Differential Privacy of Mathematical Functions

Genqiang Wu\textsuperscript{1,2}, Xianyao Xia\textsuperscript{1}, and Yeping He\textsuperscript{1,3}

\textsuperscript{1} NFS, Institute of Software Chinese Academy of Sciences, Beijing 100190, China
\texttt{genqiang80@gmail.com, \{xianyao,yeping\}@nfs.iscas.ac.cn}
\textsuperscript{2} SIE, Lanzhou University of Finance and Economics, Lanzhou 730020, China
\textsuperscript{3} SKLCS, Institute of Software Chinese Academy of Sciences, Beijing 100190, China

\textbf{Abstract.} Differential privacy is a privacy notion in statistical databases. However, we find that the notion can be extended to mathematical functions. By studying the differential privacy problem of a simple mathematical function we find two differentially private mechanisms which are applicable for almost all query functions. By using these mechanisms each query function can have custom-designed mechanisms, which implies that our mechanisms are not based on (global or local) sensitivity-based methods—some methods extensively used to construct differentially private mechanisms. Our mechanisms are so flexible that they can be used to balance the accuracy of the outputs of different datasets. We also present some interesting impossibility results of differential privacy. The above results motivate us to explore the mathematical treatment of differential privacy. We present an abstract model of differential privacy in which a differential privacy problem is modeled as finding a randomized mapping between two metric spaces. Some basic structures of differential privacy are presented, such as the \( \{T^i\} \), \( \{R^i\} \), sequences. Some of the properties of these structures, such as the convergence, are also discussed (for some special cases). Finally, we apply our theoretical results to the subgraph counting problem and the linear query problem. The experiments show that our mechanisms (in most cases) have more accurate results than the state of the art mechanisms.
1 Introduction

Differential privacy studies how to query dataset while preserving the privacy of those people whose sensitive informations are contained in the dataset. The key of the problem is to find efficient method to query sensitive dataset to obtain (relatively) accurate results while satisfying differential privacy. The global sensitivity-based method [1] is an efficient and widely used method to achieve differential privacy. Explicitly, the global sensitivity of a query function \[ f \] or a utility function [2] is its maximum difference when evaluated on any two neighbouring datasets (differing on at most one record), and the global sensitivity-based method is to perturb the query result with noise of magnitude proportional to the global sensitivity. Due to the poor accuracy of the global sensitivity-based method when the global sensitivity is large, the smooth sensitivity-based method [3] and the local sensitivity-based method [4] among others [5,6,7] are presented, where the local sensitivity of a query \( f \) on a dataset \( x \) is the maximum difference of \( f(x) \) with the value of its any neighboring dataset, and the smooth sensitivity is a compromise between the global sensitivity and the local sensitivity. The (global or local) sensitivity has long been seen as a mark of the minimum noise needed in order to satisfy differential privacy in practice. However, there is seldom theoretical result to support this point. Our question is: Does there exist query function which has very large (global or local) sensitivity but has (relatively) accurate differentially private query results?

To answer this question, we should first understand differential privacy in a simple way. However, we find that most differential privacy problems [8,9,10,11,12,13,6] are so complicated that it is not a good idea to understand differential privacy from these problems. Therefore, we try to understand differential privacy by using a simple example which, at first glance, seems not a privacy protection problem at all.

Example 1. Let the function \( f : \mathbb{R} \rightarrow \mathbb{R} \) be defined in (1).

\[
 f(x) = \begin{cases} 
 x & \text{if } x \text{ is a rational number} \\
 -x & \text{if } x \text{ is an irrational number} 
\end{cases} \quad (1)
\]

Let \( d \) denote the Euclidean distance on \( \mathbb{R} \). Then \( f \) is a mapping between two metric spaces \((d, \mathbb{R})\) and \((d, \mathbb{R})\). We denote the first metric space as \((d, D)\), and the second as \((d, R)\) for the convenience of illustration. We define the differential privacy problem of \( f(x) \) as follows. Two elements \( x, x' \in D \) are said to be neighbors if \( 0 < d(x, x') \leq 1 \). A randomized
function $\mathcal{M}$ satisfies $\epsilon$-differential privacy if for any neighbors $x, x' \in \mathcal{D}$, and for any measurable $S \subseteq \mathcal{R}$, there is

$$\Pr[\mathcal{M}(x) \in S] \leq e^\epsilon \Pr[\mathcal{M}(x') \in S].$$

(2)

The question is: Does there exist a differentially private mechanism $\mathcal{M}$ which satisfies that $f(x)$ and $\mathcal{M}(x)$ are statistically close for all (or almost all) $x \in \mathcal{D}$?

The first choice to treat Example 1 would be the Laplace mechanism [1], which is a global sensitivity-based method. However, this technique does not work for Example 1 since the global sensitivity of $f$ is

$$\Delta f = \max_{x, x' \in \mathcal{D}: x, x' \text{ are neighbors}} |x - x'| = \infty.$$

The smooth sensitivity-based method [3] and the local sensitivity-based method [4] both also do not work due to the large local sensitivity of $f$ for most points in $\mathcal{D}$.

Can we design a mechanism to achieve high accuracy while satisfying differential privacy for Example 1? Is the large (global or local) sensitivity of $f$ deemed to poor accuracy? In the paper, we will give these questions some answers.

1.1 Contribution

The contributions are as follows.

We find a way to understand differential privacy from the mathematical view. By studying the differential privacy problem of Example 1, we find two differentially private mechanisms which are applicable for almost all query functions (not just the traditional query functions in differential privacy but also the functions in mathematics). By using these mechanisms, each query function can have its own custom-designed mechanisms. That is, these mechanisms can assign probability to each point in the codomain of $f$ according to the properties (the $\{I^x_i\}_i, \{R^x_i\}_i$ sequences in the paper) of $f$. These mechanisms are so flexible that they can be used to balance the accuracy of the outputs of different datasets (by setting different $\delta$ for different points in the $\{R^x_i\}_i$ sequences). From this aspect, we could say that these mechanisms touch the “microworld” of differential privacy that the sensitivity-based methods can’t.

We also present some interesting impossibility results of differential privacy which show a tight relation between accuracy and sensitivity. Roughly speaking, these results say that, for two neighboring datasets
\(x, y \in D\), if \(|f(x) - f(y)|\) is large, then there is no differentially private mechanism by which both the outputs of \(x, y\) are accurate. Although the proofs of these results are simple, to our knowledge, they are the first to appear in differential privacy.

Furthermore, we present an abstract model of differential privacy in which a differential privacy problem is modeled as finding a randomized mapping between two metric spaces. The privacy is modeled as the property of the randomized mapping to the first metric space and the accuracy is modeled as the property of the randomized mapping to the second metric space. Under this consideration, the abstract differential privacy problem seems more hard than the metric embedding problem which only needs to keep the distance of the first metric space. Some basic structures of differential privacy are presented, such as the \(\{I^i\}_i\), \(\{R^i\}_i\) sequences in the paper. Some of the properties of these structures, such as the convergence, are also discussed (for some special cases).

Finally, we apply our theoretical results to the subgraph counting problem and the linear query problem. The experiments show that our mechanisms (in most cases) have more accurate results than the state of the art mechanisms. The results of the linear queries show an interesting phenomenon: Being contrary to the sensitivity-based mechanisms which have monotonic (from the center to two sides) density functions (such as the Laplace density function), our mechanisms generate some density functions which are not monotonic at all.

1.2 Related Works

As discussed in Section 1, the sensitivity-based methods are extensively used to construct differentially private mechanisms. These methods include the global sensitivity-based method [12], the smooth sensitivity-based method [3], the local sensitivity-based method [4] among others [5,6,7]. Different from these methods, our mechanisms do not resort on the notion of sensitivity at all. The detail is discussed in Section 4.3. Due to this reason, our mechanisms can treat what the sensitivity-based methods can’t, such as the infinite sensitivity problem in Example 1.

