k-Anonymity via Clustering Domain Knowledge for Privacy Preservation

Taiyong Li\textsuperscript{1,2}, Changjie Tang\textsuperscript{1}, Jiang Wu\textsuperscript{1,2}, Qian Luo\textsuperscript{1}, Shengshi Li\textsuperscript{1}, Xun Lin\textsuperscript{1,2}, Jie Zuo\textsuperscript{1}

\textsuperscript{1}School of Computer Science, Sichuan University, Chengdu, Sichuan, China, 610065
tangchangjie@cs.scu.edu.cn

\textsuperscript{2}School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, Sichuan, China, 610074
lity_f@swufe.edu.cn

Abstract

Preservation of privacy in micro-data release is a challenging task in data mining. The k-anonymity method has attracted much attention of researchers. Quasi-identifier is a key concept in k-anonymity. The tuples whose quasi-identifiers have near effect on the sensitive attributes should be grouped to reduce information loss. The previous investigations ignored this point. This paper studies k-anonymity via clustering domain knowledge. The contributions include: (a) Constructing a weighted matrix based on domain knowledge and proposing measure methods. It carefully considers the effect between the quasi-identifiers and the sensitive attributes. (b) Developing a heuristic algorithm to achieve k-anonymity via clustering domain knowledge based on the measure methods. (c) Implementing the algorithm for privacy preservation, and (d) Experiments on real data demonstrate that the proposed k-anonymous methods decrease 30\% information loss compared with basic k-anonymity.

1. Introduction

Recently, data mining has been a hot topic in many applications. As a side effect, some sensitive data and privacy are exposed to the public. The micro-data itself usually does not reveal any sensitive information, but when it is linked by other public data, some sensitive information is exposed, which is so called linking attack.

To avoid the linking attack, k-anonymity was proposed\textsuperscript{[1-5]}. A data set is k-anonymity if each tuple in the data set is indistinguishable from at least (k-1) other tuples in the same data set. Clearly, since there are k tuples in the data set, they can’t be distinguished by privacy attacker. At the same time, to make the data set satisfy k-anonymity, there is some information loss in micro-data. To make the information loss as little as possible while achieving k-anonymity, many methods have been proposed. However, none of them considers clustering domain knowledge.

2. Related work

The problem of privacy preservation has been studied extensively in the framework of statistical databases. A number of information disclosure limitation techniques, such as sampling, rounding, cell suppression, data swapping and perturbation, have been designed for data publishing. These techniques, however, may perturb the characteristics of the original dataset. The k-anonymity model was proposed by Sweeney and Samarati\textsuperscript{[1-3]}. A release provides k-anonymity protection if the information for each record contained in the release cannot be distinguished from at least (k-1) other records whose information also appear in the release. Based on the basic k-anonymity, some advanced ones were proposed\textsuperscript{[7-9]}. In [7], a k-anonymity method named $\ell$-diversity was introduced which gave stronger privacy guarantees. To achieve $\ell$-diversity, each equivalence class has at least $\ell$ different sensitive attributes. p-sensitive k-anonymity requires each sensitive attribute in an equivalence class has at least p different values\textsuperscript{[8]}. The utility-based anonymization\textsuperscript{[10]} gives each attribute in the quasi-identifier a weight. But for a table, the weight given...
is never changed. In practical application, an attribute in the quasi-identifier has different effect on different sensitive attributes, which is called domain knowledge in this paper. However, none of the methods mentioned above considered clustering the domain knowledge. In this paper, we study the method of k-anonymity via clustering the domain knowledge.

3. Basic concepts and terminology

The linking attack will reveal the sensitive information, which is illustrated in the following Example.

Example 1. Two tables contain some health information and voter information, as shown in Table 1 and Table 2. From Table 1 or Table 2, a privacy attacker can not get any important privacy. When the attacker links Table 1 with Table 2, he will get Table 3. It says that Alice is an AIDS patient.

![Table 1. Microdata](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Zipcode</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>25</td>
<td>F</td>
<td>12300</td>
<td>AIDS</td>
</tr>
<tr>
<td>Linda</td>
<td>35</td>
<td>F</td>
<td>13000</td>
<td></td>
</tr>
<tr>
<td>Bill</td>
<td>21</td>
<td>M</td>
<td>12000</td>
<td>flu</td>
</tr>
<tr>
<td>Sam</td>
<td>35</td>
<td>M</td>
<td>27000</td>
<td></td>
</tr>
<tr>
<td>John</td>
<td>28</td>
<td>M</td>
<td>14000</td>
<td></td>
</tr>
</tbody>
</table>

