(α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy-Preserving Data Publishing

Raymond Chi-Wing Wong[*], Jiuyong Li[+], Ada Wai-Chee Fu[*] and Ke Wang[♯]

[*]Department of Computer Science and Engineering
The Chinese University of Hong Kong
cwwong,adafu@cse.cuhk.edu.hk

[+]Departtment of Mathematics and Computing
The University of Southern Queensland
Jiuyong.Li@usq.edu.au

[♯]Department of Computer Science
Simon Fraser University, Canada
wangk@cs.sfu.ca

ABSTRACT

Privacy preservation is an important issue in the release of data for mining purposes. The k-anonymity model has been introduced for protecting individual identification. Recent studies show that a more sophisticated model is necessary to protect the association of individuals to sensitive information. In this paper, we propose an (α, k)-anonymity model to protect both identifications and relationships to sensitive information in data. We discuss the properties of (α, k)-anonymity model. We prove that the optimal (α, k)-anonymity problem is NP-hard. We first present an optimal global-recoding method for the (α, k)-anonymity problem. Next we propose a local-recoding algorithm which is more scalable and result in less data distortion. The effectiveness and efficiency are shown by experiments. We also describe how the model can be extended to more general cases.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining; K.4.1 [**Public Policy Issues**]: Privacy

General Terms

Algorithms, Theory, Performance, Experimentation

Keywords

anonymity, privacy preservation, data publishing, data mining H.2

1. INTRODUCTION

Privacy preservation has become a major issue in many data mining applications.When a data set is released to other parties for data mining, some privacy-preserving technique is often required to reduce the possibility of identifying sensitive information about individuals. This is called the disclosure-control problem [4] in statistics and has been studied for many years. Most statistical solutions concern more about maintaining statistical invariant of data. The data mining community has been studying this problem aiming at building strong privacy-preserving models and designing efficient optimal and scalable heuristic solutions. The perturbing method [2] and the k-anonymity model [11, 10] are two major techniques for this goal. The k-anonymity model has been extensively studied recently because of its relative conceptual simplicity and effectiveness (e.g. [5, 1]).

Job	Birth	Postcode	Illness
Cat1	1975	4350	HIV
Cat1	1955	4350	HIV
Cat1	1955	5432	flu
Cat1	1955	5432	fever
Cat2	1975	4350	flu
Cat2	1975	4350	fever

Table 1: Raw Medical Data Set

Job	Birth	Postcode	Illness
Cat1	*	4350	HIV
Cat1	*	4350	HIV
Cat1	1955	5432	flu
Cat1	1955	5432	fever
Cat2	1975	4350	flu
Cat2	1975	4350	fever

Table 2: A 2-anonymous Data Set of Table 1

Job	Birth	Postcode	Illness
*	1975	4350	HIV
*	*	4350	HIV
Cat1	1955	5432	flu
Cat1	1955	5432	fever
*	*	4350	flu
*	1975	4350	fever

Table 3: An Alternative 2-anonymous Data Set of Table 1

Job	Birth	Postcode	Illness
*	*	4350	HIV
*	*	4350	HIV
*	*	5432	flu
*	*	5432	fever
*	*	4350	flu
*	*	4350	fever

Table 4: A $(0.5, 2)$-anonymous Table of Table 1 by Full-Domain Generalization

In this paper, we focus on a study on the k-**anonymity** property [11, 10]. The k-anonymity model assumes a **quasi-identifier**, which is a set of attributes that may serve as an identifier in the data set. It is assumed that the dataset is a table and that each tuple corresponds to an individual. Let Q be the quasi-identifier. An **equivalence class** of a table with respect to Q is a collection of all tuples in the table containing identical values for Q. For example, tuples 1 and 2 in Table 2 form an equivalence class with respect to attribute set $\{$Job, Birth, Postcode$\}$. The size of an equivalence class indicates the strength of identification protection of individuals in the equivalent class. If the number of tuples in an equivalence class is greater, it will be more difficult to re-identify individual. A data set D is k-*anonymous* with respect to Q if the size of every equivalence class with respect to Q is k or more. As a result, it is less likely that any tuple in the released table can be linked to an individual and thus personal privacy is preserved.

For example, we have a raw medical data set as in Table 1. Attributes job, birth and postcode[1] form the quasi-identifier. Two unique patient records 1 and 2 may be re-identified easily since

[1] We use a simplified postcode scheme in this paper. There are four single digits, representing states, regions, cities and suburbs. Postcode 4350 indicates state-region-city-suburb.