We model the differential privacy problem as finding a randomized mapping between two metric spaces. There are somewhat similar treatments about these concepts in [14,15]. Our treatment is different from theirs in the following aspect. The papers [14,15] mainly focus on either generalizing the differential privacy to a broaden scope or unifying the current problems or methods in differential privacy, whereas ours mainly
Differential Privacy of Mathematical Functions

focuses on the mathematical understanding and treatment of differential privacy. We try to find methods from mathematical views.

Traditionally, the exponential mechanism [2] is seen as a general mechanism for all query functions. However, the utility function used in the exponential mechanism is not presented explicitly, which means that one needs to design the utility function when using the exponential mechanism. However, how to design a good utility function is not known. Our mechanisms (at least theoretically) give some explicit ways about how to construct differentially private mechanisms for almost all query functions. The detailed discussions are presented in Section 4.3.

The batch linear queries problems [16-19] are studied extensively in differential privacy. These works focus on the dependence (such as linear dependence) among the batch linear queries. In order to improve the accuracy, they first compress the high-dependent linear queries to the low-dependent or independent linear queries of small numbers and then use a differentially private mechanism, such as the Laplace mechanism, on the later, or as a whole. Our work can be used to treat the later low-dependent or independent linear queries to improve the accuracy of the whole of the batch linear queries.

There are a large amount of works [20-23] which approximate the batch linear queries to a dataset $x$ by the batch linear queries to another dataset $y$, a synthetic dataset. These works are based on the statistical property of both the query functions and the arguments, i.e., the datasets. Our work does not treat these approximation problems.

Metric embedding [24] seems to have deep connection to differential privacy. The paper [25] uses the Johnson-Lindenstrauss transform to construct differentially private mechanism for some applications, such as PCA, Minimum cut query. The paper [26] uses a variant of Bourgain’s theorem to solve $k$-means distances problem in differential privacy. Our work abstract the differential privacy problem as finding a randomized mapping between two metric spaces which seems similar with the randomized metric embedding problem [24]. The detailed discussions about their differences are presented in Section 3.5.

The impossibility results in [27] and the low bounds results in [9, 8, 28] mainly focus on either the linear queries or the low-sensitivity queries, whereas our impossibility results in section 4.2 are not limited either to the linear queries or to the low-sensitivity queries.
Fig. 1. The set $N_i^x$ is shown as the blue and the green lines in $x$-axis. The set $I_i^x$ is shown as the rational points in the yellow lines and the irrational points in the red lines in $y$-axis.

1.3 Outline

The rest of the paper is organized as follows: Section 2 gives two mechanisms to achieve differential privacy of Example 1. It discusses how to accurately achieve differential privacy for Example 1, how to balance the accuracy of different datasets and what is the impossibility for the accuracy. Section 3 gives the abstract model of differential privacy, which abstract a differential privacy problem as finding a randomized mapping between two metric spaces problem. Section 4 generalizes the results in Section 2 to general differential privacy problems defined in Section 3. In Section 5 we apply the mechanisms in Section 4 to the subgraph counting problem and the linear query problem. Finally, concluding remarks and a discussion of future work are presented in Section 6.

1.4 Notations

Let $\mathbb{N}$ denote the set of the natural numbers including 0. Let $\mathbb{Q} = \mathbb{R} - \mathbb{Q}$. For a set $S \subseteq \mathbb{R}^n$, let $\text{Vol}(S)$ denote the measure of $S$ on the Lebesgue measure $m$ on $\mathbb{R}^n$. For the discrete case, let $\text{Vol}(S)$ denote the number of elements in $S$, i.e., $\text{Vol}(S) = |S|$.

2 Achieving Differential Privacy for Example 1

In this section, we see how to design differentially private mechanisms for Example 1. The results in this section greatly inspired us in obtaining the general results in Section 4.
Let $D, R$ be denoted as in Example 1. For $x \in D$ let $p^x(r), r \in R$ denote the probability density function of $M(x)$. Our idea to construct $p^x(r)$ is simple: the more far away of $y \in D$ from $x$ the more smaller of $p^x(f(y))$. The details are as follows. Let $N^x_i = \{y \in D : i-1 < |x-y| \leq i\}$ denote the set of neighbors of $x$ of distance $i$, and let $I^x_i = \{f(y) : y \in N^x_i\}$ for $i \in \mathbb{N}$. For any $r \in I^x_i$, set $p^x(r) = \frac{1}{\alpha} e^{-i\epsilon}$, where $\alpha^x = \sum_{i \in \mathbb{N}} e^{-i\epsilon} \mu(I^x_i)$ and $\mu$ is the Lebesgue measure on $\mathbb{R}$. Note that

$$I^x_i = [x-i, x-(i-1)]_{\mathbb{Q}} \cup (x+i-1, x+i]_{\mathbb{Q}} \cup [-x-i, -x-(i-1)]_{\mathbb{Q}} \cup (-x+i-1, -x+i]_{\mathbb{Q}}$$

as shown in Fig. 1, where $[a,b]_{\mathbb{Q}} = [a,b] \cap \mathbb{Q}$, $[a,b]_{\overline{\mathbb{Q}}} = [a,b] \cap \overline{\mathbb{Q}}$, and $\overline{\mathbb{Q}} = \mathbb{R} - \mathbb{Q}$. Note also that $\mu(I^x_i) = 2$ for all $i \in \mathbb{N}$ and all $x \in D$. Then,

$$\alpha^x = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(I^x_i) = \frac{2e^{-\epsilon}}{1-e^{-\epsilon}}$$

for all $x \in D$. For any neighbors $x, x' \in D$ and any $r \in R$, assuming that $r \in I^x_i \cap I^{x'}_j$, one can verify that $j \in \{i-1, i, i+1\}$. We have

$$\frac{p^x(r)}{p^{x'}(r)} = \frac{e^{-i\epsilon}}{\alpha^x} \frac{e^{-j\epsilon}}{\alpha^{x'}} = e^{(j-i)\epsilon} \leq e^\epsilon.$$

Therefore, the mechanism $M$ is $\epsilon$-differentially private.

### 2.1 Accuracy Analysis

Now we analyze the utility of the mechanism $M$. Let $T \in \mathbb{Z}_+$ be a positive integer. If $x \in \mathbb{Z}_+$ such that $T < 2x$, there is

$$\Pr[|M(x) - f(x)| < T] = \int_{x-T}^{x+T} p^x(r)\mu(dr)$$

$$= \int_{r \in [x-T, x+T]_{\overline{\mathbb{Q}}}} p^x(r)\mu(dr)$$

$$= \sum_{i=1}^{2T} e^{-(2x-T+i)\epsilon} / \alpha^x \quad \text{set } x = Tn$$

$$= \frac{1}{2}(e^{(-2n+1)T\epsilon} - e^{(-2n-1)T\epsilon}).$$

Other cases of $x \in \mathbb{Q}$ can be treated similarly as above.
If \( x \in \bar{Q} \), then
\[
\Pr[|M(x) - f(x)| < T] = \int_{-x-T}^{x+T} p^x(r) \mu(dr)
\]
\[
= \int_{r \in [-x-T, -x+T]} p^x(r) \mu(dr)
\]
\[
= 2 \sum_{i=1}^{T} e^{-i\epsilon}/\alpha^x
\]
\[
= 1 - e^{-T\epsilon}.
\]

From the above two probabilities we can see that, for the utility of the mechanism \( M \), the case of \( x \in \bar{Q} \) is largely accurate than the case of \( x \in Q \), especially when \( |x| \) is very large. Recalling that \( \mu(Q) = 0 \), we obtain a mechanism which has accurate outputs with probability measure 1 on \( D \).

2.2 Impossibility Result

Now that, by Section 2.1, the mechanism \( M \) has only points in \( D \) of zero probability measure which have inaccurate outputs, can we design a mechanism which has accurate outputs for every points in \( D \) for Example 1? The answer is negative. The reason is as follow.

