Hiding Distinguished Ones into Crowd: Privacy-Preserving Publishing Data with Outliers

Hui (Wendy) Wang, Ruilin Liu
Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ, USA
hwang@cs.stevens.edu, rliu3@stevens.edu

ABSTRACT

Publishing microdata raises concerns of individual privacy. When there exist outlier records in the microdata, the distinguishability of the outliers enables their privacy to be easier to be compromised than that of regular ones. However, none of the existing anonymization techniques can provide sufficient protection to the privacy of the outliers. In this paper, we study the problem of anonymizing the microdata that contains outliers. We define the *distinguishability-based* attack by which the adversary can infer the existence of outliers as well as their private information from the anonymized microdata. To defend against the distinguishability-based attack, we define the *plain k-anonymity* as the privacy principle. Based on the definition, we categorize the outliers into two types, the ones that cannot be hidden by any plain k-anonymous group (called *global* outliers) and the ones that can (called *local* outliers). We propose the algorithm to efficiently anonymize local outliers with low information loss. Our experiments demonstrate the efficiency and effectiveness of our approach.

		Quasi-identifiers			Sensitive
tuple ID	Name	Age	Gender	Zipcode	Income
1	Alice	20	F	06006	20K
2	Bob	20	M	06011	25K
3	**Justin**	20	M	06013	**120K**
4	Carol	30	F	06001	30K
5	Allan	30	M	06010	50K
6	**Bill**	30	M	06022	**2 Billion**
7	Ben	40	M	06004	100K
8	Susan	40	F	06002	110K
9	David	40	M	06003	130K

Figure 1: An Example of Microdata; Justin and Bill are two outliers.

1. INTRODUCTION

Recent years have witnessed increasing volume of released microdata (i.e., data in raw, non-aggregated format). The release of microdata offers significant advantages in terms of information availability, which make it particularly suitable for ad hoc analysis in a variety of domains such as public health and population studies. However, the release of microdata raises concerns of revealing private information of individuals.

There are two kinds of privacy that must be protected: *presence privacy*, which is the fact that the record of a specific individual is present in the released microdata [19], and *association privacy*, which is the association between the individual and his/her sensitive values. Simply removing explicit identifiers, e.g., name and SSN, has been shown to be insufficient to protect either kind of privacy [25]. The existence of *quasi-identifiers* (QI) attributes, e.g., combination of zipcode, gender and date of birth, that can uniquely identify individuals, enables to reveal the identity of individuals when the released dataset is joined with external public datasets (e.g., voting registration list). This is called *record linkage attack* [25, 23].

Various techniques have been proposed recently to defend against the record linkage attack. Generalization [25, 23] is a popular methodology. The idea is that the QI values are generalized in the way that there are at least k individuals of the same QI values [23, 24, 25]. An anonymized microdata table is considered sufficiently protected if it conforms the defined privacy principle. The existing privacy principles include k-anonymity [25, 23], l-diversity [17], t-closeness [16], (α, k)-anonymity [26], (c, k)-safety [18], etc.. Most of them assume that there is no outlier in the microdata.

1.1 Motivation: Impact of Outliers to Privacy

It is possible that the microdata contains outliers, i.e., individuals that do not comply with the general pattern of the population. For example, the microdata in Figure 1 contains two outliers: *Bill*, whose income is 2 billion, and *Justin*, who is a young pop star with income 120k. In fact, as [4] showed, many publicly released real datasets do contain outliers [1]. Assume all records are free of input error. Then these outliers represent the distinguished individuals in our society. In practice, the adversary may have better knowledge of these distinguished people than regular ones (e.g., we know more about Bill Gates' wealth than our neighbors'). Indeed, even with the simplest knowledge as whether a specific in-

EDBT 2009, March 24–26, 2009, Saint Petersburg, Russia.
Copyright 2009 ACM 978-1-60558-422-5/09/0003 ...$5.00

[1] A list of outliers in US Census data is available on http://www.isle.org/~sbay/papers/kdd03/

Age	Gender	Zipcode	Income
20	*	[06006, 06013]	20K
20	*	[06006, 06013]	25K
20	*	[06006, 06013]	**120K**
30	*	[06001, 06022]	30K
30	*	[06001, 06022]	50K
30	*	[06001, 06022]	2 Billion
40	*	[06002, 06004]	100K
40	*	[06002, 06004]	110K
40	*	[06002, 06004]	130K

(a) A bad anonymization scheme that shows requiring *lowerbound* of income range cannot hide the outlier Justin.

Age	Gender	Zipcode	Income
[20, 30]	*	[06001, 06011]	20K
[20, 30]	*	[06001, 06011]	25K
[20, 30]	*	[06001, 06011]	30K
[20, 30]	*	[06001, 06011]	50K
[20, 40]	*	[06002, 06013]	100K
[20, 40]	*	[06002, 06013]	110K
[20, 40]	*	[06002, 06013]	**120K**
[20, 40]	*	[06002, 06013]	130K

(b) A bad anonymization scheme that shows requiring *upperbound* of income range cannot hide the outlier Justin.
Bill's record is removed.

Figure 2: Examples of Bad Anonymization.

dividual is an outlier or not, the adversary may be able to re-identify him/her from the k-anonymous datasets. For example, in Figure 2 (a), from the first group that contains income [20K, 120K] for people of age 20, the adversary can easily infer that there must exist a young person whose income is unusually high compared with other young people, i.e., there must exist an outlier. If he/she knows that Justin is an outlier and he is the only candidate whose information matches the quasi-identifier values of this group (i.e., age=20, male, Zipcode∈[06006, 06013]), he/she can explicitly re-identify Justin with his income $120k$. This example shows that with the outlierness as a bit of additional external knowledge, both the presence and association privacy of the outliers are easier to be attacked compared with the regulars. Unfortunately simply removing these outliers may cause high information loss (as shown in Section 8). Thus it is necessary to protect the peculiarity of outliers so that they can be safely hidden in the crowd. Unfortunately, none of the existing privacy-preserving data publishing work ever considered outliers and the possible threats to their privacy due to their distinguishability.

In this paper, we consider numerical sensitive values (e.g., income). We assume the "abnormality" of the outliers involves these sensitive values. Recently some privacy preserving work has been done on numerical sensitive values (e.g., [30], [15]). Most of them define a *lowerbound* of the range size for the sensitive values in the same group (i.e., the group that contains the tuples of identical quasi-identifier values), so that the adversary cannot conclude with high probability that the sensitive values fall into a small interval. However, requiring the lowerbound of the range of the sensitive values alone is not sufficient to protect the privacy of outliers. For instance, in Figure 2 (a), the range of income for people of age 20 is [20K, 120K], which definitely conforms the "large range" requirement by both [30] and [15]. However, we have shown above that the adversary still can disclose Justin's record. Indeed, since outliers are far from the regulars, anonymizing outliers together with normal tuples will likely result in the sensitive values in the same groups are of the range size that is *too large*, which may catch the adversary's attention.

A seemingly straightforward approach to hide the distinguishability of outliers is to further define an *upperbound* of the range size for the sensitive values in the same group.

Age	Gender	Zipcode	Income
[20, 40]	*	[06002, 06010]	20K
[20, 40]	*	[06002, 06010]	50K
[20, 40]	*	[06002, 06010]	100K
[20, 40]	*	[06002, 06010]	110K
[20, 40]	*	[06001, 06013]	25K
[20, 40]	*	[06001, 06013]	30K
[20, 40]	*	[06001, 06013]	**120K**
[20, 40]	*	[06001, 06013]	130K

Figure 3: A good anonymization scheme that hides Justin's record; Bill's record is removed.

For instance, with the requirement that the range size of the sensitive values in the same group must be in the range [30k, 60K], instead of grouping with people of similar age (shown in Figure 2 (a)), Justin's record is grouped together with those of similar income (Figure 2 (b)[2]). The income range [100K, 130K] is of size $30K$, which satisfies the size requirement. However, with the age range [20, 40] for this income range, the scheme still reveals the fact that there must exist a young outlier who has abnormally higher income (at least 100K) than the other young people. Therefore, the "goodness" of the anonymization cannot be achieved by simply controlling the lowerbound/upperbound of the range size of the sensitive values in the same group.