KDD'06, August 20–23, 2006, Philadelphia, Pennsylvania, USA.
Copyright 2006 ACM 1-59593-339-5/06/0008 ...$5.00.

their combinations of job, birth and postcode are unique. The table is generalized as a *2-anonymous* table as in Table 2. This table makes the two patients less likely to be re-identified.

In the literature of k-anonymization, there are two main models. One model is *global recoding* [11, 7, 5, 10] while the other is *local recoding* [11, 1].

We assume that each attribute has a corresponding conceptual hierarchy or taxonomy. A lower level domain in the hierarchy provides more details than a higher level domain. For example, birth date in D/M/Y (e.g. 15/Mar/1970) is a lower level domain and birth date in Y (e.g. 1970) is a higher level domain. We assume such hierarchies for numerical attributes too. In particular, we have a hierarchical structure defined with $\{$value, interval, ?$\}$, where value is the raw numerical data, interval is the range of the raw data and ? is a symbol representing any values. Generalization replaces lower level domain values with higher level domain values. For example, birth D/M/Y is replaced by M/Y.

In **global recoding**, all values of an attribute come from the same domain level in the hierarchy. For example, all values in Birth date are in years, or all are in both months and years. One advantage is that an anonymous view has uniform domains but it may lose more information. For example, a global recoding of Table 1 may be Table 4 and it suffers from *over-generalization*. With **local recoding**, values may be generalized to different levels in the domain. For example, Table 2 is a 2-anonymous table by local recoding. In fact one can say that local recoding is a more general model and global recoding is a special case of local recoding. Note that, in the example, known values are replaced by unknown values (*). This is called *suppression*, which is one special case of generalization, which is in turn one of the ways of recoding.

Let us return to the earlier example. If we inspect Table 2 again, we can see that though it satisfies 2-anonymity property, it does not protect two patients' sensitive information, HIV infection. We may not be able to distinguish the two individuals for the first two tuples, but we can derive the fact that both of them are HIV infectious. Suppose one of them is the mayor, we can then confirm that the mayor has contracted HIV. Surely, this is an undesirable outcome. Note that this is a problem because the other individual whose generalized identifying attributes are the same as the mayor also has HIV. Table 3 is an appropriate solution. Since (*,1975,4350) is linked to multiple diseases (i.e. HIV and fever) and (*,*,4350) is also linked to multiple diseases (i.e. HIV and flu), it protects individual identifications and hides the implication.

We see from the above that protection of $relationship$ to sensitive attribute values is as important as identification protection. Thus there are two goals for privacy preservation: (1) to protect individual identifications and (2) to protect sensitive relationships. Our focus in this paper is to build a model to protect both in a disclosed data set. We propose an (α, k)-anonymity model, where α is a fraction and k is an integer. In addition to k-anonymity, we require that, after anonymization, in any equivalence class, the frequency (in fraction) of a sensitive value is no more than α. We first extend the well-known k-anonymity algorithm Incognito [7] to our (α, k)-anonymity problem. As the algorithm is not scalable to the size of quasi-identifier and may give a lot of distortions to the data since it is global-recoding based, we also propose an efficient local-recoding based method.

This proposal is different from the work of association rules hiding [12] in a transactional data set, where the rules to be hidden have to be known beforehand and each time only one rule can be hidden. Also, the implementation assumes that frequent itemsets of rules are disjoint, which is unrealistic. Our scheme blocks all rules from quasi-identifications to a sensitive class.

This work is also different from the work of template-based privacy preservation in classification problems [13], which considers hiding strong associations between some attributes and sensitive classes and combines k-anonymity with association hiding. There, the solution considers global recoding by suppression only and the aim is to minimize a distortion effect that is designed and dedicated for a classification problem. The model defined in this paper is more general in that we allow local recoding and that we aim at minimizing the distortions of data modifications without any attachment to a particular data mining method such as classification.

The (c, l)-diversity model [8] is proposed to solve the above problem, which is called the homogeneity attack. However, the (c, l)-diversity model also aims at countering another kind of attack, which is assuming that the attacker has background knowledge to rule out some possible values in a sensitive attribute for the targeted victim. Parameter l describes the level of diversity of sensitive values. If l is larger, there will be more different sensitive values in a group. The idea of using parameters c and l is to ensure that the most frequent sensitive value in a group should not be too frequent after the next p most frequent sensitive values are removed, where p is related to parameter l. It is quite difficult for users to set parameters c and l. Though we anticipate attacks with background knowledge, it is not clear what background knowledge an attacker may have. For example, it is possible that the attacker can rule out 90% of the possibilities if he/she judges from the symptoms (e.g. coughing). An attacker knows that his/her neighbor should have either one of a few diseases (e.g. lung cancer), among tens or even hundreds of other diseases that have no relationship to the symptoms. To keep such background knowledge at bay, we must prepare for the elimination of a large amount of possible values. Setting c and l to fortify against the exclusion of say over 90% of all possibilities would require massive generalization, if not simply impossible. Besides we do not know what other kinds of background knowledge an attacker may have. Hence we believe that background knowledge attack should be handled by more special treatment and not by a general anonymization mechanism. Also, the proposed algorithm in [8] is based on a global-recoding exhaustive algorithm Incognito, which is not scalable and may generate more distortion compared to local recoding.