![Table 2. Voter registration list](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Zipcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>25</td>
<td>F</td>
<td>12300</td>
</tr>
<tr>
<td>Linda</td>
<td>35</td>
<td>F</td>
<td>13000</td>
</tr>
<tr>
<td>Bill</td>
<td>21</td>
<td>M</td>
<td>12000</td>
</tr>
<tr>
<td>Sam</td>
<td>35</td>
<td>M</td>
<td>27000</td>
</tr>
<tr>
<td>John</td>
<td>28</td>
<td>M</td>
<td>14000</td>
</tr>
</tbody>
</table>

![Table 3. Linking result](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Zipcode</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>25</td>
<td>F</td>
<td>12300</td>
<td>AIDS</td>
</tr>
</tbody>
</table>

To explain this question formally, we give the following definitions and examples.

Definition 1 (Quasi-Identifier and Sensitive Attribute, QI and SA). Given a table \( T = (A_1, A_2, ..., A_n) \), a quasi-identifier is a minimal subset of attributes \( A_1, A_2, ..., A_n \) that can potentially reveal sensitive information through linking attack. A sensitive attribute is the attribute that cannot be revealed.

Example 2. In Table 1, \{Age, Sex, Zipcode\} is a quasi-identifier. When linked with Table 2, it reveals some patients’ diseases (e.g. Alice’s AIDS). The attribute disease is a sensitive attribute, and it cannot be revealed but be preserved.

Definition 2 (k-Anonymity). Formally, given the parameter \( k \) and QI \( q_{i1}, ..., q_{in} \), a table \( T \) is said to satisfy k-anonymity if for each tuple \( t \), there exist at least another \((k-1)\) tuples \( t_1, ..., t_{k-1} \) which have the same projection on the QI, i.e. \( t(q_{i1}, ..., q_{in}) = t_1(q_{i1}, ..., q_{in}) = ... = t_{k-1}(q_{i1}, ..., q_{in}) \). Here, \( t, t_1, ..., t_{k-1} \) form an equivalence class.

Example 3. Table 4 is a 2-anonymous table of Table 1, where ID is only for reference. Each tuple has at least 1 another tuples which have the same projection on the QI\{Age, Sex, Zipcode\}. The tuples whose ID values are 1 and 2 form an equivalence class.

![Table 4. 2-Anonymous table of Table 1](image)

<table>
<thead>
<tr>
<th>ID</th>
<th>Age</th>
<th>Sex</th>
<th>Zipcode</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[25,26]</td>
<td>*</td>
<td>12***</td>
<td>AIDS</td>
</tr>
<tr>
<td>2</td>
<td>[25,26]</td>
<td>*</td>
<td>12***</td>
<td>flu</td>
</tr>
<tr>
<td>3</td>
<td>[37,40]</td>
<td>M</td>
<td>13***</td>
<td>bronchitis</td>
</tr>
<tr>
<td>4</td>
<td>[37,40]</td>
<td>M</td>
<td>13***</td>
<td>flu</td>
</tr>
<tr>
<td>5</td>
<td>[37,40]</td>
<td>M</td>
<td>13***</td>
<td>bronchitis</td>
</tr>
</tbody>
</table>

4. Domain knowledge measurement: QI-SA Matrix

The QI’s effect level on SA is represented by a matrix \( T(m \times n) \) called QI-SA Matrix, where \( m \) is the number of sensitive attribute’s value, and \( n \) is the number of the attributes in quasi-identifier. We partition the effect of the QIs on sensitive attribute into 5 levels: Very High(VH), High(H), Normal(N), Low(L), Very Low(VL). To make the problem easy, we let VH - VL equal 5-1 respectively.

The QI-SA Matrix is illustrated as follows:

\[
T = (t_{ij})_{m \times n} = \begin{bmatrix}
g_1 & g_2 & \cdots & g_n \\
s_1 & 5 & 2 & \cdots & 1 \\
s_2 & 1 & 4 & \cdots & 2 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
s_m & 3 & 1 & \cdots & 2
\end{bmatrix}
\]

It represents QI’s effect level on SA, where \( s_i (1 \leq i \leq m) \) is the i-th value of sensitive attribute and \( q_j (1 \leq j \leq n) \) is the j-th attribute in quasi-identifier. The entry \( T[i,j] \) denotes the j-th attribute’s effect in quasi-identifier on the i-th value of sensitive attribute.

Definition 3 (QI’s Effect on SA). Given the QI-SA Matrix, the effect of QI on the i-th value of SA is
where \( n \) is the number of the attributes in quasi-identifier, i.e., the effect of QI on the \( i \)-th value of sensitive attribute equals the sum of each attribute’s effect in QI in the \( i \)-th value of sensitive attribute.

