Suppression Strikes Back: On the Interaction of Thresholding and Differential Privacy

Xi Wu[†] Wentao Wu[†] Chen Zeng[‡] Jeffrey F. Naughton[†]
[†]University of Wisconsin-Madison
[‡]Google, Inc.
{xiwu, wentaowu, naughton}@cs.wisc.edu, zengc@google.com

ABSTRACT

Thresholding is a ubiquitous privacy technique that is based on *suppression*. Differential privacy, on the other hand, is a recent probabilistic privacy notion that relies on *perturbation*. Given the vast difference between the ideas and approaches underlying these two, it is interesting to ask if each has anything to contribute to the other, or if they are in fact orthogonal and separate. In this paper, we consider two challenges: (1) *ensuring data fidelity* and (2) *privacy in the presence of a dynamic universe* that arise in the context of differential privacy over evolving data sets. We find that, for both problems, thresholding is crucial for achieving the desired privacy-utility tradeoff. Perhaps more importantly, for the "dynamic universe" problem where one considers an *open world* of an evolving data universe, we show that thresholding is indeed *integral* for achieving differential privacy. Motivated by these observations, we extend previous work by establishing a more general framework within which to study the effectiveness of suppression in achieving desired privacy-utility tradeoff. Our results suggest that thresholding and differential privacy do interact in a meaningful way, and that perhaps further study of this interaction is warranted.

1. INTRODUCTION

Thresholding is an old but ubiquitous privacy technique that is still used by many data publishers [11, 21]. Specifically, thresholding publishes a value only if it is above certain predefined threshold. Differential privacy [3, 4], on the other hand, is a recent, probabilistic privacy notion currently studied by the data management and theoretical research community almost to the exclusion of all others. Due to the vast difference between them, one might reasonably ask if thresholding is even useful at all given the current move to differential privacy.

In this paper we study this question. Our answer is that, not only is thresholding a useful tool for improving *data utility* given differential privacy, but sometimes it can also be *integral* in achieving differential privacy. We start with two observations on using thresholding to improve data utility. Both observations exploit the fact that the magnitude of noise for differential privacy only depends on the sensitivity of the query.

- *Maintaining data sparsity*. Suppose we want to publish a high dimensional vector where most values are zero. If we inject noise to achieve differential privacy, the vector will no longer be sparse. This is unsatisfactory because the resulting private vector will require much more storage than the original vector. Thresholding can eliminate this problem, because with high probability, the values that were formerly zero will be small after perturbation, so suppressing values that are below some low threshold will keep the vector sparse.

- *Reducing relative error*. Relative error is a natural measure for the precision of a perturbed value. Given an original value, say v, and some noise to be added, say κ, the relative error will be $|\kappa/v|$. Because κ does not depend on v, suppressing small values in the perturbed vector means that all the remaining values have small relative error with high probability.

Note that differential privacy is trivially retained in the above two scenarios. This is because thresholding is a *post-processing* step in both cases, and it is well known that differential privacy is preserved by post-processing.

We now turn to cases where thresholding has a deeper effect on the privacy-utility tradeoff. Specifically, we examine two challenges: (1) *ensuring data fidelity* and (2) *privacy in the presence of a dynamic universe*. Both show up as important concerns when we consider differential privacy in a dynamic data environment.

Data Fidelity. While differential privacy is powerful for *encouraging participation*, a problem arises when we try to convince *data consumers* of the fidelity of the data. Using standard differential privacy mechanisms, the guarantee on noisy data is essentially the following: it is likely to be only somewhat perturbed from the true value, but it could be arbitrarily far from the true value with *positive* probability. Data consumers sometimes consider such data to be unreliable.

There are a couple of reasons behind this view. For a *static* data set, the fidelity problem might not be a significant concern, because the randomness is effectively injected only once. However, when one turns to a dynamic data set, the problem of "bad randomness" will almost surely happen. Moreover, from the view point of data users, beyond knowing that the noise is probably small, they have no idea about how much a value has actually been perturbed. Finally, the data fidelity problem is made worse due to the long-tail property of many real data sets. Because standard differential-privacy mechanisms add noise independent of the magnitude of the original data values, data values at the tail may suffer from higher relative errors.

Dynamic Universe with Multiple Releases of Data. The standard computation model of differential privacy, namely the *centralized model* [8], assumes a trusted data curator who holds a *static* dataset. As a result of this assumption, one can always assume that databases are drawn from a *fixed* universe U. The privacy guarantee is thus defined for any two databases D, D' *drawn from* U which differ by one element. This model applies in many cases, e.g., when publishing the results of a completed medical study.

However, this model can break down if data is collected and published at multiple points in time, so the universe can change between D and D'. In effect, universe change is common in practice. For example, consider a company that wishes to publish keyword search frequencies. At time t let S be the set of keywords already searched for, and suppose that at time $t+1$ someone searches for $w \notin S$. Then, comparing the published results at t and $t+1$ immediately breaks differential privacy, because the latter contains a new entry labeled by w.

We now give an overview of our results. We first give a range-based mechanism for the data fidelity problem. Given a histogram, instead of presenting users with a noisy value for each point in the universe, we report a range that is guaranteed to enclose the actual value. A natural idea to achieve this is using thresholding with respect to the *noise* of the perturbed value. In other words, we suppress a perturbed value if its injected noise is above certain threshold T. Then, for any noisy value v that survives, a range $[v-T, v+T]$ is guaranteed to include the original value. Unfortunately, we show that one has to sacrifice ε-differential privacy in order to publish non-trivial ranges. Nevertheless, we show that one can still achieve strong (ε, δ)-differential privacy with a small threshold on noise.

Next, we show how to ensure differential privacy under a different model of *neighboring databases* that captures an evolving universe. More specifically, we consider a model where neighboring databases can have different universes, and queries over neighboring databases may have different ranges. To this end, we first show that nontrivial (ε, δ)-differential privacy is not achievable unless we suppress some query results. Indeed, in this case one can prove something akin to "thresholding is essential for achieving differential privacy" (see Theorem 4 in Section 4.1). We then consider a simple mechanism that works as follows. It first injects Laplacian or geometric noise into the data as usual, and then suppresses dimensions with respect to the threshold based on the perturbed values. We prove that this simple strategy can ensure strong (ε, δ)-differential privacy in the case of a dynamic universe.

In both of our above mechanisms, one needs to *suppress* some data in order to achieve the desired privacy-utility tradeoff. Thus, it is interesting to study the interaction between suppression and differential privacy in general. From this perspective, we further provide two extensions of our basic results. In the setting of a dynamic universe, we give a mechanism whose only task is to stabilize the universe based on the utility of universe elements but otherwise publishes exact data. This allows us to *sequentially* compose it with a more advanced mechanism, such as the exponential mechanism [17], for scenarios where one can hope for better privacy-utility tradeoff. We also give a simple framework to study, under the fixed-universe assumption, the benefits of suppression as well as more general data transformations. In this vein, we generalize the results by Zeng et al. [22] and show that differential privacy is oblivious to a large class of probabilistic transformation mechanisms, including many filter-group-by aggregations.

Because suppression means a loss of data, an important question is how suppression affects data utility. We present a theoretical analysis and a calculation hinting that suppression is likely to have only a small effect on utility when publishing large data sets. In the spirit of Dwork and Pottenger [7], our observations provide more evidence that privacy may be "easy" to achieve for large-scale data. It is important future work to consider medium or small-sized data sets, or to evaluate the impact of suppression on utility in specific, concrete settings.

The rest of the paper is organized as follows. We start by providing necessary background on differential privacy in Section 2. Then in Section 3 and 4 we present basic results on data fidelity and dynamic universe, respectively. Extensions of our basic results are given in Section 5. We present theoretical observations about our mechanisms in Section 6, summarize related work in Section 7, and conclude the paper with future directions in Section 8.

2. PRELIMINARIES

We follow Dwork and Roth [8] for standard definitions and notations in differential privacy. Most work on differential privacy considers the standard model of computation, namely the *centralized model*, where one assumes a trusted data curator who holds a *single* and *static* database. As a result, one can always assume that the universe $\mathcal{U}$ of all possible tuples is fixed. We use a non-negative integer to encode a distinct tuple, and call it a *tuple type*.

A database D is encoded as a histogram, that is, a vector in $\mathbb{N}^{\mathcal{U}}$. For example, if $\mathcal{U} = \{2, 4, 6\}$, then the vector $\langle 2 : 1000, 4 : 100, 6 : 50\rangle$ represents a database where there are 1000 instances of tuple 2, 100 instances of tuple 4, and 50 instances of tuple 6. Because we use non-negative integers to represent tuples, there is a natural correspondence between the elements of the universe and the positions in the vector. For example, we will also write the database as $\langle 1000, 100, 50\rangle$ for short if there is no ambiguity about the universe (the first position corresponds to tuple 2, the second to 4, and the third to 6). In this paper, we need to consider databases where the universe can change. For this purpose, we define a *universe* to be any *finite* subset of non-negative integers. We then define the *universe of a database* to be the set of tuple types that appear in the database. For any given database D, this set is denoted as $\mathrm{univ}(D)$.

Consider the case of a fixed universe. Given two databases D and D' in $\mathbb{N}^{\mathcal{U}}$, they are *neighboring* (written as $D \sim D'$) if

$$\|D - D'\|_1 \leq 1,$$

where $\|\cdot\|_1$ is the ℓ_1 norm of a vector. Let $\mathcal{D}$ be the collection of all databases and $\mathcal{M}$ be a probabilistic algorithm. We say that $\mathcal{M}$ is (ε, δ)-differentially private if for any $S \subseteq \mathrm{Range}(\mathcal{M})$ and $D \sim D'$ we have

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta.$$

Moreover, we say that $\mathcal{M}$ is ε-differentially private if it is $(\varepsilon, 0)$-differentially private. Finally, for a database $D \in \mathcal{D}$, the *support* of $\mathcal{M}(D)$ is the set of points that receive nonzero probability mass. Formally, $\mathrm{supp}(\mathcal{M}(D)) = \big\{r \mid \Pr[\mathcal{M}(D) = r] > 0\big\}$. We need the following simple but important fact about ε-differential privacy.

