Differential Privacy with δ-Neighbourhood for Spatial and Dynamic Datasets[*]

Chengfang Fang
Huawei International
fang.chengfang@huawei.com

Ee-Chien Chang
School of Computing
National University of Singapore
changec@comp.nus.edu.sg

ABSTRACT

Differential privacy provides a strong guarantee in protecting privacy of individuals who contributed to a published dataset. In this paper, we focus on spatial datasets and dynamic datasets, and attempt to exploit the intuition that farther-apart entities should have lesser influences to each other, and thus more privacy budget should be invested to protect close-by entities. To capture such intuition, we propose embedding the underlying spatial or temporal distance function into the notion of dataset neighbourhood. We called the proposed neighbourhood δ-*neighbourhood*, and discuss its implications in both spatial and dynamic datasets. For dynamic datasets, while there are known negative results on the standard differential privacy, it is possible to continuously and indefinitely publish under δ-neighbourhood by reusing the privacy budgets. Although known mechanisms, by definition, are also differentially private under δ-neighbourhood, they are not designed to exploit the relaxed notion for better utility. For spatial datasets, we propose an approach on 2D spatial points that re-allocates more budgets to nearby entities and thus obtains significantly higher utility. In addition, we give mechanisms that achieve "sustainable privacy" on dynamic datasets under both online and offline setting.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database Administration—*Security, integrity, and protection*; K.4.1 [**Computers and Society**]: Privacy

Keywords

Differential Privacy, Bounded Neighbourhood, Spatial and Temporal Datasets

1. INTRODUCTION

Many datasets contain useful statistical information for public usage. To publish such information while preserving privacy of each contributor is challenging. The recent notion of differential privacy provides a strong form of assurance in protecting individual contributors.

A probabilistic publishing mechanism $\mathcal{A}$ is ϵ-differentially private if the published data is sufficiently noisy, so that it is difficult to distinguish the membership of an entity in a group. More specifically, the following bound holds for any $R \subseteq \text{range}(\mathcal{A})$:

$$Pr(\mathcal{A}(D_1) \in R) \leq \exp(\epsilon) \cdot Pr(\mathcal{A}(D_2) \in R), \qquad (1)$$

for any two *neighbouring datasets* D_1 and D_2 that differ on at most one entity. A useful property of the formulation is that, for two datasets D_1 and D_2 that differ in more than one entity, there is still protection but with a weaker bound $\exp(c\epsilon)$, where c is the "distance" between D_1 and D_2.

In this paper, we focus on spatial datasets and dynamic datasets, and attempt to exploit the intuition that farther-apart entities should have lesser influence to each other, and thus more budget should be invested to protect close-by entities. To capture the notion of spatial and temporal distance, we adopt a alternative definition of neighbourhood. Under the original neighbourhood (let us call it the standard neighbourhood), two neighbouring datasets D_1 and D_2 differ by one entity, in the sense that $D_1 = D_2 - \{x\}$, or $D_1 = D_2 - \{y\} + \{z\}$ for some x, y, z. In other words, D_1 and D_2 are neighbours if one can be obtained from another by either adding a new entity x or replacing an entity y by z.[1] We propose an alternative form of neighbourhood: instead of having arbitrary entity x and z, they have to meet some spatial conditions. The new x must be near to some "sources" and the replacement z must near to y within a threshold δ. Such neighbourhood naturally arises from spatial datasets, for example locations of Twitter users [12] where the distance between two entities is the geographical distance between them. We call this variant δ-*neighbourhood*, where δ is the threshold.

Similar definitions of dataset neighbourhood have been considered before. For example, Kifer et al. [14] considered the *attribute differential privacy*, where two datasets are considered neighbours iff they differ at one attribute in one record. Konstantinos et al. [2] consider broadening the neighbourhood relationship with arbitrary metrics. Our no-

[*] This work is supported by the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO.

ASIA CCS'14, June 4–6, 2014, Kyoto, Japan.
Copyright 2014 ACM 978-1-4503-2800-5/14/06 ...$15.00.
http://dx.doi.org/10.1145/2590296.2590320.

[1] There are a few versions of standard neighbourhood, for example, unbounded differential privacy [11, 25], and bounded differential privacy [24, 14].

tion of δ-neighbourhood can be viewed as a variant that stresses on spatial and temporal locality.

There are a few ways to view the assurance provided by the proposed neighbourhood. In some applications, the data are subject to some constraints and thus not all possible datasets are valid. For example, Blocki et al. [1] consider social network graphs where the degree of any node is bounded, instead of all possible graphs. When such constraints are captured by δ-neighbourhood, the guarantee provided by both notions are equivalent.

Viewing from another perspective, if the domain (where the entities of the datasets are drawn from) is connected and bounded w.r.t. the underlying metric, then a mechanism that is differentially private under δ-neighbourhood is also differentially private under the standard neighbourhood. However, the guaranteed bound (as in inequality (1)) is stronger when the entities are close-by. Hence, the proposed δ-neighbourhood can be viewed as a "redistribution" of assurance, instead of a relaxation of assurance when compares to the standard neighbourhood. Illustrating examples will be discussed in Section 4 and 6.

In addition, the δ-neighbourhood can also be adopted for dynamic datasets where entities are added and removed over time. One example is the scenario considered by Dwork et al. [7], where aggregated information on users' health conditions in a region or building (say airport) is to be monitored over time. Under the standard neighbourhood, due to the fixed budget, it is impossible to publish the dataset repeatedly with high utility. However, there are scenarios where entities do not stay in the dataset for long and thus, intuitively, the effect of information published earlier would diminish over time, and hence we should be able to continuously publish with high utility. We can define a δ-neighbourhood that captures the above intuition, so as to achieve *sustainable privacy* on dynamic datasets.

Existing differential private mechanisms designed for the standard neighbourhood are, by definition, also differentially private under the δ-neighbourhood. However, these mechanisms may not fully exploit the δ-neighbourhood for better utility. For example, publishing equi-width histogram of 1D datasets induces the same amount of sensitivity under both standard and δ-neighbourhood, and thus following the well-known method of adding Laplace noise proportional to the sensitivity would not achieve higher utility. We propose an optimization approach and give a mechanism for 2D spatial points that achieve significantly higher utility. Whereas for dynamic dataset, we investigate how to allocate the privacy budget to sustain the publishing process over time, so as to minimize the expected total amount of noise in both offline and online settings. On the other hand, some mechanisms can be naturally extended to δ-neighbourhood, such as publishing sorted 1D points, and median publishing as described in Section 5.5.

The rest of the paper is organized as follow: Section 2 introduces the background on differential privacy, followed by the proposed notion of δ-neighbourhood in Section 3. The motivating examples on spatial datasets are given in section 4, and the mechanisms catered for δ-neighbourhood are given in Section 5. Section 6 and 7 are devoted to dynamic datasets. We discuss the related works in Section 8 and conclude our work in Section 9.

2. BACKGROUND

In this section, we briefly describe the notion of differential privacy.

2.1 Neighbourhood and Differential Privacy

A dataset is a multiset (i.e. a set with possibly repeating elements) of entities from the domain $\mathcal{M}$, and let us denote the collection of all datasets as $\mathbf{D}$.

Definition (**ϵ-differential privacy** [4]) A mechanism $\mathcal{A}$ satisfies ϵ-differential privacy if for all $R \subseteq \text{range}(\mathcal{A})$, and any pair of neighbours (D_1, D_2), we have:

$$Pr(\mathcal{A}(D_1) \in R) \leq \exp(\epsilon) \cdot Pr(\mathcal{A}(D_2) \in R). \tag{2}$$

In the above definition, two datasets D_1 and D_2 are neighbours if they "differ on one entity". There are a few interpretations of the above statement: some interpret it as $D_1 = D_2 \cup \{x\}$ or $D_2 = D_1 \cup \{x\}$, i.e. one dataset is a proper subset of the other with one less in size [11][25]; and in some literatures [24][14], it is interpreted as $D_1 - \{x\} = D_2 - \{y\}$ for some x, y. The former interpretation is also known as the *unbounded differential privacy*, whereas the latter as *bounded differential privacy.* In this paper, we consider both, i.e. D_1 and D_2 are neighbours iff $D_1 = D_2 \cup \{x\}$ or $D_1 - \{x\} = D_2 - \{y\}$ for some $x, y \in \mathcal{M}$, and call this the *standard* neighbourhood. Such definition of neighbourhood is also considered by Roth et al. [19].