Let \( T \) be a large positive integer. Let \( x \in Q, y \in \bar{Q} \) such that \( |x-y| < 1 \) (meaning that \( x, y \) are neighbors) and \( x, y \gg 2T \). Assume that there exists one \( \epsilon \)-differentially private mechanism \( M' \) such that \( \Pr[|M'(y) - f(y)| \leq T] \geq 1 - \eta \), where \( f(x) \) is shown in Example 1. We have
\[
1 - \eta \leq \int_{-y-T}^{y+T} p^y(r) \mu(dr) \leq e^\epsilon \int_{-y-T}^{y+T} p^x(r) \mu(dr).
\]

Therefore,
\[
\Pr[|M'(x) - f(x)| < T] = \int_{x-T}^{x+T} p^x(r) dr \leq 1 - \int_{-y-T}^{y+T} p^x(r) dr \leq 1 - e^{-\epsilon} + e^{-\epsilon} \eta.
\]

The above results show that it is impossible to find an \( \epsilon \)-differentially private mechanism which has good accuracy at both \( x \) and \( y \) when \( \epsilon \) is small.

Now that, for any differentially private mechanism, it is impossible to achieve good accuracy at both \( x, y \), can we balance the accuracy between \( x \) and \( y \)?
2.3 Balance Accuracy

In this section, we present a differentially private mechanism \( M'' \) which is a variant of \( M \) in Section 2. The mechanism \( M'' \) can balance the accuracy between the points in \( Q \) and the points in \( \bar{Q} \) compared to \( M \). For the clarity of illustration, we only treat the case \( x \in \mathbb{Z}_+ \) and the case \( x \in \mathbb{Q}_+ \) with \( x \approx \lfloor x \rfloor \). Other cases can be treated similarly.

Our idea of constructing \( M'' \) is direct: The inaccuracy at the points \( x \in \mathbb{Q} \) of \( M \) is due to the fact that most of the values \( f(y) \) are assigned a small density value if \( y \in D \) is near to \( x \). Therefore, we can set a large density value to the values of most of those points near to \( x \). The details of constructing \( M'' \) are as follows. For any \( x \in \mathbb{D} \), set \( R^x_0 = \{ r \in R : |f(x) - r| \leq 1 \} = [f(x) - 1, f(x) + 1] \). Set \( R^x_i = \cup_{y \in \mathbb{N}^2} R^y_0 - \cup_{j=1}^{i-1} R^y_j \).

Note that the construction of \( R^x_i \) is some what complicated than \( T^x_i \) of the mechanism \( M \). This is due to the fact that each point \( f(y) \in R \) is extended to the interval \( f(y) + [-1, 1] \) and that \( R^x_i \) may have overlapped points with \( R^x_j \) when \( i > j \) and therefore these overlapped points should be removed from \( R^x_i \). As in Section 2, for any \( r \in R^x_i \), set \( p^x(r) = \frac{1}{\alpha^x} e^{-ix} \), where \( \alpha^x = \sum_{i \in \mathbb{R}} e^{-ix} \). In this way, we can balance \( M \)'s accuracy between the points in \( Q \) and the points in \( \bar{Q} \). The details of the analysis are as follows.

In the following of this section we set \( |x| > 2 \) for the clarity of illustration. If \( x \in \mathbb{Z}_+ \), then \( R^x_0 = [x - 1, x + 1] \) and

\[
R^x_1 = \cup_{y \in \mathbb{N}^2} R^y_0 - R^x_0 = \cup_{y \in \mathbb{N}^2 \cap Q} R^y_0 \cup_{y \in \mathbb{N}^2 \cap \bar{Q}} R^y_0 - R^x_0 = [-x - 2, -x + 2] \cup [x - 2, x - 1] \cup (x + 1, x + 2].
\]

For \( x - 1 \geq i \geq 2 \), we have

\[
R^x_i = \cup_{y \in \mathbb{N}^2} R^y_0 - \cup_{j=1}^{i-1} R^y_j = [-x - (i + 1), -x - i] \cup (-x + i, -x + i + 1] \cup [x - (i + 1), x - i] \cup (x + i, x + i + 1].
\]

For \( i \geq x \), we have

\[
R^x_i = [-x - (i + 1), -x - i] \cup (x + i, x + i + 1].
\]

If \( x \in \mathbb{Q}_+ \) with \( x \approx \lfloor x \rfloor \) (meaning that we can substitute \(-\lfloor x \rfloor \) for \( f(x) = -x \)), then \( R^x_0 = [-x - 1, -x + 1] \) and

\[
R^x_1 = [-x - 2, -x - 1] \cup (-x + 1, -x + 2] \cup [x - 2, x + 2].
\]

For \( x - 1 \geq i \geq 2 \), we have

\[
R^x_i = [-x - (i + 1), -x - i] \cup (-x + i, -x + i + 1] \cup [x - (i + 1), x - i] \cup (x + i, x + i + 1].
\]
For $i \geq x$, we have
\[ R^x_i = [-x - (i + 1), -x - i] \cup (x + i, x + i + 1). \]

For $x \in \mathbb{Z}_+$ or $x \in \bar{\mathbb{Q}}_+$ with $x \approx [x]$, we have
\[ \alpha^x = 2e^0 + 6e^{-\epsilon} + \sum_{i=2}^{x-1} 4e^{-i\epsilon} + \sum_{i=x}^{\infty} 2e^{-i\epsilon} = \frac{2 + 4e^{-\epsilon} - 2e^{-2\epsilon} - 2e^{-x\epsilon}}{1 - e^{-\epsilon}}. \]

Then, one can prove that $\frac{p^x(r)}{p^y(r)} \leq e^\epsilon \times \alpha^y / \alpha^x \leq e^{2\epsilon}$ for any neighbors $x, y \in D$ and any $r \in R$ as in \([3]\). Therefore, the mechanism $\mathcal{M}'$ is $2\epsilon$-differentially private.

We now analyze the accuracy of the above mechanism $\mathcal{M}''$. Let $T$ be a positive integer such that $T \ll 2x$. If $x \in \mathbb{Z}_+$, we have
\[
\Pr[|\mathcal{M}''(x) - f(x)| < T] = \int_{x-T}^{x+T} p^x(r)\mu(dr) = \frac{1 - e^{-T\epsilon}}{1 + 2e^{-\epsilon} - e^{-2\epsilon} - e^{-x\epsilon}}.
\]

For $x \in \bar{\mathbb{Q}}_+$ with $x \approx [x]$, we have
\[
\Pr[|\mathcal{M}''(x) - f(x)| < T] = 2 \sum_{i=0}^{T-1} e^{-i\epsilon} / \alpha^x = \frac{1 - e^{-T\epsilon}}{1 + 2e^{-\epsilon} - e^{-2\epsilon} - e^{-x\epsilon}}.
\]

Other cases can be treated similarly. Note that the above two probabilities are approximately the same and therefore can be seen as a balance of the corresponding two probabilities in Section 2.1.

### 3 The Abstract Model of Differential Privacy

In Section 2, we see that Example 1, a pure mathematical example, has the similar problems, i.e., privacy-accuracy tradeoffs, as the differential privacy in the computer science \([1]\). Therefore, it is needed to unify the two fields for common development. In this section, we give an abstract model of differential privacy, whose intention is to understand differential privacy from theoretical or mathematical view. There are somewhat similar treatments in \([14,15]\). The difference of ours from theirs is discussed in Section 1.2.

#### 3.1 The Dataset Metric Space and the Value Metric Space

The dataset universe is modeled as a set $D$ over which a metric $\bar{d}$ is defined.
Definition 1 (Dataset Metric Space). Let \( f \) be a function defined on the set \( D \) on which a metric \( \bar{d} \) is defined. Then the metric space \((D, \bar{d})\) is called the dataset metric space of \( f \). Two elements \( x, y \in D \) are said to be neighbors of distance \( k \) if \( k - 1 < \bar{d}(x, y) \leq k \), for \( k \in \mathbb{N} \). When \( k = 1 \), \( x, y \) are said to be neighbors.

