Interactive Privacy via the Median Mechanism

Aaron Roth[*]
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15217
alroth@cs.cmu.edu

Tim Roughgarden[†]
Stanford University
353 Serra Mall
Stanford, CA 94305
tim@cs.stanford.edu

ABSTRACT

We define a new interactive differentially private mechanism — the *median mechanism* — for answering arbitrary predicate queries that arrive online. Given fixed accuracy and privacy constraints, this mechanism can answer exponentially more queries than the previously best known interactive privacy mechanism (the Laplace mechanism, which independently perturbs each query result). With respect to the number of queries, our guarantee is close to the best possible, even for non-interactive privacy mechanisms. Conceptually, the median mechanism is the first privacy mechanism capable of identifying and exploiting correlations among queries in an interactive setting.

We also give an efficient implementation of the median mechanism, with running time polynomial in the number of queries, the database size, and the domain size. This efficient implementation guarantees privacy for all input databases, and accurate query results for almost all input distributions. The dependence of the privacy on the number of queries in this mechanism improves over that of the best previously known efficient mechanism by a super-polynomial factor, even in the non-interactive setting.

Categories and Subject Descriptors

F.2 [**ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY**]: Miscellaneous

General Terms

Theory, Algorithms

1. INTRODUCTION

Managing a data set with sensitive but useful information, such as medical records, requires reconciling two objectives: providing utility to others, perhaps in the form of aggregate statistics; and respecting the privacy of individuals who contribute to the data set. The field of *private data analysis*, and in particular work on *differential privacy*, provides a mathematical foundation for reasoning about this utility-privacy trade-off and offers methods for non-trivial data analysis that are provably privacy-preserving in a precise sense. For a recent survey of the field, see Dwork [Dwo08].

More precisely, consider a domain X and database size n. A *mechanism* is a randomized function from the set X^n of databases to some range. For a parameter $\alpha > 0$, a mechanism M is *α-differentially private* if, for every database D and fixed subset S of the range of M, changing a single component of D changes the probability that M outputs something in S by at most an e^α factor. The output of a differentially private mechanism (and any analysis or privacy attack that follows) is thus essentially independent of whether or not a given individual "opts in" or "opts out" of the database.

Achieving differential privacy requires "sufficiently noisy" answers [DN03]. For example, suppose we're interested in the result of a query f — a function from databases to some range — that simply counts the fraction of database elements that satisfy some predicate φ on X. A special case of a result in Dwork et al. [DMNS06] asserts that the following mechanism is α-differentially private: if the underlying database is D, output $f(D)+\Delta$, where the output perturbation Δ is drawn from the Laplace distribution $\text{Lap}(\frac{1}{n\alpha})$ with density $p(y) = \frac{n\alpha}{2}\exp(-n\alpha|y|)$. Among all α-differentially private mechanisms, this one (or rather, a discretized analog of it) maximizes user utility in a strong sense [GRS09].

What if we care about more than a single one-dimensional statistic? Suppose we're interested in k predicate queries $f_1, \ldots, f_k$, where k could be large, even super-polynomial in n. A natural solution is to use an independent Laplace perturbation for each query answer [DMNS06]. To maintain α-differential privacy, the magnitude of noise has to scale linearly with k, with each perturbation drawn from $\text{Lap}(\frac{k}{n\alpha})$. Put another way, suppose one fixes "usefulness parameters" ϵ, δ, and insists that the mechanism is *(ϵ, δ)-useful*, meaning that the outputs are within ϵ of the correct query answers with probability at least $1 - \delta$. This constrains the magnitude of the Laplace noise, and the privacy parameter α now suffers linearly with the number k of answered queries. This

[*]Supported in part by an NSF Graduate Research Fellowship. Portions of this work were done while visiting Stanford University.

[†]Supported in part by NSF CAREER Award CCF-0448664, an ONR Young Investigator Award, an ONR PECASE Award, an AFOSR MURI grant, and an Alfred P. Sloan Fellowship.

STOC'10, June 5–8, 2010, Cambridge, Massachusetts, USA.

dependence limits the use of this mechanism to a sublinear $k = o(n)$ number of queries.

Can we do better than independent output perturbations? For special classes of queries like predicate queries, Blum, Ligett, and Roth [BLR08] give an affirmative answer (building on techniques of Kasiviswanathan et al. [KLN$^+$08]). Specifically, in [BLR08] the exponential mechanism of McSherry and Talwar [MT07] is used to show that, for fixed usefulness parameters ϵ, δ, the privacy parameter α only has to scale *logarithmically* with the number of queries.[1] This permits simultaneous non-trivial utility and privacy guarantees even for an exponential number of queries. Moreover, this dependence on $\log k$ is necessary in every differentially private mechanism (see the full version of [BLR08]).

The mechanism in [BLR08] suffers from two drawbacks, however. First, it is *non-interactive*: it requires all queries $f_1, \ldots, f_k$ to be given up front, and computes (noisy) outputs of all of them at once.[2] By contrast, independent Laplace output perturbations can obviously be implemented interactively, with the queries arriving online and each answered immediately. There is good intuition for why the non-interactive setting helps: outperforming independent output perturbations requires correlating perturbations across multiple queries, and this is clearly easier when the queries are known in advance. Indeed, prior to the present work, no interactive mechanism better than independent Laplace perturbations was known.

Second, the mechanism in [BLR08] is *inefficient*. Here by "efficient" we mean has running time polynomial in n, k, and $|X|$; Dwork et al. [DNR$^+$09] prove that this is essentially the best one could hope for (under certain cryptographic assumptions). The mechanism in [BLR08] is not efficient because it requires sampling from a non-trivial probability distribution over an unstructured space of exponential size. Dwork et al. [DNR$^+$09] recently gave an efficient (non-interactive) mechanism that is better than independent Laplace perturbations, in that the privacy parameter α of the mechanism scales as $2^{\sqrt{\log k}}$ with the number of queries k (for fixed usefulness parameters ϵ, δ).

Very recently, Hardt and Talwar [HT10] gave upper and lower bounds for answering noninteractive linear queries which are tight in a related setting. These bounds are not tight in our setting, however, unless the number of queries is small with respect to the size of the database. When the number of queries is large, our mechanism actually yields error significantly less than required in general by their lower bound [3]. This is not a contradiction, because when translated into the setting of [HT10], our database size n becomes a sparsity parameter that is not considered in their bounds.

1.1 Our Results

We define a new interactive differentially private mechanism for answering k arbitrary predicate queries, called the *median mechanism.*[4] The basic implementation of the median mechanism interactively answers queries $f_1, \ldots, f_k$ that arrive online, is (ϵ, δ)-useful, and has privacy α that scales with $\log k \log |X|$; see Theorem 4.1 for the exact statement. These privacy and utility guarantees hold even if an adversary can adaptively choose each f_i after seeing the mechanism's first $i - 1$ answers. *This is the first interactive mechanism better than the Laplace mechanism, and its performance is close to the best possible even in the non-interactive setting.*

The basic implementation of the median mechanism is not efficient, and we give an efficient implementation with a somewhat weaker utility guarantee. (The privacy guarantee is as strong as in the basic implementation.) This alternative implementation runs in time polynomial in n, k, and $|X|$, and satisfies the following (Theorem 5.1): for every sequence $f_1, \ldots, f_k$ of predicate queries, for all but a negligible fraction of input distributions, the efficient median mechanism is (ϵ, δ)-useful.

This is the first efficient mechanism with a non-trivial utility guarantee and polylogarithmic privacy cost, even in the non-interactive setting.

1.2 The Main Ideas

The key challenge to designing an interactive mechanism that outperforms the Laplace mechanism lies in determining the appropriate correlations between different output perturbations on the fly, without knowledge of future queries. It is not obvious that anything significantly better than independent perturbations is possible in the interactive setting.

