CERIAS Tech Report 2006-10

EFFICIENT K-ANONYMITY USING CLUSTERING TECHNIQUE

by Ji-Won Byun and Ashish Kamra and Elisa Bertino and Ninghui Li

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

Efficient k-Anonymization using Clustering Techniques

Ji-Won Byun[1] Ashish Kamra[2] Elisa Bertino[1] Ninghui Li[1]

[1] Computer Science, Purdue University, {byunj, bertino, ninghui}@cs.purdue.edu

[2] Electrical and Computer Engineering, Purdue University, akamra@purdue.edu

ABSTRACT

k-anonymization techniques are a key component of any comprehensive solution to data privacy and have been the focus of intense research in the last few years. An important requirement for such techniques is to ensure anonymization of data while at the same time minimizing the information loss resulting from data modifications such as generalization and suppression. Current solutions, however, suffer from one or more of the following limitations: reliance on pre-defined generalization hierarchies; generation of anonymized data with high information loss and with high classification errors; and the inference channel arising from lack of diversity in the sensitive information.

In this paper we propose an approach that addresses these limitations. Our approach uses the idea of clustering to minimize information loss and thus ensure good data quality. The key observation here is that data records that are naturally close with respect to each other should be part of the same equivalence class. Current clustering techniques, however, are not directly applicable in this context because they do not consider the requirement that each cluster should contain at least k records. We thus formulate a specific clustering problem, referred to as *k-member clustering problem*. We prove that this problem is NP-hard and present a greedy algorithm, the complexity of which is in $O(n^2)$. As part of our approach we develop a suitable metric to estimate the information loss introduced by generalizations, which works for both numeric and categorical data. We also present extensions to our proposed algorithm that minimize classification errors in the anonymized data and eliminate the inference channel arising from lack of diversity in the sensitive attributes. We experimentally compare our algorithm with two recently proposed algorithms. The experiments show that our algorithm outperforms the other two algorithms with respect to information loss, classification errors, and diversity.

1. INTRODUCTION

Since the early age of information technology, privacy has been a challenging issue. Among many aspects of privacy, *data privacy*, which refers to the privacy of the individual to which the data[1] is related, has been one of the difficult research problems. The difficulty comes from the fact that data utility (i.e., data quality) and data privacy conflict with each other. Intuitively, data privacy can be enhanced by hiding more data values, but it decreases data utility; on the other hand, revealing more data values increases data utility, but it decreases data privacy. Therefore, we need to devise solutions that best address both utility and privacy requirements of data.

A recent approach addressing data privacy relies on the notion of *k-anonymity* [26, 30]. In this approach, data privacy is guaranteed by ensuring that any record in the released data is indistinguishable from at least $(k-1)$ other records with respect to a set of attributes called the *quasi-identifier*. Although the idea of k-anonymity is conceptually straightforward, the computational complexity of finding an optimal solution for the k-anonymity problem has been shown to be NP-hard, even when one considers only cell suppression [1, 23]. The k-anonymity problem has recently drawn considerable interest from research community, and a number of algorithms have been proposed [5, 11, 18, 19, 29]. We categorize these algorithms into two classes: *hierarchy-based generalization* and *hierarchy-free generalization*.

In the hierarchy-based generalization [5, 11, 18, 29], a generalization-hierarchy is first defined for each quasi-identifier attribute. Then an algorithm tries to efficiently find an optimal (or good) solution which is allowed by the generalization hierarchies. Here an optimal solution is a solution that satisfies the k-anonymity requirement and at the same time maximizes data utility (i.e., minimizes a particular cost metric). On the other hand, the hierarchy-free generalization technique [19] does not require any generalization hierarchy. Instead, the algorithm in this class tries to find an optimal (or good) solution without any restriction of generalization hierarchies from a much larger solution space. Each of these two approaches, however, has significant limitations. In the hierarchy-based generalization, the quality and usefulness of results depends heavily on the choice of the generalization hierarchy, which relies on the user's knowledge of the specific domain [2]. This approach also suffers from relatively high information loss due to unnecessary generalizations. In the hierarchy-free generalization, the existing algorithm requires a total order for each attribute domain, which makes it not applicable to most categorical data. Moreover, the focus is too heavily placed on preserving the quality of the quasi-identifier. As a result, the algorithm is unable to address other issues such as classification and diversity. We further discuss these two classes of generalization techniques in Section 3.

The main goal of our work is to develop a new k-anonymization approach that addresses these limitations. The key idea underlying our approach is that the k-anonymization problem can be viewed as a clustering problem. Intuitively, the k-anonymity requirement can be naturally transformed into a clustering problem where we want to find a set of clusters (i.e., equivalence classes), each of which contains at least k records. In order to maximize data quality, we also want the records in a cluster to be as similar to each other as possible. This ensures that less distortion is required when the records in a cluster are modified to have the same quasi-identifier value. Current clustering

[1] Note that although the concept can be easily broaden, in this paper we consider mainly person-specific data often referred to as micro-data.

techniques, however, are not directly applicable in this context because they do not consider the requirement that each cluster should contain at least k records. We thus formulate a specific clustering problem, which we call *k-member clustering problem*. We prove that this problem is NP-hard and present a greedy algorithm which runs in time $O(n^2)$. Although our approach does not rely on generalization hierarchies, if there exist some natural relations among the values in a domain, our algorithm can incorporate such information to find more desirable solutions. We note that while many quality metrics have been proposed for the hierarchy-based generalization, a metric that precisely measures the information loss introduced by the hierarchy-free generalization has not yet been introduced. For this reason, we define a data quality metric for the hierarchy-free generalization, which we call *information loss metric*. We also show that with a small modification, our algorithm is able to reduce classification errors [15] and eliminate the inference channel arising from lack of diversity in the sensitive attributes [21].

In summary, the key contributions of our work are as follows.

- We model the k-anonymization problem as a new clustering problem, referred to as *k-member clustering problem*, and show that the problem is NP-hard.
- We introduce data quality metrics which naturally capture the effect of data generalization.
- We discuss how to minimize classification errors in the anonymized data and how to eliminate the inference channel arising from lack of diversity.
- We present a greedy k-member clustering algorithm which has time complexity $O(n^2)$. Our experimental results show that our algorithm outperforms existing algorithms with respect to information loss, classification errors, and diversity.

The remainder of this paper is organized as follows. We review the basic concepts of the k-anonymity model in Section 2. We then discuss existing generalization techniques and their limitations in Section 3. We formally define the problem of k-anonymization as a clustering problem and introduce our approach in Section 4. In Section 5, we describe how we extend our approach to address the issues of classification errors and inference channel. Then we evaluate our approach based on the experimental results in Section 6. We survey related work in Section 7 and conclude our discussion in Section 8 with some suggestions for future work.

2. BASIC CONCEPTS

The k-anonymity model assumes the relational data model; that is, data are stored in a table (or a relation) of columns (or attributes) and rows (or records). We assume that the table to be anonymized contains entity-specific information and that each record in the table corresponds to a unique real-world entity (e.g., a person or an organization). Thus, each attribute value in a record represents a piece of information about an entity. The process of anonymizing a table starts with removing all the explicit identifiers, such as name and SSN, from the table. However, even though a table is free of explicit identifiers, some of the remaining attributes in combination could be specific enough to identify individuals if the values are already known to the public. For example, as shown by Sweeney and Samarati [30, 26], most individuals in the United States can be uniquely identified by a set of attributes such as {ZIP, gender, date of birth}. That is, even if each attribute alone may not be specific enough to identify individuals, the combination of a group of attributes may be. When one individual's values of these attributes are known, then one can link a record to that individual. The set of such attributes called *quasi-identifier*.