FACT 1. *Let $\mathcal{M}$ be an ε-differentially private mechanism computing certain query for databases in $\mathcal{D}$. As long as an event $\mathcal{E}$ can happen for* some *database, $\mathcal{M}$ must put nonzero probability mass on $\mathcal{E}$ for* any *database in the collection. By considering $\mathcal{E}$ for every output of $\mathcal{M}$, this indicates that for any $D, D' \in \mathcal{D}$, $\mathrm{supp}(\mathcal{M}(D)) = \mathrm{supp}(\mathcal{M}(D'))$.*

Indeed, from the definition of ε-differential privacy, we have that, for any event S, $\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S]$. This forces that if $\Pr[\mathcal{M}(D') \in S] = 0$, then $\Pr[\mathcal{M}(D) \in S] = 0$. The other direction follows by symmetry.

Another important property of differential privacy is that it is preserved by postprocessing.

FACT 2. *Let $\mathcal{M}$ be an (ε, δ)-differentially private mechanism, and $\mathcal{L}$ be a possibly randomized algorithm with* private randomness *that only accesses the* output of $\mathcal{M}$ *as* input. *Then $\mathcal{L} \circ \mathcal{M}$ is (ε, δ)-differentially private. Here,* private randomness *means that all the coin tosses used by $\mathcal{L}$ are independent from those used in $\mathcal{M}$.*

Here is a simple proof when $\mathcal{L}$ is deterministic. Suppose that $\mathcal{M} : \mathcal{D} \mapsto \mathcal{R}$ and $\mathcal{L} : \mathcal{R} \mapsto \mathcal{R}'$. For any $r' \in \mathcal{R}'$, define $\mathcal{L}^{-1}(r') = \{r \in \mathcal{R} : \mathcal{L}(r) = r'\}$. Then, for any $D \sim D'$ and $S' \subseteq \mathrm{Range}(\mathcal{L})$, we have

$$\begin{aligned}\Pr[\mathcal{L}(\mathcal{M}(D)) \in S'] &= \Pr[\mathcal{M}(D) \in \mathcal{L}^{-1}(S')] \\ &\le e^{\varepsilon} \Pr[\mathcal{M}(D') \in \mathcal{L}^{-1}(S')] + \delta \\ &= e^{\varepsilon} \Pr[\mathcal{L}(\mathcal{M}(D')) \in S'] + \delta.\end{aligned}$$

When $\mathcal{L}$ is probabilistic with private randomness, the same proof works by expanding $\mathcal{L}$ by conditioning on each of its coin toss sequence and using the independence assumption.

We will use the geometric mechanism. Let q be a query that maps a database $D \in \mathcal{D}$ to a vector in $\mathbb{R}^d$, for some constant d. First, we define the *sensitivity* of q:

DEFINITION 1 (SENSITIVITY). *Let $q : \mathcal{D} \mapsto \mathbb{R}^d$. The ℓ_1 sensitivity of q is defined as*

$$\Delta q = \max_{D \sim D'} \|q(D) - q(D')\|_1.$$

Let $G(\omega)$ be a two-sided geometric distribution parameterized by $\omega > 0$ such that, for any integer σ,

$$\Pr[G = \sigma] = \frac{1 - \exp(-\omega)}{1 + \exp(-\omega)} \exp(-\omega|\sigma|).$$

Then we have the following:

THEOREM 1 (GEOMETRIC MECHANISM [8]). *The geometric mechanism $\mathcal{M}$ that adds independent noise from $G(\varepsilon/\Delta q)$ to each dimension of q is ε-differentially private.*

Finally, we will also use the Laplace distribution, defined below:

DEFINITION 2 (LAPLACE DISTRIBUTION [8]). *The Laplace distribution (centered at 0) parameterized by scale $b > 0$ is the distribution with probability density function:*

$$\mathrm{Lap}(x|b) = \frac{1}{2b} \exp\big(-\frac{|x|}{b}\big).$$

We need the following fact about Laplace distribution:

FACT 3. *If $X \sim \mathrm{Lap}(b)$ then*

$$\Pr[|X| \ge t \cdot b] \le \exp(-t).$$

The two-sided Geometric distribution $G(\omega)$ has a similar behavior, if one views $1/\omega$ as the scale parameter.

3. DATA FIDELITY

In this section, we provide a case study of the data fidelity challenge to exhibit the power of thresholding in boosting utility. As we discussed in the introduction, in practice it could be unsatisfactory or even unacceptable to release inaccurate data with no *hard* quality guarantee. To address this issue, in this section we study a simple strategy which publishes a *range* that encloses the original value for each dimension in the output vector.

In order to do this with differential privacy, a natural idea is to apply a thresholding *with respect to the noise of the perturbed value*. In other words, we suppress a perturbed value if its injected noise is above certain threshold T. Then, for any noisy value v that survives, a range $[v - T, v + T]$ is guaranteed to include the original value. However, ε-differential privacy has to be sacrificed for such guarantees. To see this, let β be the true value and suppose that we want to publish a range $R = [\alpha, \gamma]$ such that $\beta \in R$. Then ε-differential privacy cannot be achieved for any non-trivial R.

FACT 4. *It is impossible to achieve ε-differential privacy and $\beta \in [\alpha, \gamma]$ simultaneously unless $[\alpha, \gamma] = (-\infty, \infty)$.*

To see this intuitively, let us consider a one-dimensional integral query of sensitivity one. Let $[\alpha, \gamma]$ be any range that is an output of a purported ε-differentially private mechanism $\mathcal{M}$ on input β. Because $\mathcal{M}$ assigns $[\alpha, \gamma]$ with non-zero probability on input β, it must also assign it with non-zero probability on input $\beta - 1$, the neighboring of β. This, however, indicates that $\beta - 1 \in [\alpha, \gamma]$ by our range requirement. Now if we repeat the same argument, but for more "remote" neighbors $\beta - 2, \beta - 3, \ldots$, and $\beta + 1, \beta + 2, \ldots$, it follows that they must be all in the range $[\alpha, \gamma]$. Therefore $[\alpha, \gamma] = (-\infty, \infty)$.

Now we describe a range mechanism $\mathcal{M}$ for integral queries of sensitivity Δ (that is, each dimension of the result vector is an integer). Its extension to arbitrary numeric queries is trivial by using Laplacian noise. Let L be the range size parameter. Given the output vector we wish to sanitize, $\mathcal{M}$ works as follows. We first inject independent geometric noise $G(\varepsilon/\Delta)$ into each dimension. Then for each dimension with original value v and perturbed value v', we publish $R = [v' - L, v' + L]$ if $v \in R$ and $\perp$ otherwise. In the following we show that strong (ε, δ)-differential privacy can be achieved with small L. We start with queries of sensitivity one, and then generalize to arbitrary sensitivity.

The Case of Sensitivity One. Consider the case of a one dimensional query q of sensitivity one, let D, D' be two neighboring databases, where $q(D) = \beta$ and $q(D') = \beta - 1$. Consider $\mathrm{supp}(\mathcal{M}(D))$ and $\mathrm{supp}(\mathcal{M}(D'))$, which consist of ranges that contain β and $\beta - 1$, respectively. We say that a range R is *bad* if it is one of the following two types:

type-(1) $R \in \mathrm{supp}(\mathcal{M}(D))$ but $R \notin \mathrm{supp}(\mathcal{M}(D'))$;

type-(2) $R \notin \mathrm{supp}(\mathcal{M}(D))$ but $R \in \mathrm{supp}(\mathcal{M}(D'))$.

Basically, the bad ranges are points where one cannot ensure ε-differential privacy. Our goal is to bound the probability that these bad points appear and thus the additive loss in differential privacy. Now observe that the *only* type-(1) bad range is $R_1 = [\beta, \beta + 2L]$. In this case, the perturbed value is $\beta + L$, which happens with small probability for sufficiently large L. The same argument applies to the type-(2) bad range. Combining these two, we have:

LEMMA 1. *For one-dimensional integral queries of sensitivity one, the mechanism $\mathcal{M}$ is (ε, δ)-differentially private for counting queries, where*

$$\delta = \frac{1 - e^{-\varepsilon}}{1 + e^{-\varepsilon}} e^{-L\varepsilon}.$$

By a simple union bound, we have the following lemma for d-dimensional integral query of sensitivity one:

LEMMA 2. *The mechanism $\mathcal{M}$ is (ε, δ)-differentially private for d-dimensional integral queries of sensitivity one, where*

$$\delta = d \cdot \frac{1 - e^{-\varepsilon}}{1 + e^{-\varepsilon}} e^{-L\varepsilon}.$$

By Lemma 2, to ensure (ε, δ)-differential privacy, one can set $L = \big(\ln d + \ln(1/\delta)\big)/\varepsilon$. If $\delta = 1/n^2$ (a common setting in practice) where n is the size of the database, then $L = (\ln d + 2 \ln n)/\varepsilon$, which is relatively small even when n is huge.

Arbitrary Sensitivity. More generally, let us further consider d-dimensional integral queries with arbitrary sensitivity Δ (not necessarily $\Delta = 1$). Let $\omega = \varepsilon/\Delta$. Without loss of generality, suppose that $q(D) = \beta$ and $q(D') = \beta - \Delta$. Then type-(1) bad ranges are now not unique. Rather, there are Δ of them:

$$[\beta - \Delta + 1, \beta - \Delta + 2L + 1], \ldots, [\beta, \beta + 2L].$$

As before, the probability of generating each range is equal to the probability of generating the center of that range. Given that these ranges are disjoint, the probability of generating at least one of them is therefore

$$\frac{1 - e^{-\omega}}{1 + e^{-\omega}} \sum_{i=1}^{\Delta} e^{-\omega(-\Delta + L + i)}.$$

By a union bound across all dimensions, and using the fact that we have ensured ε-differential privacy for all other points, we have the following theorem:

THEOREM 2. *The mechanism $\mathcal{M}$ is (ε, δ)-differentially private for d-dimensional integral queries with sensitivity Δ, where $\delta \leq de^{-\omega(L+\Delta-1)}$ with $\omega = \varepsilon/\Delta$.*

Readers may notice that our range mechanism seems to be a "postprocessing" of the differentially private output after injecting geometric noise. However, after this "postprocessing" an ε-differentially private mechanism becomes (ε', δ')-differentially private for some $\delta' > 0$. Given the well-known fact (see Fact 2) that "differential privacy is preserved by postprocessing," is there any contradiction here? The answer is *no*. The reason is that, for Fact 2 to hold, the postprocessing can only access the noisy output and must use *private* randomness. However, the "postprocessing" here needs to access the original value *before* perturbation. Therefore, it implicitly accesses the noise (i.e., the randomness) used by the perturbation phase — it does not use private randomness! Hence, it is not surprising that this will change the privacy guarantee.