Our median mechanism and our analysis of it can be summarized, at a high level, by three facts. First, among any set of k queries, we prove that there are $O(\log k \log |X|)$ "hard" queries, the answers to which completely determine the answers to all of the other queries (up to $\pm\epsilon$). Roughly, this holds because: (i) by a VC dimension argument, we can focus on databases over X of size only $O(\log k)$; and (ii) every time we answer a "hard" query, the number of databases consistent with the mechanism's answers shrinks by a constant factor, and this number cannot drop below 1 (because of the true input database). Second, we design a method to privately release an indicator vector which distinguishes between hard and easy queries online. We note that a similar private 'indicator vector' technique was used by Dwork et al. [DNR$^+$09]. Essentially, the median mechanism deems a query "easy" if a majority of the databases that are consistent (up to $\pm\epsilon$) with the previous answers of the mechanism would answer the current query accurately. The median mechanism answers the small number of hard queries using independent Laplace perturbations. It answers an easy query (accurately) using the median query result given by databases that are consistent with previous answers. A key intuition is that if a user knows that query i is easy, *then it can generate the mechanism's answer on its own.* Thus answering an easy query communicates only a single new bit of information: that the query is easy. Finally, we show how to release the classification of queries as "easy" and "hard" with

[1] More generally, linearly with the VC dimension of the set of queries, which is always at most $\log_2 k$.

[2] Or rather, it computes a compact representation of these outputs in the form of a synthetic database.

[3] We give a mechanism for answering k "counting queries" with coordinate-wise error $O(n^{2/3} \log k (\log |X|/\alpha)^{1/3})$. This is less error than required by their *lower* bound of roughly $\Omega(\sqrt{k \log(|X|/k)}/\alpha)$ unless $k \leq \tilde{O}((n\alpha/\log |X|)^{4/3})$. We can take k to be as large as $k = \tilde{\Omega}(2^{(n\alpha/\log |X|)^{1/3}})$, in which case our upper bound is a significant improvement – as are the upper bounds of [BLR08] and [DNR$^+$09].

[4] The privacy guarantee is (α, τ)-differential privacy for a negligible function τ; see Section 2 for definitions.

low privacy cost; intuitively, this is possible because (independent of the database) there can be only $O(\log k \log |X|)$ hard queries.

Our basic implementation of the median mechanism is not efficient for the same reasons as for the mechanism in [BLR08]: it requires non-trivial sampling from a set of super-polynomial size. For our efficient implementation, we pass to *fractional* databases, represented as fractional histograms with components indexed by X. Here, we use the random walk technology of Dyer, Frieze, and Kannan [DFK91] for convex bodies to perform efficient random sampling. To explain why our utility guarantee no longer holds for every input database, recall the first fact used in the basic implementation: every answer to a hard query shrinks the number of consistent databases by a constant factor, and this number starts at $|X|^{O(\log k)}$ and cannot drop below 1. With fractional databases (where polytope volumes play the role of set sizes), the lower bound of 1 on the set of consistent (fractional) databases no longer holds. Nonetheless, we prove a lower bound on the volume of this set for almost all fractional histograms (equivalently probability distributions), which salvages the $O(\log k \log |X|)$ bound on hard queries for databases drawn from such distributions.

2. PRELIMINARIES

We briefly formalize the setting of the previous section and record some important definitions. We consider some finite domain X, and define a database D to be an unordered set of elements from X (with multiplicities allowed). We write $n = |D|$ to denote the size of the database. We consider the set of Boolean functions (predicates) $f : X \to \{0, 1\}$. We abuse notation and define a predicate query $f(D) : X^* \to [0, 1]$ as $|\{x \in D : f(X) = 1\}|/|D|$, the function that computes the fraction of elements of D that satisfy predicate f. We say that an answer a_i to a query f_i is ϵ-*accurate* with respect to database D if $|f_i(D) - a_i| \leq \epsilon$. A mechanism $M(D, (f_1, \ldots, f_k))$ is a function from databases and queries to distributions over outputs. In this paper, we consider mechanisms that answer predicate queries numerically, and so the range of our mechanisms is $\mathbb{R}^k$.[5]

Definition 1. A mechanism M is (ϵ, δ)-*useful* if for every sequence of queries $(f_1, \ldots, f_k)$ and every database D, with probability at least $1 - \delta$ it provides answers $a_1, \ldots, a_k$ that are ϵ-accurate for $f_1, \ldots, f_k$ and D.

Recall that differential privacy means that changing the identity of a single element of the input database does not affect the probability of any outcome by more than a small factor. Formally, given a database D, we say that a database D' of the same size is a *neighbor* of D if it differs in only a single element: $|D \cap D'| = |D| - 1$.

Definition 2. A mechanism M satisfies (α, τ)-*differential privacy* if for every subset $S \subseteq \mathbb{R}^k$, every set of queries $(f_1, \ldots, f_k)$, and every pair of neighboring databases D, D':

$$\Pr[M(D) \in S] \leq e^{\alpha} \cdot \Pr[M(D') \in S] + \tau.$$

We are generally interested in the case where τ is a *negligible function* of some of the problem parameters, meaning one that goes to zero faster than x^{-c} for every constant c.

Finally, the *sensitivity* of a real-valued query is the largest difference between its values on neighboring databases. For example, the sensitivity of every non-trivial predicate query is precisely $1/n$.

3. THE MEDIAN MECHANISM: BASIC IMPLEMENTATION

We now describe the median mechanism and our basic implementation of it. As described in the Introduction, the mechanism is conceptually simple. It classifies queries as "easy" or "hard", essentially according to whether or not a majority of the databases consistent with previous answers to hard queries would give an accurate answer to it (in which case the user already "knows the answer"). Easy queries are answered using the corresponding median value; hard queries are answered as in the Laplace mechanism.

To explain the mechanism precisely, we need to discuss a number of parameters. We take the privacy parameter α, the accuracy parameter ϵ, and the number k of queries as input; these are hard constraints on the performance of our mechanism.[6] Our mechanism obeys these constraints with a value of δ that is inverse polynomial in k and n, and a value of τ that is negligible in k and n, provided n is sufficiently large (at least polylogarithmic in k and $|X|$, see Theorem 4.1). Of course, such a result can be rephrased as a nearly exponential lower bound on the number of queries k that can be successfully answered as a function of the database size n.[7]

The median mechanism is shown in Figure 1, and it makes use of several additional parameters. For our analysis, we set their values to:

$$m = \frac{160000 \ln k \ln \frac{1}{\epsilon}}{\epsilon^2}; \tag{1}$$

$$\alpha' = \frac{\alpha}{720 m \ln |X|} = \Theta\left(\frac{\alpha\epsilon^2}{\log |X| \log k \log \frac{1}{\epsilon}}\right); \tag{2}$$

$$\gamma = \frac{4}{\alpha' \epsilon n} \ln \frac{2k}{\alpha} = \Theta\left(\frac{\log |X| \log^2 k \log \frac{1}{\epsilon}}{\alpha \epsilon^3 n}\right). \tag{3}$$

The denominator in (2) can be thought of as our "privacy cost" as a function of the number of queries k. Needless to say, we made no effort to optimize the constants.

The value r_i in Step 2(a) of the median mechanism is defined as

$$r_i = \frac{\sum_{S \in C_{i-1}} \exp(-\epsilon^{-1} |f_i(D) - f_i(S)|)}{|C_{i-1}|}. \tag{4}$$

For the Laplace perturbations in Steps 2(a) and 2(d), recall that the distribution $\mathrm{Lap}(\sigma)$ has the cumulative distribution function

$$F(x) = 1 - F(-x) = 1 - \frac{1}{2} e^{-x/\sigma}. \tag{5}$$

[5] From ϵ-accurate answers, one can efficiently reconstruct a synthetic database that is consistent (up to $\pm\epsilon$) with those answers, if desired [DNR$^+$09].