ZIP	Gender	Age	Diagnosis
47918	Male	35	Cancer
47906	Male	33	HIV+
47918	Male	36	Flu
47916	Female	39	Obesity
47907	Male	33	Cancer
47906	Female	33	Flu

Figure 1: Patient records of a hospital

ZIP	Gender	Age	Diagnosis
4791$*$	Person	[35-39]	Cancer
4790$*$	Person	[30-34]	HIV+
4791$*$	Person	[35-39]	Flu
4791$*$	Person	[35-39]	Obesity
4790$*$	Person	[30-34]	Cancer
4790$*$	Person	[30-34]	Flu

Figure 2: 3-anonymous version of patient records

Definition 1. (**Quasi-identifier**) A quasi-identifier of table $\mathcal{T}$, denoted as $Q_{\mathcal{T}}$, is a set of attributes in $\mathcal{T}$ that can be potentially used to link a record in $\mathcal{T}$ to a real-world identity with a significant probability. □

In addition to the quasi-identifier, the table may contain publicly unknown attributes some of which are highly sensitive. In other words, a table typically consists of three types of attributes: publicly known attributes (i.e., quasi-identifier), sensitive and publicly unknown (simply, sensitive) attributes, and non-sensitive and publicly unknown (simply, non-sensitive) attributes. For instance, consider Figure 1, where the table contains patient records of a fictional hospital. Obviously, *Diagnosis* is an example of sensitive attribute. The quasi-identifier and non-sensitive attributes are determined based on background information such as previous data releases, knowledge of potential adversary, or the content of externally available data. The main objective of the k-anonymity problem is to anonymize a table so that no one can make high-probability associations between records in the table and the corresponding entities.

Definition 2. (**k-anonymity requirement**) Table $\mathcal{T}$ is said to be *k-anonymous* with respect to a quasi-identifier $Q_{\mathcal{T}}$ if and only if each distinct set of values in $Q_{\mathcal{T}}$ appears at least k times in $\mathcal{T}$. □

In other words, any record in a k-anonymous table is indistinguishable from at least $(k-1)$ other records with respect to the *quasi-identifier*. A group of records that are indistinguishable to each other is often referred to as an *equivalence class*. Through the k-anonymity requirement, it is guaranteed that even though an adversary knows that a k-anonymous table $\mathcal{T}$ contains the record of a particular individual and also knows some of the quasi-identifier attribute values of the individual, she cannot determine which record in $\mathcal{T}$ corresponds to the individual with a probability greater than $1/k$. For example, a 3-anonymous version of the table in Figure 1 is shown in Figure 2.

3. HIERARCHY VS. NO-HIERARCHY

The k-anonymity requirement is typically enforced through generalization, where real values are replaced with "less specific but semantically consistent values" [30]. Given a domain, there are various ways to generalize the values in the domain. For instance, ZIP code '47907' can be generalized to '4790$*$' (i.e., '[47900-47909]'), or even (suppressed) to '$*$' (i.e., a range covering every possible ZIP codes). Notice that we consider only the quasi-identifier of a table as the target of generalization. That is, the publicly unknown attributes are not allowed to be altered or omitted in order to make the released table as useful as possible.

Another important technique often used for k-anonymization is tuple suppression. Suppressing a tuple means that the entire tuple is removed from the table. Note that although tuple suppression reduces the number of tuples in the table (thus, results in relatively high information loss), this technique can often enhance the overall data quality. For example, suppose that we want to k-anonymize a table of n tuples with respect to quasi-identifier Q. For the purpose of illustration, let us assume that $n = k + 1$ and that k records have the same quasi-identifier value; that is, there exists one outlier with respect to Q. Without tuple suppression, all n records have to be generalized until the table satisfies the k-anonymity requirement. However, if the outlier is removed from the table, no record needs to be modified at all as the table already satisfies the k-anonymity requirement. Thus, by allowing a limited number of outliers to be suppressed, one can often achieve k-anonymity with much less generalization.

In this paper, we consider only generalization, leaving the support for tuple suppression as part of our future work. In the following sections, we briefly discuss existing algorithms, categorizing them into two classes; *hierarchy-based generalization* and *hierarchy-free generalization*.

3.1 Hierarchy-based generalization

The approaches in this class require that the generalization strategy of each attribute in the quasi-identifier be predefined by a *domain generalization hierarchy* (DGH), which is a totally ordered set of domains that indicates how the corresponding attribute should be generalized in multiple steps. Based on a DGH, one can derive a *value generalization hierarchy* (VGH), which is a partially ordered set of values. Figure 3 illustrates examples of DGH's and VGH's defined for attributes *ZIP*, *gender* and *age*.

The k-anonymization problem is to find a generalization for the entire quasi-identifier, which satisfies the k-anonymity requirement and also maximizes the data quality (or minimizes the desired cost metric). Although finding an optimal solution has been proven to be NP-hard [1, 23], recent research effort has produced many algorithms that efficiently find acceptable k-anonymizations [5, 11, 18, 26, 29, 33]. While most of earlier work on k-anonymization is based on the hierarchy-based generalization, this approach has some significant limitations. First of all, the hierarchy-based generalization suffers from relatively high information loss due to unnecessary generalizations. For instance, assume that we want to 3-anonymize the table in Figure 1 with respect to the quasi-identifier $Q = \{age\}$, using the DGH and VGH given in Figure 3 (e) and (f). Observe that although the records with value 33, which are already 3-anonymous, all have to be generalized to have $[30 - 34]$ as the other records should be generalized to have $[35-39]$. Another, and perhaps more significant, limitation is that the quality and usefulness of anonymized data depends heavily on the choice of generalization hierarchies [2]. As the construction of a domain generalization hierarchy often relies on the user's knowledge of the specific domain, it is difficult to guarantee the usefulness of results.

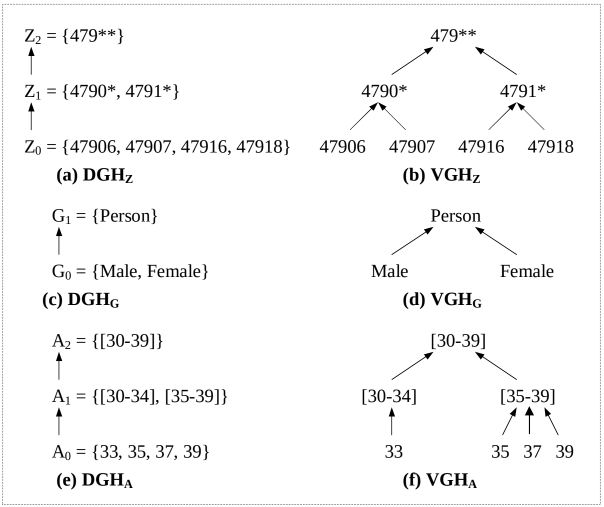


Figure 3: Domain and value generalization hierarchies

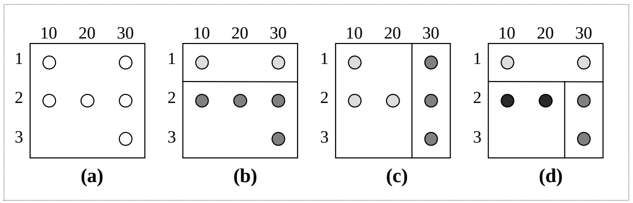


Figure 4: Various partitions of a record space

3.2 Hierarchy-free generalization

Recently, a k-anonymity technique that does not require pre-defined generalization hierarchies has been proposed [19]. In this *multidimensional* approach, the k-anonymity problem is transformed into a partitioning problem. Specifically, the approach consists of the following two steps. The first step is to find a partitioning scheme of the d-dimensional space, where d is the number of attributes in the quasi-identifier, such that each partition contains at least k records. Then the records in each partition are generalized so that they all share the same quasi-identifier value. Clearly the solution space is exponentially large, and as an approximate algorithm, they propose a greedy algorithm that recursively splits a partition at the median value (of a selected dimension) until no more split is allowed with respect to the k-anonymity requirement.