A consequence of the bound provided by differential privacy is that, when two datasets D_1 and D_2 differ by c entities, then if a mechanism $\mathcal{A}$ is ϵ-differentially private, we have:

$$Pr(\mathcal{A}(D_1) \in R) \leq \exp(c\epsilon) \cdot Pr(\mathcal{A}(D_2) \in R), \tag{3}$$

for all possible $R \subseteq \text{range}(\mathcal{A})$. In other words, although the definition only explicitly dictates the relationship among neighbours, there are still protections on datasets that are far apart, but with a weaker bound.

2.2 Sensitivity and Laplace Mechanism

It is shown [6] that given a function $f : \mathbf{D} \to \mathbb{R}^k$ for some $k \geq 1$, the probabilistic mechanism $\mathcal{A}$ that outputs:

$$f(D) + (Lap(\triangle_f/\epsilon))^k,$$

achieves ϵ-differential privacy, where $(Lap(\triangle_f/\epsilon))^k$ is a vector of k independently and randomly chosen values from the Laplace distribution, and $\triangle_f$ is the *sensitivity* of the function f. The sensitivity of f is defined as the least upper bound on the ℓ_1 difference of all possible neighbours:

$$\triangle_f := \sup\|f(D_1) - f(D_2)\|_1,$$

where the supremum is taken over pairs of neighbours D_1 and D_2. Here, $Lap(b)$ denotes the zero mean distribution with variance $2b^2$, and a probability density function:

$$\ell(x) = \frac{1}{2b}e^{-|x|/b}.$$

3. δ-NEIGHBOURHOOD

We assume that there is a distance function $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ on the domain that captures the distance between a pair of entities, and there is a set of *sources* $S \subseteq \mathcal{M}$. With this distance function and sources, for a threshold δ, we say that two datasets D_1, D_2 are δ-neighbours if, and only if the following holds:

1. there exists x_1 and $x_2 \in \mathcal{M}$, such that $d(x_1, x_2) \leq \delta$, and $D_1 - \{x_1\} = D_2 - \{x_2\}$, or

2. there exists an x_3 and $s \in S$ s.t. $d(x_3, s) \leq \delta$, and $D_1 - \{x_3\} = D_2$ or $D_2 - \{x_3\} = D_1$.

In other words, either D_1 can be obtained from D_2 by replacing an entity x_2 with a nearby entity x_1, or by adding a new entity x_3 emerged near a source s. Note that if S is empty, then the size of D_1 and D_2 must be the same.

Given two datasets $D_1, D_2 \in \mathbf{D}$, we say that D_1 and D_2 are *connected* if there exists a finite sequence $E_0, E_1, E_2, \ldots,$ E_m with $E_0 = D_1$ and $E_m = D_2$ s.t. for any i, the consecutive E_i and E_{i+1} are δ-neighbours, and call the smallest such m the distance between D_1 and D_2. If any two datasets in $\mathbf{D}$ are connected, we say that $\mathbf{D}$ is connected, and call the least upper bound on the distance, if it exists, the *diameter* of $\mathbf{D}$.

3.1 Differential Privacy under δ-Neighbourhood

We say that a mechanism $\mathcal{A}$ is ϵ-differential privacy under δ-neighbourhood if for all $R \subseteq \text{range}(\mathcal{A})$ and any pair of δ-neighbours (D_1, D_2):

$$Pr(\mathcal{A}(D_1) \in R) \leq \exp(\epsilon) \cdot Pr(\mathcal{A}(D_2) \in R). \quad (4)$$

Similar to standard neighbourhood, we can define the sensitivity of a function f: $\mathbf{D} \rightarrow \mathbb{R}$ with respect to the δ-neighbourhood, which is

$$\sup \|f(D_1) - f(D_2)\|_1,$$

where the supremum is taken over all pairs (D_1, D_2) of δ-neighbours.

3.2 Properties

Since δ-neighbours are also neighbours under the standard neighbourhood, thus an ϵ-differentially private mechanism under standard neighbourhood is also ϵ-differential private mechanism under δ-neighbourhood. The converse also holds but with a weaker bound, as stated in the following lemma(proof omitted):

Lemma 1 *If a mechanism $\mathcal{A}$ is ϵ-differential private under the δ-neighbourhood and the diameter of $\mathbf{D}$ is d, then it is $(d\epsilon)$-differential private under the standard neighbourhood.*

Sequential composition: The composition of two differentially private mechanisms is also differentially private. It is easy to show that this property also holds under δ-neighbourhood: given a sequence of k mechanisms, $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_k$, where $\mathcal{A}_i$ is ϵ_i-differentially private under δ-neighbourhood, then the mechanism

$$\mathcal{A}^*(D) = \mathcal{A}_1(D, \mathcal{A}_2(D, \ldots))$$

is $(\sum_{i=1}^{k} \epsilon_i)$-differentially private under δ-neighbourhood.

4. SPATIAL DATASETS

The δ-neighbourhood can be naturally defined on spatial points, say $\mathcal{M} = [0,1]^k$ for some $k \geq 1$. The underlying distance function $d(\cdot, \cdot)$ can be the Euclidean distance and the sources can be the boundary of $\mathcal{M}$, which implies that entities enter through the boundary, or simply none, corresponding to the bounded differential privacy. Let us investigate a few scenarios where the proposed notion is meaningful.

4.1 Example 1

Consider a situation where the dataset is constrained, in the sense that not all multisets of entities from $\mathcal{M}$ are in $\mathbf{D}$ (recall that $\mathbf{D}$ is the set of all possible datasets). Let us call the multisets that are not in $\mathbf{D}$ *invalid datasets*. If those invalid datasets are excluded by the restriction on δ-neighbourhood, then essentially the two assurances, either under standard neighbourhood or δ-neighbourhood, are equivalent. For instance, consider a D that contains the locations of a vehicle sampled at periodic intervals, say at time $1, 2, \ldots, n$, and is represented as a set of tuples where each tuple (t, x) indicates that the vehicle is at location x on time t. Suppose D is to be published by a mechanism $\mathcal{A}$ that is ϵ-differentially private under the standard neighbourhood, then for any possible output r, any D, (t, x) and (t, y), we have

$$Pr(\mathcal{A}(D + \{(t,x)\}) = r) \leq exp(\epsilon) Pr(\mathcal{A}(D + \{(t,y)\}) = r).$$

Since D essentially represents a sequence, the two tuples in the above inequality must have the same t.

We can take a step further. Due to speed limit of the vehicles (which is public knowledge), some datasets are invalid. For example, if D_1 is a valid dataset, a location y that is far from x will lead to an invalid dataset. Since the bound is not required to hold for the invalid datasets, thus, with an appropriate metric and a sufficiently large δ, the assurance provided under δ-neighbourhood is equivalent to the assurance provided under the standard neighbourhood. Similar examples are considered by Blocki et al. [1]. They consider social networks where the maximum degree of any node is likely to be bounded by a number k that is much smaller than the network size n. In such situations, an interesting question is whether the utility can be improved by exploiting the constraints. To illustrate, it is not clear how to improve the well-known histogram-based mechanisms on the constrained datasets, since the sensitivity incurred under the standard and δ-neighbourhood is the same.

4.2 Example 2

In this example, we want to publish a dataset D which contains locations of n entities drawn from the domain $\mathcal{M}$. Consider in an extreme case, where an adversary knows the locations of n-1 entities in D (let us denote this D'). The adversary wants to guess whether x is in D, in other words, whether the unknown entity is x. Under the standard neighbourhood, differential privacy guarantees that the published data does not help the adversary in guessing whether x or y is in D, where y is another location the adversary does not know, by bounding the distance of the two probabilities $P(A(D' + \{x\}) \in R)$ and $P(A(D' + \{y\}) \in R)$. Hence, from the perspective of any contributor Bob, if Bob accepts the assumption that there is at least one entity of whom the adversary does not have background information, he is comfortable in contributing his location.