[6] We typically think of α, ϵ as small constants, though our results remain meaningful for some sub-constant values of α and ϵ as well. We always assume that α is at least inverse polynomial in k. Note that when α or ϵ is sufficiently small (at most c/n for a small constant c, say), simultaneously meaningful privacy and utility is clearly impossible.

[7] In contrast, the number of queries that the Laplace mechanism can privately and usefully answer is at most linear.

```
1. Initialize $C_0 = \{$ databases of size $m$ over $X$ $\}$.
2. For each query $f_1, f_2, \ldots, f_k$ in turn:
    (a) Define $r_i$ as in (4) and let $\hat{r}_i = r_i + \text{Lap}(\frac{2}{\epsilon n \alpha'})$.
    (b) Let $t_i = \frac{3}{4} + j \cdot \gamma$, where $j \in \{0, 1, \ldots, \frac{1}{\gamma}\frac{3}{20}\}$ is
        chosen with probability proportional to $2^{-j}$.
    (c) If $\hat{r}_i \geq t_i$, set $a_i$ to be the median value of $f_i$ on $C_{i-1}$.
    (d) If $\hat{r}_i < t_i$, set $a_i$ to be $f_i(D) + \text{Lap}(\frac{1}{n\alpha'})$.
    (e) If $\hat{r}_i < t_i$, set $C_i$ to the databases $S$ of $C_{i-1}$ with
        $|f_i(S) - a_i| \leq \epsilon/50$; otherwise $C_i = C_{i-1}$.
    (f) If $\hat{r}_j < t_j$ for more than $20m \log |X|$ values of $j \leq i$,
        then halt and report failure.
```

Figure 1: The Median Mechanism.

The motivation behind the mechanism's steps is as follows. The set C_i is the set of size-m databases consistent (up to $\pm\epsilon/50$) with previous answers of the mechanism to hard queries. The focus on databases with the small size m is justified by a VC dimension argument, see Proposition 4.6. Steps 2(a) and 2(b) choose a random value $\hat{r}_i$ and a random threshold t_i. The value r_i in Step 2(a) is a measure of how easy the query is, with higher numbers being easier. A more obvious measure would be the fraction of databases S in C_{i-1} for which $|f_i(S) - f_i(D)| \leq \epsilon$, but this is a highly sensitive statistic (unlike r_i, see Lemma 4.9). The mechanism uses the perturbed value $\hat{r}_i$ rather than r_i to privately communicate which queries are easy and which are hard. In Step 2(b), we choose the threshold t_i at random between 3/4 and 9/10. This randomly shifted threshold ensures that, for every database D, there is likely to be a significant gap between r_i and t_i; such gaps are useful when optimizing the privacy guarantee. Steps 2(c) and 2(d) answer easy and hard queries, respectively. Step 2(e) updates the set of databases consistent with previous answers to hard queries. We prove in Lemma 4.7 that Step 2(f) occurs with at most inverse polynomial probability.

Finally, we note that the median mechanism is defined as if the total number of queries k is (approximately) known in advance. This assumption can be removed by using successively doubling "guesses" of k; this increases the privacy cost by an $O(\log k)$ factor.

4. ANALYSIS OF MEDIAN MECHANISM

This section proves the following privacy and utility guarantees for the basic implementation of the median mechanism.

THEOREM 4.1. *For every sequence of adaptively chosen predicate queries $f_1, \ldots, f_k$ arriving online, the median mechanism is (ϵ, δ)-useful and (α, τ)-differentially private, where τ is a negligible function of k and $|X|$, and δ is an inverse polynomial function of k and n, provided the database size n satisfies*

$$n \geq \frac{30 \ln \frac{2k}{\alpha} \log_2 k}{\alpha' \epsilon} = \Theta\left(\frac{\log |X| \log^3 k \log \frac{1}{\epsilon}}{\alpha \epsilon^3}\right). \qquad (6)$$

We prove the utility and privacy guarantees in Sections 4.1 and 4.2, respectively.[8]

4.1 Utility of the Median Mechanism

Here we prove a utility guarantee for the median mechanism.

THEOREM 4.2. *The median mechanism is (ϵ, δ)-useful, where $\delta = k \exp(-\Omega(\epsilon n \alpha'))$.*

Note that under assumption (6), δ is inverse polynomial in k and n.

We give the proof of Theorem 4.2 in three pieces: with high probability, every hard query is answered accurately (Lemma 4.4); every easy query is answered accurately (Lemmas 4.3 and 4.5); and the algorithm does not fail (Lemma 4.7). The next two lemmas follow from the definition of the Laplace distribution (5), our choice of δ, and trivial union bounds.

LEMMA 4.3. *With probability at least $1 - \frac{\delta}{2}$, $|r_i - \hat{r}_i| \leq 1/100$ for every query i.*

LEMMA 4.4. *With probability at least $1 - \frac{\delta}{2}$, every answer to a hard query is $(\epsilon/100)$-accurate for D.*

The next lemma shows that median answers are accurate for easy queries.

LEMMA 4.5. *If $|r_i - \hat{r}_i| \leq 1/100$ for every query i, then every answer to an easy query is ϵ-accurate for D.*

PROOF. For a query i, let $G_{i-1} = \{S \in C_{i-1} : |f_i(D) - f_i(S)| \leq \epsilon\}$ denote the databases of C_{i-1} on which the result of query f_i is ϵ-accurate for D. Observe that if $|G_{i-1}| \geq .51 \cdot |C_{i-1}|$, then the median value of f_i on C_{i-1} is an ϵ-accurate answer for D. Thus proving the lemma reduces to showing that $\hat{r}_i \geq 3/4$ only if $|G_{i-1}| \geq .51 \cdot |C_{i-1}|$.

Consider a query i with $|G_{i-1}| < .51 \cdot |C_{i-1}|$. Using (4), we have

$$\begin{aligned} r_i &= \frac{\sum_{S \in C_{i-1}} \exp(-\epsilon^{-1}|f_i(D) - f_i(S)|)}{|C_{i-1}|} \\ &\leq \frac{|G_{i-1}| + e^{-1}|C_{i-1} \setminus G_{i-1}|}{|C_{i-1}|} \\ &\leq \frac{(\frac{51}{100} + \frac{49}{100e})|C_{i-1}|}{|C_{i-1}|} \\ &< \frac{74}{100}. \end{aligned}$$

Since $|r_i - \hat{r}_i| \leq 1/100$ for every query i by assumption, the proof is complete. □

Our final lemma shows that the median mechanism does not fail and hence answers every query, with high probability; this will conclude our proof of Theorem 4.2. We need the following preliminary proposition, which instantiates the standard uniform convergence bound with the fact that the VC dimension of every set of k predicate queries is at most $\log_2 k$ [Vap96]. Recall the definition of the parameter m from (1).

[8] If desired, in Theorem 4.1 we can treat n as a parameter and solve for the error ϵ. The maximum error on any query (normalized by the database size) is $O(\log k \log^{1/3} |X| / n^{1/3} \alpha^{1/3})$; the unnormalized error is a factor of n larger.

PROPOSITION 4.6 (UNIFORM CONVERGENCE BOUND). *For every collection of k predicate queries $f_1, \dots, f_k$ and every database D, a database S obtained by sampling points from D uniformly at random will satisfy $|f_i(D) - f_i(S)| \le \epsilon$ for all i except with probability δ, provided*

$$|S| \ge \frac{1}{2\epsilon^2}\left(\log k + \log\frac{2}{\delta}\right).$$

In particular, there exists a database S of size m such that for all $i \in \{1, \dots, k\}$, $|f_i(D) - f_i(S)| \le \epsilon/400$.

In other words, the results of k predicate queries on an arbitrarily large database can be well approximated by those on a database with size only $O(\log k)$.