Both anonymization schemes in Figure 2 (a) and Figure 2 (b) fail to protect the privacy of Justin because they reveal some "abnormality". Such abnormality of the anonymization groups is mostly likely caused by the presence of outliers in the groups. Based on this, from the anonymized groups that bear irregularity, the adversary may be able to infer the existence of outliers in the anonymized groups, and further disclose the privacy of outliers. Therefore, a good anonymization scheme that can hide outliers must be the one that behaves "normally". Figure 3 shows such an example. The group that matches Justin's quasi-identifier values does not infer that there must exist a young person who has abnormally high income than the other peers, thus the outlier Justin's record is successfully hidden.

[2] Bill's record is removed from Figure 2 (b) since any group that includes it will fail the range size constraint

1.2 Contributions

In this paper, we study the problem of how to publish the microdata that contains the outliers so that the privacy of the outliers are adequately protected. We have the following contributions.

First, to the best of our knowledge, we are the first to study privacy-preserving publishing of data that contains outliers. We define the *distinguishability-based* attack, by which the adversary can identify the outliers as well as their private information as long as he/she can confirm the presence of outliers in the original microdata. We formally study how the adversary can infer the existence of outliers in the original microdata from the released anonymized dataset.

Second, to defend against the distinguishability-based attack, we propose a robust privacy criteria, *plain k-anonymity*. Besides requiring every individual tuple is included in a QI-group that contains at least k distinct sensitive values, plain k-anonymity further requires that by applying the distinguishability-based attack, there is no privacy leakage of either the presence of the outliers in the original microdata or their associated sensitive values.

Third, we categorize outliers into two types, namely *global* outliers that cannot be hidden in any plain k-anonymous group, and *local* outliers that can. To apply the appropriate anonymization actions on these two types of outliers, we characterize them and discuss how to efficiently distinguish them.

Fourth, we design an efficient algorithm to construct plain, k-anonymous QI-groups that effectively anonymize the local outliers. Our anonymization algorithm efficiently builds the anonymization scheme without expensive pre-checking of outliers.

Last but not least, we demonstrate the efficacy of our approach with an extensive set of experiments. Our experimental results show that our approach can efficiently anonymize the microdata that contains outliers with low information loss.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 introduces the preliminaries including anonymization framework. Section 4 defines the distinguishability-based attack on outliers. To defend against the distinguishability-based attack, Section 5 defines the privacy principle, namely plain k-anonymity. Section 6 studies both global and local outliers. Section 7 proposes the algorithm that efficiently construct plain, k-anonymity anonymization scheme. Section 8 presents the experimental results. Section 9 concludes the paper and discusses the future work.

2. RELATED WORK

Privacy-preserving data publishing has received considerable attention in recent years. K-anonymity is the first anonymization principle in the literature [23, 24, 25]. It requires that in the published data, every combination of quasi-identifiers can be indistinctly matched to at least k tuples. However, it may result in that all tuples possess exactly the same sensitive value. To address this defect, l-diversity [17] is proposed recently. It ensures that every QI-group contains at least l "well-represented" sensitive values. Other variants of k-anonymity, e.g., (α, k)-anonymity[26], (c, k)-safety[18], etc., are defined to address different privacy requirements. Most of the above work focuses on categorical sensitive values. None of them assume the existences of outliers in the microdata.

Recently some attention has been shifted to numerical sensitive values. Zhang et al. [30] discussed the information leakage when the numerical sensitive values fall into a narrow range. For example, salaries in the range [20k, 21k] enables the adversary to estimate the salary of attacked target with a high probability. To defend against the attack, it proposed (k, e)-anonymous model, which requires that each QI-group must have at least k different sensitive values, while the range of the group must be at least e. Li et al. [15] pointed out the possibility of the *proximity breach*, i.e., the adversary can conclude with high confidence that some sensitive value must fall in a small interval. They proposed (ϵ, m)-anonymity to defend against proximity breach. Specifically, (ϵ, m)-anonymity requires that for every sensitive value x, at most $1/m$ of the tuples in its QI-group can have sensitive values "similar" to x, i.e., its difference from x should be no greater than ϵ. Both techniques focus on the *lowerbound* of the range of all sensitive values in the same QI-group and require it should be *large enough* so that the adversary cannot conclude with high probability that the sensitive values fall into a small interval. However, the ranges that are too large may still incur privacy breach when outliers exist in the microdata. Section 1 has given an example. Therefore, (k, e)-anonymous and (ϵ, m)-anonymity models cannot provide sufficient protection to the privacy of outliers. Besides the above work, Li et al. proposed *t-closeness*, which required that the distribution of the sensitive values in the released microdata should be close to that of the original [16]. However, if the outliers exist in the original microdata, to satisfy t-closeness, they must be outstanding in the distribution of both the original and released microdata. As the result, the outliers can be easily identified by investigating the distribution alone. Therefore, t-closeness fails to defend against the attack on outliers as well.

3. PRELIMINARIES

In this section, we introduce the preliminaries.

3.1 Distance-based Outliers

There are a few definitions of outliers in the literature, for example, distance-based outliers [3, 20, 22] and density-based outliers [8]. In this paper, we consider distance-based outliers as defined in [20].

Definition 3.1. **[Distance-based Outlier]** *[20]* A tuple $o \in D$ is a *(p, d)-distance outlier* if at least $p\%$ of the tuples in D lie at a distance greater than d from o.

We consider *Euclidean distance* between tuples. To avoid scaling problems, all distances are standardized. In particular, we standardized values x of attribute $A = (x - min(A))/(max(A) - min(A))$.

In general, the outlierness can lie on a single attribute (e.g., income) or multiple attributes (e.g., age together with income). Since we assume the outlierness always involve sensitive values, for outliers on single attribute, they are outstanding on the sensitive attribute. We call these outliers *sensitive-attribute outliers.* Specifically,

Definition 3.2. **[Sensitive-attribute Outliers and Multi-attribute Outliers]** An outlier tuple o is a *sensitive-attribute* (p, d)*-outlier* if there exists at least $p\%$ tuples T

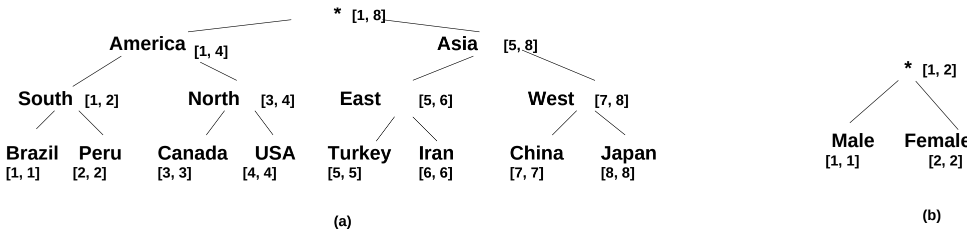


Figure 4: Examples: (a) Taxonomy Tree of *Country*, (b) Taxonomy Tree of *Gender*

such that $\forall t \in T$, $\mid t[S] - o[S] \mid \geq d$, where S is the sensitive attribute. Otherwise, o is a *multi-attribute outlier*.

Bill's record in Figure 1 is an example of sensitive-attribute outlier due to its extremely high income 2 billion, while Justin's record in Figure 1 is an example of multi-attribute outlier; It is not outstanding on the income value alone, but very notable if Justin's age is also taken into consideration.

3.2 Anonymization Framework

Let D be a microdata table that stores private information of a set of individuals. There are three types of attributes in D: identifiers (ID), quasi-identifiers $\mathcal{QI}$ $\{QI_1, \ldots, QI_m\}$ and sensitive attributes $\mathcal{S}$ $\{S_1, \ldots, S_n\}$. For simplicity, we assume there is only one sensitive attribute S in D, and focus on numeric sensitive attributes. Our techniques can be easily adapted to multiple sensitive attributes and categorical sensitive attributes. For each tuple $t \in D$, we use $t[QI_i]$ $(1 \leq i \leq m)$ and $t[S]$ to denote the value of the i-th QI-attribute and the sensitive attribute of t.