Under δ-neighbourhood, the same guarantee holds when x and y are close-by. From the perspective of Bob, if he accepts the assumption that there is at least one entity y near Bob, say within δ =40km, of whom the adversary has no background knowledge, then differential privacy with δ-neighbourhood suffices for Bob to contribute.

Now let us consider another more resourceful adversary who has more accurate background information on region near Bob. With respect to this background information, the indistinguishable entities similar to Bob can be 40 to 80 km away. In this case, when D is published by an ϵ-differential private mechanism under 40 km-neighbourhood, Bob's privacy is still protected but with a weaker assurance similar to a 2ϵ-differential privacy. Thus, compared to the standard neighbourhood, δ-neighbourhood "redistributes" the assurance by placing more emphasis on close-by entities, with the value of δ determines the rate the assurance decreases over distance. Hence, we can view the δ-neighbourhood not as a relaxation of the standard neighbourhood, but as a redistribution of assurance, where δ is a parameter controlling the rate of redistribution.

5. PUBLISHING SPATIAL DATASET

Although an ϵ-differentially private mechanism under the standard neighbourhood is also ϵ-differentially private under δ-neighbourhood, it may not achieve our intention of investing more budget on nearby entities. In this section, we consider publishing histogram of 2D spatial points to be used for subsequent range queries. Essentially, we want to determine an "optimal" linearly transformed histogram similar to the work by Li et al. [15], but with a different sensitivity function derived from the δ-neighbourhood. We observe that the sensitivity function in 2D leads to an interesting combinatoric structure in the design of the linear transformation, and propose a few constructions. We also note that for 1D spatial points, a known technique under standard neighbourhood [9] can be easily modified to achieve high utility under δ-neighbourhood, as shown in Section 5.5.

5.1 Illustration

Let us demonstrate how to capitalize the notion of δ-neighbourhood with the following simple example in 1D. Consider a dataset containing (possibly with repetitions) 4 possible values: $\{\frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1\}$. Let c_i be the number of points with value $i/4$. Table 1 gives a 1-differentially private mechanism under the standard neighbourhood that publishes the counts (c_1, c_2, c_3, c_4).

Let us compare the case under 0.25-neighbourhood, and with a source of $\{0\}$, i.e. points are added/removed within distance of 0.25 from 0. The mechanism in Table 1 is also 1-differentially private under 0.25-neighbourhood.

Now let us publish the counts as shown in Table 2, where a linear transformation is applied before adding noise. Our main observation is that, the sensitivity of publishing the values (a_1, a_2, a_3, a_4) is 1 with respect to the 0.25-neighbourhood: changing a single entity by a distance of 0.25, or adding an entity within 0.25 to the source affects only one a_i for some i. Hence, a Laplace noise of $Lap(1)$ is sufficient to guarantee 1-differential privacy under 0.25-neighbourhood. However, under the standard neighbourhood, an entity changing from value $\frac{1}{4}$ to 1 will decrease each a_1, a_2, a_3 by 1, leading to a sensitivity of 3.

By publishing the a'_i's in Table 2, we can answer range queries with higher accuracy through linear combination of the a_i's. For example, when a query asks for the frequency counts in the range $[0.4, 0.6]$, reporting the value c'_2 leads to an unbiased estimator with variance 8, which is the variance of $Lap(2)$. On the other hand, from Table 2, it can be estimated by $a'_1 - a'_2$ giving an unbiased estimator with a smaller variance of 4, which is the variance of the sum of two independent Laplace noises, $Lap(1) + Lap(1)$. Such difference is more significant for larger query range. The comparisons are shown in Table 3: row i of the table shows the noise variances when the query range covers exactly i number of the counts c_i's.

Table 1: Publishing c_i's directly.

Actual Values	Published values
c_1	$c'_1 = c_1 + Lap(2)$
c_2	$c'_2 = c_2 + Lap(2)$
c_3	$c'_3 = c_3 + Lap(2)$
c_4	$c'_4 = c_4 + Lap(2)$

Table 2: Publishing a linearly transformed histogram.

Actual values	Published values
$a_1 = c_1 + c_2 + c_3 + c_4$	$a'_1 = a_1 + Lap(1)$
$a_2 = c_2 + c_3 + c_4$	$a'_2 = a_2 + Lap(1)$
$a_3 = c_3 + c_4$	$a'_3 = a_3 + Lap(1)$
$a_4 = c_4$	$a'_4 = a_4 + Lap(1)$

Table 3: Variance of the estimator for different range size.

Number of c_i's covered	number of queries	Derived from Table 1	Derived from Table 2
1	4	8	4
2	3	16	4
3	2	24	4
4	1	32	2

By exhaustive checking, it can be verified that, in terms of minimizing the total variance of all possible range queries, i.e. the weighted sum of the variance in the rightmost column with the weights in the second column in Table 3, the construction in Table 2 is optimal among all linear combinations of c_1, c_2, c_3 and c_4 where the coefficients are binary, i.e. either 0 or 1.

Note that the above methods estimate the query results using linear combinations of the published values. One could enforce the constraints that all c_i's are non-negative, leading to a non-linear estimator. Although this may create bias, it further lowers the variance of the estimation.

5.2 Generalization

Let us generalize the illustrating example. The method shown in Table 1 corresponds to the direct method of adding noise to the frequency counts of an equi-width histogram, whereas Table 2 corresponds to a method that applies a linear transformation before adding noise. Li et al. [15] studied such general form of publishing under the standard neighbourhood. In this section, we extend it to δ-neighbourhood. As illustrated in the example, the key difference of our method is the lower sensitivity incurred under δ-neighbourhood.

Formally, a histogram $\mathcal{H}_B(D)$ for a partition of the domain $B = \{b_1, \ldots, b_k\}$ on D gives a column vector of frequency counts $\mathbf{c} = (c_1, \ldots, c_k)^t$ where $c_i = |D \cap b_i|$. We call each set in the partition B a bin. In particular, an equi-width histogram corresponds to a partition whose bins are of the same size. Since all the bins do not overlap, the effect of replacing an entity in D affects frequency counts in at most two bins, and thus the sensitivity of $\mathcal{H}_B(\cdot)$ is 2 under the standard neighbourhood. Hence the mechanism of publishing $\mathbf{c} + Lap(2/\epsilon)^k$ is ϵ-differential private under the standard neighbourhood.

We consider queries whose answers are linear combination of counts in $\mathbf{c}$, and can be expressed as $\mathbf{qc}$ where $\mathbf{q}$ is a row vector. For example, a range query can be a summation of counts in some bins. For a sequence of m queries, let us express it as an m by k matrix $\mathbf{Q}$ and hence the answer to the queries are the coefficients in $\mathbf{Qc}$. As proposed by Li et al., to answer the query $\mathbf{Q}$, one may employ a *strategy* $\mathbf{A}$, which is represented as a k by n matrix for some n, and publish

$$\widetilde{\mathbf{c}} = \mathbf{Ac} + Lap(\triangle_{\mathbf{A}}/\epsilon)^n,$$

where $\triangle_{\mathbf{A}}$ is the sensitivity of the function that on input D, returns $\mathbf{Ac}$. From the published $\widetilde{\mathbf{c}}$, we want to estimate the query results. It can be shown [21] that the following estimate is unbiased:

$$\mathbf{A}^{+}\widetilde{\mathbf{c}},$$

where $\mathbf{A}^{+} = (\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t$ is the pseudo-inverse of $\mathbf{A}$, and the variance of the estimator is

$$(\triangle_{\mathbf{A}})^2 trace(\mathbf{Q}(\mathbf{A}^t\mathbf{A})^{-1}\mathbf{Q}^t). \tag{5}$$

Now, given $\mathbf{Q}$, we want to find the $\mathbf{A}$ s.t. the variance is minimized. In the illustrating example, $\mathbf{Q}$ is a 10 by 4 matrix where each row corresponds to a range queries, and

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Now let us look at the problem in the context of δ-neighbourhood. The sensitivity of $\mathbf{A}$ under δ-neighbourhood leads to an interesting combinatoric structure that is not present in the standard neighbourhood. Under the standard neighbourhood, the sensitivity of $\mathbf{A}$ is

$$\max_{i,j \in \mathbb{Z}_n} \{ \ \|\mathbf{a}_i - \mathbf{a}_j\|_1 \ \}, \tag{7}$$

where each $\mathbf{a}_i$'s is a column vector in $\mathbf{A}$, that is, $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$. To understand the above expression, note that $\|\mathbf{a}_i - \mathbf{a}_j\|_1$ is the sum of L_1 difference when an entity change between bin i and bin j. Since the sensitivity is the least upper bound on all possible pairs of neighbouring datasets, we have the above expression.