Notice that previously proposed k-anonymization algorithms all assume the single-dimensional generalization where a necessary generalization for each attribute is often separately determined and uniformly applied to its entire domain space. However, in the multidimensional approach, it is possible that the values in a domain may be generalized into different levels. For instance, while two values 31 and 40 may be generalized into an interval $[31 - 40]$, the other values in the same range may be generalized to different intervals or even may not be generalized at all. Although this approach may introduce some inconsistency in the domain, it allows much more flexible generalization strategies. Consider the records in Figure 4 (a), where the quasi-identifier is assumed to consist of two attributes for simplicity. While a single-dimensional approach may produce solutions such as Figure 4 (b) and (c), a multidimensional approach can consider a finer-grained generalization such as Figure 4 (d) as well as (b) and (c).

While this new approach has addressed some of the limitations of the earlier work, the proposed algorithm has some major drawbacks. First, it requires a total order for each attribute domain. This requirement works well for numeric or continuous data, but does not work for most categorical data. We note that this is not a problem that can be dismissed easily, as most person-specific records consist of categorical attributes. For example, in the Adult dataset [25], eight out of fourteen attributes contain categorical domains that do not have any natural ordering. Second, although there may exist natural relations among values in a particular domain, the proposed algorithm cannot incorporate such information to find more desirable solutions. Moreover, the focus is too heavily placed on preserving the quality of the quasi-identifier. As a result, the algorithm lacks the ability to address other issues (e.g., classification and diversity) to find more desirable solutions. Finally, because the proposed algorithm tries to minimize the number of records in each partition, the quality of the resulting anonymized data cannot be guaranteed.

4. ANONYMIZATION AND CLUSTERING

In this section, we present a new k-anonymization approach that is free from the limitations of existing algorithms. The key idea underlying our approach is that the k-anonymization problem can be viewed as a clustering problem. Clustering is the problem of partitioning a set of objects into groups such that objects in the same group are more similar to each other than objects in other groups with respect to some defined similarity criteria [14]. Intuitively, an optimal solution of the k-anonymization problem is indeed a set of equivalence classes such that records in the same equivalence class are very similar to each other, thus requiring a minimum generalization. To be precise, the k-anonymization problem is a specific clustering problem with an additional requirement that each cluster (i.e., equivalence class) contains at least k records.

4.1 k-Anonymization as a clustering problem

Typical clustering problems, such as *k-center* [12] and *k-means* [22], require that a specific number of clusters be found in solutions. However, the k-anonymity problem does not have a constraint on the number of clusters; instead, it requires that each cluster contains at least k records. To the best of our knowledge, this particular constraint has not been addressed in the existing clustering literature. Thus, we pose the k-anonymity problem as a new clustering problem, referred to as *k-member clustering problem*.

Definition 3. (***k*-member clustering problem**) The k-member clustering problem is to find a set of clusters from a given set of n records such that each cluster contains at least k ($k \leq n$) data points and that the sum of all intra-cluster distances is minimized. Formally, let $\mathcal{S}$ be a set of n records and k the specified anonymization parameter. Then the optimal solution of the k-clustering problem is a set of clusters $\mathcal{E} = \{e_1, \ldots, e_m\}$ such that:

1. $\forall\, i \neq j \in \{1, \ldots, m\},\ e_i \cap e_j = \emptyset$,
2. $\bigcup_{i=1,\ldots,m} e_i = \mathcal{S}$,
3. $\forall\, e_i \in \mathcal{E},\ |e_i| \geq k$, and
4. $\sum_{\ell=1,\ldots,m} |e_\ell| \cdot MAX_{i,j\,=\,1,\ldots,|e_\ell|}\, \Delta(p_{(\ell,i)}, p_{(\ell,j)})$ is minimized.

Here $|e|$ is the size of cluster e, $p_{(\ell,i)}$ represents the i-th data point in cluster e_ℓ, and $\Delta(x, y)$ is the distance between two data points x and y. □

Note that in Definition 3, we consider the sum of all intra-cluster distances, where an intra-cluster distance of a cluster is defined as the maximum distance between any two data points in the cluster (i.e., the diameter of the cluster). As we describe in the following section, this sum captures the total information loss, which is the amount of data distortion that generalizations introduce to the entire table.

4.2 Distance and cost metrics

At the heart of every clustering problem are the distance functions that measure the dissimilarities among data points and the cost function which the clustering problem tries to minimize. The distance functions are usually determined by the type of data (i.e., numeric or categorical) being clustered, while the cost function is defined by the specific objective of the clustering problem. In this section, we describe our distance and cost functions which have been specifically tailored for the k-anonymization problem.

As previously discussed, a distance function in a clustering problem measures how dissimilar two data points are. As the data we consider in the k-anonymity problem are person-specific records that typically consist of both numeric and categorical attributes, we need a distance function that can handle both types of data at the same time. In [13, 14], Huang introduces such a measure that uses the *simple matching similarity measure* [16] for categorical attributes. We generalize the approach in [13, 14] by considering the effect of taxonomy trees.

For a numeric attribute, the difference between two values (e.g., $|x - y|$) naturally describes the dissimilarity (i.e., distance) of the values. This measure is also suitable for the k-anonymization problem. To see this, recall that when records in the same equivalence class are generalized, the generalized quasi-identifier must subsume all the attribute values in the equivalence class. That is, the generalization of two values x and y in a numeric attribute is typically represented as a range $[x,\ y]$, provided that $x < y$. Thus, the difference captures the amount of distortion caused by the generalization process to the respective attribute (i.e., the length of the range).

Age	Country	Occupation	Salary	Diagnosis
41	USA	Armed-Forces	≥50K	Cancer
57	India	Tech-support	<50K	Flu
40	Canada	Teacher	<50K	Obesity
38	Iran	Tech-support	<50K	Flu
24	Brazil	Doctor	≥50K	Cancer
45	Greece	Salesman	<50K	Fever

Figure 5: Sample data

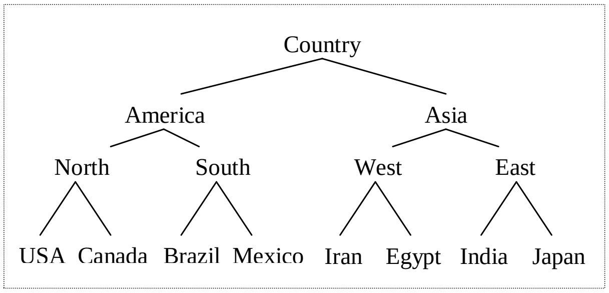


Figure 6: Taxonomy tree of *Country*

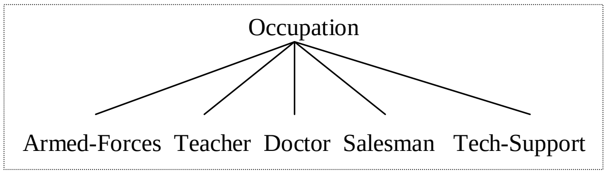


Figure 7: Taxonomy tree of *Occupation*

Definition 4. (**Distance between two numeric values**) Let $\mathcal{D}$ be a finite numeric domain. Then the normalized distance between two values $v_i,\ v_j \in \mathcal{D}$ is defined as:

$$\delta_N(v_1,\ v_2)\ =\ |v_1 - v_2|\ /\ |\mathcal{D}|,$$

where $|\mathcal{D}|$ is the domain size measured by the difference between the maximum and minimum values in $\mathcal{D}$. □

Example 1. Consider the records in Figure 5. The distance contributed by the *Age* attribute for the first two records is $|57 - 41|/|57 - 24| = 0.485$, while the distance between the last two records with respect to the same attribute is $|24 - 45|/|57 - 24| = 0.636$. Note that a small distance between two values implies that the values are similar to each other.