4. DYNAMIC UNIVERSE

Perhaps surprisingly, not only is thresholding useful for improving utility, in certain context it is also *indispensable* for the purpose of achieving differential privacy. In this section we provide such a case study by considering a different data collecting and publishing model where the universe of data may change. Universe change may break differential privacy and cause privacy leakage. Consider the following example:

EXAMPLE 1. *Suppose that we have a recommendation system which recommends videos to customers. For simplicity, suppose that the system maintains a list of video pairs $\langle v_1, v_2 \rangle$ such that there is some user who watches v_2 after v_1. One day, Alice shares a link to her family-friendly video v_1^* to Bob and requires him not to redistribute. Bob, on the other hand, watches v_2^* after v_1^*, where v_2^* is not nearly as family-friendly. However, now in the recommendation system there is a new pattern $\langle v_1^*, v_2^* \rangle$. When Alice re-watches v_1^*, the recommendation system tells her "people who watched v_1^* also watched v_2^*." Upon seeing this recommendation, Alice soon realizes that it is Bob who watches v_2^*.*

The key problem here is that Bob's participation breaks his differential privacy. That is, before he participates, there is no pattern $\langle v_1^*, v_2^* \rangle$ in the database of the recommendation system, while there is one after his participation. A careless recommendation system that directly utilizes the new universe element thus causes a significant privacy loss.

The rest of this section is organized as follows. We present our modeling of a dynamic universe and prove negative results in Section 4.1. Then, in Section 4.2, we consider a natural mechanism for integral queries. Specifically, it first injects geometric noise into the data and then applies thresholding with respect to the perturbed values. We show that this mechanism can ensure strong (ε, δ)-differential privacy with a small threshold. Finally, we close this section with a discussion of non-integral queries.

4.1 Modeling and Negative Results

Because the universe of the data set can change in our case, we now have an issue in defining the ℓ_1 distance between two databases of *different* universes. To tackle this issue, we represent each database as a vector in $(\mathbb{N} \cup \{\perp\})^{\mathbb{N}}$, with the understanding that only a finite number of dimensions have non-$\perp$ counts[1]. We stress that one cannot simply take the union of the universes of D and D' and then assume a fixed universe. This is because we are considering a dynamic data set with multiple collections and publishing, and the universe *cannot be fixed a priori*.

Using this representation, we can now define the ℓ_1 distance between two databases with different universes. Given two databases $D, D' \in (\mathbb{N} \cup \{\perp\})^{\mathbb{N}}$, their ℓ_1 distance is the normal ℓ_1 distance between these two by just treating $\perp$ to be 0. Clearly, this is a finite number because both D and D' have finite universes. Finally, we define that two databases are *neighboring* if their ℓ_1 distance is bounded by one. In the following, we still use the familiar notation $D \sim D'$ if D and D' are neighboring.

Now, we say that a query is *universe sensitive* if its range changes as the universe of the underlying database changes. It then immediately follows that for such queries, ε-differential privacy is not achievable. Specifically, let $\mathcal{M}$ be a mechanism computing a certain query q, D and D' be two databases so that $\mathrm{Range}(\mathcal{M}(D)) \neq \mathrm{Range}(\mathcal{M}(D'))$. Then Fact 1 is violated and thus $\mathcal{M}$ cannot be ε-differentially private.

Indeed, a stronger negative result holds. Consider a case where the query computes a vector of numeric values. Furthermore, with universe change of the underlying database, the dimensions of the vector change. We model this by a query that maps a database to a vector in $\mathbb{R}^S$, where S is some finite subset of $\mathbb{N}$. Then, with a dynamic universe, the set S also evolves. We call such queries *unbounded dimensional queries*. Such queries are fairly general. For example, it captures publishing histograms where the universe of the histogram can change. We next show that, if one always publishes *all* the dimensions of the vector, then even (ε, δ)-differential privacy is not achievable.

THEOREM 3. *Let q be an unbounded dimensional query. If one always publishes all the dimensions of the query result, (ε, δ)-differential privacy is not achievable for q unless $\delta \geq 1$.*

[1] We note that this "infinity" only happens in our modeling as a way of viewing a database. When used in an algorithm, a database D is still a finite vector in $\mathbb{N}^{\mathrm{univ}(D)}$.

The reason here is that, if $q(D')$ has a dimension that is not in $q(D)$ for some $D \sim D'$, then insisting that a mechanism $\mathcal{M}$ must publish all dimensions forces that $\mathrm{supp}(\mathcal{M}(D)) \cap \mathrm{supp}(\mathcal{M}(D')) = \emptyset$. This theorem indicates that suppression of some information is necessary if we want to achieve non-trivial differential privacy for publishing new dimensions. Next we show that thresholding is necessary in the sense that, if a dimension gets published with high probability, the statistic it is associated with must be significant. Let us consider a single dimension (w.l.o.g. suppose this is dimension 0) where its statistic is $\perp$ initially. The following theorem holds.

THEOREM 4. *Let v be a positive numeric statistic to publish and suppose that the sensitivity of computing the statistic is Δ. If $\mathcal{M}$ is (ε, δ)-differentially private and it publishes dimension 0 with probability at least p, then*

$$v \geq \Delta(\ln(p/\delta + 1)/\varepsilon - 1).$$

4.2 Thresholding Mechanism

Let q be an unbounded dimensional query. In this section, for integral q, we give a differentially private mechanism, which is a combination of noise injection and thresholding, to compute q. In the following, let the sensitivity of q be Δ. Without loss of generality, suppose that

$$q(D) = (a_1, \dots, a_d)$$

and

$$q(D') = (a'_1, \dots, a'_d, a'_{d+1}, \dots, a'_{d'})$$

where $\|q(D) - q(D')\|_1 \leq \Delta$. Furthermore, $d' - d \leq \Delta$ because q is integral. The mechanism is as follows:

DEFINITION 3 (THRESHOLDING MECHANISM). *Let T be the threshold and ω be the parameter of the geometric mechanism.*

(1) Perturbation (P): *Add independent geometric noise drawn from $G(\omega)$ to each dimension of $q(D)$;*

(2) Thresholding ($\mathcal{T}$): *Let $\mathrm{P}(D)$ be the output of* P *on D. For each $v \in \mathrm{P}(D)$, publish v if $v \geq T$; publish $\perp$ otherwise.*

We now prove that this mechanism is (ε, δ)-differentially private for appropriate T and ω. Our goal is that, for any $S \subseteq (\mathbb{R} \cup \{\perp\})^{d'}$,

$$\Pr[\mathcal{T}(\mathrm{P}(D)) \in S] \leq e^{\varepsilon} \Pr[\mathcal{T}(\mathrm{P}(D')) \in S] + \delta,$$
$$\Pr[\mathcal{T}(\mathrm{P}(D')) \in S] \leq e^{\varepsilon} \Pr[\mathcal{T}(\mathrm{P}(D)) \in S] + \delta.$$

Before moving to the formal justification, let us first give an informal description. We will use a *hybrid argument* by considering a *hybrid* query output $q(\widetilde{D}) = (a'_1, \dots, a'_d)$. $q(\widetilde{D})$ is "hybrid" in the sense that it resembles $q(D)$ since they have the same number of dimensions, and it resembles $q(D')$ since they have the same data in the first d dimensions. At the highest level, when we apply the thresholding mechanism $\mathcal{A}$ to D, D', and $\widetilde{D}$, the following hold:

(i) $\mathcal{A}(D)$ and $\mathcal{A}(\widetilde{D})$ should be close because they have the same number of dimensions and $\|q(D) - q(\widetilde{D})\|_1$ is small;

(ii) $\mathcal{A}(\widetilde{D})$ and $\mathcal{A}(D')$ should also be close. First, their first d dimensions have the same values. Second, the extra dimensions in D' have insignificant values as $\|q(D) - q(D')\|_1$ is small. Therefore, they will be suppressed by a sufficiently large threshold T with high probability.

Combining (i) and (ii), it follows that $\mathcal{A}(D)$ and $\mathcal{A}(D')$ are close to each other as well. We now carry out this argument formally. (i) follows from the standard argument of injecting Geometric noise.

LEMMA 3. *For any $S \subseteq (\mathbb{R} \cup \{\perp\})^{d'}$,*

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\delta\Delta} \Pr[\mathcal{A}(\widetilde{D}) \in S], \quad (1)$$
$$\Pr[\mathcal{A}(\widetilde{D}) \in S] \leq e^{\delta\Delta} \Pr[\mathcal{A}(D) \in S]. \quad (2)$$

We next formalize (ii) in two steps: we show that $\Pr[\mathcal{A}(\widetilde{D}) \in S]$ can be upper bounded by $\Pr[\mathcal{A}(D') \in S]$ and vice versa. The following lemma provides an upper bound for $\Pr[A(\widetilde{D}) \in S]$ by using $\Pr[A(D') \in S]$.

LEMMA 4. *For any $S \subseteq (\mathbb{R} \cup \{\perp\})^{d'}$,*

$$\Pr[\mathcal{A}(\widetilde{D}) \in S] \leq \left(1 - \frac{e^{-\omega(T-\Delta)}}{1 + e^{-\omega}}\right)^{-\Delta} \Pr[\mathcal{A}(D') \in S]. \quad (3)$$

Similarly, we can upper bound $\Pr[\mathcal{A}(D') \in S]$ using $\Pr[\mathcal{A}(\widetilde{D}) \in S]$, as indicated by the following lemma.