We denote \( N^x_i = \{ y \in D : i - 1 < \bar{d}(x, y) \leq i \} \), for \( i \in \mathbb{N} \), and denote \( \bar{N}^x_i = N^x_i \) for abbreviation. We denote \( N^x_1 = N^x \) for abbreviation. The codomain of a query function \( f \) on \( D \) is modeled as a set \( R \) over which a metric \( d \) is defined.

Definition 2 (Value Metric Space). For a function \( f \) on \( D \), set \( R = \{ f(x) : x \in D \} \). Defining a metric \( d \) on \( R \), then \((R, d)\) is called the value metric space of \( f \). Equipping \( R \) with the Borel \( \sigma \)-algebra \( B \) generated by the open sets in \( R \) (in the metric topology), then \((R, B)\) is a measurable space.

The product metric space and the product probability space are used to model the batch query functions.

Definition 3 (Product Metric Space). If \((R_1, d_1), \ldots, (R_n, d_n)\) are metric spaces, and \( N \) is the Euclidean norm on \( \mathbb{R}^n \), then \((R_1 \times \cdots \times R_n, N(d_1, \ldots, d_n))\) is a metric space, where the product metric is defined by

\[
N(d_1, \ldots, d_n)((x_1, \ldots, x_n), (y_1, \ldots, y_n)) = N(d_1(x_1, y_1), \ldots, d_n(x_n, y_n)),
\]

and the induced topology agrees with the product topology.

Definition 4 (Product Probability Space). Let \((R_1, B_1, \mu_1), \ldots, (R_n, B_n, \mu_n)\) be \( n \) probability spaces. Then the probability space \((R, B, \mu)\), defined by \( R = R_1 \times \cdots \times R_n, B = B_1 \times \cdots \times B_n \) and \( \mu = \mu_1 \times \cdots \times \mu_n \), is called the product probability space of the \( n \) probability spaces.

For \( n \) query functions \( f_1, \ldots, f_n \) over the dataset metric space \((D, \bar{d})\), let \((R_1, d_1), \ldots, (R_n, d_n)\) be their value metric spaces respectively. Then the product metric space \((R_1 \times \cdots \times R_n, N(d_1, \ldots, d_n))\) is called the (product) value metric space of \( f_1, \ldots, f_n \).

3.2 The Definition of Differential Privacy

Definition 5 (\( \epsilon \)-Differential Privacy). For a dataset metric space \((D, \bar{d})\), let \( \Delta(R) \) denote the set of all the probability measures on \((R, B)\). A mapping \( M : D \to \Delta(R) \) gives \( \epsilon \)-differential privacy if for all neighbors
where we abuse the notation $\mathcal{M}(x)$ as either denoting a probability distribution in $\Delta(\mathcal{R})$ or denoting a random variable following the probability distribution.

**Theorem 1 (Combination Theorem).** For a dataset metric space $(\mathcal{D}, d)$, let $\mathcal{M}_i$ be $\epsilon_i$-differentially private on $(\mathcal{R}_i, \mathcal{B}_i)$ for $i \in \{1, \ldots, n\}$. Then the combination of $\mathcal{M}_1, \ldots, \mathcal{M}_n$, defined by $\mathcal{M}(x) = (\mathcal{M}_1(x), \ldots, \mathcal{M}_n(x))$, $x \in \mathcal{D}$, is $\sum_{i=1}^{n} \epsilon_i$-differentially private on the product probability space $(\mathcal{R}, \mathcal{B})$.

**Proof.** The proof is similar with the one of Theorem 3.14 in [29] and is omitted.

### 3.3 Utility Model

Let $(\mathcal{R}, d)$ be the value metric space of $f$. Set $C^x_T = \{r \in \mathcal{R} : d(f(x), r) \leq T\}$ for all $x \in \mathcal{D}$, where $T > 0$. We use the probability $P^x_T$ to measure the accuracy of $\mathcal{M}$ at the point $x$, where

$$P^x_T = \Pr[\mathcal{M}(x) \in C^x_T].$$

We use the set $\{P^x_T : x \in \mathcal{D}\}$ to measure the accuracy of $\mathcal{M}$. An alternative to $P^x_T$ is to use the expected value of the distance $d(\mathcal{M}(x), f(x))$, i.e.,

$$\mathbb{E}[d(\mathcal{M}(x), f(x))] = \int_{r \in \mathcal{R}} d(r, f(x))dF^x(r),$$

where $F^x(r)$ is the cumulative distribution function of $\mathcal{M}(x)$.

Note that the accuracy of the batch queries is measured on their product value metric space. Note also that the functions compression method is applicable in our model to improve the batch queries’ accuracy as discussed in Section 1.2.

### 3.4 Query Function

Linear function is known to be one kind of the simplest query functions in differential privacy, which is a generalization of the sum function or the counting function.
**Definition 6 (Linear Function).** Assume the value metric space \((\mathcal{R}, d)\) is a linear space. For all \(x, y \in \mathcal{D}\), if \(\{f(x') - f(x) : x' \in \mathcal{N}\} = \{f(y') - f(y) : y' \in \mathcal{N}\}\), the function \(f\) is said to be a linear (query) function.

Since for a linear function \(f\) the set \(\{f(x') - f(x) : x' \in \mathcal{N}\}\), denoted as \(V_f\), is the same for all \(x \in \mathcal{D}\) and seems different from other linear functions, we may use it to denote the function \(f\) if without ambiguity. We call the set \(V_f\) the *neighborhood set* of the linear function \(f\). Any query function, which is not a linear function, is said to be a non-linear (query) function.

**Definition 7 (Monotonic Function).** The function \(f\) is said to be a monotonic (query) function if \(d(f(x), f(y)) \geq d(f(x), f(z))\) for any \(x \in \mathcal{D}\) and any \(y, z \in \mathcal{D}\) such that \(\bar{d}(x, y) > \bar{d}(x, z)\).

**Definition 8 (Permutation Function and Identity Function).** An injective function \(f\) is called a permutation function if its value metric space \((\mathcal{R}, d)\) is the same as the dataset metric space \((\mathcal{D}, \bar{d})\). Moreover, if \(f(x) = x\) for all \(x \in \mathcal{D}\), then \(f\) is called an identity function.

The identity function is used to model the data publication problem \([30, 31]\) in differential privacy.

### 3.5 Discussion

By the above definitions, the differential privacy of a function \(f\) is to find a randomized mapping \(\mathcal{M}\) from the dataset metric space \((\mathcal{D}, \bar{d})\) to the value metric space \((\mathcal{R}, d)\). Note that \(\mathcal{M}\) is a differentially private approximation of \(f\). Differential privacy needs the mapping \(\mathcal{M}\) should control the upper bound of the distance of the outputs (random variables) according to the distance of the inputs. The distance between the random variables is measured by the inequality \([4]\). On the other hand, the utility needs the mapping \(\mathcal{M}\) should, when inputting \(x \in \mathcal{D}\), output a random variable \(\mathcal{M}(x)\) which is a good approximation to \(f(x)\). This implies that high utility means that the outputs of the mapping \(\mathcal{M}\) should keep the distance of the metric space \((\mathcal{R}, d)\). That is, if \(d(f(x), f(y))\) is large, then the two random variables \(\mathcal{M}(x), \mathcal{M}(y)\) should also have large distance. Note that there would be a contradiction if both high privacy and high utility need holding when \(d(x, y)\) is small but \(d(f(x), f(y))\) is large. Therefore, there should be a tradeoff between privacy and utility. Furthermore, from Section 2 we can see that there is another tradeoff in differential privacy which balances the accuracy of \(\mathcal{M}\) among the different elements of \((\mathcal{D}, \bar{d})\).
In the next section we will give some concrete solutions for these tradeoffs and give some impossibility results.