We propose to handle the issues of k-anonymity with protection of sensitive values for sensitive attributes.

Our Contributions:

- We propose a simple and effective model to protect both identifications and sensitive associations in a disclosed data set. The model extends the k-anonymity model to the (α, k)-anonymity model to limit the confidence of the implications from the quasi-identifier to a sensitive value (attribute) to within α in order to protect the sensitive information from being inferred by strong implications. We prove that the optimal (α, k)-anonymity by local recoding is NP-hard.

- We extend Incognito[7], a global-recoding algorithm for the k-anonmity problem, to solve this problem for (α, k)-anonymity. We also propose a local-recoding algorithm, which is scalable and generate less distortion. In our experiment, we show that, on average, the local-recoding based algorithm performs about 4 times faster and gives about 3 times less distortions of the data set compared with the extended Incognito algorithm. We also describe how the model can be extended to more general cases.

2. PROBLEM DEFINITION

The k-anonymity model requires that every value set for the quasi-identifier attribute set has a frequency of zero or at least k. For example, Table 1 does not satisfy 2-anonymity property since tuples {Cat1, 1975, 4350} and {Cat1, 1955, 4350} occur once. Table 2 satisfies 2-anonymity property. Consider a large collection of patient records with different medical conditions. Some diseases are sensitive, such as HIV, but many diseases are common, such as cold and fever. Only associations with sensitive diseases need protection. To start with, we assume only one sensitive value, such as HIV. We introduce the α-deassociation requirement for the protection.

DEFINITION 1 (α-DEASSOCIATION REQUIREMENT). *Given a data set D, an attribute set Q and a sensitive value s in the domain of attribute $S \notin Q$. Let (E, s) be the set of tuples in equivalence class E containing s for S and α be a user-specified threshold, where $0 < \alpha < 1$. Data set D is* α-deassociated *with respect to attribute set Q and the sensitive value s if the relative frequency of s in every equivalence class is less than or equal to α. That is, $|(E, s)|/|E| \leq \alpha$ for all equivalence classes E.*

For example, Table 3 is 0.5-deassociated with respect to attribute set {Job, Birth, Postcode} and sensitive value HIV. There are three equivalence classes: $\{t_1, t_6\}$, $\{t_2, t_5\}$ and $\{t_3, t_4\}$. For each of the first two equivalent classes of size two, only one tuple contains HIV and therefore $|(E, s)|/|E| = 0.5$. For the third equivalence class, no tuple contains HIV and therefore $|(E, s)|/|E| = 0$. Thus, for any equivalence classes, $|(E, s)|/|E| \leq 0.5$.

Our objective is therefore to anonymize a data set so that it satisfies both the k-anonymity and the α-deassociation criteria.

DEFINITION 2 ((α, k)-ANONYMIZATION). *A view of a table is said to be an (α, k)*-anonymization *of the table if the view modifies the table such that the view satisfies both k-anonymity and α-deassociation properties with respect to the quasi-identifier.*

For example, Table 3 is a (0.5, 2)-anonymous view of Table 1 since the size of all equivalence classes with respect to the quasi-identifier is 2 and each equivalence class contains at most half of the tuples associating with HIV.

Both parameters α and k are intuitive and operable in real-world applications. Parameter α caps the confidence of implications from values in the quasi-identifier to the sensitive value while parameter k specifies the minimum number of identical quasi-identifications.

DEFINITION 3 (LOCAL RECODING). *Given a data set D of tuples, a function c that convert each tuple t in D to $c(t)$ is a* local recoding *for D.*

Local recoding typically distorts the values in the tuples in a data set. We can define a measurement for the amount of distortion generated by a recoding, which we shall call the **recoding cost**. If a suppression is used for recoding of a value which modifies the value to an unknown *, then the cost can be measured by the total number of suppressions, or the number of *'s in the resulting data set. Our objective is to find local recoding with a minimum cost. We call it the problem of optimal (α, k)-anonymization. The corresponding decision problem is defined as follows.