LEMMA 5. *Let γ be defined as*

$$\gamma = \frac{\Delta \cdot e^{-\omega(T-\Delta)}}{(1 + e^{-\omega})}.$$

Then for any $S \subseteq (\mathbb{R} \cup \{\perp\})^{d'}$,

$$\Pr[\mathcal{A}(D') \in S] \leq \Pr[\mathcal{A}(\widetilde{D}) \in S] + \gamma. \quad (4)$$

We are ready to prove our main theorem of this section:

THEOREM 5. *Let $\varepsilon > 0$, $\gamma > 0$ be some constants, n be the database size such that $n \geq (1 + \gamma)/\gamma\varepsilon$, and $\delta = o(1/n)$. Let q be an integral query of sensitivity Δ,* P *be a geometric mechanism computing q with parameter ω, and $\mathcal{T}$ be a thresholding mechanism with parameter T. If $\omega \leq \varepsilon/(1 + \gamma)\Delta$ and*

$$T \geq \Delta\big(1 + (1 + \gamma)\ln(\Delta/\delta)/\varepsilon\big),$$

then $\mathcal{T}(\mathrm{P}(\cdot))$ is (ε, δ)-differentially private.

On Non-Integral Queries. For Theorem 5 to hold for non-integral queries, we need an additional assumption that the participation of any single individual can only introduce a small number of new dimensions. The proof is essentially the same, with geometric noise replaced by Laplacian noise. However, if an individual is allowed to introduce an unbounded number of dimensions (unbounded in the sense that there is no a priori bound), then the probability that at least one of the perturbed dimensions exceeds the threshold can asymptotically converge to 1. More formally, suppose the probability that an arbitrary dimension exceeds the threshold after perturbation is p and there are N dimensions. Since the noise in each dimension is independent, the probability that at least one of them can exceed the threshold is $1 - (1 - p)^N$, which goes to 1 as N goes to infinity. As a result, we can only set $\delta = 1$.

5. EXTENSIONS

In this section we provide extensions of our basic results from the last two sections. Section 5.1 proposes a mechanism whose only task is to stabilize the universe. This extends the thresholding mechanism to queries with arbitrary ranges and allows us to sequentially compose with a more advanced mechanism, such as the exponential mechanism, for scenarios where one can hope for a better privacy-utility tradeoff. Then, in Section 5.2, under the usual assumption of a fixed universe, we give a framework to study the usefulness of data transformation for the privacy-utility tradeoff.

5.1 Stabilizing Universe

Most differential privacy mechanisms work under the assumption that the universe is fixed. In this section we consider an extension of our results from the last section in which we first stabilize the universe of a dataset, and then apply other mechanisms under the fixed universe assumption. Specifically, we consider a *sequential* composition where in the first step we only stabilize the universe of the data and publish a dataset restricted to the stabilized universe with *exact* values, then in the second step we apply another more specialized mechanism over the data.

5.1.1 Sequential Composition

We note that, compositions considered in traditional differential privacy are usually *parallel* compositions. Specifically, given mechanisms $\mathcal{M}_i : \mathcal{D} \mapsto \mathcal{R}_i$, one considers a composed mechanism $\mathcal{M} : \mathcal{D} \mapsto \prod_i \mathcal{R}_i$. For example, a basic result in differential privacy [8] says that if $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-differentially private, then $\mathcal{M}$ is $(\sum_i \varepsilon_i, \sum_i \delta_i)$-differentially private. Indeed, sequentially composing mechanisms usually does not make too much sense, because if we have achieved differential privacy in the first step, then the second step in the composition will not affect differential privacy (due to Fact 2).

There are two reasons why we consider sequential composition and do not aim for differential privacy in the first step. First, stabilizing the universe will be a common component in the presence of dynamic universes so we may want to do it early in the data processing pipeline. Second, perhaps more importantly, there are situations, like scenarios considered in the exponential mechanism [17], in which adding noise directly to the quantity may completely destroy utility. Thus in such cases, we might want to invoke an advanced mechanism (such as the exponential mechanism mentioned above) on the *exact* data. In view of this, it is better that we first combine perturbation and thresholding to stabilize the universe, but then publish exact (rather than perturbed) data restricted to the stabilized universe.

5.1.2 A Utility based Formulation

Instead of sticking to queries that are numeric vectors, we now consider a query q that maps a database to some arbitrary and evolving range $\mathcal{R}$ (that is, the ranges can be different for two different databases). More specifically, for any database D, let $\mathcal{R}(D)$ be the range of the query over D. Compared to the case of an output numeric vector where one can suppress some dimensions, a difficulty we now have is that it is unclear how to "stabilize" an element $r \in \mathcal{R}$. However, observe that the range evolves only if the underlying database histogram changes, so one can instead stabilize the universe of the underlying database histogram.

In this case, an immediate approach is to invoke our thresholding mechanism on the database histogram to produce a noisy database histogram, and then publish a database histogram with the same universe as the noisy one but with *exact counts*. However, such a mechanism ignores the potential relationship between a database tuple $t \in D$ and a query result $r \in \mathcal{R}$, and thus may incur a loss of utility. In general, we have the following utility notion which assigns a utility score to each universe element.

DEFINITION 4 (LOCAL UTILITY FUNCTION). *Let $\mathcal{D}$ be the collection of all databases in $\mathbb{N}^{\mathbb{N}}$ with* finite universe*, and let $\mathcal{R}$ be some arbitrary range that may depend on the universe of the database. A* local utility function *u is a mapping from $\mathcal{D} \times \mathbb{N} \times \mathcal{R}$ to $\mathbb{R}^+$* [2]*. Moreover, a local utility function is called* natural *if it satisfies one more condition: for any $D \in \mathcal{D}$, $i \notin \operatorname{univ}(D)$ and $r \in \mathcal{R}$, $u(D, i, r) = 0$.*

In other words, a utility function u assigns a score to each element i in the universe with respect to a particular database $D \in \mathcal{D}$ and a particular outcome $r \in \mathcal{R}(D)$. Intuitively, a natural local utility function means that an element provides no utility at all if it is not in the database. Our discussion above indicates that a good local utility function should take two aspects into consideration: (1) how important an element is for the utility of the result, and (2) how much privacy concern it might incur by including an element in the published universe. In the following, we present an example of local utility function.

EXAMPLE 2. *In a slightly more general setting, let us consider a local utility function where one can force the inclusion of some universe element by assigning it a utility score ∞. Consider two functions u_r, u_s from $\mathcal{D} \times \mathbb{N} \times \mathcal{R}$ to $\mathbb{R}^+ \cup \{\infty\}$. u_r is the "result utility function" which assigns a high score to i if i is important to the utility of the query result. u_s is the "safety utility function" which assigns a high score to i if including i into the published universe causes little privacy concern. Then we can define a local utility function $u = u_r \cdot u_s$ with the understanding that, if some element i has local utility ∞, then it will never be suppressed unless $u_s = 0$. We define that $\infty \cdot 0 = 0$. This is to capture that, if we have $u_r(D, i, r) = \infty$ while $u_s(D, i, r) = 0$, then we still suppress i.*

Now we define the *sensitivity* of a local utility function.

DEFINITION 5. *Consider the same setting as in Definition 4. Let u be a local utility function. The* sensitivity *of u is defined as*

$$\max_{D \sim D'} \max_{i \in \operatorname{univ}(D) \cup \operatorname{univ}(D')} \max_{r \in \mathcal{R}(D) \cup \mathcal{R}(D')} \left| u(D, i, r) - u(D', i, r) \right|.$$

In the following, we denote this quantity by Δu.

Intuitively, sensitivity captures how significant the change could be when introducing a new element into the universe. If this quantity is really large for a privacy-sensitive element (e.g., a new pattern in Example 1), then there is no hope of designing a mechanism with good privacy-utility tradeoff.

DEFINITION 6. *Let D be a database, and u be a natural local utility function. For any $i \in \mathbb{N}$, its* maximal utility *with respect to u and database D, written as $u_D(i)$, is defined as $\max_{r \in \mathcal{R}} u(D, i, r)$.*

LEMMA 6. *In the same setting as above, let D' be any database such that $D \sim D'$. Then for any element $i \in \operatorname{univ}(D')$ such that $i \notin \operatorname{univ}(D)$, we have that*

(a) $u_D(i) = 0$;

(b) $u_{D'}(i) \leq \Delta u$.

We are now ready to describe our transformation mechanism for stabilizing universe.

DEFINITION 7 (LOCAL TRANSFORMATION MECHANISM). *Given a database D, a natural local utility function u, a scale parameter b, and a utility bound κ, the local transformation mechanism $\mathcal{P}$ works as follows:*

(1) *It constructs a set I by including each tuple type $i \in \operatorname{univ}(D)$ into the output if $u_D(i) + \operatorname{Lap}(b) \geq \kappa$.*

(2) *After that, it outputs the database D restricted to I, denoted as $D|_I$. That is, it simply shrinks the universe of D to I.*

[2] $\mathbb{R}^+ = \{x \geq 0 \mid x \in \mathbb{R}\}$.

In the following let $\mathcal{P}$ denote the local transformation mechanism. We next argue that, by picking appropriate b and κ, $\mathcal{P}(D)$ and $\mathcal{P}(D')$ can be very close to each other. Therefore, one can apply an (ε, δ)-differentially private mechanism (which works under the usual fixed-universe assumption) to the restricted database to obtain differential privacy in the presence of universe change. We first observe that, if $D \sim D'$ are two nontrivial (namely it is not that $D = D'$) neighboring databases, then there are two cases:

Case (1) There is a unique i such that $\mathrm{univ}(D)$ and $\mathrm{univ}(D')$ only differ on i and either $D_i = 1$ or $D'_i = 1$. By symmetry, we will only consider that $i \in \mathrm{univ}(D')$ and $i \notin \mathrm{univ}(D)$.

Case (2) $\mathrm{univ}(D) = \mathrm{univ}(D')$ and there is a unique i such that $|D_i - D'_i| = 1$.

Now let $\mathcal{M}$ be any (ε, δ)-differentially private mechanism mapping from $\mathbb{N}^{\mathrm{univ}(D')}$ to some dynamic range $\mathcal{R}$. We have the following two lemmas.

LEMMA 7. *For* case (1)*, the composition mechanism $\mathcal{M} \circ \mathcal{P}$ is (ε', δ')-differentially private where*

$$\varepsilon' = -\ln\big(1 - \exp(-\frac{\kappa - \Delta u}{b})\big), \text{ and}$$
$$\delta' = \exp(-\frac{\kappa - \Delta u}{b}).$$

COROLLARY 1. *In the setting above, let $\varepsilon = O(1)$ and $\delta = o(1/n)$ where n is the database size. Then, in order to achieve (ε, δ)-differential privacy, it suffices to set*

$$\kappa \geq \Delta u + b\ln(1/\delta)$$

LEMMA 8. *For* case (2)*, the composition mechanism $\mathcal{M} \circ \mathcal{P}$ is $(\frac{\Delta u}{b} + \varepsilon, \exp(\frac{\Delta u}{b})\delta)$-differentially private.*

Combining these two, we have the following theorem which says that $\mathcal{M} \circ \mathcal{P}$ is differentially private.