Under the δ-neighbourhood, the sensitivity of $\mathbf{A}$ is

$$\max_{(i,j) \in N} \{ \ \|\mathbf{a}_i - \mathbf{a}_j\|_1 \ \},$$

where N is a set induced from the requirement on δ-neighbourhood,

$$N = \{(i,j) \mid \exists \ x \in b_i, y \in b_j, \ s.t. \ d(x,y) \le \delta\}.$$

Compare to the expression in (7), the maximum is taken over a smaller set N and thus could be smaller.

For the matrix $\mathbf{A}$ in the illustrating example, we have $N = \{(1,2),(2,3),(3,4)\}$ under 0.25-neighbourhood, and thus the sensitivity of $\mathbf{A}$ is 1; whereas the sensitivity under the standard neighbourhood is 3, as $\|\mathbf{a}_1 - \mathbf{a}_4\|_1 = 3$.

Solving the optimization problem in general is difficult for standard neighbourhood, partly due to the fact that the sensitivity $\triangle_{\mathbf{A}}$ as a function of $\mathbf{A}$, is non-differentiable. Likewise it is difficult for δ-neighbourhood. We will show in the next section how we can improve $\mathbf{A}$ for range queries by exploiting the combinatoric structure.

5.3 Proposed Algorithm

The main idea of our proposed algorithm can be illustrated with a graphical representation of the bins when the entries in $\mathbf{A}$ are binary, i.e. either 0 or 1. Let us treat each bin as a vertex in the graph. Hence there are k vertices $v_1, v_2, \ldots, v_k$. There is an edge between two vertices v_i and v_j iff $(i,j) \in N$.

For a matrix $\mathbf{A}$, since the entries are binary, each row corresponds to a subset of bins. Hence, $\mathbf{A}$ can be viewed as a collection of sets $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ where each set in $\mathbf{A}$ is a set of bins. For an edge (v_i, v_j), we say it is being *cut* by a set $\mathbf{a}$ iff

$$(v_i \in \mathbf{a} \wedge v_j \notin \mathbf{a}) \ \vee \ (v_i \notin \mathbf{a} \wedge v_j \in \mathbf{a}).$$

For each edge e, let us call the number of sets in $\mathbf{A}$ that cut the edge e the number of cuts on e, denoted as $C(e)$. Now, the sensitivity of $\mathbf{A}$ is the maximum number of cuts over all edges, i.e. $\max_e C(e)$.

Note that given a particular $\mathbf{A}$, it may be possible to insert a set into $\mathbf{A}$ without increasing its sensitivity. That is, it may be possible to find a subset that only cuts edges that have not been cut by subsets in $\mathbf{A}$. Since sensitivity is not increased, it would not hurt to add this set into $\mathbf{A}$ which in turn publishes this extra information[2]. This observation leads to a simple greedy algorithm that improves a strategy: simply add rows to $\mathbf{A}$ until it is not possible to do so without increasing the sensitivity.

For instance, consider a 2D histogram with bins $\{b_{i,j} \mid i,j \in \mathbb{Z}_n\}$, with neighbourhood $N = \{((i,j),(i',j')) \mid |i-i'| + |j-j'| \le 1\}$ as shown in Figure 1 where each bin is a bullet(blue), and the neighbours are connected by a dotted(red) line. Consider the set $\mathbf{A}$ that contains $\mathbf{a}_{i,j} = \{b_{2i-1,2j-1}, \ b_{2i-1,2j}, \ b_{2i,2j-1}, \ b_{2i,2j}\}$, for $i,j = 1,2,\ldots \frac{n}{2}$, that is, each $a_{i,j}$ is a solid(blue) square that contains four blue vertices. Note that the solid(blue) squares do not "cut" all the neighbouring edges, and therefore, if we add $\mathbf{a}'_{i,j} = \{b_{2i,2j}, \ b_{2i,2j+1}, \ b_{2i+1,2j}, \ b_{2i+1,2j+1}\}$ to $\mathbf{A}$, for $i,j = 1,2,\ldots \frac{n}{2} - 1$, (i.e. the dash(black) squares containing 4 vertices each), the sensitivity remains the same.

On the other hand, for some $\mathbf{A}$, inserting any additional $\mathbf{a}'$ into $\mathbf{A}$ will increase the sensitivity of $\mathbf{A}$. Therefore, the key question lies on whether the noise reduced by the additional $\mathbf{a}'$s is more significant than the noise introduced by the increment of sensitivity. Such comparison is application dependent, i.e. it depends on the queries $\mathbf{Q}$.

For 2D spatial data, we consider random range queries and propose publishing a series of equi-width histograms, similar to the construction illustrated in Figure 1. We consider datasets whose elements are in the normalized domain $[0,1)^2$. Our construction is build on publishing equi-width histograms. An equi-width histogram in 2D corresponds to the partition $B = \{b_{1,1}, b_{1,2}, \ldots, b_{k,k}\}$, where each bin $b_{i,j}$ is a square region $[\frac{i-1}{k}, \frac{i}{k}) \times [\frac{j-1}{k}, \frac{j}{k})$.

[2] One may see this from expression (5), where adding a row to $\mathbf{A}$ without increasing $\triangle_{\mathbf{A}}$ will not increase the variance.

We propose publishing a series of overlapping histograms where each histogram is shifted by an offset δ from the previous histogram in the series. Specifically, let $B_0, B_1 \ldots B_{m-1}$ be a sequence of partitions, where $m = \lceil \frac{1}{k\delta} \rceil$ and B_x is a partition $\{b^x_{1,1}, b^x_{1,2}, \ldots, b^x_{k+1,k+1}\}$ with each $b^x_{i,j}$ is a square region $[\frac{i-2}{k} + x\delta, \frac{i-1}{k} + x\delta) \times [\frac{j-2}{k} + \delta, y + \frac{j-1}{k} + x\delta)$.

Note that the sensitivity of $\mathbf{A}$ constructed in this way is 4, instead of 2 as demonstrated in Figure 1. This is because in 2D spatial data, there are edges connected the vertices $b_{i,j}$ and $b_{i+1,j+1}$. However, we will show in the next section that, when δ is relatively small, the insertion of additional $\mathbf{a}'$ can overcome the increment of the sensitivity.

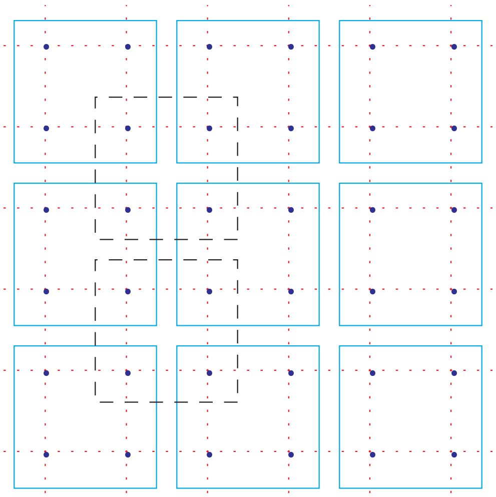


Figure 1: Demonstration of adding a' to A without increasing sensitivity.

5.4 Evaluation

5.4.1 1D range query

The earlier example described in Section 5.1 can be generalized to publish linear transformation of histograms with n bins. The transformation $\mathbf{A}$ is a lower triangular matrix of size $n \times n$ and the entries on and below diagonal are 1. Essentially, row i of $\mathbf{A}$ cumulates the counts for bin 1 to bin i. Let us call this strategy C_n. The answer to a range query that covers bin i to j can be obtained by subtracting the j-th row and $(i-1)$-th row. We are interested in how accurate C_n performs in answering 1D range queries, i.e. in answering the set of all range queries, $\mathbf{Q}$.

