Multi-dimensional K-anonymity based on Mapping for Protecting Privacy[1]

Qian Wang
College of Computer Science, Chongqing University, Chongqing, China
Email: wangqian@cqu.edu.cn

Cong Xu and Min Sun
College of Computer Science, Chongqing University, Chongqing, China
Email: {cong.xu, taoyu_610}@163.com

***Abstract*—Data release has privacy disclosure risk if not taking any protection policy. Although attributes that clearly identify individuals, such as Name, Identity Number, are generally removed or decrypted, attackers can still link these databases with other released database on attributes (Quasi-identifiers) to re-identify individual's private information. K-anonymity is a significant method for privacy protection in microdata release. However, it is a NP-hard problem for optimal k-anonymity on dataset with multiple attributes. Most partitions in k-anonymity at present are single-dimensional. Research on k-anonymity focuses on getting high quality anonymity while reducing the time complexity. This paper proposes a new multi-dimensional k-anonymity algorithm based on mapping and divide-and-conquer strategy. Multi-dimensional data are mapped to single-dimensional, and then k-anonymity on multiple attributes is implemented employing the divide-and-conquer strategy in polynomial time. Divided dimension selection is prioritized based on information dependency, which significantly reduces the information loss. The experiment shows that the proposed algorithm is feasible and performs much better in k-anonymity.**

***Index Terms*—privacy protection, k-anonymity, multi-dimension, mapping, partitioning**

I. INTRODUCTION

With the development of information technology, large amount of data are stored and released. Organizations get useful information with mining from published data while individual's privacy is threatened. K-anonymity [1] is a significant method for privacy protection when releasing microdata. It requires that in the published data tables, quasi-identifiers of every record must be the same as the ones of other k-1 records, which can prevent the attackers from linking data tables together to get the record linking to an individual. K-anonymity is applied for publication of information, such as medical treatment information [2], shopping habits, credit card record and so on. To achieve K-anonymity, quasi-identifier attributes are needed to be processed, so as to reduce the probability that attackers get the sensitive information by linking databases on quasi-identifier attributes [3]. How to improve the anonymity and minimize information loss, meanwhile reduce the time and space complexity, is now the focus of k-anonymity.

Sweeney L and Samarati proposed the k-anonymity method [4][5][6], and gave some algorithms. Many researchers do research on it and have proposed various ways to implement k-anonymity. Methods for k-anonymity can be divided into two groups: bottom-up and top-down.

Bottom-up adopts technologies of generalization and suppression. Generalization in bottom-up method can be full-domain generalization or local recoding. Full-domain generalization generalizes attributes in the whole given column of quasi-identifiers into a general domain. The representative heuristic algorithm Datafly[6] implements k-anonymity by full-domain generalization. Incognito [7] is also based on full-domain generalization. However, full-domain generalization often results with much information loss due to the fact that some satisfying k-anonymity records are still being generalized. Local recoding is partial anonymity for the generalized subtree. Reference [8] uses searching approach based on genetic algorithm, which do single-dimensional generalization by full-subtree recoding, yet its runtime is not acceptable. Reference [9] gives anonymity based on clustering a name of k-member, and also introduces two methods based on greedy algorithm and distance measure.

Top-down firstly partitions the data into equivalence classes, and then generalizes. Partition can be single-dimensional or multi-dimensional, and most partitions in k-anonymity at present are based on single-dimensional, for instance, in [10]. Single-dimensional partition is not so flexible, which can reduce the quality of anonymity. It doesn't perform well in preventing privacy attacks. When the data is scattered or k values small, single-dimensional one often falls to one attribute partitioning. In addition, single-dimensional method is so sensitive to the data that small change can influence the results significantly. Reference [11] proposes a multi-dimensional k-anonymity algorithm Mondrian that can do the partition on multiple attributes at the same time. It realizes anonymity by greedy algorithm, which helps to reduce

[1] Supported by Chongqing Municipal Natural Science Foundation (CSTC 2009BB2046)

© 2011 ACADEMY PUBLISHER
doi:10.4304/jsw.6.10.1937-1944

the time complexity, however makes no improvement for quality of anonymity.

To reduce the information loss, the data should be refined while satisfying the demand of data anonymity. Reference [12] makes a comparison of approximation algorithms, such as between APPROX-GR and GREEDY-K. Reference [13] and [14] give a proof that optimal k-anonymity on multiple attributes set is a NP-hard problem. Reference [15] employs the information entropy also proves the optimal k-anonymity on multiple attributes set to be a NP-hard problem.

To improve the k-anonymity while reduce the time complexity, the paper proposes a multi-dimensional k-anonymity algorithm based on mapping and divide-and-conquer strategy, which is top-down and multi-dimensional partition. Map data from multi-dimensional to single-dimensional, then implement k-anonymity on multiple attributes in polynomial time, finally employ the divide-and-conquer strategy.