THEOREM 6. *Let $\varepsilon = O(1)$ and $\delta = o(1/n)$ where n is the database size. Set $b = \Delta u/\varepsilon$, and $\kappa \geq \Delta u + b\ln(1/\delta) = \Delta u(1 + \ln(1/\delta)/\varepsilon)$. Then, for any (ε, δ)-differentially private mechanism $\mathcal{M}$, $\mathcal{M} \circ \mathcal{P}$ is $(2\varepsilon, e^{\varepsilon}\delta)$-differentially private.*

We remark that our analysis of differential privacy for the composed mechanism here, and later in Section 5.2, treats the composed mechanism *as a whole*. That is, unlike typical analysis of compositions in differential privacy, we do *not* allow a potential adversary to examine intermediate results produced by mechanisms in the composition. Indeed, if an adversary can examine the query result produced by our universe stabilizing mechanism, differential privacy is broken as we output exact data. On the other hand, the message we want to deliver here is that, in the setting of an evolving universe, one shall always consider stabilizing universe as the first step towards differential privacy. In this sense, we are treating universe stabilizing as the *first step* of any differentially private mechanism. Therefore, it is reasonable to analyze differential privacy with respect to the whole composed mechanism.

5.2 Suppression and Privacy-Utility Tradeoff

Interestingly, even under the fixed-universe assumption, previous work [6, 22] has found that suppression could be useful for achieving the desired privacy-utility tradeoff. We next give a framework to study the benefits of suppression, and more generally, data transformation, in terms of the tradeoff between privacy and utility. We begin by defining what we mean by "data transformation" and data transformations that are "oblivious to differential privacy."

DEFINITION 8 (DATA TRANSFORMATION). *Let $\mathcal{U}$ and $\mathcal{U}'$ be two finite sets in $\mathbb{N}$. A transformation mechanism is any probabilistic algorithm that maps a database in $\mathbb{N}^{\mathcal{U}}$ to a database in $\mathbb{N}^{\mathcal{U}'}$.*

DEFINITION 9 (DIFFERENTIAL PRIVACY OBLIVIOUSNESS). *Let $q : \mathbb{N}^{\mathcal{U}} \mapsto \mathcal{R}$ and $q' : \mathbb{N}^{\mathcal{U}'} \mapsto \mathcal{R}$. We say that $\mathcal{P} : \mathbb{N}^{\mathcal{U}} \mapsto \mathbb{N}^{\mathcal{U}'}$ is a* differential-privacy oblivious transformation *if for any (ε, δ)-differentially private mechanism $\mathcal{M}$ for computing q', $\mathcal{M} \circ \mathcal{P}$ is (ε, δ)-differentially private for computing q.*

The motivation here is that, because the guarantee of $\mathcal{M}$ for q' is over *neighboring databases*, $\mathcal{P}$ shall not map neighboring databases to non-neighboring ones. As a result, if $\mathcal{P}$ is differential-privacy oblivious, then for the purpose of ensuring differential privacy for computing q, one only needs to consider q'. We remark that one can easily generalize this definition to the case where we can map neighboring databases to databases of *bounded* distance. A systematic exploration of this generalization is left for future research.

Clearly, because our goal is to compute q, one needs to relate $q' \circ \mathcal{P}$ to q. However, enforcing $q(D) = q'(\mathcal{P}(D))$ is meaningless: data transformation will not improve privacy-utility tradeoff at all. This is because, if q and $q' \circ \mathcal{P}$ are equivalent, the lower bound on noise for computing q privately transfers automatically to computing $q' \circ \mathcal{P}$ privately. Therefore, our goal is to say that $q' \circ \mathcal{P}$ is a good approximation of q. This motivates the following definition.

DEFINITION 10 (APPROXIMATE PROJECTION). *In the same setting as above, let $u : \mathcal{R} \mapsto \mathbb{R}$ be a utility function and $\beta, \gamma \in (0, 1)$ be two constants. Then q admits a (β, γ)-approximation if there exists a q' such that, for any $D \in \mathbb{N}^{\mathcal{U}}$:*

$$\exp(-\gamma)u((q' \circ \mathcal{P})(D)) \leq u(q(D)) \leq \exp(\gamma)u((q' \circ \mathcal{P})(D))$$

with probability at least $(1 - \beta)$ over the coin tosses of $\mathcal{P}$. In this case, we say that q admits a (β, γ)-approximation q' with respect to u and $\mathcal{P}$.

Readers may wonder why we need all these definitions. Suppose that q admits a (β, γ)-approximation q' with respect to u and $\mathcal{P}$. First, let us observe that while $q' \circ \mathcal{P}$ does not compute q exactly, they are close in terms of utility. However, to compute q' privately, a transformation might make it much easier to construct a mechanism $\mathcal{M}$ with small noise. This is because the sensitivity of q' over $\mathcal{P}(\mathcal{D})$ may be significantly smaller than that of q over $\mathcal{D}$. In other words, the potentially large error due to the noise injected for computing q privately is now reduced by combining a small approximation error due to $q' \circ \mathcal{P}$ and a small noise for computing q' privately. Therefore, one might expect that $\mathcal{M} \circ \mathcal{P}$ achieves a better privacy-utility tradeoff than a direct mechanism for q could do.

We remark that, in order to instantiate this paradigm, even if one cannot formally guarantee that q' is a (β, γ)-approximation, some good heuristics with insights for a particular data set may be fruitful in improving privacy-utility tradeoff as well. Indeed, the work [22] gives one such example. In that paper, the authors considered the frequent itemset mining problem and identified a key barrier towards a good privacy-utility tradeoff: large transactions, which introduce significant sensitivity. They then proposed a *transformation* that basically *truncates* large transactions. That is, it probabilistically maps a large transaction to a small one. On one hand, this transformation greatly reduces the sensitivity: all transactions are now small, so the noise needed for privacy is reduced. On the other hand, in practice usually very few transactions are large because of data skew. Suppressing large transactions is then unlikely to change the output of frequent itemset mining. Therefore, after suppression, good approximation to the optimal solution is maintained while much smaller noise for privacy is needed.

In the following we generalize the results in [22] and show that a large class of data transformations, including many aggregation algorithms, are oblivious to differential privacy. First of all, we have the observation that any *deterministic* ℓ_1-norm preserving transformation is differential-privacy oblivious.

LEMMA 9. *Any deterministic ℓ_1-norm preserving algorithm that maps from $\mathbb{N}^{\mathcal{U}}$ to $\mathbb{N}^{\mathcal{U}'}$ is differential-privacy oblivious.*

By expanding $\mathcal{P}$ by conditioning on each sequence of its random coin tosses, we immediately have the following result.

THEOREM 7. *Let $\mathcal{P}$ be a data transformation such that, for any sequence σ of its coin tosses, $\mathcal{P}|_\sigma$ is an ℓ_1-norm preserving algorithm. Then $\mathcal{P}$ is differential-privacy oblivious.*

We mention, in particular, the following special case.

COROLLARY 2. *Let f be any probabilistic algorithm from $\mathcal{U}$ to $\mathcal{U}'$. Then the transformation $\mathcal{F}$ defined by mapping each element in $i \in \mathrm{univ}(D)$ to $f(i)$ is differential-privacy oblivious.*

We note that this class already includes many natural aggregations. For example, let $\mathcal{U}'$ be a set of buckets. Then all natural filter-group-by aggregations fall into this class. Therefore, they are differential-privacy oblivious. As a final remark for this section, we note that it is an interesting future direction to investigate under what circumstances approximate projections exist.

6. ON THE EFFECT OF SUPPRESSION ON DATA UTILITY

All the mechanisms proposed in this paper will *suppress some data*. Specifically, (i) our range-based mechanism needs to suppress ranges with too much noise, and (ii) our thresholding mechanism needs to suppress small perturbed values. Therefore, a key question here is *how suppression may affect data utility*. In this section we give some theoretical observations regarding this question.

On one hand, note that, in Theorem 2 and 5, the thresholds are proportional to the sensitivity Δ. This indicates that our mechanisms may suppress much data when Δ is large. This may give poor data utility for small or medium-sized data. On the other hand, there might be some hope when turning to large-scale data sets. Especially, we are interested in the case of publishing a histogram because a key goal of our mechanisms is for an internal data sanitization in an early stage of the data processing pipeline. In such cases, the sensitivity is one. One can then have, simultaneously, a small amount of noise injected to the data, and a small threshold for suppression. Thus, one might expect a small loss of data utility.

To this end, we note that publishing histograms in a differentially private way is studied extensively in the literature. Thus, one might naturally ask to empirically compare our methods with previous work on differentially private histograms (e.g., [13]). However, we note that most of this work considers different tradeoffs[3] and thus are incomparable to our mechanisms here. Moreover, as we mentioned Section 5.1.1, when a different utility notion is considered, one might want to sequentially compose the transformation of stabilizing universe with a mechanism optimized for that utility.

In the following, we quantify the magnitude of thresholds for sanitizing histograms. As we will see, the thresholds are indeed small when large-scale data is considered. For example, for the dynamic universe problem, our threshold for achieving strong differential privacy is only *several hundreds*, which is much smaller compared to that currently adopted in practice (e.g., [21], which uses thresholds at the scale of 10^4).

6.1 On Thresholding and Data Sparsity

In this section we discuss thresholding and data sparsity. Suppose we want to publish a high-dimensional vector of non-negative counts. Without loss of generality, suppose dimensions $1, 2, \ldots, n$ of the vector are 0. We inject independent geometric noise $G_i(\varepsilon)$ into dimension i in order to achieve ε-differential privacy. Let $X_i(i \in [n])$ be an indicator random variable such that $X_i = 1$ if dimension i becomes non-zero after perturbation. Then [4]

$$\Pr[X_i = 1] = \begin{cases} 1 & \text{w.p. } e^{-\varepsilon}/(1+e^{-\varepsilon}), \\ 0 & \text{w.p. } 1/(1+e^{-\varepsilon}). \end{cases}$$

Therefore, the expected number of non-zero dimensions is

$$\mathbb{E}[\sum_{i=1}^{n} X_i] = ne^{-\varepsilon}/(1+e^{-\varepsilon}).$$

Further, by applying the Chernoff bound, the number of non-zero dimensions will be highly concentrated around the expectation. This means that, with high probability, most of the zero dimensions are now non-zero, and thus a sparse vector now becomes dense.