Li et al. [15] consider the maximum error and total error of three strategies: H_n which queries a series of equi-width histograms [11], Y_n which is a Haar wavelet transformation matrix [24] and the identity matrix I_n. Figure 2 shows H_4, Y_4, I_4 and C_4. The maximum error refers to the maximum variance among all row vectors of $\mathbf{Q}$, and total error refers to the sum of the variance. The asymptotic bounds on the errors of H_n, Y_n and I_n are as shown in Table 4. The constructions do not exploit δ-neighbourhood, and the errors of H_n, Y_n and I_n are the same under either standard neighbourhood or δ-neighbourhood.

C_n benefits from δ-neighbourhood, in the sense that the sensitivity $\triangle_{C_n}$ is lower for smaller δ. The corresponding maximum error and total error of $C_{n,\delta}$ is also shown in Table 4. When $\delta = n$, it performs similar to identity matrix, but when δ is small, we can reduce the errors by exploiting the δ-neighbours.

$$\underset{H_4}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}} \quad \underset{Y_4}{\begin{bmatrix} 0 & 0 & 1 & \text{-}1 \\ 1 & \text{-}1 & 0 & 0 \\ 1 & 1 & \text{-}1 & \text{-}1 \\ 1 & 1 & 1 & 1 \end{bmatrix}} \quad \underset{I_4}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \quad \underset{C_4}{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

Figure 2: Strategy H_4, Y_4, I_4 and C_4.

Table 4: Max and total errors.

	H_n	Y_n	I_n	$C_{n,\delta}$
max error	$\Theta(\frac{log^3 n}{\epsilon^2})$	$\Theta(\frac{log^3 n}{\epsilon^2})$	$\Theta(\frac{n}{\epsilon^2})$	$\Theta(\frac{\delta}{\epsilon^2})$
total error	$\Theta(\frac{n^2 log^3 n}{\epsilon^2})$	$\Theta(\frac{n^2 log^3 n}{\epsilon^2})$	$\Theta(\frac{n^3}{\epsilon^2})$	$\Theta(\frac{n^2 \delta}{\epsilon^2})$

5.4.2 2D range query

We consider mechanisms that answer 2D range queries with fixed range size in $[0,1)^2$. A 2D range query of size s asks for the number of points in the region $[x - \frac{s}{2}, x + \frac{s}{2}) \times [y - \frac{s}{2}, y + \frac{s}{2})$. We compare the algorithm described in Section 5.3 with the equi-width histogram method as a baseline.

Recall that an equi-width histogram is the partition $B = \{b_{1,1}, b_{1,2}, \ldots, b_{k,k}\}$. Let $\widetilde{c}_x$ be the published frequency counts in bin b_x.

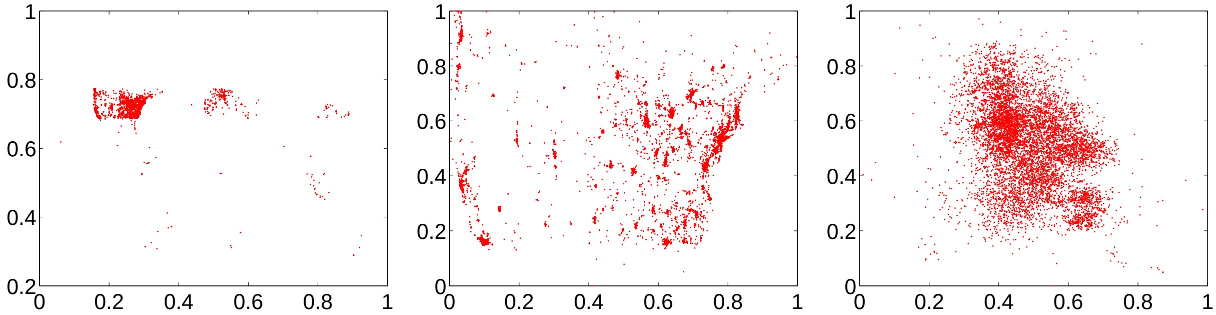
Given a range query q, we estimate the answer to q as:

$$\sum_{b_x \in B} \left(\frac{|b_x \cap q|}{|b_x|} \; c_x \right). \tag{8}$$

where $|b_x|$ is the area of b_x. Note that if the query partially intersects with a bin, that bin contributes proportionally to the answer. Our proposed method answers a range query in a similar way, but using the average of the series of equi-width histograms.

We conduct experiments on three 2D datasets. Dataset 1 contains locations of Twitter users in the world [12]. The dataset contains over 193,841 Twitter users' data from the period of March 2006 to March 2010. Dataset 2 contains the locations of users in dataset 1 cropped at the North American region. It contains locations of 183,072 users. Dataset 3 [13] contains 164,860 tuples collected from tags that continuously record the location information of 5 individuals. The data points are normalized to the space $\mathcal{M} = [0,1]^2$, and Figure 3(a), 3(b) and 3(c) illustrate the distributions of the data points for dataset 1, 2, and 3 respectively. To avoid clogging, only 5% of the points (randomly selected) are plotted for each dataset.

For the first two datasets, we consider two cases where $\delta = 0.001$ and $\delta = 0.0001$, which translate to a bound of approximately 40 and 4 kilometers for dataset 1, and 5 and 0.5 kilometers for dataset 2 respectively. For dataset 3, we consider $\delta = 0.01$ and $\delta = 0.001$, which translate to 500 and 50 meters. We evaluate the construction described in

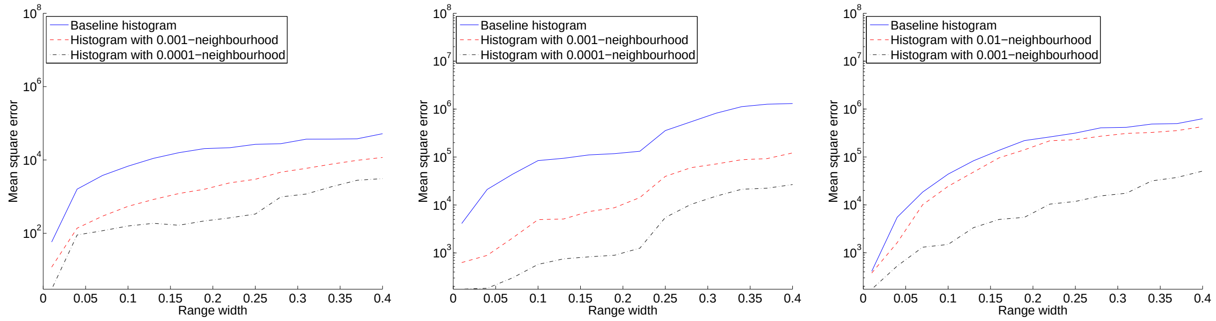


(a) Randomly selected 5% of points from Dataset 1.

(b) Randomly selected 5% of points from Dataset 2.

(c) Randomly selected 5% of points from Dataset 3.

Figure 3: The 2D location datasets.



(a) Dataset 1.

(b) Dataset 2.

(c) Dataset 3.

Figure 4: The mean square error of range queries in linear-logarithmic scale.

Table 5: Query range and corresponding best bin-width for the Dataset 1.

	Query	Mean Square Error			
	range	$k = 0.005$	$k = 0.01$	$k = 0.015$	$k = 0.02$
Dataset 1	0.01	62.01	86.16	94.12	100.69
	0.1	13218	9895.7	6726	8901.6
	0.2	72161	40301	24952	20381
Dataset 2	0.01	4112.7	6166.4	8227.1	13105
	0.1	232689	82449	86034	175203
	0.2	249344	186891	115664	103929

Section 5.3, and compare its accuracy with a baseline histogram method. For comparison purpose, we empirically choose the optimal bin width for the histogram method[3]. That is, for each query range, we computer the mean square error for $k = 0.0025$ to 0.04 with a step of 0.0025, and choose the smallest error among them. Part of the errors are shown in Table 5. Figure 4 shows the details of the experiment result. For example, when the queries is of size 0.1, the mean square error of the baseline method is 6726, where as with the construction under 0.001-neighbourhood is 543, and with 0.0001-neighbourhood is 109. By exploiting δ-neighbourhood, we can achieve significantly higher utility. For example, in dataset 1, the utility improves by a factor of 100 with 4km-neighbourhood, while many existing mechanisms can only improve the result by a factor of 2 [25, 3]. Note that our notion is orthogonal to existing techniques, and thus potentially could be combined to attain higher utility.