(α, k)-**ANONYMIZATION**: Given a data set D with a quasi-identifier Q and a sensitive value s, is there a local recoding for D by a function c such that, after recoding, (α, k)-anonymity is satisfied and the cost of the recoding is at most $\mathcal{C}$?

Optimal k-anonymization by local recoding is NP-hard as discussed in [9, 1]. Now, we show that optimal (α, k)-anonymization by local recoding is also NP-hard.

THEOREM 1. *(α, k)-anonymity is NP-hard for a binary alphabet ($\sum$ = {0, 1}).*

Proof Sketch: The proof is by transforming the problem of EDGE PARTITION INTO 4-CLIQUES to the (α, k)-anonymity problem.

EDGE PARTITION INTO 4-CLIQUES: Given a simple graph $G = (V, E)$, with $|E| = 6m$ for some integer m, can the edges of G be partitioned into m edge-disjoint 4-cliques? [6]

Given an instance of EDGE PARTITION INTO 4-CLIQUES. Set $\alpha = 0.5$ and $k = 12$. For each vertex $v \in V$, construct a non-sensitive attribute. For each edge $e \in E$, where $e = (v_1, v_2)$, create a pair of records r_{v_1,v_2} and $\widetilde{r}_{v_1,v_2}$, where the two records have the attribute values of both v_1 and v_2 equal to 1 and all other non-sensitive attribute values equal to 0, but one record r_{v_1,v_2} has the sensitive attribute equal to 1 and the other record $\widetilde{r}_{v_1,v_2}$ has the sensitive attribute equal to 0.

We define the cost of the $(0.5, 12)$-anonymity to be the number of suppressions applied in the data set. We show that the cost of the $(0.5, 12)$-anonymity is at most $48m$ if and only if E can be partitioned into a collection of m edge-disjoint 4-cliques.

Suppose E can be partitioned into a collection of m disjoint 4-cliques. Consider a 4-clique Q with vertices v_1, v_2, $v_3$and v_4. If we suppress the attributes v_1, v_2, v_3 and v_4 in the 12 records corresponding to the edges in Q, then a cluster of these 12 records are formed where each modified record has four *'s. Note that the α-deassociation requirement can be satisfied as the frequency of the sensitive attribute value 1 is equal to 0.5. The cost of the $(0.5, 12)$-anonymity is equal to $12 \times 4 \times m = 48m$.

Suppose the cost of the $(0.5, 12)$-anonymity is at most $48m$. As G is a simple graph, any twelve records should have at least four attributes different. So, each record should have at least four *'s in the solution of the $(0.5, 12)$-anonymity. Then, the cost of the $(0.5, 12)$-anonymity is at least $12 \times 4 \times m = 48m$. Combining with the proposition that the cost is at most $48m$, we obtain the cost is exactly equal to $48m$ and thus each record should have exactly four *'s in the solution. Each cluster should have exactly 12 records (where six have sensitive value 1 and the other six have sensitive value 0). Suppose the twelve modified records contain four *'s in attributes v_1, v_2, v_3 and v_4, the records contain 0's in all other non-sensitive attributes. This corresponds to a 4-clique with vertices v_1, v_2, v_3 and v_4. Thus, we conclude that the solution corresponds to a partition into a collection of m edge-disjoint 4-cliques. □

Let p be the fraction of the set of tuples that contain sensitive values. Suppose α is set smaller than p. Then no matter how we partition the data set, by the pigeon hole principle, there should be at least one partition $\mathcal{P}$ which contains p or more sensitive value, and therefore cannot satisfy α-deassociation property.

LEMMA 1 (CHOICE OF α). *α should be set to a value greater than or equal to the frequency (given in fraction) of the sensitive value in the data set D.*

Distortion Ratio or **Recoding Cost**: Since we assume the more general case of a taxonomy tree for each attribute, we define the cost of local-recoding based on this model. The cost is given by the **distortion ratio** of the resulting data set and is defined as follows. Suppose the value of the attribute of a tuple has not been generalized, there will be no distortion. However, if the value of the attribute of a tuple is generalized to a more general value in the taxonomy tree, there is a distortion of the attribute of the tuple. If the value is generalized more (i.e. the original value is updated to a value at the node of the taxonomy near to the root), the distortion

Gender	Birth	Postcode	Sens
male	May 1965	4351	n
male	Jun 1965	4351	c
male	Jul 1965	4361	n
male	Aug 1965	4362	n

Table 5: A Data Set

(a)

No	Postcode	Sens
1	4351	n
2	4351	c
3	4351	n
4	4352	n

(b)