On the other hand, suppose we do thresholding with $T = t/\varepsilon$ for some t to be determined later. Let $\beta \in (0, 1)$ be a confidence parameter, then

$$\Pr[\cup_{i=1}^{n} G_i(\varepsilon) > T] \le n \Pr[G_i(\varepsilon) > T] \le ne^{-t},$$

where the first inequality is from union bound and the second inequality is by concentration of Geometric noise. Therefore, to bound this by β, it suffices to set $t \ge \ln(n/\beta)$, which implies $T \ge \ln(n/\beta)/\varepsilon$. This guarantees that, with probability at least $1 - \beta$, all these perturbed dimensions will be suppressed by T. T could be quite small even for very large n. For example, for $\varepsilon = .1, \beta = .01$, and $n = 10^{12}$ (1 trillion), T is only roughly 320.

6.2 Thresholding Mechanism

Theorem 5 implies the threshold in this case is

$$1 + (1+\gamma)\ln(1/\delta)/\varepsilon.$$

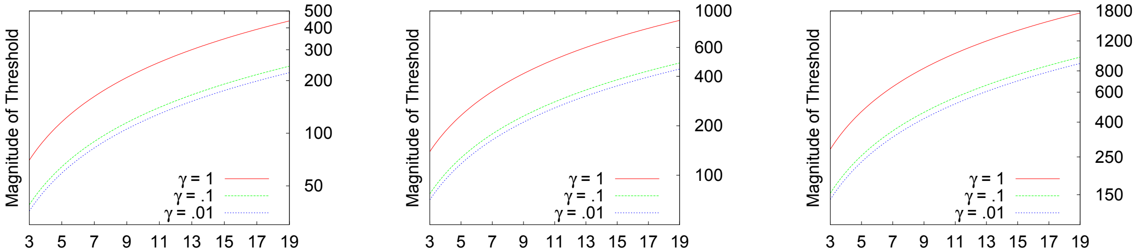
Figure 1 shows the magnitude of threshold for $\varepsilon = .2$, .1, and .05, with δ ranging from $1/10^4$ to $1/10^{19}$. For this test, we consider three possible values, 1, .1, and .01, for γ in Theorem 5. For the strongest setting of parameters (i.e., $\varepsilon = .05$ and $\delta = 1/10^{19}$), the magnitude of threshold is roughly 1800, 963, and 885 for $\gamma = 1$, .1, and .01, respectively. In such a case, let n be the total population of the histogram and suppose $\delta = 1/n^2$. Then one can ensure $(.05, 1/10^{19})$-differential privacy for a histogram with total population $10^{9.5}$. For histograms of such scale in practice, the effect of our thresholding is likely to be insignificant compared to current practice of thresholding (which uses thresholds at the scale of 10^4), while our method ensures differential privacy.

6.3 Range-Based Mechanism

Intuitively, the larger the range is, the more likely the original value will fall into this range. However, the guarantee for the consumers is weaker. Theorem 2 gives the range size $L = \ln(d/\delta)/\varepsilon$, where d is the number of dimensions in the published vector for the

[3] For example, a key tradeoff considered in [13] is how to keep answers to a *set* of histogram queries as consistent as possible, while maintaining differential privacy.

[4] We adopt the following natural strategy for publishing a non-negative count: if the noise is negative, then we round it to 0.



(a) $\varepsilon = .2$ (b) $\varepsilon = .1$ (c) $\varepsilon = .05$

Figure 1: Magnitude of threshold for a given δ. The x axis is drawn in $1/\delta$ with log scale, and tic k represents 10^k. Three settings of γ is depicted: 1, $.1$ and $.01$, and sensitivity is one.

histogram. If $G = G(\varepsilon)$ is the noise random variable, then using Fact 3, one can bound the probability that we fall out of the range:

$$\Pr[|G| > L] = \Pr[|G| > \frac{\varepsilon L}{\varepsilon}] \leq e^{-\varepsilon L} = \frac{\delta}{d},$$

where the approximation holds for small ε. By a union bound across all dimensions, it follows that, with probability at least $1 - \delta$, all dimensions will be enclosed by their corresponding ranges. Moreover, the range sizes are pretty small for large-scale data. For example, in our strongest parameter setting $\varepsilon = .05$ and $\delta = 1/10^{19}$ mentioned above, the range size is roughly $20 \ln d + 876$. Then, the range size is roughly 1290 for a histogram with a billion (10^9) dimensions. For common histograms gathered in practice, most counts are at least at the scale of 10^5. These counts thus bear small relative errors if we publish ranges with the above size.

7. RELATED WORK

Differential privacy [3, 4] has been attracting ever-growing interest in the research community (see, for example, [1, 2, 5, 6, 9, 10, 12, 18, 20]). Unlike previous proposals that define privacy as properties of the sanitized data (e.g., k-anonymity [19], ℓ-diversity [16], and t-closeness [14]), differential privacy is a definition on the query answering algorithm. Perhaps due to this, very different mathematical techniques have been developed for achieving differential privacy. The recent monograph by Dwork and Roth [8] provides a thorough treatment for differential privacy.

To the best of our knowledge, there have been rare connections between old privacy techniques and differential privacy. One notable line (as studied in [15]) shows that old privacy notions, such as k-anonymity, does give (ε, δ)-differential privacy "when done appropriately." However, it is still not very clear whether old privacy techniques and differential privacy could interact in a meaningful way. One reason, we conjecture, is that old techniques are mostly suppression-based, while differential privacy fundamentally relies on perturbation. Indeed, it is often considered an advantage of differential privacy that it does not alter the data set, but only needs to inject somewhat mild noises.

Nevertheless, some previous work has observed that information suppression can be useful for differential privacy. One example towards this end is truncating large transactions for frequent itemset mining as we mentioned earlier. As another example, suppose we want to publish a high dimensional vector. Then intuitively, if we apply thresholding with a large enough threshold, the sensitivity will only depend on *the number of dimensions with values above the threshold* (in the extreme case, if all dimensions are suppressed, no noise is needed). This significantly reduces sensitivity and thus magnitude of noise. Indeed, this intuition leads to the sparse vector technique [6, 8].

However, for both cases, suppression is still irrelevant for privacy in the sense that perturbation itself can already ensure differential privacy. By contrast, thresholding is necessary for differential privacy in the case of a dynamic universe. Another difference is that the sparse vector technique expects larger thresholds hence smaller noise, while we want to keep the threshold as small as possible so that information loss is small.

The standard computation model of differential privacy assumes a static data set. This leads to the usual assumption that the data universe is fixed. In contrast to that, we assume a model where universe cannot be fixed a priori due to multiple collections of data. From this perspective, perhaps the most relevant work is the one by Dwork et al. [5]. There they considered a *streaming* setting where users generate input events to the database and an adversary *continually* analyzes the data. They asked the question how to ensure differential privacy under this situation and provided a solution for a specific setting of maintaining a single counter. Although the setting resembles ours, there are two notable differences. First, the universe of their data analysis is still fixed (i.e., a single counter). Second, we considered a problem that is orthogonal to the single-counter problem: if a single element could induce a new dimension in the published result, what shall we do to ensure differential privacy? Our answer is that suppression is necessary and may give other benefits.

8. CONCLUSION

While traditional differential privacy theory works in a *closed-world* setting, in this paper we studied the problem of ensuring differential privacy in an open world where the universe of data can evolve due to multiple collections/releases. Thresholding, an old suppression-based technique, plays an integral role in this context. In addition, we also showed the usefulness of thresholding in improving data utility. We further extended our results from thresholding to more general data suppression/transformation.

From a larger perspective, we view our results as a case study on the interaction between old privacy techniques and the modern differential privacy. Previous work on data privacy can be broadly categorized into two kinds according to the general tradeoff between privacy and information content. One is based on information suppression, where suppressing all information achieves perfect privacy. The other is based on perturbation, where releasing complete noise achieves perfect privacy. With the popularization of differential privacy, it seems that perturbation-based methods have become dominant. However, our results demonstrate that suppression can still be useful or even necessary.

We note that, in some sense, thresholding can be thought of as a special case of k-anonymity as follows. First, perform a group-by on the quasi-identifier of the tuples to form groups. Thresholding leaves these quasi-identifiers as is, and only publishes groups whose count exceeds a threshold. On the other hand, k-anonymity generalizes the quasi-identifiers to create groups of at least k. However, both involve ensuring that groups are sufficiently large before publication. Given that thresholding is useful for addressing both utility and privacy issues in the current differential privacy regime, it is natural to ask if k-anonymity (and perhaps other privacy notions) is also still useful. We hope our work can serve as a first step that could trigger further research in this broad, fertile ground.

9. REFERENCES

[1] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *PODS*, pages 273–282, 2007.

[2] A. Blum, K. Ligett, and A. Roth. A learning theory approach to noninteractive database privacy. *J. ACM*, 60(2):12, 2013.

[3] C. Dwork. Differential privacy. In *ICALP (2)*, pages 1–12, 2006.

[4] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[5] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *STOC*, pages 715–724, 2010.

[6] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390, 2009.

[7] C. Dwork and R. Pottenger. Toward practicing privacy. *JAMIA*, 20(1):102–108, 2013.

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

[9] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.

[10] A. Ghosh, T. Roughgarden, and M. Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM J. Comput.*, 41(6):1673–1693, 2012.

[11] Google search analytics help: Data that are subject to thresholds. https://support.google.com/analytics/answer/2954071?hl=en&ref_topic=2799375.

[12] A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. *SIAM J. Comput.*, 42(4):1494–1520, 2013.

[13] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *PVLDB*, 3(1):1021–1032, 2010.

[14] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, pages 106–115, 2007.

[15] N. Li, W. Qardaji, and D. Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, pages 32–33, 2012.

[16] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *ICDE*, page 24, 2006.

