2017

Differential privacy for growing databases

Gi Heung (Robin) Kim
University of Richmond

Recommended Citation
Kim, Gi Heung (Robin), "Differential privacy for growing databases" (2017). *Honors Theses*. 995.
https://scholarship.richmond.edu/honors-theses/995

Differential Privacy for Growing Databases

Gi Heung (Robin) Kim

Honors Thesis[*]

Department of Mathematics & Computer Science

University of Richmond

April 28, 2017

[*] Under the direction of Dr. Sara Krehbiel

The signatures below, by the thesis advisor, the departmental reader, and the honors coordinator for computer science, certify that this thesis, prepared by Robin (Gi Heung) Kim, has been approved, as to style and content.

(Dr. Sara Krehbiel, thesis advisor)

(Dr. Prateek Bhakta, departmental reader)

(Dr. Lewis Barnett, honors coordinator)

Abstract

Differential privacy [DMNS06] is a strong definition of database privacy that provides individuals in a database with the guarantee that any particular person's information has very little effect on the output of any analysis of the overall database. In order for this type of analysis to be practical, it must simultaneously preserve privacy and utility, where utility refers to how well the analysis describes the contents of the database.

An analyst may additionally wish to evaluate how a database's composition changes over time. Consider a company, for example, that accumulates data from a growing base of customers. This company may want to analyze how its customer base evolves over time. Despite the practical need to conduct private analysis on growing databases, relatively little is known about differential privacy in this setting.

In this work, we seek to expand the scope of a differentially private mechanism called the median mechanism [RR10]. The median mechanism's strength lies in its ability to answer many queries interactively, satisfying both privacy and utility constraints. We examine how these privacy and utility guarantees change in a growing database setting. First, we analyze the median mechanism when run multiple times independently as the database size increases. This approach is called sequential composition, and we show how to adjust parameters so that the privacy guarantee suffers logarithmically with the number of runs of the mechanism without any loss of utility. Having established this as a benchmark, we propose a new algorithm called the memory mechanism. In contrast to sequential composition, the memory approach preserves the history of the mechanism's responses to earlier queries as the size increases. We show that the memory mechanism's worst case performance matches that of the sequential composition, and we conjecture that the utility guarantee can be improved with natural constraints on the queries asked in each phase and on the distribution of the data. Proving such a conjecture to establish the benefit of the memory mechanism is left for future work.

Contents

1 Introduction

1.1 Differential Privacy

In June 2016, Apple announced that they had begun collecting certain user information in a manner that guarantees users a particular type of privacy called *differential privacy.* In a world in which companies must extract information from vast quantities of data to stay competitive, the field of differential privacy offers powerful tools for conducting accurate analysis while still offering strong privacy guarantees to individuals in a database.

This field seeks to design mechanisms that answer statistical queries about an input database, each row of which corresponds to the private data of a single data subject. In this work, X denotes the universe of entries in a database, $D \in X^n$ denotes a database on n entries, and $M : X^n \to R$ denotes a mechanism (algorithm) that operates on such a database and produces output in range R.[1] A good mechanism should simultaneously provide guarantees of *differential privacy* and *utility.* Utility refers to a mechanism's ability to answer queries accurately, and differential privacy refers to the property that no row in the database has too much effect on the distribution of output produced by the mechanism. The preliminaries section provides formal notation and definitions.

Definition 1.1 (Utility, informally)**.** For any $\epsilon, \delta > 0$, a mechanism $M : X^n \to \mathbb{R}$ is (ϵ, δ)-useful for query $q : X^n \to \mathbb{R}$ if for any database $D \in X^n$, we have $\Pr[|M(D) - q(D)| > \epsilon] \leq \delta$.

Definition 1.2 (Privacy, informally)**.** For any $\alpha, \tau > 0$, a mechanism $M : X^n \to \mathbb{R}$ is (α, τ)-differentially private if for any databases $D, D' \in X^n$ differing on only one row and any event $S \subseteq \mathbb{R}$, we have $\Pr[M(D) \in S] \leq e^\alpha \cdot \Pr[M(D') \in S] + \tau$.

Note that both guarantees are parametrized, and smaller ϵ, α correspond to stronger utility and privacy guarantees. We think of δ and τ as being the probabilities that the ϵ-utility and α-privacy guarantees, respectively, are not met.

[1] For the informal definitions, we let $R = \mathbb{R}$, indicating that a mechanism outputs a single real value.

The signatures below, by the thesis advisor, the departmental reader, and the honors coordinator for computer science, certify that this thesis, prepared by Robin (Gi Heung) Kim, has been approved, as to style and content.

(Dr. Sara Krehbiel, thesis advisor)

(Dr. Prateek Bhakta, departmental reader)

(Dr. Lewis Barnett, honors coordinator)

1.2 Independent Laplace Perturbation and the Exponential Mechanism

Output perturbation is the technique of adding random noise to the true answer for some database query to establish a guarantee of differential privacy. More noise creates a stronger privacy guarantee at the cost of reduced utility. This tradeoff between privacy and utility is a central focus of differentially private mechanism design. The simplest differentially private mechanism based on output perturbation is called the Laplace mechanism, which adds Laplace noise to the output of a predicate query on the input database. A predicate query $q : X^n \to [0, 1]$ calculates the proportion of rows in a database satisfying some boolean predicate over X. Note that a single row change in the database changes the answer to a predicate query by at most $1/n$. The Laplace mechanism draws this perturbation from $\text{Lap}(\frac{1}{n\alpha})$, defined by probability density $p(x) = \frac{n\alpha}{2} \exp(-n\alpha|x|)$. This mechanism provides $(\alpha, 0)$-differential privacy and $(\epsilon, n\alpha \exp(-n\alpha\epsilon))$-utility for any desired $\epsilon > 0$, which is the best possible tradeoff between privacy and utility for pure $(\tau = 0)$ differentially privacy mechanisms [GRS09].

However, if we wish to answer multiple queries on the same database using independent Laplace perturbation, the parameter of the noise added to each query must scale linearly with the number of queries to maintain privacy. This means only $O(n)$ queries can be answered with meaningful privacy and utility [RR10]. To avoid this downside of independent output perturbations, [BLR08] showed how to use the exponential mechanism [MT07] to allow α to scale only logarithmically with the number of queries with fixed α and ϵ. This approach guarantees both privacy and utility over k queries, where k may be exponential in n. However, it also has two drawbacks. First, the exponential mechanism requires all queries to be given upfront, not supporting interactive analysis, which independent Laplace perturbation can handle. Second, the exponential mechanism is inefficient since its running time is not polynomial in n, k and $|X|$.