No	Postcode	Sens
1	4351	n
2	4351	c
3	435*	n
4	435*	n

Table 6: Projected Table with Quasi-identifier = Postcode: (a) Original Table and (b) Generalized Table

will be greater. Thus, the *distortion* of this value is defined in terms of the *height* of the value generalized. For example, if the value has not been generalized, the height of the value generalized is equal to 0. If the value has been generalized one level up in the taxonomy, the height of the value generalized is equal to 1. Let $h_{i,j}$ be the height of the value generalized of attribute A_i of the tuple t_j. The *distortion* of the whole data set is equal to the sum of the distortions of all values in the generalized data set. That is, distortion = $\sum_{i,j} h_{i,j}$. *Distortion ratio* is equal to the distortion of the generalized data set divided by the distortion of the *fully* generalized data set, where the fully generalized data set is one with all values of the attributes are generalized to the root of the taxonomy.

3. GLOBAL-RECODING

In this section, we extend an existing global-recoding based algorithm called Incognito [7] for the (α, k)-anonymous model. Incognito algorithm [7] is an optimal algorithm for the k-anonymity problem. It has also been used in [8] for the l-diversity problem. [7] and [8] make use of *monotonicity property* in searching the solution space. The searches can be made efficient if a stopping condition is satisfied. The stopping condition is that, if table T' is satisfied with the privacy requirements, then every generalization of T' is also satisfied with the privacy requirement.

Algorithm: The algorithm is similar to [7, 8]. The difference is in the testing criteria of each *candidate* in the solution space. [7] tests for the k-anonymity property and [8] tests the k-anonymity and l-diversity properties. Here, we check the (α, k)-anonymity property.

4. LOCAL-RECODING

The extended Incognito algorithm is an exhaustive global recoding algorithm which is not scalable and may generate excessive distortions to the data set. Here we propose a scalable local-recoding algorithm called *top-down* approach.

In this section, we present a top-down approach to tackle the problem. For ease of illustration, we first present the approach for a quasi-identifier of size 1. Then, the method is extended to handle quasi-identifiers of size greater than 1. The idea of the algorithm is to first generalize all tuples *completely* so that, initially, all tuples are generalized into one equivalence class. Then, tuples are *specialized* in iterations. During the specialization, we must maintain (α, k)-anonymity. The process continues until we cannot specialize the tuples anymore.

Let us illustrate with an example in Table 5. Suppose the quasi-identifer contains Postcode only. Assume that $\alpha = 0.5$ and $k = 2$. Initially, we generalize all four tuples completely to an equivalence class with Postcode = **** (Figure 1 (a)). Then, we specialize each tuple one level down in the generalization hierarchy. We obtain the branch with Postcode = 4*** in Figure 1 (b). In the next iterations, we obtain the branch with Postcode = 43** and the branch with Postcode = 435* in Figure 1 (c) and Figure 1 (d), respectively. As the Postcode of all four tuples starts with the prefix ”435”, there is only one branch for each specialization of the postcode with prefix ”435”. Next, we can further specialize the tuples into the two branches as shown Figure 1 (e). Hence the specialization processing can be seen as the growth of a tree.

If each leaf node satisfies (α, k)-anonymity, then the specialization will be successful. However, we may encounter some problematic leaf nodes that do not satisfy (α, k)-anonymity. Then, all tuples in such leaf nodes will be pushed upwards in the generalization hierarchy. In other words, those tuples cannot be specialized in this process. They should be kept *unspecialized* in the *parent* node. For example, in Figure 1 (e), the leaf node with Postcode = 4352 contains only one tuple, which violates (α, k)-anonymity, where $k = 2$. Thus, we have to move this tuple back to the parent node with Postcode = 435*. See Figure 1 (f).

After the previous step, we move all tuples in problematic leaf nodes to the parent node. However, if the collected tuples in the parent node do not satisfy (α, k)-anonymity, we should further move some tuples from other leaf nodes L to the parent node so that the parent node can satisfy (α, k)-anonymity while L also maintain the (α, k)-anonymity. For instance, in Figure 1 (f), the parent node with Postcode = 435* violates (α, k)-anonymity, where $k = 2$. Thus, we should move one tuples upwards in the node B with Postcode = 4351 (which satisfies (α, k)-anonymity). In this example, we move tuple 3 upwards to the parent node so that both the parent node and the node B satisfy the (α, k)-anonymity.

Finally, in Figure 1 (g), we obtain a data set where the Postcode of tuples 3 and 4 are generalized to 435* and the Postcode of tuples 1 and 2 remains 4351. We call the final allocation of tuples in Figure 1 (g) the final *distribution* of tuples after the specialization. The results can be found in Table 6 (b).