Note that the differential privacy problem presented above is different from the randomized metric embedding problem in \cite{24} where one needs to find a randomized mapping $\mathcal{M}$ between two metric spaces $(\mathcal{D}, \bar{d})$, $(\mathcal{R}, d)$ such that, for any $x, y \in \mathcal{D}$, there is

$$\bar{d}(x, y)/c \leq d(\mathcal{M}(x), \mathcal{M}(y)) \leq \bar{d}(x, y)$$

(5)

with probability $1 - \eta$. Although both of them treat the randomized mapping problem between two metric spaces, there are several major differences. First, in the differential privacy there is one query function $f$ between two metric spaces but none in the randomized metric embedding. Second, in the randomized metric embedding the randomized mapping $\mathcal{M}$ should keep the distance of the first metric space by the inequality \cite{5}. However, in the differential privacy, the privacy requirement needs the randomized mapping $\mathcal{M}$ only control the upper bound of the distance of the outputs according to the distance of the inputs using the inequality \cite{4}, which implies that the case where the outputs of the randomized mapping are the same is allowable. Third, in the differential privacy the high utility implies the outputs of $\mathcal{M}$ should keep the distance of the second spaces. However, the above implication is not reversible since there may be the case where the outputs of $\mathcal{M}$ keeps the distance of the second metric but the utility is low.

One drawback of the abstract model of differential privacy is that there are many differential privacy problems whose utility functions are not metrics \cite{30,31}. However, we believe that the model captures the core issue of differential privacy, i.e., the privacy-utility tradeoff. Figuring out the problems in the model would be the cornerstone to understand the whole field of differential privacy.

Note that, by the results of this section, any function $f$ can have its own differential privacy problem so long as its domain and codomain both are metric spaces. Therefore, an SQL query function may have the same (or similar) differential privacy problem as a mathematical function. From this aspect we can say that it is the obligation not just of computer scientists but also of mathematicians to figure out differential privacy.

4 General Result

In this section, we see how to generalize the ideas and the methods in Section 2 to treat the abstract query problems in differential privacy defined
in Section 3. We first construct the set sequence \( \{I^x_i, i \in \mathbb{N}\} \) for general query functions.

Let \((\mathcal{D}, d), (\mathcal{R}, d)\) and \((\mathcal{R}, \mathcal{B})\) be defined as in Definition 1 and Definition 2 and let \( f \) be the corresponding query function. Let \((\mathcal{R}, \mathcal{B}, \mu)\) be a measure space. For \( x \in \mathcal{D} \), let \( I^x_0 = \{f(x)\} \). Let \( I^x_i = \bigcup_{y \in N^x_i} I^y_i - A^x_{i-1} \), where \( A^x_{i-1} = \bigcup_{y \in N^x_{i-1}} I^y_0, N^x_i \) and \( N^x_{i-1} \) are defined as in Section 3.1, \( i \in \mathbb{N} \).

We have the following lemma for the the set sequence \( \{I^x_i\}_i \).

**Lemma 1.** Let \( x, x' \in \mathcal{D} \) be neighbors. Then, for any \( s, t \in \mathbb{N} \) such that \(|s - t| \geq 2\), there have that \( \cup_{i \in \mathbb{N}} I^x_i = \mathcal{R} \) and that \( I^x_i \cap I^x'_i = \emptyset \).

**Proof.** We first prove \( \cup_{i=0}^t I^x_i = A^x_i \) for all \( t \in \mathbb{N} \). Note that \( I^x_i \subseteq \cup_{y \in N^x_i} I^y_i \subseteq A^x_i \) for all \( i \leq t \). We have \( \cup_{i=0}^t I^x_i \subseteq A^x_i \) for all \( t \in \mathbb{N} \). We next prove \( \cup_{i=0}^t I^x_i \supseteq A^x_i \) by induction. Obviously, \( \cup_{i=0}^0 I^x_i = A^x_0 \); Assume that \( \cup_{i=0}^t I^x_i \supseteq A^x_i \) for all \( t < k \); For any \( r \in A^x_k \), there exist \( y \in N^x_k \) and a minimum \( j \in \mathbb{N} \) such that \( r \in A^x_j \) and \( d(x, y) = j \leq k \). If \( j < k \) we have \( r \in A^x_j \subseteq \cup_{i=0}^j I^x_i \subseteq \cup_{i=0}^k I^x_i \) by the assumption. If \( j = k \) then there is no \( i < k \) such that \( r \in A^x_i \) by the minimality of \( j \). We have \( r \in A^x_k - A^x_{k-1} = \cup_{y \in N^x_k} I^y_0 - A^x_{k-1} = I^x_k \subseteq \cup_{i=0}^k I^x_i \). In conclusion, \( \cup_{i=0}^t I^x_i = A^x_i \) for all \( x \in \mathcal{D}, t \in \mathbb{N} \).

Note that \( \cup_{i \in \mathbb{N}} I^x_i = \mathcal{R} \) is a direct corollary of \( \cup_{i=0}^t I^x_i = A^x_i \) for all \( x \in \mathcal{D}, t \in \mathbb{N} \). We next prove \( I^x_i \cap I^x'_i = \emptyset \). Without loss of generality, set \( s \leq t - 2 \). Since \( I^x_t = \cup_{y \in N^x_t} I^y_0 - A^x_{t-1} \), for any \( r \in I^x_t \), we have that there exists \( y \in \mathcal{D} \) such that \( d(x, y) = t \) and \( r = f(y) \), and that for any \( y' \in \mathcal{D} \) such that \( d(x, y') \leq t - 1 \) there has \( r \neq f(y') \). On the other hand, for any \( r' \in I^x_s \), there exists \( \hat{x} \in \mathcal{D} \) such that \( d(\hat{x}, x') = s \) and \( r' = f(\hat{x}) \). Since \( d(x, \hat{x}) \leq d(x, x') + d(x', \hat{x}) = 1 + s \leq t - 1 \), we have \( r \neq r' \), which implies \( I^x_t \cap I^x_s = \emptyset \).

The claims are proved.

Similar as in Section 2 we now construct a mechanism \( \mathcal{M} \) as follows.

For any \( x \in \mathcal{D} \) and any \( r \in \mathcal{R} \) set the density function of \( \mathcal{M}(x) \) as \( p^x(r) = \frac{1}{\alpha x} e^{-ir} \) if \( r \in I^x_i \), where the normalizer \( \alpha x = \sum_{i=0}^{\infty} e^{-ir} \mu(I^x_i) \).

**Theorem 2.** Let \( \mathcal{M} \) be defined as above. Then \( \mathcal{M} \) is \( 2\epsilon \)-differentially private.

**Proof.** Let \( x, x' \in \mathcal{D} \) be neighbors. For any \( r \in \mathcal{R} \), assuming \( r \in I^x_i \cap I^x'_i \), there is \( j \in \{i - 1, i, i + 1\} \) by Lemma 1. Therefore,

\[
\frac{p^x(r)}{p^{x'}(r)} = \frac{e^{-ir}}{e^{-jir}} \times \frac{\alpha x}{\alpha x'} \leq e^\epsilon \times e^\epsilon = e^{2\epsilon}
\]

(6)

The proof is complete.
4.1 Balance Accuracy

In this section, we construct a differentially private mechanism, which is a variant of the one in Theorem 2. The construction of the mechanism is similar with the one in Section 2.3.

For each \( x \in \mathcal{D} \) and a nonnegative real number \( \delta^x \), set \( R_{0,x,\delta^x} = \{ r \in \mathcal{R} : d(f(x), r) \leq \delta^x \} \). For \( i \in \mathbb{N} \), set \( R_{i,x,\delta^x} = \bigcup_{y \in N_{i,x}} R_{y,\delta^y} \setminus B_{i-1,x,\delta^x} \), where \( B_{i-1,x,\delta^x} = \bigcup_{y \in N_{i-1,x}} R_{y,\delta^y} \). In the following, we denote \( R_{i,x} = R_{i,x,\delta^x} \) and \( B_{i,x} = B_{i-1,x,\delta^x} \) for notational simplicity when there is no ambiguity.