1.3 The Median Mechanism

To circumvent these drawbacks, Roth and Roughgarden [RR10] developed a new mechanism called the median mechanism. Compared to the Laplace mechanism whose privacy parameter α scales linearly with the number of queries k, the median mechanism allows α to scale only logarithmically with the number of queries k and the size of X, like the exponential mechanism. Moreover, the median mechanism also avoids the drawbacks of the exponential mechanism, allowing an analyst to supply queries interactively and admitting an efficient implementation.[2]

At its core, the median mechanism works by categorizing each incoming query in real-time as either *easy* or *hard*. At a high level, the categorization works as follows. If the approximate answer to a query can be derived from the answers to all previous queries, then the query is deemed easy and the mechanism simply reports this fact to the analyst revealing no additional information about the database. Otherwise, the query is deemed hard and the mechanism applies independent Laplace perturbation. [RR10] proves that there can be only $O(\log k \log |X|)$ hard queries. By only perturbing a small fraction of the total queries, the median mechanism allows α to scale only logarithmically with the number of queries answered. The details are covered in Section 3.

1.4 Our Results

This paper aims to broaden the scope of [RR10]. As with the vast majority of differentially private mechanisms, the median mechanism works on a fixed database of size n. In this paper we are considering a dynamic setting of a database where the content is constantly changing and accumulating. We consider a simple model in which a database grows by n entries in each of K phases. An analyst requests answers to k queries in each phase, and it is possible that the content of new entries in a particular phase is completely different from the initial database. Our contributions fall into three categories:

1. We first analyze sequential composition of the median mechanism. We show that we can run the median mechanism independently K times, decreasing the privacy parameter in each phase to maintain utility at a cost of only a $\log K$ factor in privacy. See Theorem 4.1 for the formal statement of this result.

2. Next we propose a new mechanism called the memory mechanism, which retains the information provided to the analyst across phases. We analyze this mechanism and show that it

[2] This work actually focuses on the *less efficient* implementation of the median mechanism described in [RR10], but we expect our results to extend to their efficient implementation.

achieves the same performance as sequential composition of the median mechanism.

3. Finally, we conjecture that under natural assumptions about the distribution of new data entries across phases and the queries requested in each phase, the memory mechanism can provide stronger guarantees in later phases than sequential composition.

In Section 2, we formalize the notion of differential privacy and other background information necessary for the rest of the paper. In Section 3, we restate the median mechanism from [RR10] and reproduce their theorems and proofs. In Section 4 we formalize the sequential composition of the median mechanism and show that the privacy parameter decreases with each additional phase. In Section 5, we propose the memory mechanism, show that its performance matches that of sequential composition of the median mechanism, and we conjecture that the memory mechanism can provide a stronger utility guarantee. Finally in Section 6, we discuss future work to be done of the paper.

2 Preliminaries

This section presents the formal notation and definitions used throughout. We use $\mathbb{R}$ to denote the reals, $\mathbb{Z}$ to denote the integers, and $\mathbb{Z}^+$ to denote the positive integers. For $n \in \mathbb{Z}^+$ we write the nth harmonic number as $H_n = \sum_{i \in [n]} 1/i$. For any set X and $n \in \mathbb{Z}^+$, X^n denotes the set of n-tuples of elements in X.

In our database setting, sach of n data subjects has information described as an element in data universe X. The data for each subject is stored as a row in database $D \in X^n$. We consider mechanisms that operate on a database $D \in X^n$ and answer a sequence of $k \in \mathbb{Z}^+$ queries $f = (f_1, \ldots, f_k)$. We let $M(D, f)$ denote the random variable describing the distribution of outputs on the specified inputs. Differential privacy bounds how much this random variable can change due to a change in a single row of the database. Databases D and D' are said to be *neighboring* if they differ on a single row, and in this case we write $D \sim D'$. Using this notation, we can now state formal definitions of (ϵ, δ)-usefulness and (α, τ)-privacy:

Definition 2.1. For $\alpha, \tau > 0$, a mechanism $M : X^n \to \mathbb{R}^k$ that responds to k queries is (α, τ)-differentially private if for any pair of neighboring databases $D, D' \in X^n$, any sequence of queries

$f_1, \ldots, f_k : X^n \to \mathbb{R}$, and any subset $S \subseteq \mathbb{R}^k$:

$$\Pr[M(D, f_1, \ldots, f_k) \in S] \leq e^{\alpha} \cdot \Pr[M(D', f_1, \ldots, f_k) \in S] + \tau.$$

Definition 2.2. For $\epsilon, \delta > 0$, a mechanism $M : X^n \to \mathbb{R}^k$ that responds to k queries is (ϵ, δ)-useful if for any database $D \in X^n$ and any sequence of queries $f_1, \ldots, f_k : X^n \to \mathbb{R}$, it provides answers $a_1, \ldots, a_k$ such that with all but probability at most δ each answer is ϵ-accurate, i.e.,

$$\Pr[\forall\ i \in [k], |f_i(D) - a_i| \leq \epsilon] \geq 1 - \delta$$

We strive for δ to be inverse polynomial in k and n so that the mechanism outputs answers within ϵ of the true answer with high probability. We strive for τ to be negligible in k and n so that changing a single element of the input database impacts the probability of any outcome by at most a small factor e^{α} with overwhelming probability.

For any real-valued query $q : X^n \to \mathbb{R}$ and any desired $\alpha > 0$, [DMNS06] show how to construct a mechanism that is $(\alpha, 0)$-differentially private (often simply called α-differentially private) by computing $q(D)$ and adding Laplace noise that is calibrated to the *sensitivity* of the query. The sensitivity of any real-valued query q is the maximum amount it can change due to a single row change, denoted $\Delta(q) = \max_{D \sim D'} |q(D) - q(D')|$. The definition of the Laplace distribution and description of how to calibrate noise to sensitivity are as follows:

Definition 2.3. For any $b > 0$, let $\mathrm{Lap}(b)$ denote the Laplace distribution, with probability density $p(x) = \frac{1}{2b} \exp(-|x|/b)$ for any $x \in \mathbb{R}$.

Theorem 2.4. For any real-valued query $q : X^n \to \mathbb{R}$ and any $\alpha > 0$, the Laplace mechanism $M(D) = q(D) + \mathrm{Lap}(\Delta(q)/\alpha)$ is α-differentially private.

A mechanism can reply to multiple queries via independent output perturbation, also called sequential composition, but the privacy parameter suffers linearly with the number of queries answered. It is not hard to show the following more general result:

Lemma 2.5 (Composition lemma)**.** Let $\alpha_i, \tau_i > 0$ for $i \in [k]$, and let $M_i : X^n \to \mathbb{R}$ be a $(\alpha_i, \tau_i$-

differentially private for each $i \in [k]$. Then the mechanism $M(D) = (M_1(D), \ldots, M_k(D))$ concatenating the outputs of each M_i is $(\sum_{i \in [k]} \alpha_i, \sum_{i \in [k]} \tau_i)$-differentially private.