LEMMA 4.7. *If $|r_i - \hat{r}_i| \le 1/100$ for every query i and every answer to a hard query is $(\epsilon/100)$-accurate for D, then the median mechanism answers fewer than $20m \log |X|$ hard queries (and hence answers all queries before terminating).*

PROOF. The plan is to track the contraction of C_i as hard queries are answered by the median mechanism. Initially we have $|C_0| \le |X|^m$. If the median mechanism answers a hard query i, then the definition of the mechanism and our hypotheses yield

$$r_i \le \hat{r}_i + \frac{1}{100} < t_i + \frac{1}{100} \le \frac{91}{100}.$$

We then claim that the size of the set $C_i = \{S \in C_{i-1} : |f_i(S) - a_i| \le \epsilon/50\}$ is at most $\frac{94}{100}|C_{i-1}|$. For if not,

$$\begin{aligned} r_i &= \frac{\sum_{S \in C_{i-1}} \exp(-\epsilon^{-1}|f_i(S) - f_i(D)|)}{|C_{i-1}|} \\ &\ge \frac{94}{100} \cdot \exp\left(-\frac{1}{50}\right) > \frac{92}{100}, \end{aligned}$$

which is a contradiction.

Iterating now shows that the number of consistent databases decreases exponentially with the number of hard queries:

$$|C_k| \le \left(\frac{94}{100}\right)^h |X|^m \tag{7}$$

if h of the k queries are hard.

On the other hand, Proposition 4.6 guarantees the existence of a database $S^* \in C_0$ for which $|f_i(S^*) - f_i(D)| \le \epsilon/100$ for every query f_i. Since all answers a_i produced by the median mechanism for hard queries i are $(\epsilon/100)$-accurate for D by assumption, $|f_i(S^*) - a_i| \le |f_i(S^*) - f_i(D)| + |f_i(D) - a_i| \le \epsilon/50$. This shows that $S^* \in C_k$ and hence $|C_k| \ge 1$. Combining this with (7) gives

$$h \le \frac{m \ln |X|}{\ln(50/47)} < 20m \ln |X|,$$

as desired. □

4.2 Privacy of the Median Mechanism

This section establishes the following privacy guarantee for the median mechanism.

THEOREM 4.8. *The median mechanism is (α, τ)- differentially private, where τ is a negligible function of $|X|$ and k when n is sufficiently large (as in (6)).*

We can treat the median mechanism as if it has two outputs: a vector of answers $a \in \mathbb{R}^k$, and a vector $d \in \{0,1\}^k$ such that $d_i = 0$ if i is an easy query and $d_i = 1$ if i is a hard query. A key observation in the privacy analysis is that answers to easy queries are a function only of the previous output of the mechanism, and incur no additional privacy cost beyond the release of the bit d_i. Moreover, the median mechanism is guaranteed to produce no more than $O(m \log |X|)$ answers to hard queries. Intuitively, what we need to show is that the vector d can be released after an unusually small perturbation.

Our first lemma states that the small sensitivity of predicate queries carries over, with a $2/\epsilon$ factor loss, to the r-function defined in (4).

LEMMA 4.9. *The function $r_i(D) = (\sum_{S \in C} \exp(-\epsilon^{-1}|f(D) - f(S)|)/|C|$ has sensitivity $\frac{2}{\epsilon n}$ for every fixed set C of databases and predicate query f.*

PROOF. Let D and D' be neighboring databases. Then

$$\begin{aligned} r_i(D) &= \frac{\sum_{S \in C} \exp(-\epsilon^{-1}|f(D) - f(S)|)}{|C_i|} \\ &\le \frac{\sum_{S \in C_i} \exp(-\epsilon^{-1}(|f(D') - f(S)| - n^{-1}))}{|C_i|} \\ &= \exp\left(\frac{1}{\epsilon n}\right) \cdot r_i(D') \\ &\le \left(1 + \frac{2}{\epsilon n}\right) \cdot r_i(D') \\ &\le r_i(D') + \frac{2}{\epsilon n} \end{aligned}$$

where the first inequality follows from the fact that the (predicate) query f has sensitivity $1/n$, the second from the fact that $e^x \le 1 + 2x$ when $x \in [0,1]$, and the third from the fact that $r_i(D') \le 1$. □

The next lemma identifies nice properties of "typical executions" of the median mechanism. Consider an output (d, a) of the median mechanism with a database D. From D and (d, a), we can uniquely recover the values $r_1, \dots, r_k$ computed (via (4)) in Step 2(a) of the median mechanism, with r_i depending only on the first $i - 1$ components of d and a. We sometimes write such a value as $r_i(D, (d, a))$, or as $r_i(D)$ if an output (d, a) has been fixed. Call a possible threshold t_i *good* for D and (d, a) if $d_i = 0$ and $r_i(D, (d, a)) \ge t_i + \gamma$, where γ is defined as in (3). Call a vector t of possible thresholds *good* for D and (d, a) if all but $180m \ln |X|$ of the thresholds are good for D and (d, a).

LEMMA 4.10. *For every database D, with all but negligible ($\exp(-\Omega(\log k \log |X|/\epsilon^2))$) probability, the thresholds t generated by the median mechanism are good for its output (d, a).*

PROOF. The idea is to "charge" the probability of bad thresholds to that of answering hard queries, which are strictly limited by the median mechanism. Since the median mechanism only allows $20m \ln |X|$ of the d_i's to be 1, we only need to bound the number of queries i with output $d_i = 0$ and threshold t_i satisfying $r_i < t_i + \gamma$, where r_i is the value computed by the median mechanism in Step 2(a) when it answers the query i.

Let Y_i be the indicator random variable corresponding to the (larger) event that $r_i < t_i + \gamma$. Define Z_i to be 1 if and only if, when answering the ith query, the median mechanism chooses a threshold t_i and a Laplace perturbation Δ_i such that $r_i+\Delta_i < t_i$ (i.e., the query is classified as hard). If the median mechanism fails before reaching query i, then we define $Y_i = Z_i = 0$. Set $Y = \sum_{i=1}^{k} Y_i$ and $Z = \sum_{i=1}^{k} Z_i$. We can finish the proof by showing that Y is at most $160m \ln |X|$ except with negligible probability.

Consider a query i and condition on the event that $r_i \geq \frac{9}{10}$; this event depends only on the results of previous queries. In this case, $Y_i = 1$ only if $t_i = 9/10$. But this occurs with probability $2^{-3/20\gamma}$, which using (3) and (6) is at most $1/k$.[9] Therefore, the expected contribution to Y coming from queries i with $r_i \geq \frac{9}{10}$ is at most 1. Since t_i is selected independently at random for each i, the Chernoff bound implies that the probability that such queries contribute more than $m \ln |X|$ to Y is

$$\exp(-\Omega((m \log |X|)^2)) = \exp(-\Omega((\log k)^2 (\log |X|)^2/\epsilon^4)).$$

Now condition on the event that $r_i < \frac{9}{10}$. Let T_i denote the threshold choices that would cause Y_i to be 1, and let s_i be the smallest such; since $r_i < \frac{9}{10}$, $|T_i| \geq 2$. For every $t_i \in T_i$, $t_i > r_i - \gamma$; hence, for every $t_i \in T_i \setminus \{s_i\}$, $t_i > r_i$. Also, our distribution on the j's in Step 2(b) ensures that $\Pr[t_i \in T_i \setminus \{s_i\}] \geq \frac{1}{2}\Pr[t_i \in T_i]$. Since the Laplace distribution is symmetric around zero and the random choices Δ_i, t_i are independent, we have

$$\begin{aligned} \mathrm{E}[Z_i] &= \Pr[t_i > r_i + \Delta_i] \\ &\geq \Pr[t_i > r_i] \cdot \Pr[\Delta_i \leq 0] \\ &\geq \tfrac{1}{4}\Pr[t_i > r_i - \gamma] \\ &= \tfrac{1}{4}\mathrm{E}[Y_i]. \end{aligned} \tag{8}$$

The definition of the median mechanism ensures that $Z \leq 20m \ln |X|$ with probability 1. Linearity of expectation, inequality (8), and the Chernoff bound imply that queries with $r_i < \frac{9}{10}$ contribute at most $159m \ln |X|$ to Y with probability at least $1 - \exp(-\Omega(\log k \log |X|/\epsilon^2))$. The proof is complete. □

We can now prove Theorem 4.8.