For categorical attributes, however, the difference is no longer applicable as most of the categorical domains cannot be enumerated in any specific order. The most straightforward solution is to assume that every value in such a domain is equally different to each other; e.g., the distance of two values is 0 if they are the same, and 1 if different. However, some domains may have some semantic relationships among the values. In such domains, it is desirable to define the distance functions based on the existing relationships. Such relationships can be easily captured in a *taxonomy tree*. We assume that a taxonomy tree of a domain is a balanced tree of which the leaf nodes represent all the distinct values in the domain. For example, Figure 6 illustrates a natural taxonomy tree for the *Country* attribute. However, for some attributes such as *Occupation*, there may not exist any semantic relationship which can help in classifying the domain values. For such domains, all the values are classified under a common value as in Figure 7. Note that this is equivalent to the straightforward solution where we assume that every value in the domain is equally different to each other. To accommodate all these cases, we define the distance function for categorical values as follows:

Definition 5. (**Distance between two categorical values**) Let $\mathcal{D}$ be a categorical domain and $T_{\mathcal{D}}$ be a taxonomy tree defined for $\mathcal{D}$. The normalized distance between two values $v_i,\ v_j \in \mathcal{D}$ is defined as:

$$\delta_C(v_1,\ v_2)\ =\ H(\Lambda(v_i,\ v_j))\ /\ \ H(T_{\mathcal{D}}),$$

where $\Lambda(x,\ y)$ is the subtree rooted at the lowest common ancestor of x and y, and $H(T)$ represents the height of tree T. □

Example 2. Consider attribute *Country* and its taxonomy tree in Figure 6. The distance between *India* and *USA* is $3/3 = 1$, while the distance between *India* and *Iran* is $2/3 = 0.66$. On the other hand, for attribute *Occupation* and its taxonomy tree in Figure 7 which goes up only one level, the distance between any two values is always 1.

Combining the distance functions for both numeric and categorical domains, we define the distance between two records as follows:

Definition 6. (**Distance between two records**) Let $\mathcal{Q}_{\mathcal{T}} = \{N_1, \ldots, N_m, C_1, \ldots, C_n\}$ be the quasi-identifier of table $\mathcal{T}$, where $N_i (i = 1, \ldots, m)$ is an attribute with a numeric domain and $C_i (i = 1, \ldots, n)$ is an attribute with a categorical domain. The distance of two records $r_1,\ r_2 \in \mathcal{T}$ is defined as:

$$\Delta(r_1, r_2) = \sum_{i=1,\ldots,m} \delta_N(r_1[N_i], r_2[N_i]) + \sum_{j=1,\ldots,n} \delta_C(r_1[C_i], r_2[C_i]),$$

where $r_i[A]$ represents the value of attribute A in r_i, δ_N is the distance function defined in Definition 4, and δ_C is the distance function defined in Definition 5. □

Example 3. Assume that the quasi-identifier of the table in Figure 5 is $\{Age,\ Country,\ Occupation\}$. Then the distance between the first two records is $(16/33) + (3/3) + 1 = 2.485$, while the distance between the first and third record is $(1/33)\ +\ (1/3)\ +\ 1\ =\ 1.363$.

We note that our distance function in Definition 6 can be viewed as a *Manhattan distance* [4]. We also note that one can easily control the contribution of each attribute to the distance by adding a non-negative weight for each attribute as desired.

Now we discuss the cost function which the k-members clustering problem tries to minimize. As the ultimate goal of our clustering problem is the k-anonymization of data, we formulate the cost function to represent the amount of distortion (i.e., information loss) caused by the generalization process. Recall that, records in each cluster are generalized to share the same quasi-identifier value that represents every original quasi-identifier value in the cluster. We assume that the numeric values are generalized into a range [min, max] [19] and categorical values into a set that unions all distinct values in the cluster [5]. With these assumptions, we define a metric, referred to as *Information Loss* metric (IL), that measures the amount of distortion introduced by the generalization process to a cluster.

Definition 7. (**Information loss**) Let $e = \{r_1, \ldots, r_k\}$ be a cluster (i.e., equivalence class) where the quasi-identifier consists of numeric attributes $N_1, \ldots, N_m$ and categorical attributes $C_1, \ldots, C_n$. Let T_{C_i} be the taxonomy tree defined for the domain of categorical attribute C_i. Let MIN_{N_i} and MAX_{N_i} be the min and max values in e with respect to attribute N_i, and let $\cup_{C_i}$ be the union set of values in e with respect to attribute C_i. Then the amount of information loss occurred by generalizing e, denoted by $IL(e)$, is defined as:

$$\begin{aligned} \mathrm{IL}(e) &= |e| \cdot \mathrm{D}(e) \\ \mathrm{D}(e) &= \sum_{i=1,\ldots,m} \frac{(MAX_{N_i} - MIN_{N_i})}{|N_i|} \\ &\quad + \sum_{j=1,\ldots,n} \frac{H(\Lambda(\cup_{C_j}))}{H(T_{C_j})} \end{aligned}$$

where $|e|$ is the number of records in e, $|N|$ represents the size of numeric domain N, $\Lambda(\cup_{C_j})$ is the subtree rooted at the lowest common ancestor of every value in $\cup_{C_j}$, and $H(T)$ is the height of tree T. □

Using the definition above, the total information loss of the anonymized table is defined as follows:

Definition 8. (**Total information loss**) Let $\mathcal{E}$ be the set of all equivalence classes in the anonymized table $\mathcal{AT}$. Then the amount of total information loss of $\mathcal{AT}$ is defined as:

$$\text{Total-IL}(\mathcal{AT}) = \textstyle\sum_{e \in \mathcal{E}} IL(e). \qquad \square$$

We note that like the distance metric, one can easily control the contribution of each attribute to the information loss by adding a non-negative weight for each attribute.

Recall that the cost function of the k-members problem is the sum of all intra-cluster distances, where an intra-cluster distance of a cluster is defined as the maximum distance between any two data points in the cluster. Now, if we consider how records in each cluster are generalized, minimizing the total information loss of the anonymized table intuitively minimizes the cost function for the k-members clustering problem as well. Therefore, the cost function that we want to minimize in the clustering process is Total-IL.

4.3 Anonymization algorithm

Armed with the distance and cost functions, we are now ready to discuss the k-member clustering algorithm. As in most clustering problems, an exhaustive search for an optimal solution of the k-member clustering is potentially exponential.

In order to precisely characterize the computational complexity of the problem, we define the k-member clustering problem as a decision problem.

Definition 9. (k**-member clustering decision problem**) Given n records, is there a clustering scheme $\mathcal{E} = \{e_1, \ldots, e_\ell\}$ such that

1. $|e_i| \geq k$, $1 < k \leq n$: the size of each cluster is greater than or equal to a positive integer k, and
2. $\sum_{i=1,\ldots,\ell} \mathrm{IL}(e_i) < c$, $c > 0$: the Total-IL of the clustering scheme is less than a positive constant c. □

Theorem 1. *The k-member clustering decision problem is NP-complete.*

Proof. That the k-member clustering decision problem is in NP follows from the observation that if such a clustering scheme is given, verifying that it satisfies the two conditions in Definition 9 can be done in polynomial time.

In [1], Aggarwal et al. proved that optimal k-anonymity by suppression is NP-hard, using a reduction from the Edge Partition Into Triangles problem. In the reduction, the table to be k-anonymized consists of n records; each record has m attributes, and each attribute takes a value from $\{0, 1, 2\}$. The k-anonymization technique used is to suppress some cells in the table. Aggarwal et al. showed that determining whether there exists a 3-anonymization of a table by suppressing certain number of cells is NP-hard.