We first have the following definition of QI-groups adopted from [28].

Definition 3.3. **[QI-group]** *[28]* Given a microdata D, a partition consists of several subsets of D, such that each tuple in D belongs to exactly one subset. Further, all tuples in the same subset have identical (generalized) QI-values. We refer to these subsets as *QI-groups*, and denote them as $G_1, \ldots, G_m$. Namely, $\cup_{j=1}^{m} G_j = D$, and for any $1 \leq i \neq j \leq m$, $G_i \cap G_j = \oslash$.

For the categorical attributes, we assume that every domain has semantic relationships among the values. Such relationships can be be easily captured by a *taxonomy* tree. For example, Figure 4 (a) illustrates a natural taxonomy tree for the *Country* attribute. For the attributes that do not have any semantic relationship, for example, the attribute *Gender*, their values still can be classified under a common value in a taxonomy tree (Figure 4 (b)). We assign all the leaf nodes in the taxonomy tree with an integer as their topological order. Based on this order, every node in the taxonomy tree, including leaf nodes and non-leaf nodes, can be represented as an interval. We have:

Definition 3.4. **[Intervals in Taxonomy Trees]** In a taxonomy tree $\mathcal{T}$ with all leaf nodes assigned an integer as its order, each node n corresponds to an interval $[l, u]$ such that: (1) If n is a leaf node, then $l = u = i$, where i is the order assigned to n. (2) Otherwise, $l = \min(c_1.l, \ldots, c_m.l)$, and $u = \max(c_1.u, \ldots, c_m.u)$, where $c_1, \ldots, c_m$ are the children of node n in $\mathcal{T}$.

Figure 4 illustrates the examples of intervals on the taxonomy tree nodes. We say the interval $I_1[l_1, u_1] \preceq I_2[l_2, u_2]$ if $l_1 \geq l_2$ and $u_1 \leq u_2$.

A popular anonymization technique is *generalization*. In particular, in the released microdata, numerical QI-values will be recoded as an interval (e.g., age 20 is recorded as [20, 40]). The categorical QI-values will be replaced with the domain values on higher level in the taxonomy tree, which can be represented as an interval too. Based on this, we formally define anonymization.

Definition 3.5. **[Generalization]** *Generalization* is a one-to-one mapping function f from a microdata D to an anonymized table D^*, such that for any tuple $t \in D$ and any attribute A of t, $I_{t[A]} \preceq I_{f(t[A])}$, where $I_{t[A]}$ and $I_{f(t[A])}$ are the intervals of $t[A]$ and the $f(t[A])$.

For instance, generalizing *Female* in Figure 4 (b) to *** is equivalent to replacing its interval $[2, 2]$ with $[1, 2]$. Note that the sensitive values are always unchanged. Therefore, every anonymized QI-group consists of a set of generalized QI-values and a set of original sensitive values. Based on this, we define the *anonymized QI-group*. We use $G_i[S]$ to denote the sensitive value of the i-th tuple in the QI-group G.

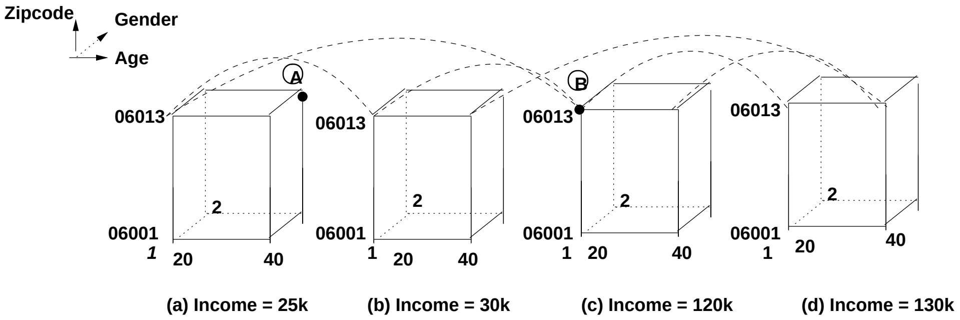
Definition 3.6. **[Anonymized QI-group]** Given a microdata D of m QI-attributes and the sensitive attribute S, let $G \subseteq D$ be a QI-group. Then the anonymized QI-group $G^* = QI^* \cup S^*$, where $QI^* = \{[l_i, u_i] \mid 1 \leq i \leq m\}$, with $[l_i,\, u_i]$ as the interval of the i-th QI-attribute of G, and $S^* = \{G_i[S] \mid 1 \leq i \leq \mid G \mid\}$. We say QI-value q matches G^* if $\forall i (1 \leq i \leq m)$, $q_i \in [l_i, u_i]$.

Justin's QI-group in Figure 3 is $\{$[20, 40], [1, 2], [06001, 06013]$\}$ $\cup$ $\{$25K, 30k, 120K, 130K$\}$. The interval [1, 2] comes from the taxonomy tree of *Gender* in Figure 4 (b).

When a table D is anonymized to a more generalized table D^*, it is important to measure the incurred information loss. A variety of metrics to measure the information loss by generalization have been proposed recently. The ones that are defined based on taxonomy trees are the Generalized Loss Metric [12] and the similar Normalized Certainty Penalty (NCP) [29]. For both metrics, the information loss is measured as a ratio. We adapt these two information loss models to our paper. Specifically,

Definition 3.7. **[Information Loss]**

- For any categorical QI-attribute A, let $\mathcal{T}$ be its taxonomy tree. Let v and v' be a data value of A before



(a) Income = 25k (b) Income = 30k (c) Income = 120k (d) Income = 130k

Figure 5: (3, 4) QI-hypercube group of Justin's QI-group in Figure 3

and after generalization. Let P be the corresponding node of v' in $\mathcal{T}$. Then the information loss of v is:

$$IL_v = (M_P - 1)/(M - 1),$$

where M is the total number of leaf nodes in $\mathcal{T}$, and M_P is the number of leaf nodes in the subtree rooted at node P.

- For any numerical QI-attribute A, let Min and Max be the minimum and maximum of the values of the attribute A. For any value v of A that is generalized to an interval $[l_i, u_i]$, the information loss of v

$$IL_v = (u_i - l_i + 1)/(Max - Min + 1).$$

Given the microdata D, the average information loss $IL_t = \frac{\sum_{t \in D} \sum_{v_i \in t} IL_{v_i}}{|D|}$.

For instance, in Figure 4, assume the value *USA* is generalized to *America*, the information loss is 4/8.

3.3 QI-hypercube

Given a microdata D of m QI-attributes, we can consider it as an m-dimension space. Then for any anonymized QI-group, their generalized intervals can be illustrated as a hypercube in the m-dimension space. Formally, we have:

Definition 3.8. [**QI-hypercube**] Given a microdata D of m QI-attributes, let $G \in D$ be a QI-group, and $G^* = QI^* \cup S^*$ be the anonymization group of G. Let c ($c \leq m$) be the number of QI-attributes whose values are generalized in QI^*. Then QI^* corresponds to a hypercube H of c dimensions, i.e., an *c-cube*, in which the edge on the i-th dimension corresponds to the interval on the i-th attribute ($1 \leq i \leq c$) in G^*. We call H the *QI-hypercube* of G^*.

Figure 5 illustrates four QI-hypercubes, each of them is of the same QI-values {`[20, 40], [1, 2], [06001, 06013]`} (Justin's QI-group in Figure 3). Each QI-hypercube consists of three dimensions, corresponding to generalization on the attributes `Age`, `Zipcode`, and `Gender` respectively. The interval [1, 2] comes from the taxonomy tree of *Gender* in Figure 4 (b).

Each QI-group that consists of n distinct sensitive values can be represented as a group of n QI-hypercubes, with each QI-hypercube corresponding to the same QI-values but different sensitive value s_i ($1 \leq i \leq n$). Formally,

Definition 3.9. [(c, n) **QI-hypercube Group**] Given an anonymized QI-group $G^* = QI^* \cup S^*$, let c be the number of QI-attributes whose values are generalized in QI^*, and n be the number of unique sensitive values in S^*. Then G^* corresponds to a *(c, n) QI-hypercube group* that consists of n c-dimension QI-hypercubes, each corresponding to a sensitive value $s \in S^*$.