Finally we note that this work is primarily concerned with mechanisms for approximating *predicate queries* on databases. A predicate over X maps each element in X to a bit. For predicate $f : X \to \{0, 1\}$, we also use f to denote the corresponding predicate query over a database, evaluated as $f(D) = \frac{|\{x \in D : f(X)=1\}|}{|D|}$, which computes the fraction of elements of the database D that satisfies predicate f. Note that predicate queries have sensitivity $1/n$ for databases of size n.

3 The Median Mechanism

3.1 Mechanism

The median mechanism [RR10] is parametrized by privacy and utility parameters $\alpha, \epsilon > 0$, data universe X, and query budget $k \in \mathbb{Z}^+$. It takes as input a database $D \in X^n$ and first initializes a set C of all databases of size $m = \Theta(\frac{\log k \log \frac{1}{\epsilon}}{\epsilon^2})$. Throughout the life of the mechanism, C represents the set of all databases consistent with D based only on the information that the mechanism provides to the analyst.

Queries $f_1, \ldots, f_k$ arrive online. The mechanism categorizes each query f_i as either *easy* ($d_i = 0$) or *hard* ($d_i = 1$) based on how well the query answer on the true database coheres with C. To do this while respecting privacy, the mechanism compares a noisy version of the query's easiness r_i (a measure of similarity between $f_i(D)$ and $f_i(S)$ for each $S \in C$) to a noisy threshold t_i. The mechanism replies to hard queries via output perturbation and further removes from C all databases far away from the noisy reply. The mechanism replies to easy queries with the median value of the query on databases in C.

Note that after receiving a noisy answer to a hard query, the analyst knows exactly how the mechanism updates C, and so upon learning that a query is easy, the analyst already knows the query's median value on C. This observation is important in the privacy analysis of the mechanism, and it is the central reason the mechanism is able to answer exponentially many queries interactively while maintaining meaningful privacy and utility.

Algorithm 1 The median mechanism for privacy and utility parameters $\alpha, \epsilon > 0$, data universe X, and query budget $k \in \mathbb{Z}^+$

```
• Upon initialization with database $D \in X^n$:
      Let $m = \frac{160000 \ln k \ln \frac{1}{\epsilon}}{\epsilon^2}$.
      Let $\alpha' = \frac{\alpha}{720m \ln|X|}$.
      Let $\gamma = \frac{4}{\alpha' \epsilon n} \ln \frac{2k}{\alpha}$.
      Let $C$ be the set of all databases of size $m$.
      Let $i, h = 0$.
• Upon receipt of a new query with $i < k$ and $h < 20m \log|X|$:
      Increment $i$ and let $f_i$ be the new query.
      Let $r_i = \frac{\sum_{S \in C} \exp(-|f_i(D) - f_i(S)|/\epsilon)}{|C|}$ and $\hat{r}_i = r_i + \mathrm{Lap}(\frac{2}{\epsilon n \alpha'})$.
      Let $t_i = \frac{3}{4} + \xi \cdot \gamma$ for $\xi \in \{0, 1, \ldots, \frac{3}{20\gamma}\}$ chosen with probability proportional to $2^{-\xi}$.
      If $\hat{r}_i \geq t_i$,
        Let $d_i = 0$ (easy).
        Let $a_i = \mathrm{median}\{f_i(S) : S \in C\}$.
      Otherwise,
        Let $d_i = 1$ (hard) and increment $h$.
        Let $a_i = f_i(D) + \mathrm{Lap}(\frac{1}{n\alpha'})$.
        Remove from $C$ all $S \in C$ with $|f_i(S) - a_i| > \epsilon/50$.
      Output $(d_i, a_i)$.
```

In this section, we provide a more detailed proof of the below theorem for the median mechanism from [RR10], which serves as the starting point for the analysis of sequential composition of the median mechanism and the memory mechanism described in the following sections.

Theorem 3.1. There exist constants $c_\tau, c_\delta, c_n > 0$ such that for any privacy and utility parameters $\alpha, \epsilon > 0$, data universe X, and query budget $k \in \mathbb{Z}^+$, the median mechanism satisfies (α, τ)-differential privacy and (ϵ, δ)-utility for $\tau = \exp(-\frac{c_\tau \ln k \ln \frac{1}{\epsilon} \ln|X|}{\epsilon^2})$ and $\delta = k \exp(-\frac{c_\delta n \alpha \epsilon^3}{\ln k \ln \frac{1}{\epsilon} \ln|X|})$ when run on databases of size $n \geq \frac{c_n \ln \frac{2k}{\alpha} \ln^2 k \ln \frac{1}{\epsilon} \ln|X|}{\alpha \epsilon^3}$.

Note that for n as above, τ and δ are negligible and inverse polynomial, respectively, in k and n. We separately prove the privacy and utility guarantees in the following two subsections. All of the following lemmas are with respect to arbitrary parameters $\alpha, \epsilon > 0$, X, and $k \in \mathbb{Z}^+$, arbitrary input data $D \in X^n$ for sufficiently large n, and arbitrary predicate queries $f_1, \dots, f_k$. The values of m, α', γ used in the proofs are as initialized by the mechanism.

3.2 Utility Analysis

To establish utility, [RR10] shows that with all but probability $\delta = k \exp(-c_\delta \epsilon n \alpha')$ for sufficiently small constant c_δ, sufficient conditions hold to ensure that every query is answered with ϵ-accuracy and the mechanism does not exceed the hard query budget $h = 20m \log|X|$. Lemmas 3.2 and 3.3 respectively show that with high probability, not too much noise is added to any query's easiness measure and not too much noise is added to any hard query answer. Lemma 3.4 shows that accurate easiness measures guarantee ϵ-accuracy for easy queries. Finally Lemma 3.6 uses a uniform convergence bound to show that if easiness measures are accurate, then the mechanism will not categorize more than the maximum allowed number of queries as hard.

Lemma 3.2. With probability at least $1 - \frac{\delta}{2}$, $|r_i - \hat{r_i}| \leq \frac{1}{100}$ for every query i.

Proof. Recall $\hat{r_i} = r_i + \text{Lap}(\frac{2}{\epsilon n \alpha'})$. Then by definition of the Laplace distribution, we have

$$\Pr[|\hat{r}_i - r_i| > 1/100] = \Pr[|\text{Lap}(2/(\epsilon n \alpha'))| > 1/100] = \exp(-\frac{1}{100} \cdot \frac{\epsilon n \alpha'}{2}).$$

By union bound over every query $i \in [k]$, this event occurs for *any* query with $\leq \delta/2$ probability. □

Lemma 3.3. With probability at least $1 - \frac{\delta}{2}$, the answer to every hard query is $(\frac{\epsilon}{100})$-accurate.