Proof of Theorem 4.8: Recall Definition 2 and fix a database D, queries $f_1, \ldots, f_k$, and a subset S of possible mechanism outputs. For simplicity, we assume that all perturbations are drawn from a discretized Laplace distribution, so that the median mechanism has a countable range; the continuous case can be treated using similar arguments. Then, we can think of S as a countable set of output vector pairs (d, a) with $d \in \{0,1\}^k$ and $a \in \mathbb{R}^k$. We write $MM(D, f) = (d, a)$ for the event that the median mechanism classifies the queries $f = (f_1, \ldots, f_k)$ according to d and outputs the numerical answers a. If the mechanism computes thresholds t while doing so, we write $MM(D, f) = (t, d, a)$. Let $G((d, a), D)$ denote the vectors that would be good thresholds for (d, a) and D. (Recall that D and (d, a) uniquely define the corresponding $r_i(D, (d, a))$'s.)

We have

$$\Pr[MM(D,f) \in S] = \sum_{(d,a) \in S} \Pr[MM(D,f) = (d,a)]$$

$$\leq \tau + \sum_{(d,a) \in S} \Pr[MM(D,f) = (t,d,a)]$$

$$= \tau + \sum_{(d,a) \in S} \sum_{t \in G((d,a),D)} \Pr[MM(D,f) = (t,d,a)]$$

with some t good for $(d, a), D$, and where τ is the negligible function of Lemma 4.10. We complete the proof by showing that, for every neighboring database D', possible output (d, a), and thresholds t good for (d, a) and D,

$$\Pr[MM(D,f) = (t,d,a)] \leq e^{\alpha} \cdot \Pr[MM(D',f) = (t,d,a)]. \tag{9}$$

Fix a neighboring database D', a target output (d, a), and thresholds t good for (d, a) and D. The probability that the median mechanism chooses the target thresholds t is independent of the underlying database, and so is the same on both sides of (9). For the rest of the proof, we condition on the event that the median mechanism uses the thresholds t (both with database D and database D').

Let $\mathcal{E}_i$ denote the event that $MM(D, f)$ classifies the first i queries in agreement with the target output (i.e., query $j \leq i$ is deemed easy if and only if $d_j = 0$) and that its first i answers are $a_1, \ldots, a_i$. Let $\mathcal{E}'_i$ denote the analogous event for $MM(D', f)$. Observe that $\mathcal{E}_k, \mathcal{E}'_k$ are the relevant events on the left- and right-hand sides of (9), respectively (after conditioning on t). If (d, a) is such that the median mechanism would fail after the ℓth query, then the following proof should be applied to $\mathcal{E}_\ell, \mathcal{E}'_\ell$ instead of $\mathcal{E}_k, \mathcal{E}'_k$. We next give a crude upper bound on the ratio $\Pr[\mathcal{E}_i|\mathcal{E}_{i-1}]/\Pr[\mathcal{E}'_i|\mathcal{E}'_{i-1}]$ that holds for every query (see (10), below), followed by a much better upper bound for queries with good thresholds.

Imagine running the median mechanism in parallel on D, D' and condition on the events $\mathcal{E}_{i-1}, \mathcal{E}'_{i-1}$. The set C_{i-1} is then the same in both runs of the mechanism, and $r_i(D), r_i(D')$ are now fixed. Let b_i (b'_i) be 0 if $MM(D, f)$ ($MM(D', f)$) classifies query i as easy and 1 otherwise. Since $r_i(D') \in [r_i(D) \pm \frac{2}{\epsilon n}]$ (Lemma 4.9) and a perturbation with distribution $\mathrm{Lap}(\frac{2}{\alpha' \epsilon n})$ is added to these values before comparing to the threshold t_i (Step 2(a)),

$$\Pr[b_i = 0 \,|\, \mathcal{E}_{i-1}] \leq e^{\alpha'} \Pr[b'_i = 0 \,|\, \mathcal{E}'_{i-1}]$$

and similarly for the events where $b_i, b'_i = 1$. Suppose that the target classification is $d_i = 1$ (a hard query), and let s_i and s'_i denote the random variables $f_i(D) + \mathrm{Lap}(\frac{1}{\alpha' n})$ and $f_i(D') + \mathrm{Lap}(\frac{1}{\alpha' n})$, respectively. Independence of the Laplace perturbations in Steps 2(a) and 2(d) implies that

$$\Pr[\mathcal{E}_i|\mathcal{E}_{i-1}] = \Pr[b_i = 1 \,|\, \mathcal{E}_{i-1}] \cdot \Pr[s_i = a_i \,|\, \mathcal{E}_{i-1}]$$

and

$$\Pr[\mathcal{E}'_i|\mathcal{E}'_{i-1}] = \Pr[b'_i = 1 \,|\, \mathcal{E}'_{i-1}] \cdot \Pr[s'_i = a_i \,|\, \mathcal{E}'_{i-1}].$$

Since the predicate query f_i has sensitivity $1/n$, we have

$$\Pr[\mathcal{E}_i \,|\, \mathcal{E}_{i-1}] \leq e^{2\alpha'} \cdot \Pr[\mathcal{E}'_i \,|\, \mathcal{E}'_{i-1}] \tag{10}$$

when $d_i = 1$.

Now suppose that $d_i = 0$, and let m_i denote the median value of f_i on C_{i-1}. Then $\Pr[\mathcal{E}_i|\mathcal{E}_{i-1}]$ is either 0 (if $m_i \neq a_i$) or $\Pr[b_i = 0 \,|\, \mathcal{E}_{i-1}]$ (if $m_i = a_i$); similarly, $\Pr[\mathcal{E}'_i|\mathcal{E}'_{i-1}]$ is either 0 or $\Pr[b'_i = 0 \,|\, \mathcal{E}'_{i-1}]$. Thus the bound in (10) continues to hold (even with $e^{2\alpha'}$ replaced by $e^{\alpha'}$) when $d_i = 0$.

[9] For simplicity, we ignore the normalizing constant in the distribution over j's in Step 2(b), which is $\Theta(1)$.

Since α' is not much smaller than the privacy target α (recall (2)), we cannot afford to suffer the upper bound in (10) for many queries. Fortunately, for queries i with good thresholds we can do much better. Consider a query i such that t_i is good for (d, a) and D and condition again on $\mathcal{E}_{i-1}, \mathcal{E}'_{i-1}$, which fixes C_{i-1} and hence $r_i(D)$. Goodness implies that $d_i = 0$, so the arguments from the previous paragraph also apply here. We can therefore assume that the median value m_i of f_i on C_{i-1} equals a_i and focus on bounding $\Pr[b_i = 0 \,|\, \mathcal{E}_{i-1}]$ in terms of $\Pr[b'_i = 0 \,|\, \mathcal{E}'_{i-1}]$. Goodness also implies that $r_i(D) \geq t_i + \gamma$ and hence $r_i(D') \geq t_i + \gamma - \frac{2}{\epsilon n} \geq t_i + \frac{\gamma}{2}$ (by Lemma 4.9). Recalling from (3) the definition of γ, we have

$$\begin{aligned} \Pr[b'_i = 0 \,|\, \mathcal{E}'_{i-1}] &\geq \Pr[r_i - \hat{r}_i < \tfrac{\gamma}{2}] \\ &= 1 - \tfrac{1}{2} e^{-\gamma \alpha' \epsilon n/4} \\ &= 1 - \frac{\alpha}{4k} \qquad (11) \end{aligned}$$

and of course, $\Pr[b_i = 0 \,|\, \mathcal{E}_{i-1}] \leq 1$.