Nodes: Given a (c, n) QI-hypercube group, each c-dimension hypercube in the group consists of 2^c nodes. Thus the group consists of $n * 2^c$ nodes. Each node corresponds to a tuple that may or may not exist in the original microdata. We define the *lowerbound* and *upperbound* of the values of the attribute QI_i in G^* as $t_1[QI_i]$ and $t_2[QI_i]$, where t_1 and t_2 correspond to two nodes on an edge on dimension QI_i that only differ the values on QI_i. Figure 5 illustrates the QI-hypercube group of Justin's QI-group in Figure 3. It consists of four QI-hypercubes, corresponding to four sensitive values on *Income*. In this group, node A corresponds to the tuple `(Age=40, Gender=F, Zipcode = 06013, Salary = 25k)`. The lowerbound and upperbound of the *Age* in this group is *20* and *40*.

Edges: We consider two types of edges in the QI-hypercube group, *intra-hypercube* and *inter-hypercube* edges. Intra-hypercube edges connect the nodes in the same hypercube whose corresponding tuples only differ on one QI-attribute, while inter-hypercube edges connect the nodes in different hypercubes whose corresponding tuples are of the same QI-values. Figure 5 illustrates all intra-hypercube edges (in solid lines) and a subset of the inter-hypercube edges (in dotted lines).

4. DISTINGUISHABILITY-BASED ATTACK

In this section, we define the *distinguishability-based attack* on the outliers. We assume the adversary has the QI-values of all individuals from the external public dataset, e.g., voter lists. We also assume that the adversary knows which individuals are outliers. This piece of adversary knowledge can be expressed as a set of entries in the form of `(QI, outlierness)`, where *QI* is the QI-value of an individual, and *outlierness* is valued "true" or "false". For simplicity, in the following, we use (o, `'T'`) to denote the adversary knowledge that `(QI=o, outlierness='True')`. We call such o the *outlier QI-value*. It is possible that adversary may know more than true/false of the outliers, for example, he/she also knows the domains that the outlierness lies (e.g., Justin stands out on the combination of his age and income). But in this paper, we only focus on the simplest adversary knowledge as the true/false of outlierness. We will show that even with this small bit of information, there exists privacy breach.

We consider two types of privacy: *presence privacy*, which is the fact that the individual's record is present in the released microdata, and *association privacy*, which is the association between the individual and the sensitive values. Based on the adversary knowledge of outliers, both types of privacy can be disclosed by the inference of the distinguishability of the outliers. Formally,

DEFINITION 4.1. **[Distinguishability-based Attack]** Given a microdata D and the adversary knowledge $(o, `T')$, let $G^* = QI^* \cup S^*$ be the anonymized group that matches o. If the adversary can infer that there must exist at least an outlier in G^*, then the probability of *presence privacy leakage*

$$Pr(o \in D \mid G^*) = 1/h,$$

where h is the number of outliers whose QI-values match G^*. Further, the probability of *association privacy leakage*

$$Pr((o, s) \in D \mid G^*) = 1/l,$$

where l is the number of distinct sensitive values $s \in S^*$ such that (o, s) is a (p, d)-outlier.

If the adversary cannot infer the existence of any outlier in D, both $Pr(o \in D \mid G^*)$ and $Pr((o, s) \in D \mid G^*)$ equal 0. Therefore, the key to protect both the presence and association privacy of outliers is to prevent the adversary from inferring the existence of outliers. To address this, we study how the adversary can infer the existence of the outliers from the released G^*. First, we define *bad* QI-hypercube nodes and edges.

DEFINITION 4.2. **[Bad QI-hypercube Nodes and Edges]** A node of a QI-hypercube is *bad* if it corresponds to a (p, d)-outlier tuple. An (intra-hypercube or inter-hypercube) edge is *bad* if it connects two bad nodes.

For instance, node B in Figure 5 is a bad node, since it corresponds to an outlier tuple (`Age = 20, Gender = M, Zipcode = 06013, Income = 120K`).

Second, we define *distance-constrained tuples.* The intuition of this definition is that such tuples can enable the inference of existence of outliers (as shown in later Lemma 4.1).

DEFINITION 4.3. **[Distance-constrained Tuple]** Given a microdata D and an anonymization group $G^* = QI^* \cup S^*$ of D, let H^s be a QI-hypercube of G^* on the sensitive value $s \in S^*$. Then a tuple that matches G^* is *distance-constrained* if it satisfies that $\forall$ tuple $t' \in D$ and $\forall$ QI-attribute QI_i, either $t[QI_i] + l_i \geq 2 * t'[QI_i]$ or $t[QI_i] + u_i \leq 2 * t'[QI_i]$, where l_i and u_i are the lowerbound and upperbound values on dimension QI_i in H^s.

By knowing all QI-values of the individuals from the external knowledge, the adversary can easily check whether a tuple is distant-constrained. An important property of *distance-constrained tuple* is explained in the following Lemma.

LEMMA 4.1. Given a distance-constrained tuple t, $\forall$ tuple $t' \in D$, it must satisfy that $\forall$ QI-attribute QI_i, either $\mid t[QI_i] - t'[QI_i] \mid \geq \mid l_i - t'[QI_i] \mid$, or $\mid t[QI_i] - t'[QI_i] \mid \geq \mid u_i - t'[QI_i] \mid$, where l_i and u_i are the lowerbound and upperbound values on dimension QI_i of G^* that t matches.

Proof sketch: For simplicity, let $q = t[QI_i]$ and $q' = t'[QI_i]$. We have $l_i \leq q \leq u_i$. There are two possibilities of q': (1) $q' \notin [l_i, u_i]$, and (2) $q' \in [l_i, u_i]$. For Case (1), $q' < l_i$ or $q' > u_i$. If $q' < l_i$, then $\mid q - q' \mid \geq \mid l_i - q' \mid$. Similarly if $q' > u_i$, then $\mid q - q' \mid \geq \mid u_i - q' \mid$. Then for Case (2), we prove by contraction. Assume that both $\mid q - q' \mid < \mid l_i - q' \mid$ and $\mid q - q' \mid < \mid u_i - q' \mid$. Then we have $q + l_i < 2 * q'$ and $q + u_i > 2 * q'$, which contradicts the assumption that t is a *distance-constrained* tuple. The correctness of the lemma then follows. ▮

Based on Lemma 4.1, we can show that single QI-hypercubes that contain bad nodes can be used to infer the existence of outliers. Specifically, we have:

THEOREM 4.1. **(Inference of Existence of Outliers on Single QI-hypercube) :** Given a microdata D and the adversary knowledge $(o, `T')$, let $G^* = QI^* \cup S^*$ be the anonymized QI-group that o matches, and H^s be the QI-hypercube of G^* on sensitive value $s \in S^*$. If: (1) o is *distance-constrained*, and (2) H^s only contains bad nodes, then o must exist in G^*.

Proof sketch: We prove the correctness by contradiction; if o is a non-outlier, then H^s must contain at least a good node. Let $T = \{t \mid \forall \text{ node } n \in H^s,\ n \text{ corresponds to } t\}$. It is straightforward that $\forall t \in T$, it must be true that $\forall$ QI-attribute QI_i, $t[QI_i]$ is either the lowerbound or the upperbound value of QI_i in G^*. From Lemma 4.1, there must exist a tuple $t \in T$ s.t. for each dimension QI_i, $\forall$ tuple $t' \in D$,

$$\mid o[QI_i] - t'[QI_i] \mid \geq \mid t[QI_i] - t'[QI_i] \mid (*).$$

Since o is a non-outlier, there are more than $p\%$ tuples t' whose $dist(o, t') < d$. Thus based on (*), the tuple t must also be a non-outlier, which contradicts the assumption that every node in H^s corresponds to an outlier. Thus o must exist in the microdata. The correctness of the theorem then follows. ▮

We assume the adversary can decide the outlierness of QI-hypercube nodes from his/her common knowledge. Then Theorem 4.1 shows that by combining the outlierness of the individuals from adversary knowledge, and the distance-constrainedness of the outlier nodes, the adversary can infer the existence of outliers. The anonymization group of Justin's record in Figure 2 (a) is such an example. In this example, all nodes of the QI-hypercube of sensitive value $120K$ are bad. Since Justin's record is distance-constrained, the adversary can infer that Justin's record must exist in the microdata. Since Justin is the only outlier whose QI-values matches the group, the adversary's probability of Justin's presence privacy is 100%. Further, since $120K$ is the only sensitive value that makes Justin's record an outlier, the adversary's probability of his association privacy is 100%. In other words, Justin's privacy has been completely revealed.