In this approach, we have to un-specialize some tuples which have already satisfied the (α, k)-anonymity. Which tuples should we select in order to produce a generalized data set with less distortion? We tackle this issue by the following additional steps. We further specializing all tuples in all candidate nodes. We repeat the specialization process until we cannot further specialize the tuples. Then, for each tuple t, we record the number of times of specializations. If the tuple t has fewer times of specializations, it should be considered as a good choice for un-specialization since it is evident that it cannot be specialized deeply in later steps.

Quasi-identifier of Size More Than 1: Next we extend the top-down algorithm to handle the case where the quasi-identifier has a size greater than one. Again, all attributes of the tuples are generalized fully in the first step. Then, for each iteration, we find the ”best” attribute for specialization and perform the specialization for the ”best” attribute. The iteration continues until no further specialization is available.

Consider a group P. We will specialize the group P by specializing with one attribute. We have to find the ”best” attribute for specialization. For each attribute in the quasi-identifer, our approach ”tries” to specialize P. Then, among those specializations, we find the ”best” attribute for final specialization. Our criteria of choosing the ”best” attributes are described as follows.

Criteria 1 (Greatest No of Tuples Specialized): During the specialization of P, we obtain a final distribution of the tuples. Some are specialized and some may still remain in P. The ”best” specialization yields the greatest number of tuples specialized be-

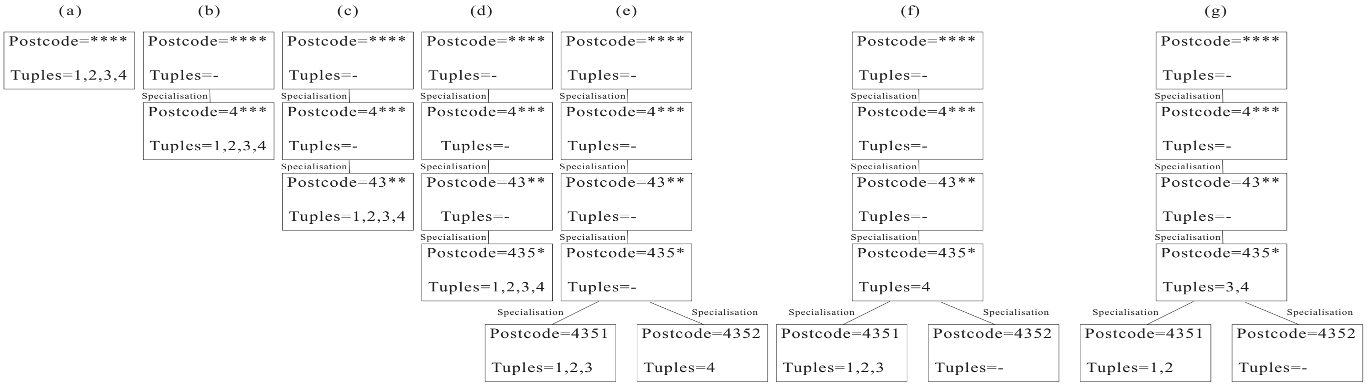
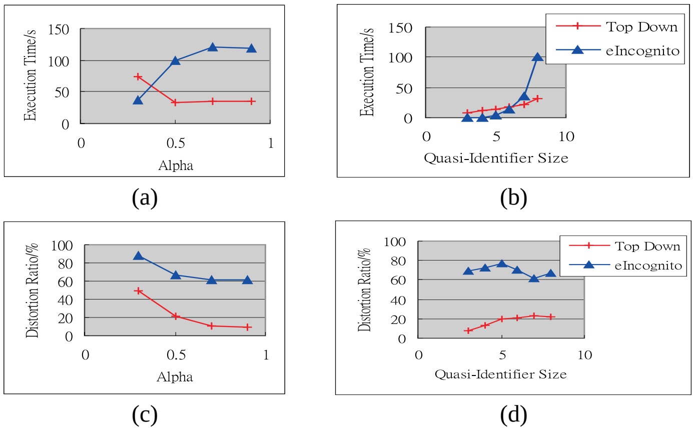


Figure 1: Top-Down Algorithm for Quasi-identifier of Size 1

	Attribute	Distinct Values	Generalizations	Height
1	Age	74	5-, 10-, 20-year ranges	4
2	Work Class	7	Taxonomy Tree	3
3	Education	16	Taxonomy Tree	4
4	Martial Status	7	Taxonomy Tree	3
5	Occupation	14	Taxonomy Tree	2
6	Race	5	Taxonomy Tree	2
7	Sex	2	Suppression	1
8	Native Country	41	Taxonomy Tree	3
9	Salary Class	2	Suppression	1

Table 7: Description of Adult Data Set



(a) (b)

(c) (d)

Figure 2: Execution Time and Distortion Ratio Versus Quasi-identifier Size and α ($k = 2$)

cause that corresponds to the least overall distortion.