5.5 Other Publishing Mechanisms

For some mechanisms, it is easier to apply the notion of δ-neighbourhood. In this Section we analyze their performance under δ-neighbourhood.

5.5.1 *Publishing Sorted 1D Points*

Fang et al. [9] propose a method of publishing 1D histogram by directly publishing the sorted point. The sensitivity of this mechanism depends on the size of the domain, say m.

Under δ-neighbourhood, the sensitivity of the publishing method is reduced to δm and therefore the Laplace noise required to achieve ϵ-differential privacy is reduced from $Lap(m/\epsilon)$ to $Lap(m\delta/\epsilon)$. Thus, there is significant improvement when applying the publishing method as it is. Figure 5 shows the improvement for expected mean square error for range query of 10,000 runs for each range size.

[3] The step of revealing the optimal bin width could reveal more information and thus the whole process may not be differentially private. Nevertheless, it can serve as a baseline to compare the performance of the purposed mechanisms.

Although the error is significantly decreased (the factor of improvement on mean square error is approximately $(1/\delta^2)$ for δ-neighbourhood), it is not clear how to generalize the construction to higher dimensions. The method of using locality preserving transformation as described by Fang et al. [9] would not help since here we are required to preserve locality in the "difficult" direction.

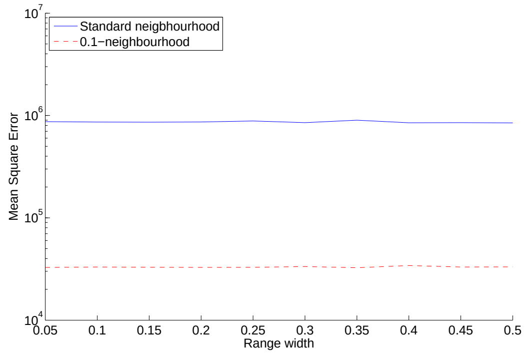


Figure 5: Average error for range query.

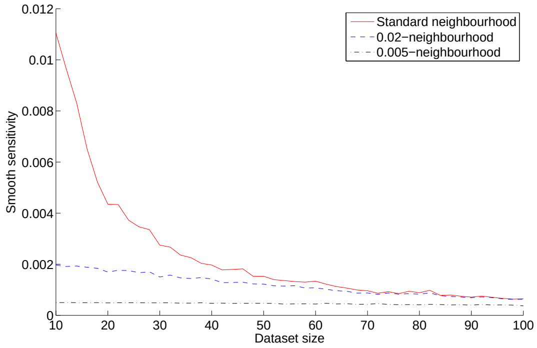


Figure 6: Average error for median.

5.5.2 *Publishing Median*

Publishing median differentially privately is technically challenging. To publish the median of a set of 1D points in $[0, m]$, a noise of $Lap(m/\epsilon)$ is required, although for most database instances, the "local sensitivity" is low, i.e. changing any element in that particular database instance will not significantly change the value of the median. Nissim et al. [18] proposed a method that adds noise proportional to the "smooth sensitivity" (a smooth bound of the local sensitivity) of a database instance. He showed that this mechanism has high accuracy when the smooth sensitivity is low.

The δ-neighbourhood can further reduce the noise requirement when "local sensitivity" can be still large. With δ-neighbourhood, we can reduce the global sensitivity, and thus bound the smooth sensitivity for some worst case scenarios. Figure 6 shows the noise required to publish the median of a synthesized dataset with random 1D points generated under the exponential distribution and then scaled to $[0, 1]$. For each size of the dataset, we repeat the process 300 times and the average smooth sensitivity is recorded under different neighbourhood definitions.

6. DYNAMIC DATASETS

We now investigate dynamic datasets. Consider situations where information on entities are collected periodically over time, say at discrete time $1, 2 \ldots$. Occasionally, statistics are to be published. Intuitively, with limited budget, it is impossible to continuously publish meaningful information indefinitely, in fact, Dwork et al. [7] showed a negative result under a setting that captures this intuition. However, in some scenarios, the entities are not required to contribute at all collection times, and are likely to leave within a short period. With such restriction, it should be now possible to continuously publish with low noise indefinitely, as effect of information contributed earlier would diminish in time.

6.1 Example 1

One situation where publishing dynamic dataset can benefit from δ-neighbourhood is when sensitive information only last for a short period. Consider a regional flu response organization who wants to continuously collect daily information on the health conditions of visitors, and release the information occasionally. Alice wants to infer whether Bob has been to the region based on the released information. If the publishing mechanism $\mathcal{A}$ is ϵ-differential privacy, then Alice's inference is bounded by:

$$Pr(\mathcal{A}(D_0 + \{\mathbf{x}\}) \in R) \leq exp(2\epsilon) Pr(\mathcal{A}(D_0) \in R),$$

where $\mathbf{x}$ is Bob's information. If all visitors must leave within 14 days, then $\mathbf{x}$ must be near the source, i.e. $d(\mathbf{x}, \perp) <$ 14 days, otherwise the dataset is invalid. Hence, under this constraint on the datasets, the guarantee under the standard neighbourhood and δ-neighbourhood are equivalent.

6.2 Example 2

Let us revisit Example 1. Suppose the authority allows some visitors to stay for a longer period, say 28 days, even if the dataset is published under 14-neighbourhood, there is still protection. If Bob indeed stayed for 28 days, the bound is relaxed to $exp(2\epsilon)$. Hence, similar to the spatial datasets, the protection is being redistributed with more protection to entities with shorter stay.

6.3 Formulation

Let a sequence $x_1, x_2, \ldots$ be the data contributed by an entity, where each $x_i \in \mathbf{U} + \{\perp\}$ is the data contributed at time i, with $\mathbf{U}$ being the domain of the contributed data, and $\perp$ being a special symbol indicating that the entity is not contributing at that time. Let us call a sequence containing only the symbol $\perp$ a *null sequence*. A dataset D is a set of the aforementioned sequences. We assume that every entity in D has contributed a data in $\mathbf{U}$ at some time, and thus D does not contain null sequence. The prefix of a sequence x contains data contributed by the entity up to time n, denoted $x_{[1..n]}$, where n is the length of the prefix. Let us denote $D_{[1..n]}$ the set of such prefixes in D that are not null sequence. In addition, denote D_n the set of all n-th elements of the sequences in D that is not $\perp$, that is, D_n contains all data contributed at time n.

At certain time, say time t, some information on D_t is to be published. We assume that information is published at any time i, and let $\mathcal{A}_i$ be the publishing mechanism employed at time i. Hence, the data published are $\mathcal{A}_1(D_1)$, $\mathcal{A}_2(D_2), \ldots$. Combining all the data published before time $n+1$, we can treat the "effects" of mechanisms $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n$ as a single mechanism $\mathcal{A}^*_n$ that operates on $D_{[1..n]}$.

6.4 δ-Neighbour on Dynamic Dataset

Given two datasets, D and D', under the standard neighbourhood, they are neighbours if, and only if they differ by one entity. That is, there is a sequence x and y s.t. $D + \{x\} = D'$, or $D + \{x\} = D' + \{y\}$. This is essentially the same notion of neighbourhood for *user-level privacy* studied by Dwork et al. [7][8].

For two sequences $\mathbf{x} = \langle x_1, x_2, \ldots\rangle$ and $\mathbf{y} = \langle y_1, y_2, \ldots\rangle$, let us define $d(\mathbf{x}, \mathbf{y})$ to be the value $i_s - i_t$ where i_s is the smallest index s.t. $x_{i_s} \neq y_{i_s}$ and i_t is the largest index s.t. $x_{i_t} \neq y_{i_t}$. That is, it is the length of the smallest consecutive subsequence that contains all the differences. We take the null sequence as the source. Hence, D and D' are δ-neighbourhood if, and only if $D + \{\mathbf{x}\} = D'$, or $D + \{\mathbf{y}\} = D' + \{\mathbf{z}\}$, for some $\mathbf{y}, \mathbf{z}$ s.t. $d(\mathbf{y}, \mathbf{z}) \leq \delta$, or some $\mathbf{x}$ s.t. $d(\mathbf{x}, \hat{\perp}) \leq \delta$ where $\hat{\perp}$ denotes the null sequence. When $\delta = 1$, then providing differential privacy under δ-neighbourhood is same as the *event-level privacy* studied by Dwork et al. [7].