Next, we study how the adversary infers the existence of outliers from multiple QI-hypercubes in the same QI-group. We have:

THEOREM 4.2. **(Inference of Existence of Outliers from Multiple QI-hypercubes) :** Given a microdata

D and the adversary knowledge $(o, `T')$, let $G^* = QI^* \cup S^*$ be an anonymized QI-group. Let H be the QI-hypercube group of G^*. If there exists a node $n \in H$ such that: (1) all inter-hypercube edges that connect the nodes of the same QI-value of n are bad, and (2) $\forall t' \in D$, it satisfies that $\forall$ QI-attribute QI_i, $\mid o[QI_i]-t'[QI_i] \geq\mid t[QI_i]-t'[QI_i] \mid$, where t is the tuple that node n corresponds to, then o must exist in G^*.

Proof Sketch: We prove that if $o \notin G^*$, then there must exist at least an inter-hypercube edge that is not bad. The contradiction proof is similar to Theorem 4.1; since o is a non-outlier, there are more than $p\%$ tuples t' whose $dist(o, t') < d$. Since $\mid o[QI_i] - t'[QI_i] \geq\mid t[QI_i] - t'[QI_i] \mid$, t must be an non-outlier, which violates the assumption that every node on the inter-hypercube edges corresponds to an outlier. The correctness of the theorem then follows. ▮

An example for inference of existence of outliers from multiple QI-hypercubes is Justin's record in Figure 2 (b). All inter-hypercube edges on the nodes of the same QI-value that contains *Age=20* are bad. Thus *Justin*'s record must exist in the original microdata. Consequently Justin's presence privacy is inferred with probability 1. However, the adversary's probability of Justin's association privacy is 1/3, since all three income values can make outlier tuple.

5. PRIVACY MODEL

Theorem 4.1 and 4.2 have shown that the adversary can possibly infer the presence of the outlier tuples. Consequently he/she can disclose both the presence and association privacy of the outliers by distinguishability-based attack. To deal with this attack, we define *plain k-anonymity.*

DEFINITION 5.1. **[Plain k-anonymity]** Given the microdata D and the adversary knowledge $(o, `T')$, the anonymization group $G^*{=}QI^* \cup S^*$ is *plain* and *k-anonymous* if: (1) There are k distinct sensitive values in S^*, and (2) $Pr(o \in D \mid G^*) = Pr((o, s) \in D \mid G^*) = 0$.

Given the microdata D, our goal is to find the plain and k-anonymous groups. The key is to disable the adversary to infer the existence of outliers. As shown by both Theorem 4.1 and Theorem 4.2, the inference is always based on the badness of QI-hypercube nodes and whether the outlier tuples are distance-constrained. Since distance-constrained is not changeable with given QI-values, what we can do is to remove the badness of QI-hypercube nodes. We have:

THEOREM 5.1. **(Plain k-anonymous Groups) :** Given a QI-group G, let $G^* = QI^* \cup S^*$ be its anonymization. Let H be the QI-hypercube group of G^*. Then G^* is *plain and k-anonymous* if $\forall t \in G$: (1) There are at least k distinct values in S^*, (2) $\forall$ QI-hypercube $H^s \in H$ $(s \in S^*)$, there exists at least a node $n \in H^s$ that is not bad, and (3) $\forall$ inter-hypercube edges that connect the nodes of the same QI-values, there exists at least one that is not bad.

The correctness of Theorem 5.1 is straightforward from Theorm 4.1 and 4.2. Figure 3 is an example of plain k-anonymous groups. Now our goal is to design such anonymization scheme.

6. GLOBAL AND LOCAL OUTLIERS

Prior to studying how to design the plain k-anonymous scheme, a fundamental question must be answered: given an outlier, does there always exist a plain, k-anonymized QI-group? Unfortunately it is not true. There may exist outliers that cannot be included into any plain, k-anonymous QI-group. One example is *Bill*'s record in Figure 1. Due to his extremely high income (2 billion), including his record into any QI-group would not help to hide the fact that the record of someone who is super rich exists in the original microdata. Therefore, to address the impact of distinguishability of the outliers to their identification, the outliers are categorized into two types: *global* and *local* outliers.

DEFINITION 6.1. **[Global and Local Outliers]** Given a microdata D, let o be an (p, d)-distance outlier in D. We say o is a *global* outlier if $\nexists$ a plain k-anonymous QI-group $G \in D$ s.t. $o \in G$. Otherwise, o is a *local* outlier.

Following the definition, a naive way to find out the global outliers is to try all possible QI-group schemes, which is very costly. The challenge is to efficiently locate the global outliers in the microdata. We have:

THEOREM 6.1. **(Global Outlierness of Sensitive-attribute Outliers) :** Any sensitive-attribute outlier o is a global outlier.

The correctness of Theorem 6.1 is straightforward. Any tuple that contains a (p, d)-outlier sensitive value s must be a (p, d)-outlier. Therefore, any QI-hypercube that contains s must only contain bad nodes, which violates the condition (2) in Theorem 5.1.

To eliminate the global outlierness, one possible solution is to suppress the sensitive values. However, as other tuples publish sensitive values, such "abnormal" suppression may enable the attacker to infer the existence of outliers. Thus we have no choice but to remove the global outlier tuples, even though this may cause 100% infomraiton loss on these tuples.

Next, we answer the question that *whether a multi-attribute outlier can be a global outlier.*

We have the following theorem.

THEOREM 6.2. **(Local-outlierness of Multi-attribute Outliers) :** Any multi-attribute outlier o is a local outlier.

Proof: Assume there exists a multi-attribute outlier o that is a global outlier. Then any QI-group that matches the QI value QI of o cannot be plain, i.e., all nodes of the QI-hypercubes are bad. However, there may exist the anonymized QI-groups whose QI-hypercubes that o matches are of nodes corresponding to non-outlier tuples, which brings contradiction. ▮

7. ANONYMIZATION ALGORITHM

Given a microdata D, our goal is to split the microdata into partitions that correspond to plain, k-anonymized QI-groups, so that the adversary cannot explicitly identify neither any outlier nor any regular tuple. To achieve this goal, we propose an efficient construction mechanism. It consists of three steps, *removal of global outliers*, *expansion-based*

Tuple ID	Age
1	20
2	20
3	20
4	30
5	30
7	30
8	40
9	40

(a) Sorted list on *Age*

Tuple ID	Gender
2	M
3	M
5	M
7	M
9	M
1	F
4	F
8	F

(b) Sorted list on *Gender*

Tuple ID	Zipcode
4	06001
8	06002
9	06003
7	06004
1	06006
5	06010
2	06011
3	06013

Sorted list on *Zipcode*

Tuple ID	Income
1	20K
2	25K
4	30K
5	50K
7	100K
8	110K
3	120K
9	130K

(b) Sorted list on *Income*

Figure 6: An example of sorted lists; Bill's record (global outlier) has been removed.

grouping, and *processing of residue tuples*. Next, we elaborate the details of these three steps. The pseudo code is shown in Algorithm 1. Before we present the algorithm, we must point out that finding optimal k-anonymization with minimal information loss is NP-hard [21, 14]. Thus we focus on efficient heuristics.

7.1 Step 1: Removal of Global Outliers

Since none of the global outliers can be included into any plain QI-group, thus first, we remove all global outliers, i.e., sensitive-attribute outliers, from the microdata. Line 1 - 2 of Algorithm 1 remove the global outliers. Although the removal operation is simple, it reduces the size of the microdata and as a result it may affect the later decision of (p, d)-outlierness. Therefore, we record the size of the original microdata and in the following steps, we always use this size to check outlierness.