Criterion 2 (Smallest No of Branching Specialized): In case there is a tie when we consider the first criterion, we will further consider the number of branches specialized (i.e. non-empty branches). The ”best” specialization yields the smallest number of branches specialized. The rationale is that smallest number of branches can be an indicator of more generalized domain and it is a good choice compared to a less generalized domain.

5. EMPIRICAL STUDY

Pentium IV 2.2GHz PC with 1GM RAM was used to conduct our experiment. The algorithm was implemented in C/C++. In our experiment, we adopted the publicly available data set, Adult Database, at the UCIrvine Machine Learning Repository [3]. This data set (5.5MB) was also adopted by [7, 8, 14, 5]. We used a configuration similar to [7, 8]. We eliminated the records with unknown values. The resulting data set contains 45,222 tuples. Nine of the attributes were chosen as the quasi-identifier, as shown in Table 7. On default, we set $k = 2$ and $\alpha = 0.5$, and we chose the first eight attributes and the last attribute in Table 7 as the quasi-identifer and the sensitive attribute, respectively.

We evaluated the proposed algorithm in terms of two measurements: execution time and distortion ratio (see Section 2). We conducted the experiments five times and took the average execution time.

We denote the proposed algorithms by *Top Down* and *eIncognito*. *eIncognito* denotes the extended Incognito algorithm while *Top Down* denotes the local-recoding based top-down approach, respectively.

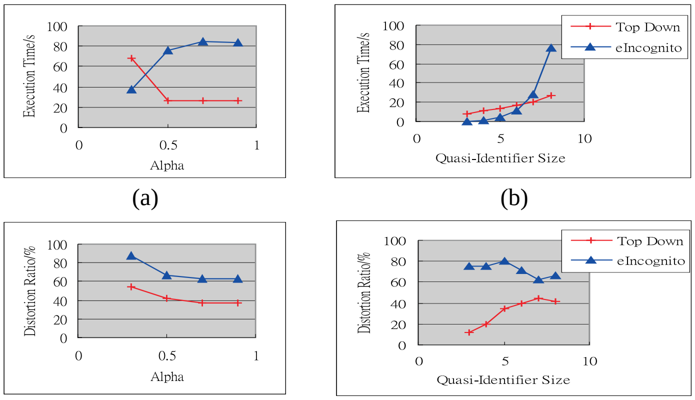
Figure 2 shows the graphs of the execution time and the distortion ratio against quasi-identifier size and α when $k = 2$. In Figure 2 (a), when α varies, different algorithms change differently. The execution time of eIncognito Algorithm increases with α. This is because, when α increases, the number of candidates (representing the generalization domain) increases, and thus the execution time increases. The execution time of Top Down Algorithm decreases when α increases. In the top-down algorithm, we may have to unspecialize some tuples in the branches satisfying (α, k)-anonymity so that the parent P satisfies (α, k)-anonymity. When α is small, it is more likely that the parent P cannot satisfy (α, k)-anonymity, triggering this step of un-specialization. As the un-specialization step is more complex, the execution time is larger when α is smaller.

In Figure 2 (b), when the quasi-identifer size increases, the execution time of the algorithm increases because the complexity of the algorithms is increased with the quasi-identifier size.

On average, among these three algorithms, eIncognito Algorithm requires the greatest execution time and Top Down Algorithm has the smallest execution time. This shows that eIncognito performs much slower compared with local-recoding based algorithm.

In Figure 2 (c), when α increases, the distortion ratio decreases. Intuitively, if α is greater, there is less requirement of α-deassociation, yielding fewer operations of generalization of the values in the data set. Thus, the distortion ratio is smaller.

In Figure 2 (d), it is easy to see why the distortion ratio increases with the quasi-identifier size. When the quasi-identifier contains more attributes, there is more chance that the quasi-identifier of two tuples are different. In other words, there is more chance that the tuples will be generalized. Thus, the distortion ratio is greater. When k is larger, it is also obvious that the distortion ratio is greater



(a) (b)

(c) (d)

Figure 3: Execution Time and Distortion Ratio Versus Quasi-identifier Size and α ($k = 10$)

because it is less likely that the quasi-identifer of two tuples are equal. On average, Top Down algorithm results in about 3 times smaller distortion ratio compared with eIncognito Algorithm.