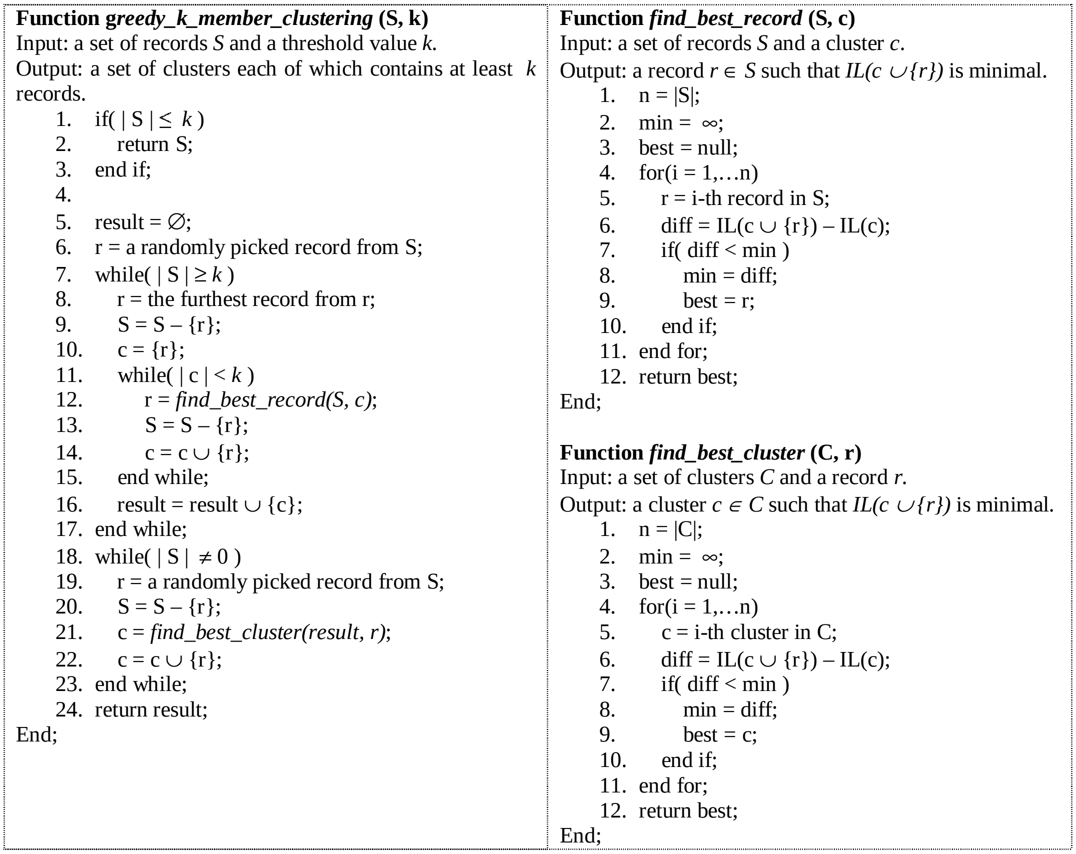
We observe that the problem in [1] is a special case of the k-member clustering problem where each attribute is categorical and has a flat taxonomy tree. It thus follows that the k-member clustering problem is also NP-hard. When each attribute has a flat taxonomy tree, the only way to generalize a cell is to the root of the flat taxonomy tree, and this is equivalent to suppressing the cell. Given such a database, the information loss of each record in any generalization is the same as the number of cells in the record that differ from any other record in the equivalent class, which equals the number of cells to be suppressed. Therefore, there exists a k-anonymization with total information loss no more than t if and and only if there exists a k-anonymization that suppresses at most t cells. □

Faced with the hardness of the problem, we propose a simple and efficient algorithm that finds a solution in a greedy manner. The idea is as follows. Given a set of n records, we first randomly pick a record r_i and make it as a cluster e_1. Then we choose a record r_j that makes $\mathrm{IL}(e_1 \cup \{r_j\})$ minimal. We repeat this until $|e_1| = k$. When $|e_1|$ reaches k, we choose a record that is furthest from r_i and repeat the clustering process until there are less than k records left. We then iterate over these leftover records and insert each record into a cluster with respect to which the increment of the information loss is minimal. We provide the core of our greedy k-member clustering algorithm, leaving out some trivial functions, in Figure 8.

Theorem 2. *Let n be the total number of input records and k be the specified anonymity parameter. Every cluster that the greedy* k*-member clustering algorithm finds has at least k records, but no more than $2k - 1$ records.*

Proof. Let $\mathcal{S}$ be the set of input records. As the algorithm finds a cluster with exactly k records as long as the number of remaining records is equal to or greater than k, every cluster contains at least k records. If there remain less than k records, these leftover records are distributed to the clusters that are already found. That is, in the worst case, $k - 1$ remaining records are added to a single cluster which already contains k records. Therefore, the maximum size of a cluster is $2k-1$. □

Theorem 3. *Let n be the total number of input records and k be the specified anonymity parameter. If $n \gg k$, the*



```
Function greedy_k_member_clustering (S, k)
Input: a set of records S and a threshold value k.
Output: a set of clusters each of which contains at least k records.
    1.  if( | S | ≤ k )
    2.      return S;
    3.  end if;
    4.
    5.  result = ∅;
    6.  r = a randomly picked record from S;
    7.  while( | S | ≥ k )
    8.      r = the furthest record from r;
    9.      S = S – {r};
    10.     c = {r};
    11.     while( | c | < k )
    12.         r = find_best_record(S, c);
    13.         S = S – {r};
    14.         c = c ∪ {r};
    15.     end while;
    16.     result = result ∪ {c};
    17. end while;
    18. while( | S | ≠ 0 )
    19.     r = a randomly picked record from S;
    20.     S = S – {r};
    21.     c = find_best_cluster(result, r);
    22.     c = c ∪ {r};
    23. end while;
    24. return result;
End;
```

```
Function find_best_record (S, c)
Input: a set of records S and a cluster c.
Output: a record r ∈ S such that IL(c ∪{r}) is minimal.
    1.  n = |S|;
    2.  min = ∞;
    3.  best = null;
    4.  for(i = 1,…n)
    5.      r = i-th record in S;
    6.      diff = IL(c ∪ {r}) – IL(c);
    7.      if( diff < min )
    8.          min = diff;
    9.          best = r;
    10.     end if;
    11. end for;
    12. return best;
End;
```

```
Function find_best_cluster (C, r)
Input: a set of clusters C and a record r.
Output: a cluster c ∈ C such that IL(c ∪{r}) is minimal.
    1.  n = |C|;
    2.  min = ∞;
    3.  best = null;
    4.  for(i = 1,…n)
    5.      c = i-th cluster in C;
    6.      diff = IL(c ∪ {r}) – IL(c);
    7.      if( diff < min )
    8.          min = diff;
    9.          best = c;
    10.     end if;
    11. end for;
    12. return best;
End;
```

Figure 8: Greedy k-member clustering algorithm

time complexity of the greedy k-member clustering algorithm is in $O(n^2)$.

Proof. Observe that the algorithm spends most of its time selecting records from the input set S one at a time until it reaches $|S| = k$ (Line 9). As the size of the input set decreases by one at every iteration, the total execution time T is estimated as:

$$T = (n-1) + (n-2) + \ldots + k \approx \frac{n(n-1)}{2},$$

when $k \ll n$. Therefore, T is in $O(n^2)$. □

5. USABILITY AND SECURITY

In this section, we address two critical issues which have not been investigated thoroughly. In most k-anonymity work, the focus is heavily placed on the quasi-identifier, and therefore the other types of attributes (i.e., non-sensitive attributes and sensitive attributes as described in Section 2) are often ignored. However, we believe that these attributes deserve more careful consideration. In fact, the anonymization (i.e., generalization) of quasi-identifier is not to protect the quasi-identifer values as they are assumed to be already known to public. The reason to anonymize the quasi-identifier is thus to protect the sensitive attributes from being inferred by using the uniqueness of quasi-identifier. For instance, in a health record, a person's age or ZIP-code may not be so sensitive, compared to the person's disease. Also, we want to minimize the distortion of quasi-identifier not only because the quasi-identifier itself is meaningful information, but also because a more accurate quasi-identifier will lead to good predictive models (e.g., association rules or classification labels) on non-sensitive attributes. Therefore, the security and the utility of these two types of attributes must be carefully considered when a k-anonymization algorithm is developed. We now describe how our greedy k-member clustering algorithm can be extended to address each of these issues.