Applying (10) to the bad queries — at most $180 m \ln |X|$ of them, since t is good for (d, a) and D — and (11) to the rest, we can derive

$$\begin{aligned} \Pr[\mathcal{E}_k] &= \prod_{i=1}^{k} \Pr[\mathcal{E}_i \,:\, \mathcal{E}_{i-1}] \\ &\leq \underbrace{e^{360\alpha' m \ln |X|}}_{\leq e^{\alpha/2} \text{ by } (2)} \cdot \underbrace{(1 - \frac{\alpha}{4k})^{-k}}_{\leq (1+\frac{\alpha}{2k})^k \leq e^{\alpha/2}} \cdot \prod_{i=1}^{k} \Pr[\mathcal{E}'_i \,:\, \mathcal{E}'_{i-1}] \\ &\leq e^{\alpha} \cdot \Pr[\mathcal{E}'_k], \end{aligned}$$

which completes the proof of both the inequality (9) and the theorem. ■

5. THE MEDIAN MECHANISM: EFFICIENT IMPLEMENTATION

The basic implementation of the median mechanism runs in time $|X|^{\Theta(\log k \log(1/\epsilon)/\epsilon^2)}$. This section provides an efficient implementation, running in time polynomial in n, k, and $|X|$, although with a weaker usefulness guarantee.

THEOREM 5.1. *Assume that the database size n satisfies (6). For every sequence of adaptively chosen predicate queries $f_1, \ldots, f_k$ arriving online, the efficient implementation of the median Mechanism is (α, τ)-differentially private for a negligible function τ. Moreover, for every fixed set $f_1, \ldots, f_k$ of queries, it is (ϵ, δ)-useful for all but a negligible fraction of* fractional *databases (equivalently, probability distributions).*

Specifically, our mechanism answers exponentially many queries for all but an $O(|X|^{-m})$ fraction of probability distributions over X drawn from the unit ℓ_1 ball, and from databases drawn from such distributions. Thus our efficient implementation always guarantees privacy, but for a given set of queries $f_1, \ldots, f_k$, there might be a negligibly small fraction of fractional histograms for which our mechanism is not useful for all k queries.

We note however that even for the small fraction of fractional histograms for which the efficient median mechanism may not satisfy our usefulness guarantee, it does not output incorrect answers: it merely halts after having answered a sufficiently large number of queries using the Laplace mechanism. Therefore, even for this small fraction of databases, the efficient median mechanism is an improvement over the Laplace mechanism: in the worst case, it simply answers every query using the Laplace mechanism before halting, and in the best case, it is able to answer many more queries.

We give a high-level overview of the proof of Theorem 5.1 which we then make formal. First, why isn't the median mechanism a computationally efficient mechanism? Because C_0 has super-polynomial size $|X|^m$, and computing r_i in Step 2(a), the median value in Step 2(c), and the set C_i in Step 2(e) could require time proportional to $|C_0|$. An obvious idea is to randomly sample elements of C_{i-1} to approximately compute r_i and the median value of f_i on C_{i-1}; while it is easy to control the resulting sampling error and preserve the utility and privacy guarantees of Section 4, it is not clear how to sample from C_{i-1} efficiently.

We show how to implement the median mechanism in polynomial time by redefining the sets C_i to be sets of probability distributions over points in X that are consistent (up to $\pm\frac{\epsilon}{50}$) with the hard queries answered up to the ith query. Each set C_i will be a convex polytope in $\mathbb{R}^{|X|}$ defined by the intersection of at most $O(m \log |X|)$ halfspaces, and hence it will be possible to sample points from C_i approximately uniformly at random in time poly$(|X|, m)$ via the grid walk of Dyer, Frieze, and Kannan [DFK91]. Lemmas 4.3, 4.4, and 4.5 still hold (trivially modified to accommodate sampling error). We have to reprove Lemma 4.7, in a somewhat weaker form: that for all but a diminishing fraction of input databases D, the median mechanism does not abort except with probability $k \exp(-\Omega(\epsilon n \alpha'))$. As for our privacy analysis of the median mechanism, it is independent of the representation of the sets C_i and the mechanisms' failure probability, and so it need not be repeated — the efficient implementation is provably private for *all* input databases and query sequences.

We now give a formal analysis of the efficient implementation.

5.1 Redefining the sets C_i

We redefine the sets C_i to represent databases that can contain points fractionally, as opposed to the finite set of small discrete databases. Equivalently, we can view the sets C_i as containing probability distributions over the set of points X.

We initialize C_0 to be the ℓ_1 ball of radius m in $\mathbb{R}^{|X|}$, $mB_1^{|X|}$, intersected with the non-negative orthant:

$$C_0 = \{F \in \mathbb{R}^{|X|} : F \geq 0, ||F||_1 \leq m\}.$$

Each dimension i in $\mathbb{R}^{|X|}$ corresponds to an element $x_i \in X$. Elements $F \in C_0$ can be viewed as fractional histograms. Note that integral points in C_0 correspond exactly to databases of size at most m.

We generalize our query functions f_i to fractional histograms in the natural way:

$$f_i(F) = \frac{1}{m} \sum_{j : f_i(x_j) = 1} F_j.$$

The update operation after a hard query i is answered is the same as in the basic implementation:

$$C_i \leftarrow \left\{ F \in C_{i-1} : |f_i(F) - a_i| \leq \frac{\epsilon}{50} \right\}.$$

Note that each updating operation after a hard query merely intersects C_{i-1} with the pair of halfspaces:

$$\sum_{j:f_i(x_j)=1} F_j \leq ma_i+\frac{\epsilon m}{50} \qquad \text{and} \qquad \sum_{j:f_i(x_j)=1} F_j \geq ma_i-\frac{\epsilon m}{50};$$

and so C_i is a convex polytope for each i.

Dyer, Kannan, and Frieze [DFK91] show how to δ-approximate a random sample from a convex body $K \in \mathbb{R}^{|X|}$ in time polynomial in $|X|$ and the running time of a membership oracle for K, where δ can be taken to be exponentially small (which is more than sufficient for our purposes). Their algorithm has two requirements:

1. There must be an efficient membership oracle which can in polynomial time determine whether a point $F \in \mathbb{R}^{|X|}$ lies in K.
2. K must be 'well rounded': $B_2^{|X|} \subseteq K \subseteq |X|B_2^{|X|}$, where $B_2^{|X|}$ is the unit ℓ_2 ball in $\mathbb{R}^{|X|}$.

Since C_i is given as the intersection of a set of explicit halfspaces, we have a simple membership oracle to determine whether a given point $F \in C_i$: we simply check that F lies on the appropriate side of each of the halfspaces. This takes time poly$(|X|, m)$, since the number of halfspaces defining C_i is linear in the number of answers to hard queries given before time i, which is never more than $20m \ln |X|$. Moreover, for each i we have $C_i \subseteq C_0 \subset mB_1^{|X|} \subset mB_2^{|X|} \subset |X|B_2^{|X|}$. Finally, we can safely assume that $B_2^X \subseteq C_i$ by simply considering the convex set $C'_i = C_i + B_2^X$ instead. This will not affect our results.

Therefore, we can implement the median mechanism in time poly$(|X|, k)$ by using sets C_i as defined in this section, and sampling from them using the grid walk of [DFK91]. Estimation error in computing r_i and the median value of f_i on C_{i-1} by random sampling rather than brute force is easily controlled via the Chernoff bound and can be incorporated into the proofs of Lemmas 4.3 and 4.5 in the obvious way. It remains to prove a continuous version of Lemma 4.7 to show that the efficient implementation of the median mechanism is (ϵ, δ)-useful on all but a negligibly small fraction of fractional histograms F.