**Definition 4 (Distance of QI’s Effect).** Given 2 different values of sensitive attribute \( s_i \) and \( s_k \), the distance of the QI’s effect is

\[
D_{ik} = \sum_{j=1}^{n} |t_{ij} - t_{kj}|
\]

where \( n \) is the number of the attributes in quasi-identifier and \( |t_{ij} - t_{kj}| \) is the absolute value of \( t_{ij} \) and \( t_{kj} \).

We do not define \( D_{ik} \) as \( |E_{i1} - E_{ik}| \) because it can not reflect the difference well. For example, consider quasi-identifier has two attributes \( q_1 \) and \( q_2 \), and their effect on \( s_j \) is 3 and 2 respectively, while their effect on \( s_k \) is 4 and 2. Obviously, \( q_1 \) and \( q_2 \) has different effect on \( s_j \) and \( s_k \). However, if we compute the distance by \( |E_{i1} - E_{ik}| \), the value will be 0. This is not what we want.

According to the distance between any two QIs, we propose our clustering algorithm of domain knowledge. The algorithm is based on the basic K-Means.

**Algorithm 1 (CLUDK) Domain Knowledge Clustering Algorithm**

Input: QI-SA Matrix \( T \), parameter \( K \)

Output: a set of \( K \) clusters

Begin

1. randomly choose \( K \) objects from \( T \) as the initial cluster centers;
2. repeat
3. (re)assign each object to the cluster to which the distance between the object and the mean value of the objects in the cluster is the shortest;
4. update the cluster means;
5. until no change;
End

The Algorithm 1 is derived from the classical K-means algorithm. It treats each line in QI-SA Matrix \( T \) as an \( n \)-dimensional object. The key steps are (a) Randomly choosing \( K \) objects from \( T \) as the initial cluster centers. (b) By the distance between an object and each cluster center, the object is assigned to the nearest cluster center. And new cluster center is computed. The process iterates until the cluster centers have no change.

**5. k-Anonymity algorithm via clustering domain knowledge**

When achieving k-anonymity, the information loss is a very important issue. In this section, we describe the information loss function for our algorithm to achieve k-anonymity based on the result of clustering domain knowledge.

To construct an equivalence class, it is obvious to group the tuples which have the nearest distance together to have the least information loss. In a privacy table, there are two types of data, i.e., numeric and categorical data. Usually, the categorical data can be discreted to numeric data, but not all categorical data can do so. Therefore, we need two distance functions to measure numeric data and categorical data respectively.

Let \( T \) be a table with QI(\( N_1,\ldots,N_m,C_1,\ldots,C_m \)), where \( N_1(1 \leq i \leq n) \) is the \( i \)-th numeric attribute and \( C_j(1 \leq j \leq m) \) is the \( j \)-th categorical attribute in the QI. Suppose a tuple \( t=(x_{N_1},\ldots,x_{N_m},x_{C_1},\ldots,x_{C_m}) \) in privacy table is generalized to \( t'=(y_{N_1},z_{N_1},\ldots,y_{N_m},z_{N_m},D_{C_1},\ldots,D_{C_m}) \) in k-anonymous table.

From \( t \) to \( t' \), there is some information loss. On numeric attribute \( N_i \), the information loss is defined as

\[
IL_{N_i}(t) = w_i \frac{z_{N_i} - y_{N_i}}{|N_i|}
\]

where \( |N_i| \) represents the range of all tuples on attribute \( N_i \) and \( w_i \) is the weight of \( N_i \)'s effect on the value of the sensitive attribute of tuple \( t \). On categorical attribute \( C_j \), the information loss is defined as

\[
IL_{C_j}(t) = w_j \frac{H(A(x_{C_j},D_{C_j}))}{H(T_{C_j})}
\]

where \( H(A(x_{C_j},D_{C_j})) \) is the height of the subtree rooted at the lowest common ancestor of \( x_{C_j} \) and \( D_{C_j} \), and \( H(T_{C_j}) \) is the height of tree of the categorical attribute \( C_j \) and \( w_j \) is the weight of \( C_j \)'s effect on the value of the sensitive attribute of tuple \( t \).

Putting things together, we give the following definition.

**Definition 5 (Information Loss).** The information loss of tuple \( t \) is defined as

\[
IL(t) = \sum_{i=1}^{n} IL_{N_i}(t) + \sum_{j=1}^{m} IL_{C_j}(t)
\]

\[
= \sum_{i=1}^{n} w_i \frac{z_{N_i} - y_{N_i}}{|N_i|} + \sum_{j=1}^{m} w_j \frac{H(A(x_{C_j},D_{C_j}))}{H(T_{C_j})}
\]

And the information loss of table \( T \) is defined as

\[
IL(T) = \frac{1}{|T|} \sum_{t \in T} IL(t)
\]

where \( |T| \) is the number of tuples in table \( T \).