Proof. Recall that an answer to a hard query i is calculated as $a_i = f_i(D) + \text{Lap}(\frac{1}{n\alpha'})$, and a query answer is $(\frac{\epsilon}{100})$-accurate if $|f_i(D) - a_i| \leq \frac{\epsilon}{100}$ by Definition 2.2. Then we have

$$\Pr[|f_i(D) - a_i| > \epsilon/100] = \Pr[|\text{Lap}(1/(n\alpha'))| > \epsilon/100] = \exp(-\frac{\epsilon}{100} \cdot n\alpha').$$

There are at most k hard queries, so union bound again gives us our desired result. □

Lemma 3.4. If $|r_i - \hat{r_i}| \leq \frac{1}{100}$ for every query i, then every answer to an easy query is ϵ-accurate.

Proof. Let C be the current set of hypothetical databases stored by the mechanism when it is about to answer some easy query f_i, and let $G = \{S \in C : |f_i(D) - f_i(S)| \leq \epsilon\}$ denote the subset of good databases in C that align with the real answer $f_i(D)$ within at most ϵ error. If $|G| \geq .51|C|$, the median value of f_i on C is ϵ-accurate. It is therefore enough to prove that if $|r_i - \hat{r_i}| \leq 1/100$ and i is categorized as easy, then $|G| \geq .51|C|$.

Noting that $t_i \geq 3/4$, and i is categorized easy if and only if $\hat{r}_i \geq t_i$, easiness implies $r_i \geq 74/100$ assuming $|\hat{r}_i - r_i| \leq 1/100$. It is therefore enough to show that $|G| < .51|C|$ implies $r_i < 74/100$.

$$
\begin{aligned}
r_i &= \frac{\sum_{S\in G} \exp(-|f_i(D) - f_i(S)|/\epsilon) + \sum_{S\in C\setminus G} \exp(-|f_i(D) - f_i(S)|/\epsilon)}{|C|} \\
&= \frac{|G| + |C\backslash G|/e}{|C|} \\
&= \frac{|G|(1 - 1/e) + |C|/e}{|C|} \\
&< \frac{.51|C|(1 - 1/e) + |C|/e}{|C|} \\
&= .51(1 - 1/e) + 1/e \\
&< 74/100
\end{aligned}
$$

□

Finally, we must show that the mechanism is not likely to abort by classifying too many queries as hard. Lemma 3.6 does this by showing that the set of hypothetical databases C shrinks substantially after every hard query, assuming the conclusion of Lemma 3.2. The following uniform convergence bound determines an appropriate value of m to ensure that one of the hypothetical databases simulates D for all k queries, guaranteeing that C contains at least one database that will survive all phases of contraction.

Proposition 3.5. (Uniform Convergence Bound). For every collection of k predicate queries $f_i, \ldots, f_k$ 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

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

In particular, some database S of size $m = \frac{160000 \ln k \ln \frac{1}{\epsilon}}{\epsilon^2}$ satisfies $|f_i(D) - f_i(S)| \le \frac{\epsilon}{400}$ for all $i \in [k]$.

Lemma 3.6. If $|\hat{r_i} - r_i| \le \frac{1}{100}$ for every query i and every answer to a hard query is $(\frac{\epsilon}{100})$-accurate, then the median mechanism classifies fewer than $20m \log|X|$ queries as hard.

Proof. We track how C contracts as we answer hard queries. For any hard query i, we have:

$$r_i \le \underbrace{\overbrace{\hat{r}_i + \frac{1}{100}}^{\text{By assumption}} \overbrace{< t_i + \frac{1}{100}}^{\xi \le \frac{3}{20\gamma} \text{ so } t_i = \frac{3}{4} + \xi\gamma \le \frac{90}{100}}}_{i \text{ is hard so } \hat{r}_i < t_i} \le \frac{91}{100}$$

At least 6% of the databases $S \in C$ have $|f_i(S) - a_i| > \epsilon/50$, because otherwise:

$$r_i = \frac{\sum_{S \in C} \exp(-|f_i(D) - f_i(S)|/\epsilon)}{|C|} > \frac{94}{100} e^{-\frac{1}{50}} > \frac{92}{100}$$

Let h be the number queries classified as hard out of the k total queries. Then noting that initially $|C| = |X|^m$, the size of C after all k queries can be bounded as:

$$|C| \le (\frac{94}{100})^h |X|^m$$

The uniform convergence bound says that some database in C must survive all hard queries, so:

$$h \le \frac{1}{\ln(\frac{100}{94})} m \ln|X| < 20m \ln|X|$$

□

3.3 Privacy Analysis

This section reproduces the proofs from [RR10] to show that the median mechanism is (α, τ)-differentially private. The median mechanism essentially outputs two values: a vector query answers

$a \in \mathbb{R}^k$ and a vector $d \in \{0,1\}^k$ indicating whether each query f_i was classified as easy ($d_i = 0$) or hard ($d_i = 1$). Since an analyst can compute easy answers herself and Laplace perturbation is added to hard answers, the privacy of a is straightforward. This section focuses on arguing that the noise encapsulated in d suffices for privacy without causing the mechanism's behavior to diverge much on neighboring inputs with more than probability τ.

In Lemma 3.7, we first reproduce the proof that the easiness measure r_i has small sensitivity, which informs how much noise should be added to r_i for privacy. Along with the correct calibration of noise added to hard query answers, this ensures that (d_i, a_i) for any particular query suffers privacy cost $2\alpha'$. However, this is not enough for our desired privacy guarantee for the overall mechanism, which must answer k queries, many of them easy. In Lemma 3.8, we show that with overwhelming probability, the thresholds t_i generated by the mechanism are good in that most queries are classified as very easy, which in turn allows [RR10] to bound the probability differences between the behavior of the mechanism on neighboring databases as required in Lemma 3.9.

Lemma 3.7. For every fixed set C of databases and predicate query f, the easiness function $r_i(D) = \frac{\sum_{S \in C} \exp(-|f(D)-f(S)|/\epsilon)}{|C|}$ has sensitivity $\Delta(r_i) = \frac{2}{\epsilon n}$.

Proof. Noting that the sensitivity of any predicate query is n, for any predicate query f, set C of databases, and neighboring databases D and D' of size $n \in \mathbb{N}$, we have:

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

where the second to last inequality holds as long as $n \geq 1/\epsilon$, which is implied by the bound on n in Theorem 3.1. □

The next lemma demonstrates that with all but probability $\tau = \exp(-c_\tau m \ln|X|)$ for sufficiently small constant $c_\tau > 0$, all but $180m \ln|X|$ of the thresholds t_i generated randomly by the mechanism are *good*. A threshold t_i is good for query i if the query was categorized as easy ($d_i = 0$) and its noiseless easiness value r_i exceeds t_i by at least γ.