5.1 Diversity

Perhaps the information that requires the highest privacy is in the sensitive attributes. However, the security of such information has not received enough attention from most k-anonymity work. In fact, the k-anonymity model is specifically designed to address *identity disclosures* (i.e., re-identification of records) and does not address *attribute disclosures* (i.e., exposure of sensitive data). However, note that an attribute disclosure can occur with or without an identification through various inference channels [17]. For instance, consider a k-anonymized table such that all records in an equivalence class have the same sensitive attribute value. Then although the records may not be re-identified with a significant probability, the sensitive attribute value can be inferred with a probability of 1. Such an inference is referred to as a *homogeneity attack* in [21]. This problem can be serious especially when there exists a strong correlation between the quasi-identifier and the sensitive attributes. The k-anonymization algorithms group records with similar quasi-identifier values together in order to minimize the amount of generalization, and as a result, records with similar sensitive attribute values are indeed grouped together.

Recently, Machanavajjhala et al. [21] pointed out such inference issues in the k-anonymity model and proposed the notion of *ℓ-diversity*, which requires records in every equivalence class to have at least ℓ distinct sensitive attribute values. Building upon this idea, we develop two versions of diversity metrics which measure the total number of records in a k-anonymized table that are vulnerable to inference. In the first version, called the *Equal Diversity* metric (ED), we assume that all sensitive attribute values are equally sensitive. Now, if every record in an equivalence class has the same sensitive attribute value, then all records in that class equally contribute to the cost of the ED metric. In the second version, called the *Sensitive Diversity* metric (SD), we assume that there are two types of values in a sensitive attribute; those that are truly-sensitive and the others not-so-sensitive. This may sound odd at first, but we believe this assumption to be realistic in many cases. For example, consider medical records which contain the name of each patient's disease. Although some diseases (e.g., HIV+, cancer, and STD) may be highly sensitive, some common diseases (e.g., flu and itchy eye) may not be so sensitive. With this assumption, we can relax the previous metric as follows. If every record in an equivalence class has the same sensitive attribute value and the value is truly-sensitive, then all these records contribute to the cost of the SD metric. In other words, the SD metric does not penalize the records with not-so-sensitive values although the values can be inferred. We now formally define our diversity metrics as follows.

Definition 10. (**Diversity Metrics**) Let $\mathcal{AT}$ be a k-anonymized table which contains a sensitive attribute s. Let $\mathcal{E} = \{e_1, \ldots, e_m\}$ represent the set of all equivalence classes in $\mathcal{AT}$. Then the Equal Diversity cost for $\mathcal{AT}$ is defined as:

$$ED(\mathcal{AT}, s) = \sum_{i=1,\ldots,m} \phi(e_i, s) \times |e_i|,$$

where $\phi(e, s) = 1$ if every record in e has the same s value; $\phi(e, s) = 0$, otherwise.
Provided that only a subset of s values is truly-sensitive, the Sensitive Diversity cost of $\mathcal{AT}$ is defined as:

$$SD(\mathcal{AT}, s) = \sum_{i=1,\ldots,m} \psi(e_i, s) \times |e_i|,$$

where $\psi(e, s) = 1$ if every record in e has the same s value that is truly-sensitive; $\psi(e, s) = 0$, otherwise. □

In order to address the issue of diversity, our greedy algorithm is modified by replacing Line 6 of Function *find_best_record* with the following.

```
For equal diversity:
  if (isDiverse(c) == true)
    diff = $IL(\{c \cup \{r\}) - IL(c)$;
  else if (majority-class(c) $\neq$ class(r))
    diff = $IL(\{c \cup \{r\}) - IL(c)$ + diversityPenalty;

For sensitive diversity:
  if (isDiverse(c) == true)
    diff = $IL(\{c \cup \{r\}) - IL(c)$;
  else if (majority-class(c) is sensitive &&
          majority-class(c) $\neq$ class(r))
    diff = $IL(\{c \cup \{r\}) - IL(c)$ + diversityPenalty;
```

The underlying idea is as follows. When choosing a record to insert into a cluster, the modified algorithm explicitly penalizes records that have the same sensitive value as the majority of the records already in the cluster; thus resulting in a diversified cluster. Observe that the magnitude of enforcement can be controlled by the weight of penalty. We demonstrate the effectiveness of this modification in Section 6.

5.2 Classification

The predictive modeling of attributes is well motivated by Iyengar [15]. In fact, the correlation between the quasi-identifier and non-sensitive attributes can be significantly weakened or perturbed due to the ambiguity introduced by the generalization of quasi-identifier. Thus, it is critical that the generalization process does not significantly weaken the discrimination of classes using quasi-identifier. Considering this issue, Iyengar also proposed the *classification* metric (CM) as:

$$CM = \textstyle\sum_{all\ rows} Penalty(row\ r)\ /\ N,$$

where N is the total number of records, and $Penalty(row\ r) = 1$ if r is suppressed or the class label of r is different from the class label of the majority in the equivalence group.

Motivated by this metric, we modify our algorithm in Figure 8 by replacing Line 6 of Function *find_best_record* with the following.

```
if (majority-class-label(c) == class-label(r))
  diff = $IL(\{c \cup \{r\}) - IL(c)$;
else diff = $IL(\{c \cup \{r\}) - IL(c)$ + classPenalty;
```

In essence, the algorithm is now forced to choose records with the same class label for a cluster, and the magnitude of enforcement is controlled by the weight of penalty. With this minor modification, our algorithm can effectively reduce the cost of classification metric without increasing much information loss. We show the result in Section 6.

6. EXPERIMENTAL RESULTS

The main goal of the experiments is to investigate the performance of our approach in terms of data quality, efficiency, and scalability. To accurately evaluate our approach, we also compare our implementation with two other algorithms, namely the *median partitioning algorithm* as described in the Mondrian approach [19] and the *k-nearest neighbor algorithm* used in the Condensation approach [2]. We note that the Condensation approach is different from most k-anonymity work in that it does not use data generalization. Instead, each equivalence class (referred to as k-indistinguishable group in [2]) is represented by comprehensive statistics. However, for our experiment, we devise their approach (which finds k-indistinguishable groups by repeatedly (randomly) selecting a record and making it a group with $k - 1$ nearest neighbors) as a clustering algorithm and compare it with our approach.

