Efficient k-Anonymization Using Clustering Techniques*

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

[1] CERIAS and Computer Science, Purdue University
{byunj, bertino, ninghui}@cs.purdue.edu
[2] CERIAS and Electrical and Computer Engineering, Purdue University
akamra@purdue.edu

Abstract. k-anonymization techniques 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. In this paper we propose an approach that 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 similar to each other should be part of the same equivalence class. 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 heuristic, 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.

1 Introduction

A recent approach addressing data privacy relies on the notion of *k-anonymity* [11,13]. 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 [1,9]. The k-anonymity problem has recently drawn considerable interest from research community, and a number of algorithms have been proposed [3,4,6,7,8,12]. Current solutions, however, suffer from high information loss mainly due to reliance on pre-defined generalization hierarchies [4,6,7,12] or total order [3,8] imposed on each attribute domain. We discuss these algorithms more in detail in Section 2.

The main goal of our work is to develop a new k-anonymization approach that addresses such 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

* This material is based upon work supported by the National Science Foundation under Grant No. 0430274 and the sponsors of CERIAS.

R. Kotagiri et al. (Eds.): DASFAA 2007, LNCS 4443, pp. 188–200, 2007.

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. 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 effectively.

The remainder of this paper is organized as follows. We review the basic concepts of the k-anonymity model and survey existing techniques in Section 2. We formally define the problem of k-anonymization as a clustering problem and introduce our approach in Section 3. Then we evaluate our approach based on the experimental results in Section 4. We conclude our discussion in Section 5.

2 Preliminaries

2.1 Basic Concepts

The k-anonymity model assumes that person-specific data are stored in a table (or a relation) of columns (or attributes) and rows (or records). The process of anonymizing such 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 [13], most individuals in the United States can be uniquely identified by a set of attributes such as {ZIP, gender, date of birth}. Thus, even if each attribute alone is not specific enough to identify individuals, a group of certain attributes together may identify a particular individual. The set of such attributes is called *quasi-identifier*.

The main objective of the k-anonymity model is thus to transform a table so that no one can make high-probability associations between records in the table and the corresponding entities. In order to achieve this goal, the k-anonymity model requires that any record in a table be indistinguishable from at least $(k-1)$ other records with respect to the pre-determined quasi-identifier. A group of records that are indistinguishable to each other is often referred to as an *equivalence class*. By enforcing the k-anonymity requirement, it is guaranteed that even though an adversary knows that a k-anonymous table contains the record of a particular individual and also knows some of the 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

Fig. 1. Patient Table

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

Fig. 2. 3-anonymous Patient table

attribute values of the individual, he/she cannot determine which record in the table corresponds to the individual with a probability greater than $1/k$. For example, a 3-anonymous version of the table in Fig. 1 is shown in Fig. 2.

2.2 Existing Techniques

The k-anonymity requirement is typically enforced through *generalization*, where real values are replaced with "less specific but semantically consistent values" [13]. Given a domain, there are various ways to generalize the values in the domain. Typically, numeric values are generalized into intervals (e.g., [12−19]), and categorical values are generalized into a set of distinct values (e.g., {USA, Canada}) or a single value that represents such a set (e.g., North-America).

Various generalization strategies have been proposed. In [7,11,12], a non-overlapping generalization-hierarchy is first defined for each attribute of quasi-identifier. Then an algorithm tries to find an optimal (or good) solution which is allowed by such generalization hierarchies. Note that in these schemes, if a lower level domain needs to be generalized to a higher level domain, all the values in the lower domain are generalized to the higher domain. This restriction could be a significant drawback in that it may lead to relatively high data distortion due to unnecessary generalization. The algorithms in [4,6], on the other hand, allow values from different domain levels to be combined to represent a generalization. Although this leads to much more flexible generalization, possible generalizations are still limited by the imposed generalization hierarchies.

Recently, some schemes that do not rely on generalization hierarchies [3,8] have been proposed. For instance, LeFevre et al. [8] transform the k-anonymity problem into a partitioning problem. Specifically, their approach consists of the following two steps. The first step is to find a partitioning 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. Although shown to be efficient, these approaches also have a disadvantage that it requires a total order for each attribute domain. This makes it impractical in most cases involving categorical data which have no meaningful order.

3 Anonymization and Clustering

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 [5]. 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.

3.1 k-Anonymization as a Clustering Problem

Typical clustering problems 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. Thus, we pose the k-anonymity problem as a clustering problem, referred to as *k-member clustering problem*.

Definition 1. (***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 1, 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.

3.2 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.

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).

Definition 2. (**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 $\mathcal{D}$. □

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* [1]. 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, Fig. 3 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 Fig. 4. We now define the distance function for categorical values as follows:

Definition 3. (**Distance between two categorical values**) Let $\mathcal{D}$ be a categorical domain and $\mathcal{T}_\mathcal{D}$ be a taxonomy tree defined for $\mathcal{D}$. The normalized distance between two values $v_i, v_j \in \mathcal{D}$ is defined as:

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

[1] Taxonomy tree can be considered similar to generalization hierarchy introduced in [7,11,12]. However, we treat taxonomy tree not as a restriction, but a user's preference.