7.2 Step 2: Expansion-based grouping

After removal of global outliers, only local outliers remain. The naive approach of QI-group construction is to first locate all local outliers, then for each local outlier, try all possible partition schemes until a plain, k-anonymous scheme is reached. This approach is extremely inefficient for two reasons: (1) finding all outliers in the dataset may be time costly, and (2) the number of possible partition schemes is exponential to the number of the tuples in the microdata. The time complexity of finding all outliers can be quadratic to the size of the dataset [20, 22], sub-quadratic to the size of the dataset [10], or linear to the size of the dataset but exponential in the number of dimensions [20]. Although the use of spatial index structures (KD-trees [6], R-trees [11], or X-trees [7]) may help to speed up the outlier detection, the index itself brings construction overhead and maintenance cost. Therefore, to efficiently construct a plain and k-anonymous QI-group scheme, it is desirable to anonymize the groups *without finding any local outlier before anonymization*. Indeed, as long as the constructed QI-hypercubes satisfies Theorem 5.1, their corresponding anonymized QI-groups must be plain and k-anonymous, no matter whether they contain outliers. Following this principle, we design our anonymization algorithm. The basic idea of the algorithm is that for every tuple t, it starts itself as the seed group. The seed group is repeatedly *expanded* by adding new tuples, until a plain, k-anonymized QI-group is reached. Our experimental results show that our expansion-based approach always achieves better performance than the approaches that need to find the local outliers before anonymization. More details can be found in Section 8.

Expansion operation is equivalent to adding tuples into the QI-group. The effect of expansion is to enlarge at least one dimension in the QI-hypercube, so that the "badness" of the nodes on these dimension will be reduced accordingly. A fundamental question is, which tuple(s) should be chosen for expansion? Randomly picking tuples may result in two tuples that are far away being put into the same QI-group, which may induce much information loss. Therefore, to reduce the information loss, we keep a sorted list for every attribute. The values in the sorted list are sorted in descending order. To record the association between tuples and their values, every value is linked with its tuple ID in the sorted list. Figure 6 shows the sorted lists of the microdata in Figure 1. Every sorted list keeps a pointer that points to the next value that will be picked. Based on the sorted lists, we expand the group in a *greedy* fashion: whenever the expansion is needed, for each QI-attribute QI_i, we pick the candidate tuple t_i that is pointed to in the sorted list L_i. If there are multiple tuples that are of the same value on QI_i as the pointed tuple, we pick all of them. We collect all candidates and pick the one such that including it into the current QI-group will make the group plain and k-anonymized. If there is no such tuple or there are multiple choices, we pick the one that yields the least information loss if it is included into the current QI-group. Line 8 - 13 of Algorithm 1 gives more details. After the added tuple is picked, the sorted lists will be updated accordingly. To be more specific, let $t(q_1, \ldots, q_m, s)$ be the picked tuple. Then in each sorted list L_i, we remove the value q_i. Furthermore, we update the pointers in the sorted list. To be specific, let G^* be the new anonymized QI-group after expansion with t. For each attribute QI_i, let $[l_i, u_i] \in G^*$ be the generalized interval. Then the pointer in the sorted list L_i will be moved to the value v right next to u_i, if $u_i - v < v - l_i$, or right before l_i otherwise. In other words, the pointer always points to the value that is the closest to the current QI-group. The updates of the sorted lists are implemented by Line 14 - 16 of Algorithm 1. When we reach a plain, k-anonymized group, we return the group, pick the next ungrouped tuple, and repeat the above procedure until all tuples are traversed.

It may be possible that at some stage of expansion, the QI-group includes a tuple that makes the bad nodes of the QI-hypercube. More seriously, further expansion of this QI-group may not be able to remove the badness of the nodes. As a result, keep expanding such QI-groups will never terminate and produce plain, k-anonymity groups. To avoid this situation, we keep track of the number of steps that the groups have been expanded. If it exceeds a given threshold δ (Line 19), we look for a unpicked non-outlier tuple t such that including t into the group will incur that a node of the QI-hypercube corresponds to t (Line 20 - 23). In other words, the QI-hypercube consists of at least a good node and consequently construct a plain and k-anonymous QI-group. If no such tuple t exists in the microdata, to reduce the "badness" of the QI-hypercube nodes, we adjust the current QI-hypercube such that at least one QI-hypercube node whose corresponding tuple is changed. To achieve this goal, we remove a tuple in the current group (Line 25). The criteria is to pick the tuple t that contributes the most to the QI-hypercube nodes, i.e., the number of attributes on which t has either the minimum value (contributing to the lowerbound) or the maximum value (contributing to the upperbound). After the removal, we repeat the whole grouping procedure.

Example 7.1. **[Expansion]**
Assume we consider 3-anonymity. Assume we have picked the tuple 1, 2 and 3, which will result in the QI-group as *Age=20, Gender=[M, F], Zipcode=[06006, 06013], income* $=\{$*20K, 25K, 120K*$\}$. This QI-group is not plain. Then we pick tuple 4 from the sorted list of the attribute *age*, tuple 5, 7 and 9 from the sorted list of *Gender* (they are of the same value M), and tuple 7 from the sorted list of *Zipcode* (06004 is the closest value to [06006, 06013]). Both tuple 5 and 7 can make the QI-group as a plain 3-anonymity group. However, since tuple 7 induces less information loss, tuple 7 is chosen for expansion. □

7.3 Step 3: Processing of Residue Tuples

Since Step 2 only considers unanonymized tuples for expansion, it is possible that some tuples cannot be grouped if their group members are anonymized already. For each such residue tuple t, first, we pick the anonymization group that produces the minimal information loss by including t as the seed group. We repeatedly merge the seed group with the other groups if the merge causes the minimal information loss, until we reach a plain and k-anonymity group. Line 19 - 24 of Algorithm 1 process the residue tuples.

7.4 Discussion

The performance of Algorithm 1 is dominated by Step 2. In this step, the time complexity of checking plain and k-anonymous group (Line 7) is $O(kn2^m)$, where k is the given threshold for k-anonymity, m is the number of QI-attributes, and n is the size of the microdata. The time complexity of each greedy expansion is $O(m)$. For the worst case it will be expanded n times. Thus the total cost of expansion is $o(mn)$. The sorted lists are updated with $O(m)$ complexity. Thus the complexity of Algorithm 1 is $O(kn2^m)$.

8. EXPERIMENTS

We ran a battery of experiments to measure the performance of our anonymization algorithm and explored various factors that impact the anonymization performance. Further, we investigated the information loss by our anonymization approach. In this section, we describe our experiments and provide an analysis of our observations.

Algorithm 1 Algorithm ExpC(): Construct the plain k-anonymous scheme based on expansion

```
Require: Microdata $D$;
Ensure: A generalized version $D^*$ that is good;
    {Step 1: Remove global outliers}
 1: for all tuple $t$ that is sensitive-attribute outlier do
 2:     remove $t$;
    {Step 2: Expansion-based grouping}
 3: $QGroup \leftarrow \{\}$;
 4: repeat
 5:     Pick a ungrouped tuple $t$;
 6:     $G = \{t\}$;
 7:     $Num \leftarrow 0$;
 8:     while $G$ is not a plain k-anonymity group do
 9:         $CandidateSet = \{\}$;
10:         for all Sorted list $L_i$ do
11:             Pick the tuple $t'$ that is pointed in $L_i$;
12:             $CandidateSet = CandidateSet \cup \{t'\}$;
13:         Let $t \in CandidateSet$ be the one that makes $G \cup \{t\}$
            of the minimum information loss;
14:         $G \leftarrow G \cup \{t\}$;
15:         for all Sorted list $L_i$ do
16:             move the pointer to the closest value to the in-
                terval of attribute $A_i$ in $G \cup \{t\}$;
17:             remove the value $a_i$ of the tuple $t$ from $L_i$;
18:         $Num \leftarrow Num + 1$;
19:         if $Num \geq \delta$ then
20:             $max_i \leftarrow$ the maximum value of all tuples in $G$ on
                attribute $QI_i$;
21:             $min_i \leftarrow$ the minimum value of all tuples in $G$ on
                attribute $QI_i$; {Expand too much; Choose a non-
                outlier tuple to be the node of the QI-hypercube;}
22:             if $\exists$ a unpicked non-outlier tuple $t$ s.t., $\forall$ QI-
                attribute $QI_i$, $t[QI_i] > max_i$ or $t[QI_i] < min_i$
                then
23:                 $G \leftarrow G \cup \{t\}$;
24:             else
25:                 Remove a tuple $t'$ from $G$ s.t. it has the max-
                    imum number of the attributes on which $t'$
                    equals either $max_i$ or $min_i$;
26:     $QGroup \leftarrow QGroup \cup G$;
27: until The sorted lists are empty;
    {Step 3: Processing of Residue Tuples}
28: for all Residue tuple $t$ do
29:     pick $G$ s.t. the information loss of $G \cup \{t\}$ is minimal
30:     while $G$ is not a plain k-anonymity group do
31:         for all $G' \neq G$ do
32:             if $G \cup G'$ is of minimal information loss then
33:                 $G \leftarrow G \cup G'$;
```