6.5 Sustainable Differential Privacy

If each mechanism $\mathcal{A}_i$ is ϵ-differentially private under either notions of neighbourhood, then the mechanism $\mathcal{A}^*_n$ is $(n\epsilon)$-differentially private under the respective neighbourhood. However, for δ-neighbourhood, we should be able to "reuse" the budget spent on much earlier published data. This observation is formulated in the following theorem:

Theorem 2 *Let D be a dynamic dataset with the mechanism $\mathcal{A}^*_n$, $\mathcal{A}_1$, $\mathcal{A}_2$, $\ldots \mathcal{A}_n$ as defined above in Section 6.3. If mechanism $\mathcal{A}_i$ is ϵ_i-differentially private under the standard neighbourhood for each $i \in \{1, \ldots, n\}$, and*

$$\sum_{i=1}^{\delta} \epsilon_{k+i} \leq \epsilon, \quad \text{for } k \in \{0, 1, \ldots, (n-\delta)\},$$

then $\mathcal{A}^*_n$ is ϵ-differentially private under δ-neighbourhood.

Proof. Consider two datasets D and D', where $D' + \{\mathbf{y}\} = D + \{\mathbf{x}\}$ and $d(\mathbf{x}, \mathbf{y}) \leq \delta$. Let i_s be the smallest index at which $\mathbf{x}$ and $\mathbf{y}$ differ. Consider an output $\mathbf{a} = \langle a_1, a_2 \ldots a_n\rangle$ of $\mathcal{A}^*_n(D)$, we have the probability that $\mathcal{A}^*_n$ gives the same output on dataset D' as:

$$\begin{aligned} Pr(\mathcal{A}^*_n(D') = r) &= \prod_{i=1}^{n} Pr(\mathcal{A}_i(D'_i) = a_i) \\ &\leq \left(\prod_{i=i_s}^{i_s+\delta-1} exp(\epsilon_i)\right) \cdot \left(\prod_{i=1}^{n} Pr(\mathcal{A}_i(D_i) = a_i)\right) \\ &\leq exp(\epsilon) Pr(\mathcal{A}^*_n(D) = r) \end{aligned}$$

Similarly argument holds for any pair D and D' where $D' = D + \{x\}$ and x is near the source. Therefore, $\mathcal{A}^*_n$ is ϵ-differentially private under δ-neighbourhood. □

For instance, if $\epsilon_i = \delta^{-1}$ for all i, under standard neighbourhood, $\mathcal{A}^*_n$ is a n/δ-differentially private, but it is a 1-differentially private mechanism under δ-neighbourhood. Note that the assurance is independent of n, and thus it is possible to continue publishing indefinitely and yet achieve ϵ-differential privacy. In general, we say that a mechanism achieves *sustainable differential privacy* when the factor in the assurance is bounded by a constant independent of n.

7. PUBLISHING DYNAMIC DATASET: ALLOCATING BUDGET

The privacy requirement ϵ is often called the privacy budget as it can be divided between and allocated to a group of mechanisms. As shown in section 6.5, sustainable ϵ-differential privacy can be achieved by ensuring budget spent in any sliding window is bounded by ϵ (Theorem 2). There are many ways to allocate the budget over the time window and yet achieving sustainable privacy. An interesting question is on how to allocate the budget ϵ_i to the mechanism A_i at each time i, so as to minimize the "total error".

We consider total error of the form $\sum_1^n (w_i \mathcal{E}rr_i(\epsilon_i))$, where $\mathcal{E}rr(\cdot)$ is a non-negative function quantifying the error incurred by the mechanism A_i in term of the budget, and the non-negative weight w_i gives the weightage of the query at time i. A zero weight at time i, i.e. $w_i = 0$, corresponds to the event that no publishing is required at time i. Now, given a weightage $w = \langle w_1, \ldots, w_n\rangle$ and the privacy requirement, we want to find an allocation of the budget ϵ_i so as to minimize the total error.

We consider two settings. Under the *offline* setting, the publisher knows all the weights at time 0, and hence the publisher can determine the allocation before publishing. This setting could be unrealistic in scenarios where the publisher does not know the queries in advance. Under the *online* setting, the value of w_i is only known at time i and the budget ϵ_i has to be committed before time $t+1$.

7.1 Offline Allocation

The offline budget allocation problem can be formulated as the following optimization problem:

Problem 1 Offline Budget Allocation

Given: $\delta \in \mathbb{Z}_n, \epsilon, \mathbf{w} = \langle w_1 \ldots w_n\rangle \in \mathbb{R}^n_{\geq 0}$

Find: $\langle \epsilon_1, \epsilon_2, \ldots, \epsilon_n\rangle$

Minimize: $\displaystyle\sum_{i=1}^{n} w_i \mathcal{E}rr_i(\epsilon_i)$

Subject to: $\displaystyle\sum_{i=1}^{\delta} \epsilon_{k+i} \leq \epsilon$, for $k = 1, 2, \ldots, (n-\delta)$.

In general, solving the above optimization problem is difficult. However, when the objective function is quadratic, it is a convex optimization problem whose solution can be found using existing optimization solvers, for example, a SDPT3 solver [22][23]. In this section, we study error function of the form $\mathcal{E}rr_i(\epsilon) = c_i\epsilon^{-2}$ for some constant c_i. This form of error function corresponds to mechanisms such as the Laplace mechanism, whereby the variance of the error is a quadratic function w.r.t. ϵ_i^{-1}. Since the constant c_i can be captured by the weight vector $\mathbf{w}$, Without loss of generality, we assume $c_i = 1$ for all i.

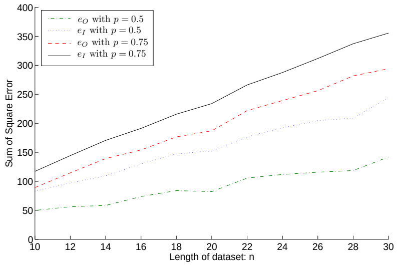


Figure 7: Improvement of offline version for $\delta = 4$.

Let $\mathbf{e}_I = \langle \frac{\epsilon}{\delta}, \ldots, \frac{\epsilon}{\delta} \rangle$, which corresponds to an allocation that divides the budget equally across time; and let $\mathbf{e}_O$ to be the optimal allocation. Note that $\mathbf{e}_I$ is in the feasible region of the problem and could be a good initial solution for a solver.

Figure 7 shows the comparison of errors between $\mathbf{e}_I$ and the optimal budget allocation $\mathbf{e}_O$, where $\mathbf{w}$ is a binary vector and each $w_i \in \{0, 1\}$ is independently randomly chosen to be 1 with probability $p = 0.5$ and $p = 0.75$, respectively.

7.2 Online Allocation

Under online setting, only $w_1, \ldots, w_i$ are available at time i, and the allocating algorithm has to commit the budget for e_i which may later turn out to be sub-optimal. Indeed, it is easy to construct a counter example to show that for any deterministic algorithm, in the worst case, there is an instance where the error incurred is twice as large than the offline optimal. In this section, we focus on average case performance where the $\mathbf{w}$ is drawn from some distribution known to the publisher.

We propose an online algorithm as follow. At time i, given the committed budget allocation $e_1, \ldots, e_{i-1}$ and $w_1, \ldots, w_i$, the following steps are carried out:

1. N (in our experiment, N =1,000) samples of weights $\mathbf{w}_1, \ldots, \mathbf{w}_N$ are drawn from the distribution on condition that the first i values are $w_1, \ldots, w_i$.

2. For each candidate of e_i (in our experiment, we try $0.01, 0.02, \ldots, 1$) and each $\mathbf{w}$ among the N weight samples, compute the "optimal" error by solving the constrained offline allocation problem given below (Problem 2). After the errors by the N samples are obtained, the average error is computed.