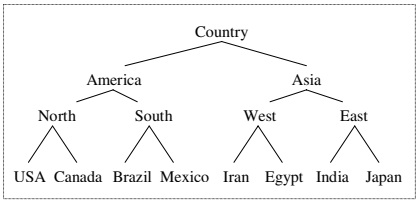

Fig. 3. Taxonomy tree of *Country*

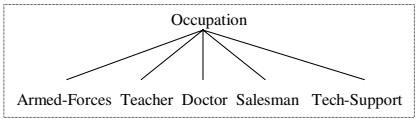

Fig. 4. Taxonomy tree of *Occupation*

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

Example 1. Consider attribute *Country* and its taxonomy tree in Fig. 3. 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 Fig. 4 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 4. (**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 T, where $N_i (i = 1, \ldots, m)$ is an attribute with a numeric domain and $C_j (j = 1, \ldots, n)$ is an attribute with a categorical domain. The distance of two records $r_1,\ r_2 \in T$ is defined as:

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

where $r_i[A]$ represents the value of attribute A in r_i, and δ_N and δ_C are the distance functions defined in Definitions 2 and 3, respectively. □

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] [8] and categorical values into a set that unions all distinct values in the cluster [3]. 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 5. (**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 $\mathcal{T}_{C_i}$ 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:

$$\mathrm{IL}(e) = |e| \cdot \Big(\sum_{i=1,\ldots,m} \frac{(MAX_{N_i} - MIN_{N_i})}{|N_i|} + \sum_{j=1,\ldots,n} \frac{H(\Lambda(\cup_{C_j}))}{H(\mathcal{T}_{C_j})} \Big)$$

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 $H(\mathcal{T})$ is the height of taxonomy tree $\mathcal{T}$. □

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

Definition 6. (**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:

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

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.

3.3 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 as follows.

Definition 7. (**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 7 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 5.

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

```
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.  result = ∅; r = a randomly picked record from S;
5.  while( | S | ≥ k )
6.      r = the furthest record from r;
7.      S = S – {r};
8.      c = {r};
9.      while( | c | < k )
10.         r = find_best_record(S, c);
11.         S = S – {r};
12.         c = c ∪ {r};
13.     end while;
14.     result = result ∪ {c};
15. end while;
16. while( | S | ≠ 0 )
17.     r = a randomly picked record from S;
18.     S = S – {r};
19.     c = find_best_cluster(result, r);
20.     c = c ∪ {r};
21. end while;
22. 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|; min = ∞; best = null;
2.  for(i = 1,…n)
3.      r = i-th record in S;
4.      diff = IL(c ∪ {r}) – IL(c);
5.      if( diff < min )
6.          min = diff;
7.          best = r;
8.      end if;
9.  end for;
10. 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|; min = ∞; best = null;
2.  for(i = 1,…n)
3.      c = i-th cluster in C;
4.      diff = IL(c ∪ {r}) – IL(c);
5.      if( diff < min )
6.          min = diff;
7.          best = c;
8.      end if;
9.  end for;
10. return best;
End;
```

Fig. 5. Greedy k-member clustering algorithm

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:

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

Therefore, T is in $O(n^2)$. □

3.4 Improvement for Classification

In most k-anonymity work, the focus is heavily placed on the quasi-identifier, and therefore other attributes are often ignored. However, these attributes deserve more careful consideration. In fact, 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 on the transformed table [6]. In fact, the correlation between the quasi-identifier and other attributes can be significantly weakened or perturbed due to the ambiguity introduced by the generalization of the quasi-identifier. Thus, it is critical that the generalization process does preserve the discrimination of classes using quasi-identifier. Considering this issue, Iyengar also proposed the *classification* metric (CM) as:

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

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.