[17] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 94–103, 2007.

[18] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84, 2007.

[19] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[20] J. Ullman. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. In *STOC*, pages 361–370, 2013.

[21] United states census bureau american community survey data suppression explained. http://www.census.gov/acs/www/Downloads/data_documentation/data_suppression/ACSO_Data_Suppression.pdf.

[22] C. Zeng, J. F. Naughton, and J.-Y. Cai. On differentially private frequent itemset mining. *PVLDB*, 6(1):25–36, 2012.

APPENDIX

A. PROOFS

A.1 Proof of Fact 4

PROOF. Suppose that a probabilistic mechanism $\mathcal{K}$ can achieve both. Consider a one dimensional integral query q. Suppose that $q(D) = \beta$ and $q(D') = \beta + 1$ where $D \sim D'$. Since $\mathcal{K}$ is ε-differentially private, we have

$$\mathrm{supp}(\mathcal{K}(D)) = \mathrm{supp}(\mathcal{K}(D'))$$

Note that here $\mathrm{supp}(\mathcal{K}(D))$ is a collections of ranges. It follows that for every $R \in \mathrm{supp}(\mathcal{K}(D))$ we have both $\beta \in R$ and $\beta + 1 \in R$. Now consider another database D'' such that $q(D'') = \beta + 2$ and $D' \sim D''$. With exactly the same argument, we can show that every range in $\mathrm{supp}(\mathcal{K}(D))$ must also contain $\beta + 2$. Repeating this procedure, it follows that every range in $\mathrm{supp}(\mathcal{K}(D))$ must contain all integers. Clearly, only the range $(-\infty, \infty)$ can satisfy this requirement. □

A.2 Proof of Lemma 1

PROOF. Let q be a counting query. Suppose that $q(D) = \beta$ and $q(D') = \beta - 1$ for two neighboring databases D and D'. Consider $\mathrm{supp}(\mathcal{M}(D))$ and $\mathrm{supp}(\mathcal{M}(D'))$, which consists of ranges that contain β and $\beta - 1$, respectively. A bad range R must be of one of the following two types:

(1) $R \in \mathrm{supp}(\mathcal{M}(D))$ but $R \notin \mathrm{supp}(\mathcal{M}(D'))$;

(2) $R \notin \mathrm{supp}(\mathcal{M}(D))$ but $R \in \mathrm{supp}(\mathcal{M}(D'))$.

Now consider a bad range R of type (1). By the mechanism $\mathcal{M}_0$, we have $R_1 = [\beta' - L, \beta' + L]$ where β' is noisy version of β after perturbation. Since $\beta \in R_1$ but $\beta - 1 \notin R_1$, it follows that $\beta' - L = \beta$ and hence $R_1 = [\beta, \beta + 2L]$ — this is the *unique* bad range of type (1). Similarly, $R_2 = [\beta - 1 - 2L, \beta - 1]$ is the unique bad range of type (2).

We now compute the probability that R_1 can be generated by $\mathcal{M}$ (actually by $\mathcal{M}_0$). With respect to $\mathcal{M}_0$, it equals the probability that $\beta' = \beta + L$, which is

$$\Pr[\beta' = \beta + L] = \Pr[\beta = \beta' - L] = \frac{1 - e^{-\varepsilon}}{1 + e^{-\varepsilon}} e^{-L\varepsilon}$$

because $\mathcal{M}_0$ adds geometric noise drawn from $G(\varepsilon)$ to β. Similarly, we can compute the probability of R_2, which turns out to be the same as the above. Since R_1 and R_2 are the only two ranges that violate ε-differential privacy, it follows that for all $S \subseteq \mathrm{Range}(\mathcal{M})$,

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \frac{1 - e^{-\varepsilon}}{1 + e^{-\varepsilon}} e^{-L\varepsilon},$$

$$\Pr[\mathcal{M}(D') \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D) \in S] + \frac{1 - e^{-\varepsilon}}{1 + e^{-\varepsilon}} e^{-L\varepsilon}.$$

Therefore, it is sufficient to set $\delta = \frac{1-e^{-\varepsilon}}{1+e^{-\varepsilon}} e^{-L\varepsilon}$. □

A.3 Proof of Lemma 2

PROOF. Let q be such a query. Since $\Delta q = 1$, only one dimension in $q(D)$ and $q(D')$ could be different. Following the proof of

Lemma 1, approximate DP holds for that particular dimension with additive error $\frac{1-e^{-\varepsilon}}{1+e^{-\varepsilon}} \cdot e^{-L\varepsilon}$. Intuitively, since the affected dimension must be one of the d dimensions, it follows that the total additive error in the d-dimensional case is at most $d \cdot \frac{1-e^{-\varepsilon}}{1+e^{-\varepsilon}} \cdot e^{-L\varepsilon}$. More formally, let B_i be the event that for dimension i type (1) bad range happens and let $B = \cup_{i\in[d]} B_i$. For any $S \subseteq \mathrm{Range}(\mathcal{M})$, we use A and A' to denote the events $\mathcal{M}(D) \in S$ and $\mathcal{M}(D') \in S$. By symmetry, we only need to consider that

$$\begin{aligned}
\Pr\left[A\right] &= \Pr\left[A \wedge \neg B\right] + \Pr\left[A \wedge B\right] \\
&\le \Pr\left[A \wedge \neg B\right] + \Pr\left[B\right] \\
&\le e^{\varepsilon} \Pr\left[A' \wedge \neg B\right] + \Pr\left[\cup_{i=1}^{d} B_i\right] \\
&\le e^{\varepsilon} \Pr\left[A'\right] + \Pr\left[\cup_{i=1}^{d} B_i\right] \\
&\le e^{\varepsilon} \Pr\left[A'\right] + d \cdot \frac{1-e^{-\varepsilon}}{1+e^{-\varepsilon}} \cdot e^{-L\varepsilon}.
\end{aligned}$$

The last inequality follows from applying the *union bound*

$$\Pr[\cup_{i=1}^{t} A_i] \le \sum_{i=1}^{t} \Pr[A_i],$$

for any probability space and events $A_1, \ldots, A_t$ in that space. Setting $\delta = d \cdot \frac{1-e^{-\varepsilon}}{1+e^{-\varepsilon}} \cdot e^{-L\varepsilon}$ completes the proof of the lemma. □

A.4 Proof of Theorem 3

PROOF. Without loss of generality, let $D \sim D'$ be two neighboring database where $q(D')$ contains one more dimension than $q(D)$. We extend $q(D)$ into a $(d+1)$-dimensional vector where the $(d+1)$-th dimension is $\perp$.

Consider any randomized mechanism $\mathcal{M}$ that is used to compute $q(D)$. The range of $\mathcal{M}(D)$ consists of vectors where the $(d+1)$-th dimension must be $\perp$, while the range of $\mathcal{M}(D')$ consists of vectors where the $(d+1)$-th dimension is not $\perp$. Now consider the event $\mathcal{E} = \mathbb{R}^{d+1}$. Then $\Pr[\mathcal{M}(D') \in \mathcal{E}] = 1$ but $\Pr[\mathcal{M}(D) \in \mathcal{E}] = 0$. For (ε, δ)-differential privacy to hold, we need that $\Pr[\mathcal{M}(D') \in \mathcal{E}] \le e^{\varepsilon} \Pr[\mathcal{M}(D) \in \mathcal{E}] + \delta$ which implies $\delta \ge 1$. □

A.5 Proof of Theorem 4

PROOF. Let $\mathcal{E}_v$ be the event that $\mathcal{M}$ releases dimension 0 when it has measure v, and denote p_v the probability that $\mathcal{E}_v$ happens. By the definition of differential privacy, we can define the following sequence $p_\perp = 0, p_\Delta \le \delta$, and $p_{w+\Delta} \le e^{\varepsilon} p_w + \delta$. Therefore

$$p_v \le \delta(1 + e^{\varepsilon} + e^{2\varepsilon} + \cdots + e^{(\lceil v/\Delta \rceil - 1)\varepsilon}) \le \delta \frac{e^{\lceil v/\Delta \rceil \varepsilon} - 1}{e^{\varepsilon} - 1}$$

Therefore for $p_v \ge p$, it suffices that $(e^{\lceil v/\Delta \rceil \varepsilon} - 1)\delta \ge p$, which implies that $v \ge \Delta(\ln(p/\delta + 1)/\varepsilon - 1)$. □

A.6 Proof of Lemma 4

For Lemma 4 and Lemma 5, let

$$S_0 = \left\{ x \in (\mathbb{R} \cup \{\perp\})^{d'} \;\middle|\; \forall i \in \{d+1, \ldots, d'\}, x_i = \perp \right\}.$$

Clearly, the supports of $\mathcal{A}(\widetilde{D})$ and $\mathcal{A}(D)$ only contain vectors in S_0. Let S_0^c be the complement of S_0 in $(\mathbb{R} \cup \{\perp\})^{d'}$.