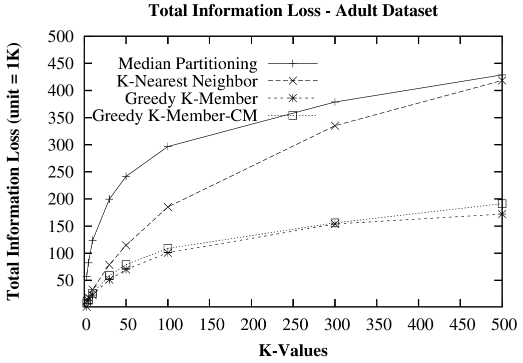


Figure 9: Information Loss Metric

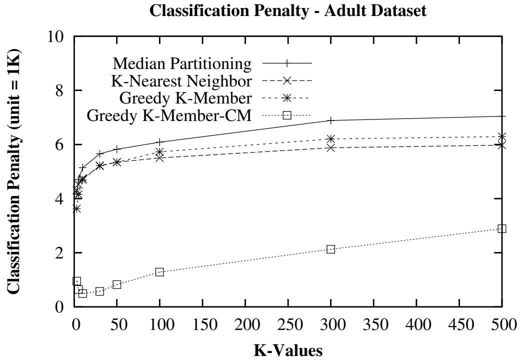


Figure 11: Classification Metric

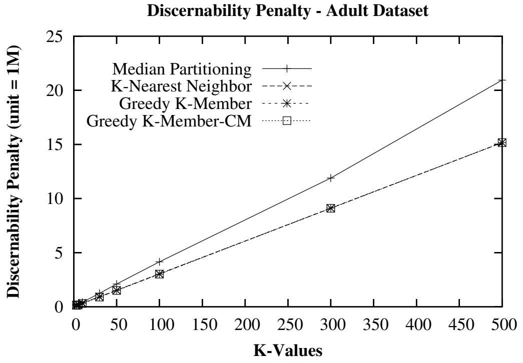


Figure 10: Discernibility Metric

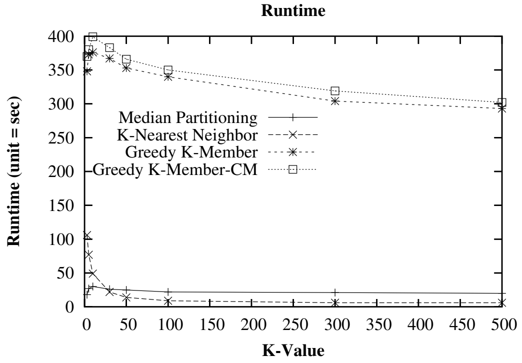


Figure 12: Execution Time

6.1 Experimental setup

The experiments were performed on a 2.66 GHz Intel *IV* processor machine with 1 GB of RAM. The operating system on the machine was Microsoft Windows XP Professional Edition, and the implementation was built and run in Java 2 Platform, Standard Edition 5.0.

For our experiments, we used the Adult dataset from the UC Irvine Machine Learning Repository [25], which is considered a de facto benchmark for evaluating the performance of k-anonymity algorithms. Before the experiments, the Adult data set was prepared as described in [5, 15, 19]. We removed records with missing values and retained only nine of the original attributes. For k-anonymization, we considered (*age*, *work class*, *education*, *marital status*, *occupation*, *race*, *gender*, and *native country*) as the quasi-identifier. Among these, *age* and *education* were treated as numeric attributes while the other six attributes were treated as categorical attributes. In addition to that, we also retained the *salary class* attribute to evaluate the classification and diversity metrics.

6.2 Data quality and efficiency

In this section, we report experimental results on the greedy k-members algorithm for data quality and execution efficiency.

Figure 9 reports the Total-IL costs of the four algorithms (median partitioning, k-nearest neighbor, greedy k-member, and greedy k-member modified to reduce classification error) for increasing values of k. As the figure illustrates, the greedy k-members algorithm results in the least cost of the Total-IL for all k values. Note also that the Total-IL cost of the modified greedy k-member is very close to the cost of the unmodified algorithm. The superiority of our algorithms over the median partitioning algorithm results from the fact that the median partitioning algorithm considers the proximity among the data points only with respect to a single dimension at each partitioning. Another interesting result from the experiment is that the greedy k-member algorithm results in much lower Total-IL costs than the k-nearest neighbor approach. This difference can be explained with the following reasons. First, the greedy k-member algorithm selects each center point using the furthest-first traversal approach [12] while the k-nearest neighbor algorithm selects center points randomly. Secondly, while the k-nearest neighbor algorithm selects $k-1$ records that are nearest to the center point, the greedy k-member algorithm selects $k-1$ records, one at a time, that are closest to the cluster.

Another metric used to measure the data quality is the *Discernibility* metric (DM) [5]. This metric measures the data quality based on the size of each equivalence class and is defined as follows:

$$DM = \sum_{e \in E} |e|^2 \ + \ \sum_{s \in S} |D||s|,$$

where E represents the set of equivalence classes, S the set of suppressed records, and D the input dataset. Intuitively data quality diminishes as more records become indistinguishable with respect to each other, and DM effectively captures this effect of the k-anonymization process.

Figure 10 shows the DM costs of the four algorithms for increasing k values. As shown, the two greedy k-member algorithms and the k-nearest neighbor algorithm perform better than the median partitioning algorithm. In fact, the greedy k-member and the k-nearest neighbor algorithms always produce equivalence classes with sizes very close to the

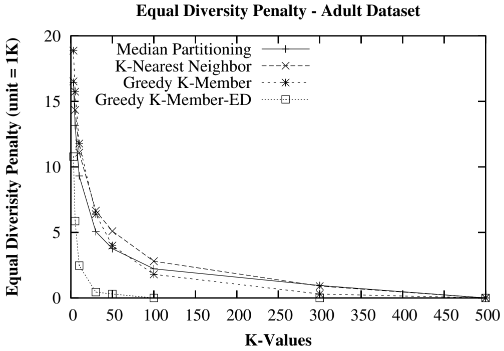


Figure 13: Equal Diversity Metric

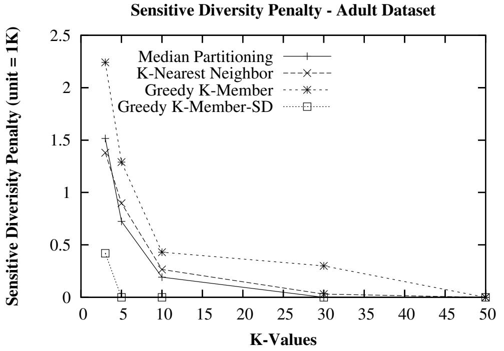


Figure 14: Sensitive Diversity Metric

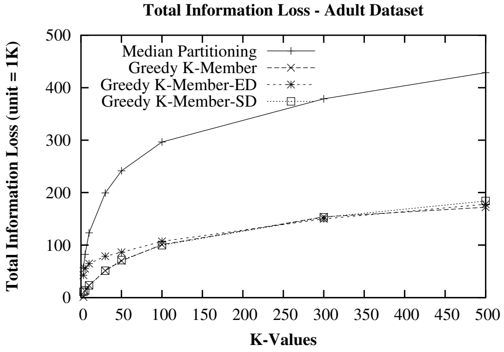


Figure 15: Diversity Algorithms and Information Loss

specified k, due to the way clusters are formed.

Figure 11 shows the experimental result with respect to the CM metric. As expected, the greedy k-member algorithm modified to minimize classification errors (as described in Section 5.2) outperforms all the other algorithms. Observe that even without the modification, the greedy k-members algorithm still produces less classification errors than the median partitioning for every k value.

We also measured the execution time of the algorithms for different k values. The results are shown in Figure 12. Even though the execution time for the greedy k-member algorithm is higher than the other two algorithms, we believe that it is still acceptable in practice[2].

6.3 Diversity

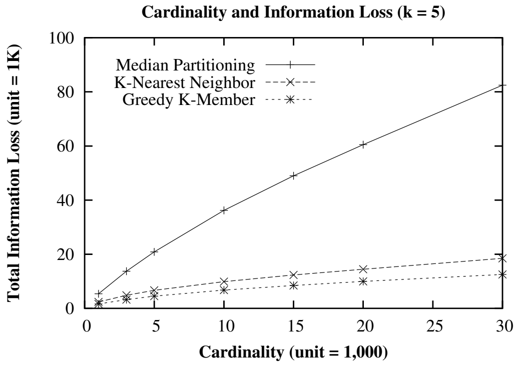


Figure 16: Cardinality and Information Loss

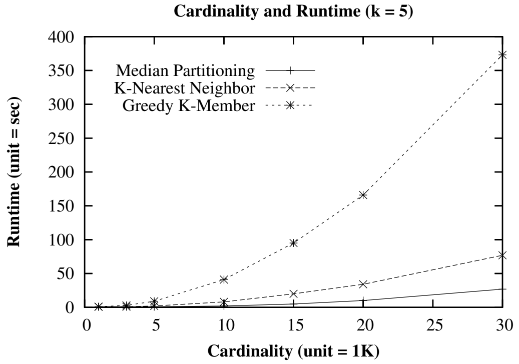


Figure 17: Cardinality and Runtime

In this section, we present experimental results for the greedy k-member algorithm modified to enhance the diversity of sensitive attribute values as described in Section 5.1. Recall that in the ED metric, if every record in an equivalence class has the same sensitive attribute value, then all records in that class equally contribute to the cost of the metric. The SD metric is more lenient in that if every record in an equivalence class has the same sensitive attribute value and the value is truly-sensitive, then all these records contribute to the cost of the metric. We considered class label ">50" as the truly-sensitive value in our experiment.