	$d_i = 0$	$d_i = 1$
$r_i \geq t_i + \gamma$	t_i is good	t_i is bad
$r_i < t_i + \gamma$	t_i is bad	t_i is bad

Lemma 3.8. For every database D, with all but τ probability, the thresholds t generated by the median mechanism are good for its output (d, a).

Proof. Lemma 3.6 shows that there are at most $20m \ln|X|$ queries i with $d_i = 1$. It suffices to show that at most $160m \ln|X|$ queries i have $r_i < t_i + \gamma$. Let Y_i be a random variable indicating $r_i < t_i + \gamma$, and let $Y = \sum_{i \in [k]} Y_i$. We first show that queries i with $r_i \geq 9/10$ contribute at most $m \ln|X|$ to Y, and then we show that queries i with $r_i < 9/10$ contribute at most $159m \ln|X|$.

Suppose $r_i \geq 9/10$. Then $Y_i = 1$ only if $t_i = 9/10$. Since $t_i = \frac{3}{4} + \gamma \cdot \xi$ with $\xi \in \{0, 1, \ldots, \frac{1}{\gamma} \cdot \frac{3}{20}\}$, the only way to have $t_i = 9/10$ is if $\xi = \frac{1}{\gamma} \cdot \frac{3}{20}$, which occurs with probability proportional to $2^{-3/(20\gamma)}$. With $\gamma = \frac{4}{\alpha' \epsilon n} \ln \frac{2k}{\alpha}$ and $n \geq \frac{30 \ln \frac{2k}{\alpha} \log k}{\alpha' \epsilon}$, as implied by the bound in Theorem 3.1, this event occurs with probability $\leq 1/k$. Therefore such queries contribute at most 1 to Y in expectation. Since the t_i are chosen independently at random for each i, the Chernoff bound implies that the probability that there are more than $m \ln|X|$ such queries is at most $\tau/2$.

Now suppose $r_i < 9/10$. Let T be the set of all possible thresholds t_i such that $r_i < t_i + \gamma$. Let s_i be the smallest threshold in T. Note that $|T| > 1$. By choosing ξ proportional to $2^{-\xi}$, we guarantee that $\Pr[t_i \in T_i] \leq 2 \Pr[t_i \in T_i \backslash \{s_i\}] + c/k$ for some constant c. Note that for every threshold $t_i \in T \backslash \{s_i\}$, $t_i > r_i$. Together, these observations give us:

$$\begin{aligned}
\Pr[t_i > \hat{r}_i] &\geq \Pr[t_i > r_i] \cdot \Pr[\text{Lap}(\frac{2}{\epsilon n \alpha'}) \leq 0] \\
&= \Pr[t_i \in T_i \backslash \{s_i\}] \cdot \frac{1}{2} \\
&\geq \frac{1}{4}(\Pr[t_i \in T_i] - c/k) \\
&= \frac{1}{4}\Pr[r_i < t_i + \gamma] - c/(4k)
\end{aligned}$$

The mechanism ensures that the total number of i with $t_i > \hat{r}_i$ is at most $20m \ln|X|$. Then with linearity of expectation, the Chernoff bound implies that queries with $r_i \leq 9/10$ contribute at most $159m \ln|X|$ to Y except with probability $\tau/2$. □

Let $MM(D, f)$ denote either the distribution of outputs (d, a) or the distribution of outputs (t, d, a) for internally chosen thresholds t, we observe that by the previous lemma, we have

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

The following lemma therefore suffices for privacy.

Lemma 3.9. For any neighboring databases D and D', queries $f = (f_1, \ldots, f_k)$, outputs (d, a), and corresponding good thresholds t, we have $\Pr[MM(D, f) = (t, d, a)] \leq e^{\alpha} \Pr[MM(D', f) = (t, d, a)]$.

Proof. For any query i, let $\mathcal{E}_i$ denote the event that $MM(D, f)$ matches the target output (d, a) on the first i queries. Let $\mathcal{E}'_i$ denote the analogous event for $MM(D', f)$. Let b_i indicate that $MM(D, f)$ classifies query i as hard, and let b'_i indicate that $MM(D', f)$ classifies query i as hard.

Both b_i and b'_i depend on C, so we condition on the events $\mathcal{E}_{i-1}$ and $\mathcal{E}'_{i-1}$ respectively to ensure that the mechanisms running on D and D' have the same C when processing query i. The randomness of the threshold is independent of the state of the mechanism, and since the mechanism adds $\text{Lap}(\frac{2}{\alpha' \epsilon n})$ noise to r_i, which by Lemma 3.7 has sensitivity $\frac{2}{\epsilon n}$, the single-query categorization process is α'-differentially private in the following sense:

$$\Pr[b_i = 0 \mid \mathcal{E}_{i-1}] \leq e^{\alpha'} \cdot \Pr[b'_i = 0 \mid \mathcal{E}'_{i-1}] \tag{3.1}$$

$$\Pr[b_i = 1 \mid \mathcal{E}_{i-1}] \leq e^{\alpha'} \cdot \Pr[b'_i = 1 \mid \mathcal{E}'_{i-1}]. \tag{3.2}$$

To evaluate the respective probabilities of a particular a_i for the mechanism running on neighboring databases, we first consider the case that the target classification of i is hard ($d_i = 1$), and then we consider the case that the target classification of i is easy ($d_i = 0$).

Suppose $d_i = 1$ and let s_i and s'_i denote the mechanism's noisy answer to query i when running on D and D', respectively. Agreeing with the target output on query i requires agreeing with both the target classification and the target answer, which are subject to independent perturbations, so:

$$\Pr[\mathcal{E}_i \mid \mathcal{E}_{i-1}] = \Pr[b_i = 1 \mid \mathcal{E}_{i-1}] \cdot \Pr[s_i = a_i \mid \mathcal{E}_{i-1}]$$
$$\Pr[\mathcal{E}'_i \mid \mathcal{E}'_{i-1}] = \Pr[b'_i = 1 \mid \mathcal{E}'_{i-1}] \cdot \Pr[s'_i = a_i \mid \mathcal{E}'_{i-1}]$$

Then by Equation 3.2 and the noise added to easiness computations and hard query answers:

$$P[\mathcal{E}_i \mid \mathcal{E}_{i-1}] = e^{2\alpha'} P[\mathcal{E}'_i \mid \mathcal{E}'_{i-1}] \tag{3.3}$$

Now suppose $d_i = 0$ and let m_i the median value of f_i on C conditioning on $\mathcal{E}_i$ or $\mathcal{E}'_i$. Then we have the following possibilities:

$$\Pr[\mathcal{E}_i \mid \mathcal{E}_{i-1}] = \begin{cases} 0 & \text{if } m_i \neq a_i \\ \Pr[b_i = 0 \mid \mathcal{E}_{i-1}] & \text{if } m_i = a_i \end{cases}$$

and similarly for $\Pr[\mathcal{E}'_i \mid \mathcal{E}'_{i-1}]$. As before, we can argue that $\Pr[\mathcal{E}_i \mid \mathcal{E}_{i-1}] \leq e^{\alpha'} \Pr[\mathcal{E}'_i \mid \mathcal{E}'_{i-1}]$, but paying this cost for all the easy queries will quickly exceed our privacy budget.