We have also conducted the experiments for $k = 10$, which is shown in Figure 3. The results are also similar to the graphs for $k = 2$ (as in Figure 2).

6. GENERAL (α, K)-ANONYMITY MODEL

In this section, we will extend the simple (α, k)-model to multiple sensitive values. When there are two or more sensitive values and they are rare cases in a data set (e.g. HIV and prostate cancer). We may combine them into one combined sensitive class and the simple (α, k)-anonymity model is applicable. The inference confidence to each individual sensitive value is smaller than or equal to the confidence to the combined value, which is controlled by α.

Next we consider the case when all values in an attribute are sensitive and require protection. It is possible to have an (α, k)-anonymity model to protect a sensitive attribute when the attribute contains many values and no single value dominates the attribute (which will be explained later). The salary attribute in employer table is an example. When each equivalent class contains three salary scales with even distribution, we have about 33% confidence to infer the salary scale of an individual in the equivalent class.

DEFINITION 4 (α-RARE). *Given an equivalence class E, an attribute X and an attribute value $x \in X$. Let (E, x) be the set of tuples containing x in E and α be a user-specified threshold, where $0 \leq \alpha \leq 1$. Equivalence class E is* α-rare *with respect to attribute set X if the proportion of every attribute value of X in the data set is not greater than α, i.e. $|(E, x)|/|E| \leq \alpha$ for $x \in X$.*

For example, in Table 3, if $X = Illness$, equivalent class $\{t_3, t_4\}$ is 0.5-rare because ”flu” and ”fever” occur evenly in the equivalent class. If every equivalent class is α-rare in the class, the data set is called α-deassociated.

DEFINITION 5 (GENERAL α-DEASSOCIATION PROPERTY). *Given a data set D, an attribute set Q and a sensitive class attribute S. Let α be a user-specified threshold, where $0 \leq \alpha \leq 1$. Data set D is generally* α-deassociated *with respect to an attribute set Q and a sensitive attribute S if, for any equivalent classes $E \subset D$, E is α-rare with respect to S.*

For example, Table 3 is 0.5-deassociated since all three equivalent classes, $\{t_1, t_6\}$, $\{t_2, t_5\}$ and $\{t_3, t_4\}$, are 0.5-rare with respect to attribute set Illness. When a data set is α-deassociated with respect to a sensitive attribute, it is α-deassociated with respect to every value in the attribute. Therefore, the upper bound of inference confidence from the quasi-identifier to the sensitive attribute is α.

The proposed algorithms in Sections 3 and 4 can be extended to the general (α, k)-anonymity model. The global-recoding based algorithm depends on the monotonicity property. The propoerty holds for the general (α, k)-anonymity. Thus, the global-recoding based algorithm can be extended by modifying the step of testing of candidates with the general model.

The top-down local-recoding algorithm can also be easily extended to the general model by modifying the condition when testing the candidates.

7. CONCLUSION

The k-anonymity model protects identification information, but does not protect sensitive relationships in a data set. In this paper, we propose the (α, k)-anonymity model to protect both identifications and relationships in data. We discuss the properties of the model. We prove that achieving optimal (α, k)-anonymity by local recoding is NP-hard. We present an optimal global-recoding method and an efficient local-encoding based algorithm to transform a data set to satisfy (α, k)-anonymity property. The experiment shows that, on average, the local-encoding based algorithm performs about 4 times faster and gives about 3 times less distortions of the data set compared with the global-recoding algorithm. We also describe how the model can be extended to more general cases.

ACKNOWLEDGEMENTS: This paper is partly supported by ARC DP0559090.

8. REFERENCES

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *ICDT*, pages 246–258, 2005.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[3] E. K. C. Blake and C. J. Merz. UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998.

[4] L. Cox. Suppression methodology and statistical disclosure control. *J. American Statistical Association*, 75:377–385, 1980.

[5] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *ICDE*, pages 205–216, 2005.

[6] I. Holyer. The np-completeness of some edge-partition problems. *SIAM J. on Computing*, 10(4):713–717, 1981.

[7] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *SIGMOD Conference*, pages 49–60, 2005.

[8] A. Machanavajjhala, J. Gehrke, and D. Kifer. l-diversity: privacy beyond k-anonymity. In *To appear in ICDE06*, 2006.

[9] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS*, pages 223–228, 2004.

[10] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[11] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International journal on uncertainty, Fuzziness and knowldege based systems*, 10(5):571 – 588, 2002.

[12] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):434–447, 2004.

[13] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. In *To appear in ICDM05*, 2005.

[14] K. Wang, P. S. Yu, and S. Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *ICDM*, pages 249–256, 2004.