Inspired by this metric, we modify our algorithm in Figure 5 by replacing Line 4 of Function *find_best_record* with the following.

```
if (majority-class-label(c) == class-label(r))
    diff = IL({c ∪ {r}) − IL(c);
else diff = IL({c ∪{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 results in Section 4.

4 Experimental Results

The main goal of the experiments was to investigate the performance of our approach in terms of data quality, efficiency, and scalability. To accurately evaluate our approach, we also compared our implementation with another algorithm, namely the *median partitioning algorithm* proposed in [8].

4.1 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 [10], 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 [3,6,8]. 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 metric.

4.2 Data Quality and Efficiency

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

Fig. 6 reports the Total-IL costs of the three algorithms (median partitioning, greedy k-member, and greedy k-member modified to reduce classification error)

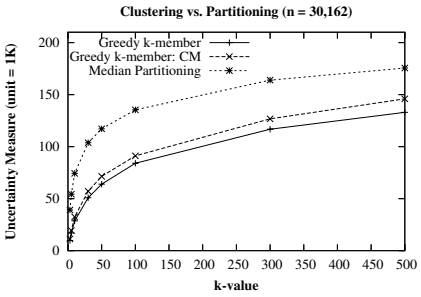

Fig. 6. Information Loss Metric

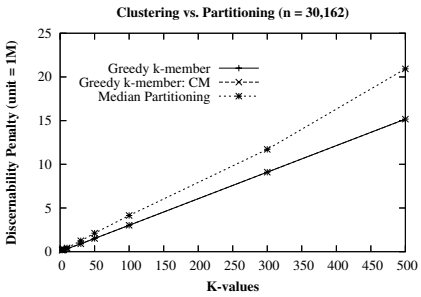

Fig. 7. Discernibility Metric

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 metric used to measure the data quality is the *Discernibility* metric (DM) [3], which measures the data quality based on the size of each equivalence class. 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. Fig. 7 shows the DM costs of the three algorithms for increasing k values. As shown, the two greedy k-member algorithms perform better than the median partitioning algorithm. In fact, the greedy k-member algorithms always produce equivalence classes with sizes very close to the specified k, due to the way clusters are formed.

Fig. 8 shows the experimental result with respect to the CM metric described in Section 3. As expected, the greedy k-member algorithm modified to minimize classification errors (as described in Section 3) 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 Fig. 9. Even though the execution time for the greedy k-member algorithm is higher than the partitioning algorithm, we believe that it is still acceptable in practice as k-anonymization is often considered an off-line procedure.

4.3 Scalability

Fig. 10 and 11 show the Total-IL costs and execution-time behaviors of the algorithms for various table cardinalities (for $k = 5$). For this experiment, we used the subsets of the Adult dataset with different sizes. As shown, the Total-IL costs increase almost linearly with the size of the dataset for both algorithms. However, the greedy k-member algorithm introduces the least Total-IL cost for

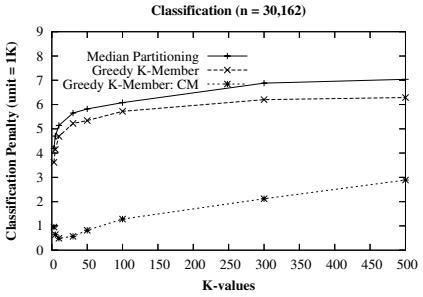

Fig. 8. Classification Metric

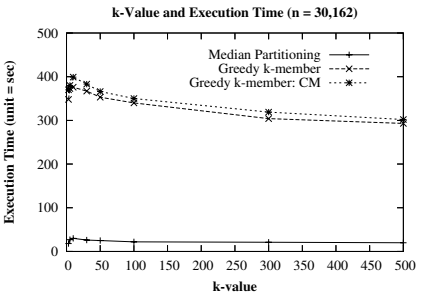

Fig. 9. Execution Time

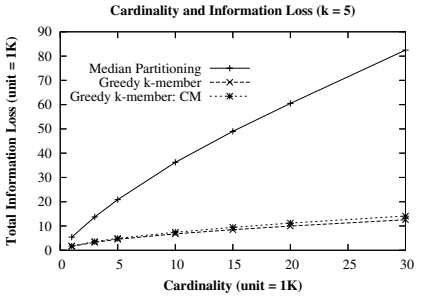

Fig. 10. Cardinality and Information Loss

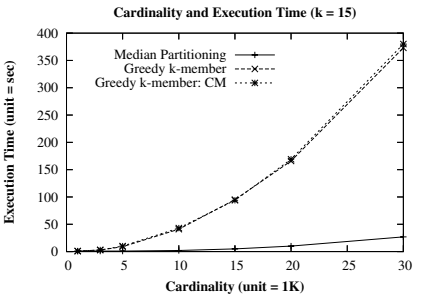

Fig. 11. Cardinality and Runtime

any size of dataset. Although the greedy k-members is slower than the partitioning algorithm, we believe that the overhead is still acceptable in most cases considering its better performance with respect to the Total-IL metric.

5 Conclusions

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.

References

1. G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *International Conference on Database Theory*, pages 246–258, 2005.
2. C. C. Aggrawal and P. S. Yu. A condensation approach to privacy preserving data mining. In *International Conference on Extending Database Technology*, 2004.

3. R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *International Conference on Data Engineering*, 2005.
4. B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *International Conference on Data Engineering*, 2005.
5. 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.
6. V. S. Iyengar. Transforming data to satisfy privacy constraints. In *ACM Conference on Knowledge Discovery and Data mining*, 2002.
7. K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *ACM International Conference on Management of Data*, 2005.
8. K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *International Conference on Data Engineering*, 2006.
9. A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *ACM Symposium on Principles of Database Systems*, 2004.
10. C. B. S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.
11. P. Samarati. Protecting respondent's privacy in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13, 2001.
12. L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.
13. L. Sweeney. K-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.