Note that since we only have to compare target outputs for possible runs of the mechanism, assuming events $\mathcal{E}_{i-1}, \mathcal{E}'_{i-1}$ and $d_i = d'_i = 0$, it is safe to also assume that $m_i = m'_i = a_i$, so it suffices to bound $\Pr[\mathcal{E}_i \mid \mathcal{E}_{i-1}] \leq 1$ with respect to $\Pr[\mathcal{E}'_i \mid \mathcal{E}'_{i-1}] = \Pr[b'_i = 0 \mid \mathcal{E}'_{i-1}]$. Let r_i and r'_i denote the true easiness of query i for $MM(D, f)$ and $MM(D', f)$ given $\mathcal{E}_{i-1}$ and $\mathcal{E}'_{i-1}$, respectively. Suppose additionally that $r_i \leq t_i + \gamma$, which is true for all but $180m \ln|X|$ thresholds by the assumption that t is good for (d, a). By the sensitivity of r_i, we also have $r'_i \geq t_i + \gamma - \frac{2}{\epsilon n} \geq t_i + \gamma/2$. This means that i is classified easy by $MM(D', f)$ whenever MM adds $> -\gamma/2$ noise to r'_i:

$$\begin{aligned}
\Pr[b'_i = 0 \mid \mathcal{E}'_{i-1}] &\geq \Pr[r'_i - \hat{r}'_i < \gamma/2] \\
&= \Pr[\mathrm{Lap}(\frac{2}{\epsilon n \alpha'}) > -\gamma/2] \\
&= 1 - \frac{1}{2} e^{-\gamma \epsilon n \alpha'/4} \\
&= 1 - \frac{\alpha}{4k}
\end{aligned}$$

Rearranging this, noting that $\Pr[\mathcal{E}_i \mid \mathcal{E}_{i-1}] \leq 1$:

$$\Pr[\mathcal{E}_i \mid \mathcal{E}_{i-1}] \leq (1 - \frac{\alpha}{4k})^{-1} \Pr[\mathcal{E}'_i \mid \mathcal{E}'{}_{i-1}]. \tag{3.4}$$

Applying Equation 3.3 to at most $180m \ln|X|$ bad queries and Equation 3.4 to all other queries, we complete the proof as follows:

$$\begin{aligned}
\Pr[MM(D, f) = (t, d, a)] &= \prod_{i=1}^{k} P[\mathcal{E}_i \mid \mathcal{E}_{i-1}] \\
&\leq e^{360\alpha' m \ln|X|} \cdot (1 - \frac{\alpha}{4k})^{-k} \cdot \prod_{i=1}^{k} \Pr[\mathcal{E}'_i \mid \mathcal{E}'_{i-1}] \\
&\leq e^{\alpha} \cdot \Pr[MM(D', f) = (t, d, a)]
\end{aligned}$$

□

4 The Median Mechanism for a Growing Database

Although the median mechanism allows an analyst to submit queries interactively, it assumes that the database is fixed. We now show how to run the mechanism multiple times so that an analyst may ask queries as the database grows, and we give privacy and utility results for this setting.

4.1 Mechanism

We consider $K \in \mathbb{Z}^+$ phases of database growth, where each phase involves n entries being added to the database. For every growth phase $j \in [K]$, we initialize a new run of the median mechanism on the larger database for up to k queries with a fixed utility parameter but a decreasing privacy parameter.

Algorithm 2 Sequential composition of the median mechanism for privacy and utility parameters $\alpha, \epsilon > 0$, data universe X, query budget $k \in \mathbb{Z}^+$, $c_K > 1$, and number of phases $K \in \mathbb{Z}^+$

```
• Upon initialization with a database size $n \in \mathbb{Z}^+$:
      Let $D$ be a database of size 0 over $X$.
      Let $j = 0$.
• Upon receipt of $\geq n$ new data entries in $X$ with $j < K$:
      Increment $j$ and let $D$ be the concatenation of itself with the new data.
      Initialize the median mechanism with parameters $c_K\alpha/j, \epsilon, X, k$ and input database $D$.
      Forward up to $k$ queries to the current instantiation of the median mechanism.
```

The following subsection sketches a generalization of the argument that if we use privacy parameter $c_K\alpha/j$ for phase $j \in [K]$, we lose only a $\log K$ factor in the privacy parameter, as long as $K \leq c_K(2k/\alpha)^{c_K-1}$. The following theorem states this result formally:

Theorem 4.1. There exist constants $c_\tau, c_\delta, c_n > 0$ such that for any privacy and utility parameters $\alpha, \epsilon > 0$, data universe X, query budget $k \in \mathbb{Z}^+$, $c_K > 1$, and number of phases $K \leq c_K(\frac{2k}{\alpha})^{c_K-1}$, sequential composition of the median mechanism as described above satisfies $(c_K H_K \alpha, K\tau)$-differential privacy and $(\epsilon, K\delta)$-utility for $\tau = \exp(-\frac{c_\tau \ln k \ln\frac{1}{\epsilon}\ln|X|}{\epsilon^2})$ and $\delta = k\exp(-\frac{c_\delta n\alpha\epsilon^3}{\ln k \ln\frac{1}{\epsilon}\ln|X|})$ when initialized with database size $n \geq \frac{c_n \ln\frac{2k}{\alpha}\ln^2 k \ln\frac{1}{\epsilon}\ln|X|}{\alpha\epsilon^3}$.