Figure 13 illustrates the performance of the algorithms with respect to the ED metric. The result is rather surprising. For small k values (under 10), the sensitive attribute values of nearly a half of anonymized records can be easily inferred for the median partitioning, the k-nearest neighbor, and the greedy k-member algorithms. Moreover, a significant number of records still remain vulnerable for higher k values. On the other hand, the greedy k-member algorithm modified for the ED metric reduces this vulnerability significantly even for small k values. Our experimental result for the SD metric is shown in Figure 14, which reports similar trends. As shown in Figure 15, the Total-IL cost is still low even when the algorithm is modified for either type of diversity metrics. Although we do not provide a figure, the modifications for diversity do not introduce any considerable overheads to the execution-time behavior.

6.4 Scalability

Figures 16 and 17 show the Total-IL costs and execution-time behaviors of the algorithms for various table cardinal-

[2] The k-anonymity model assumes static data, which means that k-anonymization is an off-line procedure.

ities[3] (for a fixed k value of 5). As shown, the Total-IL costs increase almost linearly with the size of the dataset for all the three algorithms. However, the greedy k-member algorithm introduces the least Total-IL cost for any size of dataset.

As discussed in Section 4, the time complexity of the greedy k-member algorithm is quadratic with respect to the number of records in the dataset, which is clearly shown in Figure 17. We have also analyzed the theoretical time complexities of the other two algorithms. A brief report of our analysis is provided below.

Algorithm	Time Complexity
Median partitioning	$O(nlog(n))$
k-nearest neighbor	$O(n^2)$
Greedy k-members	$O(n^2)$

Although the greedy k-members is slower than the other two algorithms, the overhead is still acceptable in most cases considering its better performance with respect to the Total-IL metric. In addition to that, we are encouraged by the fact that the current version of our algorithm is not fully optimized. This promises that the runtime behavior of our algorithm can be improved after optimization.

7. RELATED WORK

In this section, we briefly survey existing literature that addresses data privacy. Instead of providing a comprehensive survey, we discuss various aspects of data privacy. Note that we do not include the k-anonymity work here as detailed discussion can be found in Section 2.

Ensuring privacy in published data has been a difficult problem for a long time, and this problem has been studied in various aspects. In [17], Lambert provides informative discussion on the risk and harm of undesirable disclosures and discusses how to evaluate a dataset in terms of these risk and harm. In [6], Dalenius poses the problem of re-identification in (supposedly) anonymous census records and firstly introduces the notion of "quasi-identifier". He also suggests some ideas such as suppression or encryption of data as possible solutions.

Data privacy has been extensively addressed in statistical databases, which primarily aim at preventing various inference channels. One of the common techniques is data perturbation [20, 24, 31], which mostly involves swapping data values or introducing noise to the dataset. While the perturbation is applied in a manner which preserves statistical characteristics of the original data, the transformed dataset is useful only for statistical research. Another important technique is query restriction [7, 9], which restricts queries that may result in inference. In this approach, queries are restricted by various criteria such as query-set-size, query-history, and partitions. Although this approach can be effective, it requires the protected data to remain in a dedicated database at all time.

Today's powerful data mining techniques [8, 10, 27] are often considered great threats to data privacy. However, we have recently seen many privacy-preserving data mining techniques being developed. For instance, Evfimievski et al. in [3] propose an algorithm which randomizes data to prevent association rule mining [28]. There has also been much work done addressing privacy-preserving information sharing [32, 3], where the main concern is the privacy of databases rather than data subjects.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an efficient k-anonymization algorithm by transforming the k-anonymity problem to the k-member clustering problem. We also proposed two important elements of clustering, that is, distance and cost functions, which are specifically tailored for the k-anonymization problem. We emphasize that our cost metric, IL metric, naturally captures the data distortion introduced by the generalization process and is general enough to be used as a data quality metric for any k-anonymized dataset. We also proposed two diversity metrics, ED and SD metrics, which measure the level of security a k-anonymized dataset provides. Although the issue of diversity has been overlooked in most k-anonymity work, we believe that this issue deserves more careful consideration.

Our future work includes the following. Although our experiments show that our greedy algorithm produces "good" results, the algorithm does not guarantee any approximation factor. It is important to investigate this issue in order to improve the current algorithm. We are also working on the dynamic aspect of k-anonymity. So far, the k-anonymity problem has assumed that the data is static. While this may have been acceptable before, today we see the databases continuously growing everyday and every hour, and there is a strong demand for up-to-date data. Thus, we believe that mechanisms that guarantee the privacy in dynamic databases is a critical issue.

9. REFERENCES

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *Proceedings of the Tenth International Conference on Database Theory (ICDT)*, pages 246–258, 2005.

[2] C. C. Aggrawal and P. S. Yu. A condensation approach to privacy preserving data mining. In *the ninth International Conference on Extending Database Technology (EDBT)*, 2004.

[3] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *ACM International Conference on Management of Data (SIGMOD)*, 2003.

[4] M. R. Anderberg. *Cluster Analysis for Applications.* Academic Press, 1973.

[5] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *the twenty-first International Conference on Data Engineering (ICDE)*, 2005.

[6] T. Dalenius. Finding a needle in a haystack. *Journal of Official Statistics*, 2, 1986.

[7] D. Dobkin, A. K. Jones, and R. J. Lipton. Secure databases: Protection against user influence. *ACM Transactions on Database systems (TODS)*, 4, 1979.

[8] X. Dong, A. Halevy, J. Madhavan, and E. Nemes. Reference reconciliation in complex information

[3] For this experiment, we used the subsets of the Adult dataset with different sizes.

spaces. In *ACM International Conference on Management of Data (SIGMOD)*, 2005.

[9] I. P. Fellegi. On the question of statistical confidentiality. *Journal of the American Statistical Association*, 1972.

[10] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 1969.

[11] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *the twenty-first International Conference on Data Engineering (ICDE)*, 2005.

[12] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[13] Z. Huang. A fast clustering algorithm to cluster very large categorical values. In *the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 1–8, 1997.

[14] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(2):283–304, 1998.

[15] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *ACM Conference on Knowledge Discovery and Data mining (SIGKIDD)*, 2002.

[16] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: An introduction to cluster analysis*. A Wiley and Sons, Inc., 1990.

[17] D. Lambert. Measures of disclosure risk and harm. *Journal of Official Statistics*, 9, 1993.

[18] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *ACM International Conference on Management of Data (SIGMOD)*, 2005.

[19] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *To appear in the 22nd International Conference on Data Engineering (ICDE)*, 2006.

[20] C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems (TODS)*, 10, 1985.

[21] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *To appear in the 22nd International Conference on Data Engineering (ICDE)*, 2005.

[22] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *the fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[23] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *the twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2004.

[24] S. P. Reiss. Practical data-swapping: The first steps. *ACM Transactions on Database Systems (TODS)*, 9, 1980.

[25] C. B. S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.

[26] P. Samarati. Protecting respondent's privacy in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13, 2001.

[27] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2002.

[28] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *ACM International Conference on Management of Data (SIGMOD)*, 1996.

[29] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.

[30] L. Sweeney. K-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.

[31] J. F. Traub and Y. Y. H. Wozniakowski. The statistical security of statistical database. *ACM Transactions on Database Systems (TODS)*, 9, 1984.

[32] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[33] S. Zhong, Z. Yang, and R. N. Wright. Privacy-enhancing k-anonymization of customer data. In *the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2005.