilon}\right)/2$-net for C. We may therefore reason analogously to Proposition 3.4.

By the definition of an $\alpha'/2$-net, we know that there exists some $D^* \in \mathcal{R}$ such that $q(D, D^*) \geq -\alpha'/2$. By the definition of the exponential mechanism, this D^* is output with probability proportional to at least $\exp(\frac{-\epsilon\alpha'}{8})$. Similarly, there are at most $|X|^{\log|C|/\alpha^2}$ databases $D' \in \mathcal{R}$ such that $q(D, D') \leq -\alpha'$. Hence, by a union bound, the probability that the exponential mechanism outputs some D' with $q(D, D') \leq -\alpha'$ is proportional to at most $|X|^{\log|C|/\alpha^2} \exp(\frac{-\epsilon\alpha'}{4})$. Therefore, if we denote by A the event that the Net mechanism outputs some D^* with $q(D, D^*) \geq -\alpha'/2$, and denote by B the

event that the Net mechanism outputs some D' with $q(D, D') \leq -\alpha'$, we have:

$$\frac{\Pr[A]}{\Pr[B]} \geq \frac{\exp(\frac{-\epsilon\alpha'}{8})}{|X|^{\log|C|/\alpha^2}\exp(\frac{-\epsilon\alpha'}{4})} = \frac{\exp(\frac{\epsilon\alpha'}{8})}{|X|^{\log|C|/\alpha^2}}$$

Note that if this ratio is at least $2/\delta$, then we will have proven that the Net mechanism is (α', δ) useful with respect to C.

Recalling that $\alpha' = \alpha n + 4\ln(2/\delta)/\epsilon$, we have that this inequality holds whenever:

$$\frac{\epsilon}{8}\alpha n + \frac{\ln(2/\delta)}{2} \geq \frac{\log|C|\log|C|}{\alpha^2} + \ln(2/\delta)$$

Solving for α, we find that this is the case whenever:

$$\alpha \geq \left(\frac{8\log|C|\log|X|}{\epsilon n} + \frac{4}{\epsilon}\ln\left(\frac{2}{\delta}\right)\right)^{1/3}$$

Finally, we remark that whenever $n = \Omega(\ln(1/\delta)/\epsilon)$ (a necessity for the above bound to be nontrivial), the optimal value of α can be approximated within a small constant factor by using $\hat{n}$ instead of n. $\square$