4.2 Utility and Privacy Analysis

For fixed $\alpha, \epsilon > 0, X, k \in \mathbb{Z}^+ c_K > 1, K \leq c_K(2k/\alpha)^{c_K-1}$, and for constant c_n as in the above theorem, let $C = \frac{c_n \ln^2 k \ln\frac{1}{\epsilon}\ln|X|}{\epsilon^3}$. It is enough to show that $n \geq C\frac{\ln\frac{2k}{\alpha}}{\alpha}$ as required for a single phase of the median mechanism implies $jn \geq C\frac{\ln\frac{2k}{c_K\alpha/j}}{c_K\alpha/j}$ for $j \leq c_K(\frac{2k}{\alpha})^{c_K-1}$, allowing us apply the median mechanism results for phase j with privacy parameter $c_K\frac{\alpha}{j}$. Then the composition theorem will give $(c_K H_K \alpha, K\tau)$-privacy, and the results from the median mechanism's utility proof will compose with no change in ϵ and δ suffering linearly with K by union bound. We apply the bounds on n and j as follows to get this result:

$$\begin{aligned} jn &\geq jC\frac{\ln\frac{2k}{\alpha}}{\alpha} \\ &\geq C\frac{\ln(\frac{2k}{\alpha})^{c_K}}{c_K\alpha/j} \\ &\geq C\frac{\ln\frac{2k}{c_K\alpha/j}}{c_K\alpha/j} \end{aligned}$$

5 The Memory Mechanism

5.1 Mechanism

The memory mechanism seeks to preserve the information about previous phases that repeated independent application of the median mechanism ignores. For simplicity, first consider this two-phased scenario. Our mechanism will start with an initial database D_1 of size n and answers k queries, consistent with the median mechanism. In phase 2, we append another database D_2 of size n to the original database D_1 and answer another set of k queries. In order to answer the phase 2 queries, we replace the set of hypothetical databases at the end of phase 1 with its cross product with a new set of all databases of size m at the beginning of phase 2, and then we proceed to answer the phase 2 queries using this larger space of hypothetical databases. We present this idea generalized to K-phased scenario, reducing our privacy parameter as the database increases size as we did when analyzing sequential composition of the median mechanism.

The size of the databases in C increases by m for each phase, but the number of new data entries each phase needn't be the same as long as each phase it is at least $n_{\min}$. Comparing $f(D)$ to the result of the query run on database $S \in C$, we must appropriately weight the blocks of m rows in S. For predicate query f and a set C of databases of size jm for some $j \in [K]$, define:

$$f^C(S) = \sum_{\ell \in [j]} \frac{|D_\ell|}{|D|} f(S_\ell) \quad \text{where } S_\ell \text{ represents the } \ell\text{th block of } m \text{ rows of } S.$$

Algorithm 3 The memory mechanism for privacy and utility parameters $\alpha, \epsilon > 0$, data universe X, query budget $k \in \mathbb{Z}^+$, and number of phases $K \in \mathbb{Z}^+$

```
• Upon initialization:
    Let $m = \frac{160000 \ln(Kk) \ln \frac{1}{\epsilon}}{\epsilon^2}$
    Let $n_{\min} = \frac{21600 m \ln \frac{2k}{\alpha} \log_2 k \ln|X|}{\alpha\epsilon}$
    Let $D$ be a database of size 0 over $X$.
    Let $C$ be a set of databases containing a single database of size 0 over $X$.
    Let $j, i, h = 0$.

• Upon receipt of $\geq n_{\min}$ new data entries with $j < K$:
    Increment $j$.
    Replace $D$ with its concatenation with the new data $D_j$.
    Replace $C$ with its cross product with the set of all databases of size $m$.
    Let $\alpha'_j = \frac{\alpha}{720 jm \ln|X|}$.
    Let $\gamma_j = \frac{4}{\alpha'_j \epsilon |D|} \ln \frac{2k}{\alpha/j}$.
    Let $i_{\max} = i + k, h_{\max} = 20jm \log|X|$.

• Upon receipt of a new query with $i < i_{\max}$ and $h < h_{\max}$:
    Increment $i$ and let $f_i$ be the new query.
    Let $r_i = \frac{\sum_{S \in C} \exp(-|f_i(D) - f_i^C(S)|/\epsilon)}{|C|}$ and $\hat{r}_i = r_i + \mathrm{Lap}(\frac{2}{\epsilon |D| \alpha'_j})$.
    Let $t_i = \frac{3}{4} + \xi \cdot \gamma_j$ for $\xi \in \{0, 1, \ldots, \frac{3}{20\gamma_j}\}$ chosen with probability proportional to $2^{-\xi}$.
    If $\hat{r}_i \geq t_i$,
      Let $a_i = \mathrm{median}\{f_i^C(S) : S \in C\}$.
    Otherwise,
      Let $a_i = f_i(D) + \mathrm{Lap}(\frac{1}{|D|\alpha'_j})$.
      Remove from $C$ all $S \in C$ with $|f_i^C(S) - a_i| > \epsilon/50$ and increment $h$.
    Output $a_i$.
```

Our privacy and utility results for the memory mechanism are summarized in Theorem 5.1. We note that the mechanism may refuse to either accept a new set of data that contains too few entries or process queries beyond the per-phase total query budget of k in a given phase. Avoiding these events are the user's responsibility, whereas reaching $h_{\max}$ before reaching $i_{\max}$ is considered a utility failure of the mechanism, because a user should be allowed to make k arbitrary queries and cannot know before the request whether a given query will be easy or hard. This is the purpose of Lemma 5.3 in conjunction with the other lemmas in the utility section. An $(\epsilon, K\delta)$-usefulness guarantee therefore ensures that with all but probability $K\delta$, the mechanism will answer the first k queries in each phase with ϵ-accuracy. Since the minimum size increase threshold and the number of queries per phase is not affected by the data itself, no-ops caused by users failing to abide by these restrictions impose no additional privacy cost, so the privacy guarantee has the exact same meaning as in the single-phase median mechanism.

Theorem 5.1. There exist constants $c_\tau, c_\delta > 0$ such that for any privacy and utility parameters $\alpha, \epsilon > 0$, data universe X, query budget $k \in \mathbb{Z}^+$, and number of phases $K \in \mathbb{Z}^+$, the memory mechanism satisfies $(H_K\alpha, K\tau)$-differential privacy and $(\epsilon, 2K\delta)$-utility for $\tau = \exp(-\frac{c_\tau \ln k \ln \frac{1}{\epsilon} \ln|X|}{\epsilon^2})$ and $\delta = k\exp(-\frac{c_\delta n\alpha\epsilon^3}{\ln k \ln \frac{1}{\epsilon} \ln|X|})$.

5.2 Utility and Privacy Analysis

Our proof of usefulness follows that the proof structure in [RR10]. With high probability, not too much noise is added to any query's easiness r_i and all hard queries are answered $\epsilon/100$-accurately; Lemmas 3.2 and 3.3 present and prove these results for the median mechanism. Here we present the first of these results for the memory mechanism (Lemma 5.2) with proof to show how the proof must be modified in the multi-phase setting; the second result can be proven analogously. Lemma 3.4 for the median mechanism shows that the former event is enough to guarantee that all easy queries are answered ϵ-accurately; its proof is independent of database size and privacy parameters, so the analogous result for the memory mechanism is immediate. Finally we present and prove that with high probability the memory mechanism does not classify too many queries as hard in *any* of the K phases (Lemma 5.3, analogous to Lemma 3.6 for the median mechanism). Together, these results give our $(\epsilon, K\delta)$-usefulness result.

Lemma 5.2. With all but $< K\delta$ probability, $|\hat{r}_i - r_i| \leq 1/100$ for every query i.