Lemma 2. Let \( x, x' \in \mathcal{D} \) be neighbors. Then, for any \( s, t \in \overline{N}_x \) such that \( |s - t| \geq 2 \), there have that \( \bigcup_{i \in \mathbb{N}} R_{i,x} = \mathcal{R} \) and that \( R_{i,x} \cap R_{i,x'} = \emptyset \).

Proof. The proof of Lemma 2 is similar with the one of Lemma 1 and is omitted.

We now construct a mechanism \( \mathcal{M} \) as follows. For any \( x \in \mathcal{D} \) and any \( r \in \mathcal{R} \) set the density function of \( \mathcal{M}(x) \) as \( p^x(r) = \frac{1}{\alpha^x} e^{-i\epsilon} \) if \( r \in R_{i,x} \), where the normalizer \( \alpha^x = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(R_{i,x}) \).

Theorem 3. Let \( \mathcal{M} \) be defined as above. Then \( \mathcal{M} \) is \( 2\epsilon \)-differentially private.

Proof. The proof of Theorem 3 is similar with the one of Theorem 2 and is omitted.

4.2 Impossibility Result

The impossibility result in Section 2.2 is generalized as follows. Set \( C_{T,x} = \{ r \in \mathcal{R} : d(f(x), r) \leq T \} \) for all \( x \in \mathcal{D} \), where \( T > 0 \). We have the following result.

Lemma 3. Assume that there is \( y \in N_{i,x} \) such that \( f(y) \notin C_{T,x} \), where \( i \in \mathbb{N} \). Assume also that \( \mathcal{M} \) is an \( \epsilon \)-differentially private mechanism such that \( \Pr[\mathcal{M}(x) \in C_{T,x}] \geq 1 - \eta \). Then \( \Pr[\mathcal{M}(y) \in C_{T,y}] \leq 1 - e^{-i\epsilon} + e^{-i\epsilon} \eta \).

Proof. Note that

\[ 1 - \eta \leq \Pr[\mathcal{M}(x) \in C_{T,x}] \leq e^{i\epsilon} \Pr[\mathcal{M}(y) \in C_{T,y}], \quad (7) \]

by the inequality (4). Then

\[ \Pr[\mathcal{M}(y) \in C_{T,y}] \leq 1 - \Pr[\mathcal{M}(y) \in C_{T,x}] \leq 1 - e^{-i\epsilon}(1 - \eta) \]

where the first inequality is due to \( C_{T,x} \cap C_{T,y} = \emptyset \), and the second inequality is due to (7).

The claim is proved.
Lemma 3 says that if \( x, y \in \mathcal{D} \) are close in the metric space \((\mathcal{D}, \bar{d})\) and \( f(x), f(y) \) are far away in the metric space \((\mathcal{R}, d)\), then there is no differentially private mechanism which can achieve good accuracy at both \( x, y \). We have the following impossibility result by Lemma 3.

**Theorem 4.** Assume the volume of \( \mathcal{D} \) is finite, i.e., \( \text{Vol}(\mathcal{D}) < \infty \). Assume that there are two sets \( \mathcal{D}_1, \mathcal{D}_2 \subseteq \mathcal{D} \) such that \( \mathcal{D}_2 \subseteq \cap_{x \in \mathcal{D}_1} N^x \) and that, for any one \( x \in \mathcal{D}_1 \), there has \( f(y) \notin C^x_T \) for all \( y \in \mathcal{D}_2 \). Assume that \( \mathcal{M} \) is one \( \epsilon \)-differentially private mechanism. Set \( \max(\eta, 1 - e^{-\epsilon \eta} + e^{-\epsilon \eta}) < 1/2 \). There then does not exist set \( \mathcal{D}' \subseteq \mathcal{D} \), which satisfies \( \text{Vol}(\mathcal{D}') > \text{Vol}(\mathcal{D}) - \min(\text{Vol}(\mathcal{D}_1), \text{Vol}(\mathcal{D}_2)) \), such that there is \( \Pr[\mathcal{M}(x) \in C^x_T] \geq 1 - \eta \) for any \( x \in \mathcal{D}' \).

### 4.3 Discussion

The mechanism in Theorem 2 is only related to the the metric \( \bar{d} \), no needing the participation of the second metric \( d \). This is a vital feature which differentiates the mechanism from the sensitivity-base methods which need the second metric \( d \), or at least a utility function \( u^x(r) \), to compute the (global or local) sensitivity. The mechanism in Theorem 2 can be seen as a mechanism only considering the need of privacy since the metric \( d \), the measure of the utility, is not used at all.

\[
P^x_T = \Pr[\mathcal{M}(x) \in C^x_T] = \frac{\sum_{i=0}^{\infty} e^{-i \epsilon} \mu(I^x_i \cap C^x_T)}{\sum_{i=0}^{\infty} e^{-i \epsilon} \mu(I^x_i)}
\]

(8)

The accuracy of the mechanism can be measured by \( P^x_T \) in the equation (8), which indicates that, in general, the mechanism is not suitable for the non-monotonic functions (Definition 7) since non-monotonic function may have smaller value of \( \mu(I^x_i \cap C^x_T) \) for smaller \( i \).

The mechanism in Theorem 3 extends the set \( I^x_0 = \{ f(x) \} \) in Theorem 2 into a ball \( R^x_0 = \{ r \in \mathcal{R} : d(r, f(x)) \leq \delta^x \} \). This mechanism can be seen as an optimized variant of the mechanism in Theorem 2 when considering the need of the metric \( d \), i.e., the need of utility.

\[
P^x_T = \Pr[\mathcal{M}(x) \in C^x_T] = \frac{\sum_{i=0}^{\infty} e^{-i \epsilon} \mu(R^x_i \cap C^x_T)}{\sum_{i=0}^{\infty} e^{-i \epsilon} \mu(R^x_i)}
\]

(9)

The accuracy of the mechanism can be measured by \( P^x_T \) in the equation (9).

The above two mechanisms show a way to solve the abstract differential privacy problem presented in Section 3. Note that they will give each query function some custom-designed probability distributions due to the (in general) uniqueness of the \( \{I^x_i\}_i \), \( \{R^x_i\}_i \), sequences of the function.
Algorithm 1: Generate the set sequence \(\{I^g_{k}\}\) of each graph \(g_i\)

**Input**: The set of all non-isomorphic graphs \(\mathcal{D} = \{g_1, g_2, \ldots, g_m\}\) in \(\mathcal{G}_n\)

**Output**: The matrix \(I\), where \(I[k,i]\) is the set \(I^g_{k}\) of the graph \(g_i\)

/* The matrix \(adjG\) stores the neighboring relationship of graphs */

1. for \(i = 1\) to \(m\) do
   2. for \(j = 1\) to \(m\) do
      3. if If the graph \(g_j\) is a neighbor of distance \(k\) of \(g_i\) then
         4. set \(adjG[i,j] = k\)

/* The array \(tri\) stores the number of triangles of the graphs */

6. for \(i = 1\) to \(m\) do
   7. if If the number of triangles in the graph \(g_i\) is \(k\) then
      8. set \(tri[i] = k\)
      9. set \(I[0,i] = \{k\}\)

/* The set \(I[k,i]\) is the set \(I^g_{k}\) of the graph \(g_i\) */

11. for \(k = 1\) to \(\text{edgeNo}\) do
    12. for \(i = 1\) to \(m\) do
        13. \(I\text{temp}=\text{NULL}\)
        14. for \(j = 1\) to \(m\) do
            15. if \(adjG[i,j] = = k\) then
               16. \(I\text{temp}=\text{union}(I\text{temp}, tri[j])\) // Evaluate set union
        17. for \(s = 1\) to \(k\) do
            18. \(I\text{temp}=\text{setdiff}(I\text{temp}, I[s-1,i])\) // Evaluate set difference
        19. set \(I[k,i] = I\text{temp}\)

21. return \(I\)

5 Application

In this section, we apply the two mechanisms presented in Section 4 to the subgraph counting problem and the linear query problem.