8.1 Experimental Setup

Setup We use a PC machine with one processor having a speed of 2GHz and 1GB of RAM. We implement the algorithms in C++. **Datasets** We use the *Census* dataset that contains personal information of 500,000 American adults[3]. The details of the dataset are summarized in Figure 7. We construct test datasets of different sizes by picking various subsets from the *Adults* data.

Anonymization approaches We mainly compare the performance and information loss of three anonymization ap-

[3] http://www.ipums.org/

Attribute	Number of distinct values
Age	78
Gender	2
Education	17
Marital	6
Race	9
Work Class	10
Country	83
Occupation	50
Salary-class	50

Figure 7: Summary of Attributes

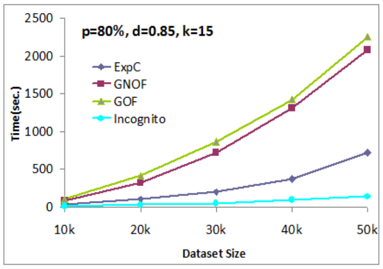


Figure 8: Performance Comparison of *ExpC*, *GNOF*, and *GOF* Approaches; p=0.8, d=0.85, k=15

proaches:

1. *ExpC* approach: our anonymization algorithm *ExpC* (Algorithm 1 in Section 7) that anonymizes the microdata by expansion-based grouping without finding any local outlier beforehand.

2. *GNOF* approach: The local outliers are located first. When constructing the anonymization groups, non-outlier tuples are grouped first. The outliers are added into the groups of non-outliers.

3. *GOF* approach: Similar to *GNOF* approach, the local outliers are located first. Then the construction of the anonymization groups start from outliers. The group is constructed by using the same expansion-based grouping approach. The non-outliers are added into the groups by expansion.

We also implemented the *Incognito* generalization approach in [13] for comparison of the performance.

Information loss measurement We use the information loss measurement defined in Section 3, i.e., the average information loss $IL = \frac{\sum_{t\in D}\sum_{v_i\in t} IL_{v_i}}{|D|}$.

8.2 Performance

8.2.1 Comparison of Four Anonymization Approaches

The first part of performance experiments is to compare the performance of four anonymization approaches, *ExpC*, *GNOF*, *GOF*, and *Incognito* [13]. Figure 8 shows the result. First, Incognito always wins, since it never checks the goodness of the anonymization groups. Second, our *ExpC* approach gets better performance than *GNOF* and *GOF*. The performance gain increases when the size of databases grows. When the dataset is of $50K$, our *ExpC* approach is around twice faster than *GNOF* and *GOF*. This is because compared with *GNOF* and *GOF* approaches, our approach avoids outlier detection, which takes considerable amounts of time.

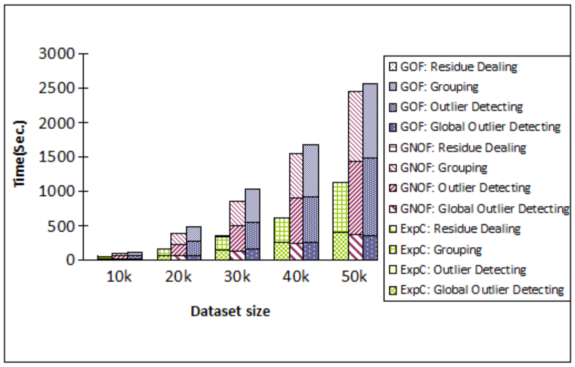


Figure 9: Performance of Four Components of *ExpC*, *GNOF*, and *GOF* Approaches; p=0.8, d=0.85, k=15

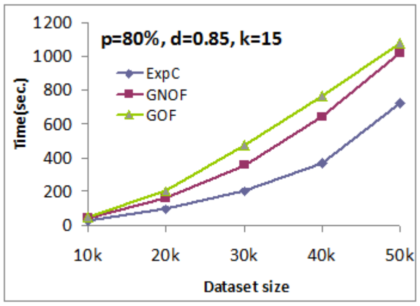


Figure 10: Performance of Expansion of *ExpC*, *GNOF*, and *GOF* Approaches

To further study more details of the performance, we measure the performance of four major components of each approach. These major components are: finding global outliers, finding local outliers, expansion-based grouping, and residue processing. Figure 9 shows the results. We observed that for all three approaches, the time cost of expansion-based grouping is dominant. This proves that it is necessary to avoid finding the local outliers before anonymization, as what we have did in the *ExpC* approach. Further, the cost of residue processing is always negligible. This is because the number of residue tuples are always very small compared with the size of the microdata. We also compare the cost of the expansion-based grouping of *ExpC* with both the *GNOF* and *GOF* approaches (Figure 10). The observation is that the cost of expansion-based grouping of the *ExpC* approach is always less than that of both *GNOF* and *GOF* approaches, which proves that anonymization without distinguishing outliers is efficient.

Furthermore, we compare the average size of QI-groups for these three approaches. The results are in Figure 11. It shows that our *ExpC* approach always produces the QI-groups of the smallest average sizes, which results in smaller information loss, as shown in Section 8.3.

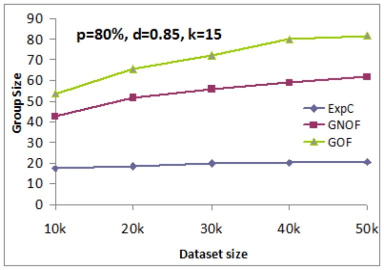


Figure 11: Average size of QI-groups

8.2.2 *Performance Study of Our* ExpC *Approach*

The second part of performance experiments is to study the impact of various configurations to the performance of our *ExpC* approach. First, we measure the performance of *ExpC* algorithm for datasets of various sizes. The result is shown in Figure 9. Unsurprisingly, the performance degrades with databases of increasing sizes. Then we change the setup of p, d values for the definition of (p, d)-outliers, and k value for k-anonymity. First, we fix the value p and change d value. The results are shown in Figure 12 (a). We observed that for most of the cases, the performance gets better with increasing d values. The reason is that the larger d value is, the less likely that the nodes of the QI-hypercubes are the outliers, and thus the fewer expansions are needed. Further, we observed that increasing k values improve performance. The reason for this is that smaller k values will result in smaller QI-groups and possibly more necessary expansions.

We also fix the value d and change p value. The result is shown in Figure 12 (b). We observed the similar phenomenon as fixing p value case.

8.3 Information Loss

8.3.1 *Comparison of Four Approaches*

We compare the information loss of the three approaches, as well as the approach that simply removing all outliers (including both global and local ones). The results are shown in Figure 13. We observed that our *ExpC* approach always achieves the best information loss. Further, removing outliers incurs the worst information loss, which proves that our original statement that simply removing outlier tuples is not a good solution regarding the utility of the anonymized data.