5.2 Usefulness for Almost All Distributions

We now prove an analogue of Lemma 4.7 to establish a usefulness guarantee for the efficient version of the median mechanism.

Definition 3. With respect to any set of k queries $f_1, \ldots, f_k$ and for any $F^* \in C_0$, define

$$\text{Good}_\epsilon(F^*) = \{F \in C_0 : \max_{i \in \{1,2,\ldots,k\}} |f_i(F) - f_i(F^*)| \leq \epsilon\}$$

as the set of points that agree up to an additive ϵ factor with F^* on every query f_i.

Since databases $D \subset X$ can be identified with their corresponding histogram vectors $F \in \mathbb{R}^{|X|}$, we can also write Good$_\epsilon(D)$ when the meaning is clear from context.

For any F^*, Good$_\epsilon(F^*)$ is a convex polytope contained inside C_0. We will prove that the efficient version of the median mechanism is (ϵ, δ)-useful for a database D if

$$\frac{\text{Vol}(\text{Good}_{\epsilon/100}(D))}{\text{Vol}(C_0)} \geq \frac{1}{|X|^{2m}}. \tag{12}$$

We first prove that (12) holds for almost every fractional histogram. For this, we need a preliminary lemma.

Lemma 5.2. *Let $\mathcal{L}$ denote the set of integer points inside C_0. Then with respect to an arbitrary set of k queries,*

$$C_0 \subseteq \bigcup_{F \in \mathcal{L}} \textit{Good}_{\epsilon/400}(F).$$

Proof. Every rational valued point $F \in C_0$ corresponds to some (large) database $D \subset X$ by scaling F to an integer-valued histogram. Irrational points can be arbitrarily approximated by such a finite database. By Proposition 4.6, for every set of k predicates $f_1, \ldots, f_k$, there is a database $F^* \subset X$ with $|F^*| = m$ such that for each i, $|f_i(F^*) - f_i(F)| \leq \epsilon/400$. Recalling that the histograms corresponding to databases of size at most m are exactly the integer points in C_0, the proof is complete. □

Lemma 5.3. *All but an $|X|^{-m}$ fraction of* fractional *histograms F satisfy*

$$\frac{\text{Vol}(\textit{Good}_{\epsilon/200}(F))}{\text{Vol}(C_0)} \geq \frac{1}{|X|^{2m}}.$$

Proof. Let

$$\mathcal{B} = \left\{F \in \mathcal{L} : \frac{\text{Vol}(\text{Good}_{\epsilon/400}(F))}{\text{Vol}(C_0)} \leq \frac{1}{|X|^{2m}}\right\}.$$

Consider a randomly selected fractional histogram $F^* \in C_0$. For any $F \in \mathcal{B}$ we have:

$$\Pr[F^* \in \text{Good}_{\epsilon/400}(F)] = \frac{\text{Vol}(\text{Good}_{\epsilon/400}(F))}{\text{Vol}(C_0)} < \frac{1}{|X|^{2m}}$$

Since $|\mathcal{B}| \leq |\mathcal{L}| \leq |X|^m$, by a union bound we can conclude that except with probability $\frac{1}{|X|^m}$, $F^* \notin \text{Good}_{\epsilon/400}(F)$ for any $F \in \mathcal{B}$. However, by Lemma 5.2, $F^* \in \text{Good}_{\epsilon/400}(F')$ for some $F' \in \mathcal{L}$. Therefore, except with probability $1/|X|^m$, $F' \in \mathcal{L} \setminus \mathcal{B}$. Thus, since $\text{Good}_{\epsilon/400}(F') \subseteq \text{Good}_{\epsilon/200}(F^*)$, except with negligible probability, we have:

$$\frac{\text{Vol}(\text{Good}_{\epsilon/200}(F^*))}{\text{Vol}(C_0)} \geq \frac{\text{Vol}(\text{Good}_{\epsilon/400}(F'))}{\text{Vol}(C_0)} \geq \frac{1}{|X|^{2m}}.$$

□

We are now ready to prove the analogue of Lemma 4.7 for the efficient implementation of the median mechanism.

Lemma 5.4. *For every set of k queries $f_1, \ldots, f_k$, for all but an $O(|X|^{-m})$ fraction of fractional histograms F, the efficient implementation of the median mechanism guarantees that: The mechanism answers fewer than $40m \log |X|$ hard queries, except with probability $k\exp(-\Omega(\epsilon n\alpha'))$,*

Proof. We assume that all answers to hard queries are $\epsilon/100$ accurate, and that $|r_i - \hat{r}_i| \leq \frac{1}{100}$ for every i. By Lemmas 4.3 and 4.4 — the former adapted to accommodate approximating r_i via random sampling — we are in this case except with probability $k\exp(-\Omega(\epsilon n\alpha'))$.

We analyze how the volume of C_i contracts with the number of hard queries answered. Suppose the mechanism answers a hard query at time i. Then:

$$r_i \leq \hat{r}_i + \frac{1}{100} < t_i + \frac{1}{100} \leq \frac{91}{100}.$$

Recall $C_i = \{F \in C_{i-1} : |f_i(F) - a_i| \leq \epsilon/50\}$. Suppose that $\mathrm{Vol}(C_i) \geq \frac{94}{100}\mathrm{Vol}(C_{i-1})$. Then

$$r_i = \frac{\int_{C_{i-1}} \exp(-\epsilon^{-1}|f_i(F) - f_i(D)|)dF}{\mathrm{Vol}(C_{i-1})}$$

$$\geq \frac{94}{100}\exp\left(-\frac{1}{50}\right) > \frac{92}{100},$$

a contradiction. Therefore, we have

$$|C_k| \leq \left(\frac{94}{100}\right)^h \mathrm{Vol}(C_0), \qquad (13)$$

if h of the k queries are hard.

Since all answers to hard queries are $\epsilon/100$ accurate, it must be that $\mathrm{Good}_{\epsilon/100}(D) \in C_k$. Therefore, for an input database D that satisfies (12) — and this is all but an $O(|X|^{-m})$ fraction of them, by Lemma 5.3 — we have

$$\mathrm{Vol}(C_k) \geq \mathrm{Vol}(\mathrm{Good}_{\epsilon/100}(D)) \geq \frac{\mathrm{Vol}(C_0)}{|X|^{2m}}. \qquad (14)$$

Combining inequalities (13) and (14) yields

$$h \leq \frac{2m\ln|X|}{\ln\frac{50}{47}} < 40m\ln|X|,$$

as claimed. □

Lemmas 4.4, 4.5, and 5.4 give the following utility guarantee.

THEOREM 5.5. *For every set $f_1, \ldots, f_k$ of queries, for all but a negligible fraction of fractional histograms F, the efficient implementation of the median mechanism is (ϵ, δ)-useful with $\delta = k\exp(-\Omega(\epsilon n\alpha'))$.*

5.3 Usefulness for Finite Databases

Fractional histograms correspond to probability distributions over X. Lemma 5.3 shows that most probability distributions are 'good' for the efficient implementation of the Median Mechanism; in fact, more is true. We next show that finite databases *sampled* from randomly selected probability distributions also have good volume properties. Together, these lemmas show that the efficient implementation of the median mechanism will be able to answer nearly exponentially many queries with high probability, in the setting in which the private database D is drawn from some 'typical' population distribution.

DatabaseSample($|D|$):

1. Select a fractional point $F \in C_0$ uniformly at random.
2. Sample and return a database D of size $|D|$ by drawing each $x \in D$ independently at random from the probability distribution over X induced by F (i.e. sample $x_i \in X$ with probability proportional to F_i).