3. The candidate that attains the smallest average error is committed to be the budget of e_i.

7.3 Evaluations

We evaluate the performance of the online algorithm, comparing to the offline optimal solution and $\mathbf{e}_I$. We consider $\epsilon = 1$, and $\delta = 4$ or 7. For each setting, we repeat the experiment for 1,000 times and record the average error of the three solutions.

We consider a $\mathbf{w}$ where each $w_i \in \{0, 1\}$ is taken to be 1 with probability $p = 0.5$. Figure 8(a) shows the errors of $\mathbf{e}_O$, $\mathbf{e}_X$ and $\mathbf{e}_I$ for $\delta = 4$, and Figure 8(b) shows errors when $\delta = 7$. Figure 8(c) consider a $\mathbf{w}$ where each $w_i \in \{0, 1\}$ is taken to be 1 with probability $p = 0.75$.

Problem 2 Constrained Offline Allocation

Given:	$\delta \in \mathbb{Z}_n, \epsilon, \mathbf{e}' = \langle \epsilon'_1, \epsilon'_2, \ldots, \epsilon'_m \rangle \in \mathbb{R}^m_{\geq 0}$, $\mathbf{w} = \langle w_1 \ldots w_n \rangle \in \mathbb{R}^n_{\geq 0}$
Find:	$\langle \epsilon_1, \epsilon_2, \ldots, \epsilon_n \rangle$
Minimize:	$\displaystyle\sum_{i=1}^{n} w_i \mathcal{E}rr_i(\epsilon_i)$
Subject to:	$\displaystyle\sum_{i=1}^{\delta} \epsilon_{k+i} \leq \epsilon, \text{ for } k = 1, 2, \ldots, (n - \delta);$ $\epsilon_k = \epsilon'_k, \text{ for } k = 1, 2, \ldots, m.$

8. RELATED WORK

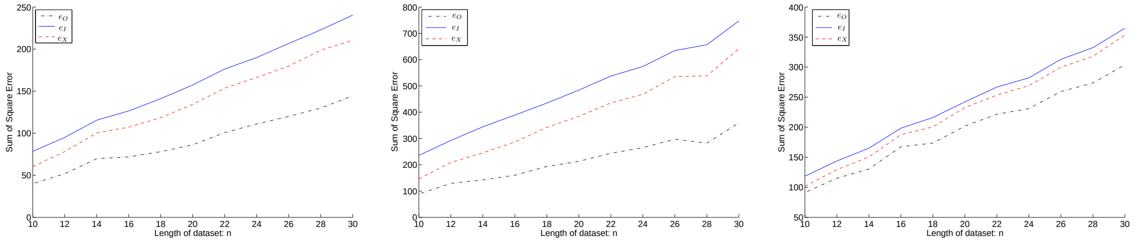
There are extensive works on privacy-preserving data publishing. The recent survey by Fung et al. [10] gives a comprehensive overview on various notions, for example, k-anonymity [20], ℓ-diversity [16], and differential privacy [4].

In practice, ϵ-differential privacy can be too strong to be achieved in some scenarios. Many relaxations capture alternative notions of "indistinguishability", in particular, on how the two conditional probabilities in the bound are compared. For example, (ϵ, δ)-differential privacy [5] relaxes the bound with an additive factor δ, and (ϵ, τ)-probabilistic differential privacy [17] allows the bound to be violated with a probability τ. Similar to our work, Konstantinos et al.[2] proposed broadening the differential privacy definition by considering different underlying metrics.

Alternative relaxations include attribute differential privacy and bit differential privacy considered by Kifer et al. [14], where two datasets are neighbours if they differ at only one attribute value or one bit. Blocki et al. [1] consider differentially private graph algorithms, with restriction that the maximum degree of any node in a social network graph is bounded. They consider the restricted datasets and show that such restricted sensitivity can be significantly lower than the smooth sensitivity for subgraph counting queries and local profile queries.

There are many mechanisms designed for histogram publishing. Xiao et al. [24] proposed a mechanism of adding Laplace noise to the coefficients of a wavelet transformation of an equi-width histogram, whereby range query can be answer with different combination of the published transformation. Hay et al. [11] proposed a method that a series of equi-width histograms for different bin-widths is to be published, and a range query can then be decomposed and answered from the histograms series different scales. Li et al. [15] gave an analysis on linear transformations to answer to a query workload. Machanavajjhala et al. [17] proposed a 2D dataset publishing method that can handle the sparse data in 2D equi-width histogram. However, it is not clear how to adapt the above-mentioned mechanisms to exploit δ-neighbourhood. One exception is the method by Fang et al. [9] as demonstrated in Section 5.5.

Dwork et al. [7] consider applications that involve repeated computations on dynamic datasets, such as monitoring data or searching thread. They gave a general transformation that converts mechanisms on static dataset to mechanisms under dynamic dataset. The idea of processing dynamic datasets also lead to a concept of pan-privacy [8], which



(a) $\delta = 4$, $p = 0.5$.

(b) $\delta = 7$, $p = 0.5$.

(c) $\delta = 4$, $p = 0.75$.

Figure 8: Comparison of offline and online algorithms.

require each datum to be discarded immediately after processing, and therefore guarantee that the internal state be differentially private as well.

9. CONCLUSION

In this paper, we propose to relax differential privacy by adopting an alternative definition of neighbourhood which "redistributes" the assurances based on the underlying distance of the entities. Although the idea is simple, for some applications, it is not clear how to exploit the relaxation to achieve higher utility. We consider two types of datasets, spatial datasets and dynamic datasets, and show that the noise level can be further reduced by constructions that exploit the δ-neighbourhood. We give a few scenarios where δ-neighbourhood would be more appropriate, and we believe the notion provides a good trade-off for better utility.

10. REFERENCES

[1] J. Blocki, A. Blum, A. Datta, and O. Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 87–96, 2013.

[2] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies*, pages 82–102, 2013.

[3] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu. Differentially private spatial decompositions. In *ICDE*, pages 20–31, 2012.

[4] C. Dwork. Differential privacy. *Automata, languages and programming*, pages 1–12, 2006.

[5] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. *Advances in Cryptology-EUROCRYPT*, pages 486–503, 2006.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, pages 265–284, 2006.

[7] C. Dwork, M. Naor, T. Pitassi, and G. Rothblum. Differential privacy under continual observation. *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 715–724, 2010.

[8] C. Dwork, M. Naor, T. Pitassi, G. Rothblum, and S. Yekhanin. Pan-private streaming algorithms. *In Proceedings of ICS*, 2010.

[9] C. Fang and E. C. Chang. Adaptive differentially private histogram of low-dimensional data. *Privacy Enhancing Technologies*, pages 160–179, 2012.

[10] B. Fung, K. Wang, R. Chen, and P. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys*, pages 14–57, 2010.

[11] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *VLDB Endowment*, pages 1021–1032, 2010.

[12] Infochimps. Twitter census: Twitter users by location [online]. http://www.infochimps.com/datasets/twitter-census-twitter-users-by-location.

[13] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, and M. Gams. An agent-based approach to care in independent living. *Ambient Intelligence*, pages 177–186, 2010.

[14] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. *Proceedings of the 2011 international conference on Management of data*, pages 193–204, 2011.

[15] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor. Optimizing linear counting queries under differential privacy. *ACM symposium on Principles of database systems of data*, pages 123–134, 2010.

[16] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *International Conference on Data Engineering*, pages 24–36, 2006.

[17] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. *International Conference on Data Engineering*, pages 277–286, 2008.

[18] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. *ACM Symposium on Theory of Computing*, pages 75–84, 2007.

[19] A. Roth and T. Roughgarden. Interactive privacy via the median mechanism. *ACM Symposium on Theory of Computing*, pages 765–774, 2010.

[20] P. Samarati. Protecting respondents identities in microdata release. *Knowledge and Data Engineering*, pages 1010–1027, 2001.

[21] S. Silvey. *Statistical inference*, volume 7. Chapman & Hall/CRC, 1975.

[22] K. Toh, M. Todd, and R. Tütüncü. Sdpt3ąła matlab software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, pages 545–581, 1999.

[23] R. Tütüncü, K. Toh, and M. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical programming*, pages 189–217, 2003.

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

[25] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially private histogram publication. *International Conference on Data Engineering*, pages 32–43, 2012.