Proof. For any query i in phase $j \in [K]$, we add $\text{Lap}(\frac{2}{\epsilon|D|\alpha'_j})$ noise to r_i. Noting that the parameter can be bounded by $\frac{2}{\epsilon|D|\alpha'_j} \le \frac{1}{15 \ln \frac{2k}{\alpha} \log_2 k}$, we have:

$$\begin{aligned}
\Pr[|\hat{r}_i - r_i| \le 1/100] &\le 2 \cdot \Pr[\mathrm{Lap}(\frac{1}{15 \ln \frac{2k}{\alpha} \log_2 k}) \le -1/100] \\
&= \exp(-15 \ln \frac{2k}{\alpha} \log_2 k/100) \\
&\le \frac{1}{k^{c_\delta \ln \frac{2k}{\alpha}}} \\
&= \delta/k.
\end{aligned}$$

Then by union bound over the maximum number of queries Kk, we get our desired result. □

Lemma 5.3. If $|\hat{r}_i - r_i| \le 1/100$ for every query i and $|a_i - f_i(D)| \le \epsilon/100$ for every hard query i, then $h \le 20jm \log|X|$ at the end of any phase $j \in [K]$.

Proof. By the argument given in RR, $|C|$ decreases by at least 6% after each hard query and increases by a multiplicative factor $|X|^m$ at the beginning of each new phase. Hence if h_j denotes h at the end of phase $j \in [K]$ and if c_j denotes $|C|$ at the end of phase $j \in [K]$, we have

$$c_j \le \left(\frac{94}{100}\right)^{h_j} |X|^{jm}$$

Applying the uniform convergence bound given in RR, we know there exists a database $S_j^* \in X^m$ for each $j \in [K]$ such that $|f_i(D_j) - f_i(S_j^*)| \le \epsilon/100$ for each hard query i in phases $j, \ldots, K$. The concatenation of these S_j^* databases remains in C since for hard query i in phase $j \in [K]$, we have:

$$\begin{aligned}
|f_i^C(S_1^*|| \ldots ||S_j^*) - a_i| &\le |f_i^C(S_1^*|| \ldots ||S_j^*) - f_i(D)| + |f_i(D) - a_i| \\
&\le |\sum_{\ell \in [j]} \frac{|D_\ell|}{|D|}(f_i(S_\ell^*) - f_i(D_\ell))| + \frac{\epsilon}{100} \\
&\le \frac{\epsilon}{100} \cdot \sum_{\ell \in [j]} \frac{|D_\ell|}{|D|} + \frac{\epsilon}{100} \\
&= \epsilon/50.
\end{aligned}$$

Thus $c_1, \dots, c_K \geq 1$, so we conclude that for each $j \in [K]$, we have

$$h_j \leq \frac{1}{\ln \frac{100}{94}} jm \ln|X| \leq 20jm \log|X|.$$

□

The privacy result for the memory mechanism follows with minimal modification to the privacy analysis for the median mechanism. A direct analog of Lemma 3.8 for the median mechanism establishes with all but $K\tau$ probability, the memory mechanism generates good thresholds. An analog of Lemma 3.9 updated to reflect the differing values of α' in each phase shows that the difference in probability of any particular output for neighboring databases is bounded conditioning on the event that the mechanism generates good thresholds. As with the median mechanism, these two results are enough to give the desired privacy guarantee.

5.3 Conjectured Utility Improvements

We remark that Theorem 5.1 for the memory mechanism does not illustrate an asymptotic improvement over the results for sequential composition of the median mechanism, and it seems likely that these results are tight in the worst case. However, the state of the memory mechanism captures strictly more information than that of the median mechanism. We therefore conjecture that for some notion of a typical use case of the memory mechanism, it is possible to improve on the utility guarantees with no further privacy cost. Here we briefly describe two classes of assumptions we might reasonably make about typical use cases that would allow us to answer more queries in subsequent phases with greater accuracy.

For many realistic settings, it may be fair to assume that the database composition does not change too much across phases. Of course if the exact same entries are received in each phase, the data after the first phase are useless. However, if any constant number of predicate queries are allowed to have arbitrarily different answers across phases, we can assume that all but a negligible fraction of the $2^{|X|}$ total possible predicate queries are meaningfully affected. For super-constant k, this means that a vanishing fraction of queries asked in a particular phase will have an answer significantly different from the answer that would have been provided in an earlier phase.

We may combine with this assumption the assumption that the analyst does not ask too many new queries in each phase. On one hand, a core function of our mechanism is to support analysis that is interactive not only in response to new information from fixed data but also in response to information learned from new data. On the other hand, an analyst's primary questions of interest will not necessarily change much, even as the answers to these questions evolve with new data. If we allow an analyst to ask only a constant number of new queries each phase, the prior assumption that the new data only affects answers for a constant number of new queries ensures that only a constant fraction of queries each phase will be hard.

We hope that these two assumptions together may allow us to provide refined utility analysis for later phases when more is known about the data. If this refined analysis is not possible for the memory mechanism as is, the assumptions may provide guidance for how to modify the memory mechanism to answer queries in a way that makes more careful use of the statefulness of the mechanism and in turn yields better utility.

6 Conclusion

In this paper, we examined the inner workings of the median mechanism by deconstructing the proofs in [RR10] and establishing a baseline for the future work, motivated by the need for a mechanism that can accommodate a dynamically growing database. To establish that it is possible to handle this setting, we analyzed the privacy and utility of both sequential composition of the median mechanism and our new memory mechanism.

We have shown that sequential composition gives a privacy guarantee that suffers only logarithmically in the number of phases of database growth. The worst case performance for the memory mechanism matches that of sequential composition. In the future, we hope to formally prove that under natural assumptions, keeping track of both the composition of smaller databases from previous phases as well as the information the analyst has learned about them will allow for improved utility in later phases.

References

[BLR08] A Blum, K Ligett, and A Roth. A learning theory approach to non-interactive database privacy. *In Proceedings of the 40th ACM Symposium on Theory of Computing (STOC)*, pages 609–618, 2008.

[DMNS06] C Dwork, F McSherry, K Nissim, and A Smith. Calibrating noise to sensitivity in private data analysis. *In Proceedings of the 3rd Theory of Cryptography Conference (TCC)*, pages 265–284, 2006.

[GRS09] A Ghosh, T Roughgarden, and M Sundararajan. Universally utility-maximizing privacy mechanisms. *In Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, pages 351–360, 2009.

[MT07] F McSherry and K Talwar. Mechanism design via differential privacy. *In Proceedings of the 48th ACM Symposium on Foundations of Computer Science (FOCS)*, pages 94–103, 2007.

[RR10] T Roughgarden and A Roth. Interactive privacy via the median mechanism. *In Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 765–774, 2010.