LEMMA 5.6. *For $|D|$ as in* (6) *(as required for the Median Mechanism), a database sampled by **DatabaseSample**($|D|$) satisfies* (12) *except with probability at most $O(|X|^{-m})$.*

PROOF. By lemma 5.3, except with probability $|X|^{-m}$, the fractional histogram F selected in step 1 satisfies

$$\frac{\mathrm{Vol}(\mathrm{Good}_{\epsilon/200}(F))}{\mathrm{Vol}(C_0)} \geq \frac{1}{|X|^{2m}}.$$

By lemma 4.6, when we sample a database D of size $|D| \geq O((\log|X|\log^3 k \log 1/\epsilon)/\epsilon^3)$ from the probability distribution induced by F, except with probability $\delta = O(k|X|^{-\log^3 k/\epsilon})$, $\mathrm{Good}_{\epsilon/200}(F) \subset \mathrm{Good}_{\epsilon/100}(D)$, which gives us condition (12). □

We would like an analogue of lemma 5.3 that holds for all but a diminishing fraction of *finite* databases (which correspond to lattice points within C_0) rather than fractional points in C_0, but it is not clear how uniformly randomly sampled lattice points distribute themselves with respect to the volume of C_0. If $n >> |X|$, then the lattice will be fine enough to approximate the volume of C_0, and lemma 5.3 will continue to hold. We now show that *small* uniformly sampled databases will also be good for the efficient version of the median mechanism. Here, small means $n = o(\sqrt{|X|})$, which allows for databases which are still polynomial in the size of X. A tighter analysis is possible, but we opt instead to give a simple argument.

LEMMA 5.7. *For every n such that n satisfies* (6) *and $n = o(\sqrt{|X|})$, all but an $O(n^2/|X|)$ fraction of databases D of size $|D| = n$ satisfy condition* (12).

PROOF. We proceed by showing that our **DatabaseSample** procedure, which we know via lemma 5.6 generates databases that satisfy (12) with high probability, is close to uniform. Note that **DatabaseSample** first selects a *probability distribution* F uniformly at random from the positive quadrant of the ℓ_1 ball, and then samples D from F.

For any particular database D^* with $|D^*| = n$ we write $\Pr_U[D = D^*]$ to denote the probability of generating D^* when we sample a database uniformly at random, and we write $\Pr_N[D = D^*]$ to denote the probability of generating D^* when we sample a database according to **DatabaseSample**. Let R denote the event that D^* contains no duplicate elements. We begin by noting by symmetry that: $\Pr_U[D = D^*|R] = \Pr_N[D = D^*|R]$ We first argue that $\Pr_U[R]$ and $\Pr_N[R]$ are both large. We immediately have that the expected number of repetitions in database D when drawn from the uniform distribution is $\binom{n}{2}/|X|$, and so $\Pr_U[\neg R] \leq \frac{n^2}{|X|}$. We now consider $\Pr_N[R]$. Since F is a uniformly random point in the positive quadrant of the ℓ_1 ball, each coordinate F_i has the marginal of a Beta distribution: $F_i \sim \beta(1, |X| - 1)$. (See, for example, [Dev86] Chapter 5). Therefore, $E[F_i^2] = \frac{2}{|X|(|X|+1)}$ and so the expected number of repetitions in database D when drawn from **DatabaseSample** is $\binom{n}{2}\sum_{i=1}^{|X|} E[F_i^2] = \frac{2\binom{n}{2}}{|X|+1} \leq \frac{2n^2}{|X|}$. Therefore, $\Pr_N[\neg R] \leq \frac{2n^2}{|X|}$.

Finally, let B be the event that database D fails to satisfy (12). We have:

$$\begin{aligned}
\Pr_U[B] &= \Pr_U[B|R] \cdot \Pr_U[R] + \Pr_U[B|\neg R] \cdot \Pr_U[\neg R] \\
&= \Pr_N[B|R] \cdot \Pr_U[R] + \Pr_U[B|\neg R] \cdot \Pr_U[\neg R] \\
&\leq \Pr_N[B|R] \cdot \Pr_U[R] + \Pr_U[\neg R] \\
&\leq \Pr_N[B] \cdot \frac{\Pr_U[R]}{\Pr_N[R]} + \Pr_U[\neg R] \\
&\leq \frac{\Pr_N[B]}{1 - \frac{2n^2}{|X|}} + \frac{n^2}{|X|} \\
&= O(\frac{n^2}{|X|})
\end{aligned}$$

where the last equality follows from lemma 5.6, which states that $\Pr_N[B]$ is negligibly small. □

We observe that we can substitute either of the above lemmas for lemma 5.3 in the proof of lemma 5.4 to obtain versions of Thoerem 5.5:

COROLLARY 5.8. *For every set $f_1, \ldots, f_k$ of queries, for all but a negligible fraction of databases sampled by* ***DatabaseSample****, the efficient implementation of the median mechanism is (ϵ, δ)-useful with $\delta = k \exp(-\Omega(\epsilon n \alpha'))$.*

COROLLARY 5.9. *For every set $f_1, \ldots, f_k$ of queries, for all but an $n^2/|X|$ fraction of uniformly randomly sampled databases of size n, the efficient implementation of the median mechanism is (ϵ, δ)-useful with $\delta = k \exp(-\Omega(\epsilon n \alpha'))$.*

6. CONCLUSION

We have shown that in the setting of predicate queries, interactivity does not pose an information theoretic barrier to differentially private data release. In particular, our dependence on the number of queries k nearly matches the optimal dependence of $\log k$ achieved in the *offline* setting by [BLR08]. We remark that our dependence on other parameters is not necessarily optimal: in particular, [DNR+09] achieves a better (and optimal) dependence on ϵ. We have also shown how to implement our mechanism in time $\text{poly}(|X|, k)$, although at the cost of sacrificing worst-case utility guarantees. The question of an interactive mechanism with $\text{poly}(|X|, k)$ runtime and worst-case utility guarantees remains an interesting open question. More generally, although the lower bounds of [DNR+09] seem to preclude mechanisms with run-time $\text{poly}(\log |X|)$ from answering a superlinear number of generic predicate queries, the question of achieving this runtime for specific query classes of interest (offline or online) remains largely open. Recently a representation-dependent impossibility result for the class of conjunctions was obtained by Ullman and Vadhan [UV10]: either extending this to a representation-independent impossibility result, or circumventing it by giving an efficient mechanism with a novel output representation would be very interesting.

Acknowledgments

The first author wishes to thank a number of people for useful discussions, including Avrim Blum, Moritz Hardt, Katrina Ligett, Frank McSherry, and Adam Smith. He would particularly like to thank Moritz Hardt for suggesting trying to prove usefulness guarantees for a continuous version of the BLR mechanism, and Avrim Blum for suggesting the distribution from which we select the threshold in the median mechanism.

7. REFERENCES

[BLR08] 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 New York, NY, USA, 2008.

[Dev86] L. Devroye. *Non-uniform random variate generation.* 1986.

[DFK91] M. Dyer, A. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM (JACM)*, 38(1):1–17, 1991.

[DMNS06] 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.

[DN03] 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.

[DNR+09] 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 Symposium on theory of computing*, pages 381–390. ACM New York, NY, USA, 2009.

[Dwo08] C. Dwork. Differential privacy: A survey of results. In *Proceedings of Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008*, volume 4978 of *Lecture Notes in Computer Science*, page 1. Springer, 2008.

[GRS09] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing*, pages 351–360. ACM New York, NY, USA, 2009.

[HT10] 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.

[KLN+08] 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.

[MT07] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual Symposium on Foundations of Computer Science*, 2007.

[UV10] J. Ullman and S. Vadhan. PCPs and the Hardness of Generating Synthetic Data . *Manuscript*, 2010.

[Vap96] V. Vapnik. Structure of statistical learning theory. *Computational Learning and Probabilistic Reasoning*, page 3, 1996.