5.1 Application in Subgraph Counting

Subgraph counting is one important problem in differential privacy 7[6][5][4]. We use the edge differential privacy as in 4. That is, two graphs are said to be neighbors if the difference of their edges is 1. We count the number of triangles in a graph. The corresponding algorithm is shown in Algorithm 1, which implements the mechanism in Theorem 2 and outputs the set sequence \(\{I^g_{k}: k \in \mathbb{N}\}\) of each graph \(g_i \in \mathcal{D}\) in \(\mathcal{G}_n\), where \(\mathcal{G}_n\) denotes
the set of all the graphs with the number of nodes equals \( n \) and \( \mathcal{D} \) denotes the set of all non-isomorphic graphs in \( \mathcal{G}_n \). The algorithm for the mechanism in Theorem 3 is similar with Algorithm 1 and is omitted.

The accuracy of the mechanism \( \mathcal{M} \) at \( g_i \) is measured using the expected value of the distance \( d(\mathcal{M}(g_i), f(g_i)) \) as in Section 3.3, i.e.,

\[
\mathbb{E}[d(\mathcal{M}(g_i), f(g_i))] = \sum_{r \in \mathcal{R}} p^\mathcal{M}(r)|r - f(g_i)|,
\]

where \( f(g_i) \) denotes the number of triangles in the graph \( g_i \), \( p^\mathcal{M}(r) \) denotes the probability of \( r \) in the mechanism \( \mathcal{M} \) when the input is \( g_i \), and \( \mathcal{R} = \{ f(g_i) : g_i \in \mathcal{D} \} \). We compare the accuracy of our mechanisms to the one in [4, Algorithm 1] (we call it Ladder mechanism). For the fairness of comparison, we set the codomain of the counting function \( f \) in [4, Algorithm 1] be \( \mathcal{R} \) as above instead of \( \mathbb{Z} \). Furthermore, we substitute 2\( \epsilon \) for \( \epsilon \) in [4, Algorithm 1] which ensures that the Ladder mechanism is 2\( \epsilon \)-differentially private as ours. We evaluate the rate \( \text{Rate}(g_i) = \frac{\text{mean}(g_i)}{\text{mean}'(g_i)} \), where \( \text{mean}(g_i) \) denotes the expected value \( \mathbb{E}[d(\mathcal{M}(g_i), f(g_i))] \) of our mechanism and \( \text{mean}'(g_i) \) denotes the corresponding expected value at \( g_i \) of the Ladder mechanism.

The details of the experiments are as follows. We set \( \mathcal{G}_n \) be \( \mathcal{G}_7 \), where \( |\mathcal{D}| = 1044 \) and \( |\mathcal{R}| = 28 \). The results are shown in Fig. 2 where the point \( i \) in the \( x \)-axis denotes the graph \( g_i \), the point \( y \) in \( y \)-axis denotes the value \( 10 \times \epsilon \) and the value at the coordinate \( (i, y) \) is the value \( \text{Rate}(g_i) \) when the input graph is \( g_i \) and \( \epsilon = y/10 \).
The top figure in Fig. (2) shows the result when comparing the Ladder mechanism and the mechanism in Theorem 2. The below figure in Fig. (2) shows the result when comparing the Ladder mechanism and the mechanism in Theorem 3, where $\delta = 1$ for all the graphs in $\{g_i : i \in \{1, 2, \ldots, 100\}\}$, and $\delta = 0$ for other graphs in $\mathcal{D}$. From Fig. (2) we can see the mechanism in Theorem 2 is better than the Ladder mechanism at most graphs. However, the mechanism in Theorem 3 is worse than the Ladder mechanism at those graphs of setting $\delta = 1$. We reason that this is due to the (almost) monotonicity of the triangle counting function.

5.2 Application in Linear Query

Linear query function (Definition 6) is a kind of well studied query functions in differential privacy [9, 8, 32, 33, 17]. Instead of treating batch linear queries, we treat a linear query. Batch linear queries can be treated by the combination of the method of this section and the functions compression method as discussed in Section 1.2.

We consider a special kind of the linear queries: the sum query. For the sum query, one dataset can be denoted as its histogram $x \in \mathbb{R}^N$, with $x_i$ denoting the number of occurrences of the $i$th element of the universe $[8, 29]$. As discussed in Section 3.4 we use the neighborhood set $\mathcal{V}_f$ to denote the linear function $f$.

The details of the experiments are as follows. We consider four linear functions (over four different dataset universes) respectively. They are $\mathcal{V}_1 = [0, 1] \cup [1000, 1001]$, $\mathcal{V}_2 = [0, 100] \cup [1000, 1001]$, $\mathcal{V}_3 = [0, 500] \cup [1000, 1001]$, $\mathcal{V}_4 = [0, 1001]$, where $[a, b]$ denotes the corresponding interval in $\mathbb{R}$. We implement the mechanisms in Theorem 2 and in Theorem 3 to these queries. The corresponding algorithms to evaluate the sequences $\{I^z_i\}_i$ and $\{R^z_i\}_i$ are similar with Algorithm 1 and are omitted. The only difference is that, since the linear query is symmetric for different datasets, each of the above two sequences is the same (when subtracting the corresponding $f(x)$) for different datasets. Therefore, we only need to evaluate the sequences for only one dataset. Note that the same value $\delta$ is assigned to different datasets due to the above symmetric property. Before giving the detailed experiments, we first give some theoretical results about linear queries.

**Corollary 1.** The mechanisms in Theorem 2 and in Theorem 3 are $\epsilon$-differentially private for the linear query.

The above corollary is due to the symmetric property presented above which leads to $\alpha^x = \alpha^y$ (the proof of Theorem 2) for different datasets.
x, y. Note that, for two neighbors x, x', the mechanisms in Corollary 7 reach the bound of the inequality (4) at all the points in $I_i \cap I_i'$ or all the points in $R_i \cap R_i'$, $i \in \mathbb{N}$.

We now discuss the convergence of the set sequences.

**Definition 9 (The convergence of set sequence).** Let $\mathcal{V}_f$ be a linear query function over $\mathbb{R}$. The corresponding set sequence $\{I_i : i \in \mathbb{N}\}$ is said to be convergent if there exist $a_n$ and $n$ such that $I_{n} = a_{n} + [0, \Delta f]$ and $I_{n+1} = a_{n} + [\Delta f, 2\Delta f]$, where $\Delta f$ is the global sensitivity of $\mathcal{V}_f$.

**Proposition 1.** Assume $\mathcal{V}_f = [0, a] \cup [b, c]$ is a linear query function, where $0 < a < b < c$. Then the sequence $\{I_i : i \in \mathbb{N}\}$ is convergent when $i \geq \frac{\Delta f}{c-\delta}$.

**Proof.** Note that the interval $[b, c]$ will generate the interval $[ib, ic]$ in $I_i$. Setting $i \geq \frac{\Delta f}{c-\delta}$, we have $[(i-1)\Delta f, i\Delta f] \subseteq [ib, ic]$. This implies that $[i\Delta f, (i+1)\Delta f] \subseteq I_{i+1}$. Then it is convergent.

The corresponding $\{R_i : i \in \mathbb{N}\}$ sequence has the similar convergence property with $\{I_i : i \in \mathbb{N}\}$ as in Proposition 1.