PROOF. We have that

$$\begin{aligned}
&\Pr[\mathcal{A}(D') \in S] \\
\ge\, &\Pr[\mathcal{A}(D') \in S \cap S_0] \\
=\, &\Pr\left[\mathcal{A}(\widetilde{D}) \in S \cap S_0, \wedge_{i=d+1}^{d'} (a_i' + X_i < T)\right] \\
=\, &\Pr[\mathcal{A}(\widetilde{D}) \in S \cap S_0] \cdot \textstyle\prod_{i=d+1}^{d'} \Pr[X_i < T - a_i'] \\
\ge\, &\Pr[\mathcal{A}(\widetilde{D}) \in S \cap S_0] \cdot \big(1 - \dfrac{e^{-\omega(T-\Delta)}}{1+e^{-\omega}}\big)^{\Delta}.
\end{aligned}$$

completing the proof. □

A.7 Proof of Lemma 5

PROOF. We have that

$$\begin{aligned}
\Pr[\mathcal{A}(D') \in S_0^c] &= \Pr\left[\vee_{i=d+1}^{d'} \mathcal{A}(D')_i \ne \perp\right] \\
&= \Pr\left[\vee_{i=d+1}^{d'} \mathrm{P}(D')_i \ge T\right] \\
&\le \sum_{i=d+1}^{d'} \Pr[a_i' + X_i \ge T] \\
&\le \sum_{i=d+1}^{d'} \Pr[\Delta + X_i \ge T] \\
&= \sum_{i=d+1}^{d'} \frac{e^{-\omega(T-\Delta)}}{(1+e^{-\omega})} \\
&\le \frac{\Delta \cdot e^{-\omega(T-\Delta)}}{(1+e^{-\omega})} \\
&\overset{\text{def}}{=} \gamma.
\end{aligned}$$

Since $\Pr[\mathcal{E}_1 \cap \mathcal{E}_2] \ge \Pr[\mathcal{E}_1] - \Pr[\neg\mathcal{E}_2]$,

$$\begin{aligned}
\Pr[\mathcal{A}(D') \in S \cap S_0] &\ge \Pr[\mathcal{A}(D') \in S] - \Pr[\mathcal{A}(D') \in S_0^c] \\
&\ge \Pr[\mathcal{A}(D') \in S] - \gamma.
\end{aligned}$$

As a result $\Pr[\mathcal{A}(D') \in S] \le \Pr[\mathcal{A}(D') \in S \cap S_0] + \gamma$. Since the first d dimensions of D' and $\widetilde{D}$ are the same,

$$\Pr[\mathcal{A}(D') \in S \cap S_0] \le \Pr[\mathcal{A}(\widetilde{D}) \in S \cap S_0]$$

The lemma follows by observing that $\Pr[\mathcal{A}(\widetilde{D}) \in S \cap S_0] = \Pr[\mathcal{A}(\widetilde{D}) \in S]$. □

A.8 Proof of Theorem 5

PROOF. Combining (1) and (3), (2) and (4), we have that

$$\Pr[\mathcal{A}(D) \in S] \le \frac{e^{\omega\Delta}}{\big(1 - \frac{e^{-\omega(T-\Delta)}}{1+e^{-\omega}}\big)^{\Delta}} \Pr[\mathcal{A}(D') \in S],$$

$$\Pr[\mathcal{A}(D') \in S] \le e^{\omega\Delta} \Pr[\mathcal{A}(D) \in S] + \frac{\Delta e^{-\omega(T-\Delta)}}{(1+e^{-\omega})}.$$

It is left to set the parameters. Our goal is that

$$e^{\omega\Delta}\Big(1 - \frac{e^{-\omega(T-\Delta)}}{1+e^{-\omega}}\Big)^{-\Delta} \le e^{\varepsilon}, \frac{\Delta e^{-\omega(T-\Delta)}}{(1+e^{-\omega})} \le \delta$$

It suffices to set

$$e^{\omega\Delta}\big(1 - e^{-\omega(T-\Delta)}\big)^{-\Delta} \le e^{\varepsilon}, \tag{5}$$

$$\Delta e^{-\omega(T-\Delta)} \le \delta \tag{6}$$

By (6) we have

$$e^{-\omega(T-\Delta)} \le \delta/\Delta = o(1/n)$$

which gives that

$$T \geq \Delta + \ln(\Delta/\delta)/\omega$$

Plugging into (5),

$$\begin{aligned} e^{\omega\Delta}\big(1 - e^{-\omega(T-\Delta)}\big)^{-\Delta} &\leq e^{\omega\Delta}(1-\delta/\Delta)^{-\Delta} \\ &\leq e^{\omega\Delta+\delta} \\ &\overset{\text{want}}{\leq} e^{\varepsilon}. \end{aligned}$$

Thus one can set $\omega = \varepsilon/(1+\gamma)\Delta$ because $\delta = o(1/n)$. Plugging back $\omega = \varepsilon/(1+\gamma)\Delta$, we have $T \geq \Delta\big(1 + (1+\gamma)\ln(\Delta/\delta)/\varepsilon\big)$ completing the proof. □

A.9 Proof of Lemma 6

PROOF. Part (a) follows directly from the fact that u is natural. Part (b) follows from our definition of Δu and the fact that D and D' are neighboring. □

A.10 Proof of Lemma 7

PROOF. We observe that, because $\mathcal{P}$ uses independent randomness to each tuple type, we have that for any $S \subseteq \mathcal{R}$,

$$\Pr[\mathcal{M}(\mathcal{P}(D)) \in S] = \Pr[\mathcal{M}(\mathcal{P}(D')) \in S | i \notin \mathrm{supp}(\mathcal{P}(D'))]$$

Therefore,

$$\begin{aligned} &\Pr[\mathcal{M}(\mathcal{P}(D') \in S] \\ =&\Pr[\mathcal{M}(\mathcal{P}(D')) \in S | i \in \mathrm{supp}(\mathcal{P}(D'))] \cdot \Pr[i \in \mathrm{supp}(\mathcal{P}(D'))] + \\ &\Pr[\mathcal{M}(\mathcal{P}(D')) \in S | i \notin \mathrm{supp}(\mathcal{P}(D'))] \cdot \Pr[i \notin \mathrm{supp}(\mathcal{P}(D'))] \\ \leq&\Pr[i \in \mathrm{supp}(\mathcal{P}(D'))] + \Pr[\mathcal{M}(\mathcal{P}(D')) \in S | i \notin \mathrm{supp}(\mathcal{P}(D'))] \\ \leq&\exp(-\frac{\kappa - \Delta u}{b}) + \Pr[\mathcal{M}(\mathcal{P}(D')) \in S | i \notin \mathrm{supp}(\mathcal{P}(D'))] \\ =&\exp(-\frac{\kappa - \Delta u}{b}) + \Pr[\mathcal{M}(\mathcal{P}(D)) \in S] \end{aligned}$$

where the second inequality is due to Lemma 6. On the other hand, note that

$$\begin{aligned} \Pr[\mathcal{M}(\mathcal{P}(D')) \in S] \geq& \Pr[\mathcal{M}(\mathcal{P}(D')) \in S | i \notin \mathrm{supp}(\mathcal{P}(D'))] \\ &\cdot \big(1 - \exp(-\frac{\kappa - \Delta u}{b})\big) \end{aligned}$$

It then follows that

$$\begin{aligned} \Pr[\mathcal{M}(\mathcal{P}(D)) \in S] \leq& \big(1 - \exp(-\frac{\kappa - \Delta u}{b})\big)^{-1} \\ &\cdot \Pr[\mathcal{M}(\mathcal{P}(D')) \in S] \end{aligned}$$

completing the proof. □

A.11 Proof of Corollary 1

PROOF. We want that

$$\big(1 - \exp(-\frac{\kappa - \Delta u}{b}\big) - 1 \leq e^{\varepsilon} \quad (1)$$

$$\exp(-\frac{\kappa - \Delta u}{b}) \leq \delta \quad (2)$$

From (2) we derive that $\kappa \geq \Delta u + b\ln(1/\delta)$. Plugging (2) back to (1)

$$\begin{aligned} &\big(1 - \exp(-\frac{\kappa - \Delta u}{b})\big)^{-1} \\ \leq&(1-\delta)^{-1} \\ \leq&(e^{-\delta})^{-1} = e^{\delta} \leq e^{\varepsilon} \end{aligned}$$

where the last inequality is due to our settings of δ and ε. □

A.12 Proof of Lemma 8

PROOF. We write

$$\begin{aligned} &\Pr[\mathcal{M}(\mathcal{P}(D)) \in S] \\ =&\sum_{Z} \Pr[\mathrm{supp}(\mathcal{P}(D)) = Z] \Pr[\mathcal{M}(D|_Z) \in S] \\ \leq&\sum_{Z} \Pr[\mathrm{supp}(\mathcal{P}(D)) = Z)] \\ &\cdot \big(e^{\varepsilon} \Pr[\mathcal{M}(D'|_Z) \in S] + \delta\big) \end{aligned}$$

Therefore it suffices to bound

$$e^{-\Delta u/b} \leq \frac{\Pr[\mathrm{supp}(\mathcal{P}(D')) = S]}{\Pr[\mathrm{supp}(\mathcal{P}(D)) = S]} \leq e^{\Delta u/b}$$

But this follows because D and D' has the same universe so we can view $\mathcal{P}$ as applying a postprocessing after Laplacian mechanism, therefore by Fact 2 the above quantity is bounded by the privacy loss of Laplacian mechanism, which is $\exp(\Delta u/b)$. Plugging back completes the proof. □

A.13 Proof of Theorem 7

PROOF. Let $D \sim D'$ be two neighboring databases in $\mathbb{N}^{\mathcal{U}}$, and $\mathcal{M}$ be an (ε, δ)-differentially private mechanism, then for any event $\mathcal{E} \subseteq \mathcal{R}$,

$$\begin{aligned} &\Pr[\mathcal{M}(\mathcal{P}(D)) \in \mathcal{E}] \\ =&\sum_{\sigma} \Pr[\mathcal{P} = \sigma] \Pr[\mathcal{M}(\mathcal{P}|_{\sigma}(D) \in \mathcal{E}] \\ \leq&\sum_{\sigma} \Pr[\mathcal{P} = \sigma](e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{P}|_{\sigma}(D')) \in \mathcal{E}] + \delta) \\ =&e^{\varepsilon} \Pr[\mathcal{M}(\mathcal{P}(D')) \in \mathcal{E}] + \delta \end{aligned}$$

completing the proof. □

A.14 Proof of Corollary 2

PROOF. Fix any sequence of coin tosses of $\mathcal{F}$, the resulting mapping is a deterministic mapping that maps each element in $\mathcal{U}$ to an element in $\mathcal{U}'$, and thus is ℓ_1-norm preserving. Thus by Theorem above, $\mathcal{F}$ is differential privacy oblivious. □

A.15 Proof of Lemma 9

PROOF. Let q' be any query from $\mathbb{N}^{\mathcal{U}'}$ to $\mathcal{R}$, and $\mathcal{M}$ be an (ε, δ)-differentially private mechanism computing q'. Now, given any two databases $D \sim D'$ in $\mathbb{N}^{\mathcal{U}}$, we have that $\mathcal{P}(D) \sim \mathcal{P}(D')$ because $\mathcal{P}$ is ℓ_1-norm preserving. Therefore the differential privacy guarantee of $\mathcal{M}$ holds for $\mathcal{P}(D) \sim \mathcal{P}(D')$. This indicates that $\mathcal{M} \circ \mathcal{P}$ is (ε, δ)-differentially private, and so $\mathcal{P}$ is differential privacy oblivious. □