8.3.2 *Information Loss of Our* ExpC *Approach*

We evaluate the information loss of our *ExpC* approach for various configurations of p, d, and k values. We start from fixed d values. The result is shown in Figure 13 (b). The first observation is that the information loss is always small (less than 1). This proves that our algorithm can achieve anonymization with small information loss. Second, the information loss increases with increasing k values. The result is straightforward: with bigger k values, the QI-groups must be larger, which incurs greater information loss. Third, it shows that the information loss is always the same with larger p values. This is because the non-outlier for (p, d_1) configuration must be non-outlier for (p, d_2), where $d_1 \leq d_2$. Consequently any plain, k-anonymous group with (p, d_1) configuration must be plain and k-anonymous for (p, d_2). We also measure the information loss for fixed p values and show the result in Figure 13 (c) and observe the similar results.

8.4 Summary of Experiments

The experimental results demonstrates that our *ExpC* approach can efficiently anonymize the microdata that contains outliers with low information loss. The key to achieve good performance is that we avoid finding outliers before anonymization.

9. CONCLUSION

In this paper, we studied the k-anonymization problem with presence of outliers. We defined the novel concept of *plain k-anonymity* so that the distinguishability of outliers are adequately protected by anonymization. We characterized outliers into two types, global outliers and local outliers, and studied anonymization technique for each type. We designed an efficient algorithm to produce plain k-anonymity schemes.

There are many interesting issues to be explored in the future. In particular, we aim at extending our model to richer external knowledge, for example, the adversary not only knows that Justin is an outlier but also his outlierness lies on the combination of age and income. Then the attack can be more sophisticated than the current distinguishability-based attack. We plan to continue on this theme. We are also interested in studying possible optimization techniques on the expansion-based approach. Extending the framework to support database updates is another interesting issue. Further, we plan to work on density-based outliers.

10. REFERENCES

[1] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining", SIGMOD 2000.

[2] C. C. Aggarwal, "On Randomization, Public Information and the Curse of Dimensionality", ICDE 2007.

[3] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces", in Proceedings of the Sixth European Conference on the Principles of Data Mining and Knowledge Discovery, 2002.

[4] S. D. Bay, M. Schwabacher, "Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule", SIGKDD 2003.

[5] R. J. Bayardo and R. Agrawal, "Data privacy through optimal k-anonymization", ICDE 2005.

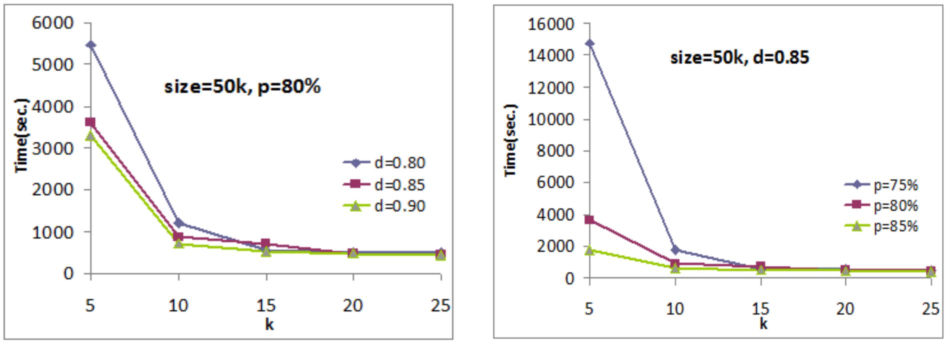
[6] J. Bentley, "Multidimensional binary search trees used for associative searching". Communications of the ACM, 1975.

[7] S. Berchtold, D. Keim, and H. Kreigel, "The X-tree: an index structure for high dimensional data", In VLDB, 1996.

[8] M. M. Breunig, H. Kriegel, R.T. Ng, J. Sander, "LOF: Identifying Density-Based Local Outliers", SIGMOD 2000.

[9] T. Dalenius and S. P. Reiss, "Data swapping: a technique for disclosure control", Journal of Statistical Planning and Inference, 1982.

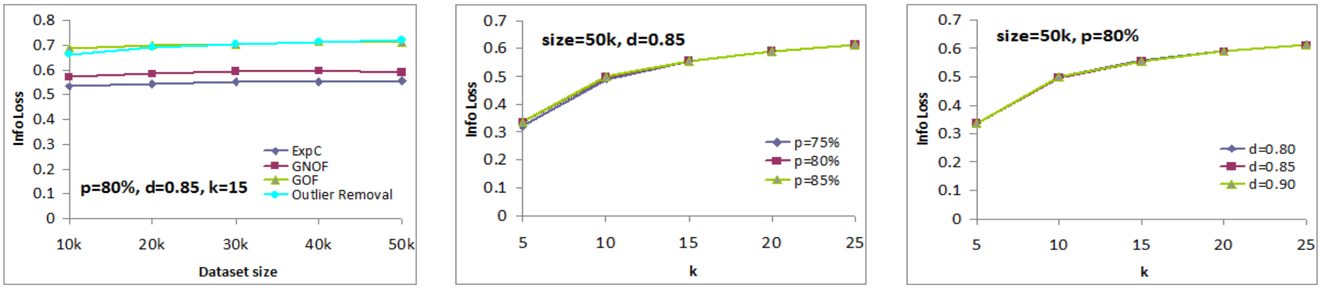
[10] A. Ghoting, S. Parthasarathy, M. E. Otey, "Fast Mining of Distance-Based Outliers in High-Dimensional Datasets", SIDM, 2006.



(a) Fix p

(b) Fix d

Figure 12: Performance of *ExpC* Approach; Dataset $50K$



(a) Four Approaches

(b) *ExpC* approach, fix d

(c) *ExpC* approach, fix p

Figure 13: Information Loss

[11] R. Guttmann, "A dynamic index structure for spatial searching", In SIGMOD, 1984

[12] V. S. Iyengar, "Transforming data to satisfy privacy constraints", SIGKDD, 2002.

[13] K., David DeWitt, and Raghu Ramakrishnan, "Incognito: Efficient Full-domain K-anonymity", SIGMOD 2005.

[14] K. LeFevre, D. DeWitt, and Raghu Ramakrishnan, "Mondrian Multidimensional K-Anonymity", ICDE 2005.

[15] J. Li, Y. Tao, X. Xiao, "Preservation of Proximity Privacy in Publishing Numerical Sensitive Data", SIGMOD 2008.

[16] N. Li, T. Li, "t-Closeness: Privacy Beyond K-anonymity and l-diversity", ICDE 2007.

[17] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. "l-Diversity: Privacy Beyond k-Anonymity", ICDE 2006.

[18] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, J. Y. Halpern, "Worst-Case Background Knowledge for Privacy-Preserving Data Publishing", ICDE 2007.

[19] M. Ercan Nergiz, Maurizio Atzori, Christopher W. Clifton, " Hiding the Presence of Individuals from Shared Databases", SIGMOD'07.

[20] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications", VLDB Journal, 8(3-4):237-253,2000.

[21] Adam Meyerson, Ryan Williams, "On the Complexity of Optimal K-anonymity", PODS, 2004.

[22] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets", SIGMOD 2000.

[23] P. Samarati, L. Sweendy, "Generalizing Data to Provide Anonymity when Disclosing Information", PODS, 1998.

[24] P. Samarati, "Protecting respondents' identities in microdata release", TKDE, 2001.

[25] L. Sweeney, "k-anonymity: a model for protecting privacy", International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5):557570, 2002.

[26] R. Wong, J. Li, A. Fu, K. Wang, "(α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy-Preserving Data Publishing", SIGKDD, 2006.

[27] R. Wong, A. Fu, K. Wang J. Pei, "Minimality Attack in Privacy Preserving Data Publishing", VLDB, 2007.

[28] X. Xiao, Y. Tao,"Anatomy: Simple and Effective Privacy Preservation", VLDB, 2006.

[29] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, A. Fu, "Utility-Based Anonymization Using Local Recoding", SIGKDD, 2006.

[30] Q. Zhang, N. Koudas, D. Srivastava, T. Yu: "Aggregate Query Answering on Anonymized Tables", ICDE 2007.