The accuracy of the mechanism $\mathcal{M}$ is measured using the expected value of the distance $d(\mathcal{M}(x), f(x))$ as in Section 3.3, i.e.,

$$E[d(\mathcal{M}(x), f(x))] = \int_{r \in \mathbb{R}} p(r)|r - f(x)|dr,$$

(11)

where set $f(x) = 0$, $p(r)$ denotes the probability of $r$ in the mechanism $\mathcal{M}$ when the input is $x$. We compare the accuracy of our mechanisms to the one in [32, Algorithm 1] (we call it Staircase mechanism). We evaluate the rate $Rate = \frac{\text{mean}}{\text{mean}'}$, where $\text{mean}$ denotes the expected value $E[d(\mathcal{M}(x), 0)]$ of our mechanism and $\text{mean}'$ denotes the corresponding expected value of the Staircase mechanism. The results are shown in Fig. 3 where the $x$-axis denotes the values of $\delta$, the $y$-axis denotes the value of $\epsilon$ and the
value at the coordinate \((\delta, \epsilon)\) is the value \(Rate = \frac{\text{mean}}{\text{mean}}\) for the definite \(\delta\) and \(\epsilon\). In Fig. 3 the results of the four queries \(V_1, V_2, V_3, V_4\) are shown in \((a), (b), (c), (d)\) respectively.

We now analyze the results in Fig. 3. The four queries \(V_1, V_2, V_3, V_4\) have the same global (and local) sensitivity \(\Delta f = 1001\). However, the volumes of their neighborhood set (Section 3.4) is different. Explicitly, \(\text{Vol}(V_1) = 2, \text{Vol}(V_2) = 101, \text{Vol}(V_3) = 501, \text{Vol}(V_4) = 1001\). The Fig. 3 shows some interesting phenomenon: The more large of the value \(\Delta f / \text{Vol}(V_f)\), the more better of our mechanisms compared to the Staircase mechanism.

Another interesting phenomenon is that the density functions of the first three functions in Fig. 3 are not monotonic (from the center to the two sides). This is counterintuitive since it is custom to assign less probability value to the point far away from \(f(x)\), such as the Laplace density function or the Gaussian density function.

\section{5.3 Discussion}

The \(\{I^x_i\}_i\) sequence and the \(\{R^x_i\}_i\) sequence contain many important properties about the corresponding query function \(f\). The convergence property defined in Definition 9 is an useful property of the linear queries. However, there is linear query function, such as the function \(f\) with the codomain \(R = \mathbb{Z}\) and the neighborhood set \(V_f = \{1, 7, 8, 10\}\), whose \(\{I^x_i\}_i\) sequence and the \(\{R^x_i\}_i\) sequence do not have the convergence property of Definition 9 but have other forms of “convergence” property: to the function \(f\) the volume of the sequences converge to \(2\Delta f = 20\). These “convergence” properties are need to be considered further. Furthermore, the property of the query function, such as the monotonic property, may have effect on the mechanism of the \(\{R^x_i\}_i\) sequence. Moreover, how to appropriately assign \(\delta\)s in \(\{R^x_i\}_i\) sequence for different datasets is still unknown.

\section{6 Conclusion and Future Work}

In differential privacy, the privacy-utility tradeoff is not well studied for general query functions except some linear queries, such as the counting query function. This paper gives a tentative exploration to the abstract treatment of differential privacy to treat the tradeoff. In this paper, the differential privacy problem is abstracted as finding a randomized mapping between two metric spaces. Two mechanisms are presented to construct the mappings for different query functions. They show a way to
study the abstract differential privacy problems. Other methods are still
needed to be explored to treat these problems.

Query functions should be categorized for the better understanding
of differential privacy. One possible way is to use the properties of the
\( \{I^x_i\}_i \) sequence and the \( \{R^x_i\}_i \) sequence of the query function, such as
the convergence property of linear queries. The monotonic functions and
the permutation functions defined in Section 3.4 seem more simpler or
fundamental than other functions and therefore need to be studied firstly.

Furthermore, the connection between the abstract differential privacy
in Section 3 and the randomized metric embedding needs to be considered.
There are only works of treating the differential privacy problems using
the metric embedding methods so far. We hope, in future, there may be
some metric embedding problems solved by methods in the differential
privacy.

Finally, one drawback of the mechanisms in this paper is the high com-
plexity of the algorithms compared to the sensitivity-based one. How to
construct low-complexity algorithms using the mechanisms of this paper
needs to be considered.

References

1. Cynthia Dwork. Differential privacy. In ICALP (2), pages 1–12, 2006.
2. Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In
FOCS, pages 94–103, 2007.
3. Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Smooth sensitivity
and sampling in private data analysis. In Proceedings of the 39th Annual ACM
Symposium on Theory of Computing, San Diego, California, USA, June 11-13,
2007, pages 75–84, 2007.
4. Jun Zhang, Graham Cormode, Cecilia M. Procopiou, Divesh Srivastava, and Xi-
akui Xiao. Private release of graph statistics using ladder functions. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of
Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015, pages 731–745, 2015.
5. Vishesh Karwa, Sofya Raskhodnikova, Adam D. Smith, and Grigory Yaroslavtsev.
Private analysis of graph structure. PVLDB, 4(11):1146–1157, 2011.
6. Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam D.
Smith. Analyzing graphs with node differential privacy. In TCC, pages 457–476,
2013.
7. Shixi Chen and Shuigeng Zhou. Recursive mechanism: towards node differential
privacy and unrestricted joins. In Proceedings of the ACM SIGMOD International
Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June
22-27, 2013, pages 655–664, 2013.
8. Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential
privacy: the sparse and approximate cases. In Symposium on Theory of Computing
Conference, STOC’13, Palo Alto, CA, USA, June 1-4, 2013, pages 351–360, 2013.
9. Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 705–714, 2010.

10. Moritz Hardt and Aaron Roth. Beating randomized response on incoherent matrices. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1255–1268, 2012.

11. Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 11–20, 2014.

12. Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *PVLDB*, 5(11):1364–1375, 2012.

13. Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: private data release via bayesian networks. In *SIGMOD Conference*, pages 1423–1434, 2014.

14. Naoise Holohan, Douglas J. Leith, and Oliver Mason. Differential privacy in metric spaces: Numerical, categorical and functional data under the one roof. *Inf. Sci.*, 305:256–268, 2015.

15. Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies - 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings*, pages 82–102, 2013.

16. Chao Li and Gerome Miklau. Optimal error of query sets under the differentially-private matrix mechanism. In *Joint 2013 EDBT/ICDT Conferences, ICDT ’13 Proceedings*, Genoa, Italy, March 18-22, 2013, pages 272–283, 2013.

17. Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *VLDB J.*, 24(6):757–781, 2015.

18. Grigory Yaroslavtsev, Graham Cormode, Cecilia M. Procopiuc, and Divesh Srivastava. Accurate and efficient private release of datacubes and contingency tables. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, pages 745–756, 2013.

19. Ziteng Wang, Kai Fan, Jiaqi Zhang, and Liwei Wang. Efficient algorithm for privately releasing smooth queries. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 782–790, 2013.

20. Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 339–356, 2012.

21. Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 25-26, 2010, Las Vegas, Nevada, USA*, pages 61–70, 2010.

22. Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 765–774, 2010.
23. Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Taho, Nevada, United States.*, pages 2348–2356, 2012.

24. Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 10–33, 2001.

25. Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 410–419, 2012.

26. Zhiyi Huang and Aaron Roth. Exploiting metric structure for efficient private query release. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 523–534, 2014.

27. Hai Brenner and Kobbi Nissim. Impossibility of differentially private universally optimal mechanisms. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 71–80, 2010.

28. Anindya De. Lower bounds in differential privacy. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 321–338, 2012.

29. Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

30. Rui Chen, Gergely Ács, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *ACM Conference on Computer and Communications Security*, pages 638–649, 2012.

31. Rui Chen, Noman Mohammed, Benjamin C. M. Fung, Bipin C. Desai, and Li Xiong. Publishing set-valued data via differential privacy. *PVLDB*, 4(11):1087–1098, 2011.

32. Quan Geng and Pramod Viswanath. The optimal noise-adding mechanism in differential privacy. *IEEE Trans. Information Theory*, 62(2):925–951, 2016.

33. Quan Geng and Pramod Viswanath. Optimal noise adding mechanisms for approximate differential privacy. *IEEE Trans. Information Theory*, 62(2):952–969, 2016.