The problem of optimal k-anonymity is NP-hard[11]. Therefore, we develop a heuristic algorithm for k-anonymity via clustering domain knowledge.
The idea of our algorithm is treating each tuple as a group, merging the groups which are in the same cluster according to clustering domain knowledge. The procedure repeats until each group contains at least k tuples.

Algorithm 2(KACLUDK) k-Anonymity Algorithm

Input: a table T, a QI-SA matrix M of domain knowledge, parameter k
Output: a k-anonymous table T'

Begin

1: cluster domain knowledge(CLUDK);
2: create a group for each tuple;
3: WHILE there exists some group G such that |G| < k DO{
4:   FOR each group G such that |G| < k DO{
5:     scan all other groups to find G' such that G and G' are in the same result of CLUDK;
6:     compute the information loss of transforming G to G' (IL1) and the information loss of transforming G' to G(IL2);
7:     if IL1 > IL2 then merge G' into G;
   else{ merge G into G'; G=G';}
5:   FOR each group G such that |G| ≥ 2k DO{
6:     split G into groups such that each group satisfies k-anonymity restriction ;
10: generalize and output the surviving groups to equivalence classes.
5: }
4: }
3: }

Line 1 clusters sensitive attributes according to QI-SA matrix via CLUDK. Each tuple is treated as a group first. When the size of one group G is smaller than k, it must be merged with another group G' or be merged into G' so that the size of G ∪ G' is larger or equals k. This process is shown in line 3 to line 7. Line 3 to line 7 terminate when there is no group G whose size is less than k. The size of each group is larger than or equals to k, so k-anonymity is achieved. However, inevitably, the sizes of some groups may be larger than 2k. To reduce information loss, these groups should be split until the size of each group is larger than or equals to k and is less than 2k. This process is in line 8 to line 9. Line 10 generalizes each group to equivalence classes and output them.

6. Experiments and performance study

We implement our algorithm and analyse its performance and then we compare it with the basic k-anonymous method without clustering domain knowledge.

6.1 Experimental setup

We use the adult census data set from the UC Irvine machine learning repository to evaluate our methods. We take age(A), gender(G), education(E), work class(WC) and salary class(SC) as Quasi-Identifiers and major occupation code(MOC) as Sensitive Attribute. We retain eight kinds of value of major occupation code: Armed Forces(AF), Executive admin and managerial(EAM), Farming forestry and fishing(FFF), Private household services(PHS), Professional specialty(PS), Sales(S), Transportation and material moving(TMM) and Technicians and related support(TRS). Therefore, the number of tuples is 54483. And we give the domain knowledge (QI-SA matrix) as shown in Table 5.

<table>
<thead>
<tr>
<th>A</th>
<th>G</th>
<th>E</th>
<th>WC</th>
<th>SC</th>
<th>MOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>AF</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>EAM</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>FFF</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>PHS</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>PS</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>S</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>TMM</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>TRS</td>
</tr>
</tbody>
</table>

The experiments were conducted on a PC with Intel Pentium 4 2.0GHz CPU and 1G main memory, and the operation system is Windows XP Professional. The development environment is Microsoft Visual C++ 6.0.

6.2 Experimental results

We use information loss of table to measure the anonymization quality. We compare the k-anonymity method in this paper which called k-anonymity CLUDK with basic k-anonymity. The experimental results are shown in Figure 1 and Figure 2.

![Figure 1. Information loss with respect to k](image-url)

Figure 1 compares the information loss between k-anonymity CLUDK and basic k-anonymity with respect to different values of k. From the curves, we can see that when k increases, k-anonymity CLUDK will has less information loss than basic k-anonymity has. Averagely, k-anonymity CLUDK decreases 30% information loss compared with basic k-anonymity.
Figure 2 shows the runtime of the two methods with respect to different values of k. Basic k-anonymity is about 4-6 times slower than k-anonymity CLUDK because the search space of the last one decreases greatly after clustering domain knowledge. The runtime of the two methods is not sensitive to k.

7. Conclusions and future work

Domain knowledge plays an important role when achieving k-anonymity, but it has not been considered in previous research. In this paper, we propose a heuristic k-anonymity via clustering domain knowledge. We give methods to measure the quasi-identifiers’ effect on sensitive attributes. We also define the information loss based on domain knowledge. Our experiments show that the proposed method causes less information loss and runtime than the basic k-anonymity method.

In future work, we will study the ways to enhance the diversity of sensitive attribute in k-anonymous table to preserve privacy and improve the performance of the algorithm.

References