Section 2 gives some definitions, and section 3 introduces the proposed algorithm and analyzes its time complexity. After that, a comparison experiment is done in section 4, and finally a conclusion is drawn in section 5.

II. RELATED DEFINITIONS

A. Definition 1: Quasi-Identifier

Given set U and relation table $T(A_1, A_2, ..., A_n)$, $f_c : U \to T$ and $f_g : T \to U'$, where $U \subseteq U'$. A quasi-identifier of T, QI is a set of attributes $(X_i, ..., X_j)$, and $(X_i, ..., X_j) \subseteq (A_1, A_2, ..., A_n)$ where: $\exists p_i \in U$ such that $f_g(f_c(p_i[QI])) = p_i$.

Quasi-identifier is a set of attributes which can identify one individual by linking extern information. These attributes value should be protected from privacy attack before release.

B. Definition 2: equivalence class

Equivalence class is a multiset on tuple[16] included in a table. $E = \{t_{r_1}, t_{r_2}, ...t_{r_m}\}$, t_{r_i} is a record in the relation T. For instance, equivalence class of relation T on set of attributes $(X_i, ..., X_j)$ is set of tuples of T on attributes $(X_i, ..., X_j)$ which share the same value. For each QI_p in QI, equivalence class satisfies $t_{r_i}[QI_p] = t_{r_j}[QI_p](i, j \in [r_1, r_m], i \neq j)$.

C. Definition 3: k-anonymity

Let $T(A_1,...,A_n)$ be a table and QI be the quasi-identifier associated with it. T is said to satisfy k-anonymity if and only if each sequence of values in T[QI] appears with at least k occurrences in T[QI].

D. Definition 4: k-anonymization

For original table T, if T' calculated from T satisfies k-anonymity property, the computing process from T to T' is K-anonymization. And the computing algorithm is k-anonymization algorithm.

E. Definition 5: k-partitioning

Let $T(A_1,...,A_n)$ be a table and QI be the quasi-identifier associated with it which contains d attributes. Tuple $t=(a_i,\ldots,a_{i+d})$ denotes d-dimensional points in d-dimension space. Partition T into m equivalence classes based on quasi-identifiers, n_i is the tuple count of the ith class. If for $\forall$ i, n_i satisfies $n_i \geq k$ and $n = \sum_{i=1}^{m} n_i$, the partitioning is k- partitioning that relation T on QI.

III. MULTI-DIMENSIONAL K-ANONYMITY ALGORITHM

Partitioning is a top-down method. Firstly, it takes a data point set S, $S = \{p_1,p_2,\ldots,p_n\}$, as an entry point, then partitions the data points into different sets according to certain rules. When doing a partitioning work, a nonempty subset A of the nonempty set S must follow these two rules: first, every elements of A is contained in S; second, if A_1 and A_2 are two different subset of S in one partition, then $A_1 \cap A_2 = \varnothing$. If a record of the data table is corresponded to a data point p_i in set S, k-anonymization based on partitioning can be realized by partitioning the data set into different subsets, according to quasi-identifier attributes $\{X_1,X_2,\ldots,X_n\}$. The subset R_i satisfier $\sum_{p_j \in R} f_{JUD}(p_j \in R_i) \geq k$,

$$f_{JUD}(e) = \begin{cases} 1 & (\text{if e exist}) \\ 0 & (\text{otherwise}) \end{cases}$$

The quality of k-anonymization depends on the size and distribution of subset (also called equivalence class).

K-anonymity can be achieved by single-dimensional or multi-dimensional partitioning method. Since single-dimensional partitioning is not flexible enough, it cannot get high-quality anonymity. It is a NP-hard problem to get an optimal k-anonymity on multiple attributes set. Thus, how to get an algorithm with better anonymity results and lower computation complexity becomes very important in research.

A. About the proposed algorithm

It is hard to directly work out the solution when the input scale n is a large number. Sd sometimes it is impossible to get the solution. The divide-and-conquer strategy works by partitioning a big problem into small ones and conquering them one by one. Usually the sub-problem is of the same kind with the big primary one, therefore, the divide-and-conquer process works by using recursive method.

Multi-dimensional k-anonymization based on mapping and divide-and-conquer strategy transforms the data table into data point set in multi-dimensional space which is constructed according to quasi-identifier attributes. If the data point set is dividable in multi-dimensional space, the result can be obtained. Otherwise, map the multi-dimensional points to each dimension to get single-dimensional data point sets. When doing the mapping, the number of data points that each dimension is mapped to single-dimensional set, expressed as Pro and number of multi-dimensional data points that each single-dimensional data point is mapped to, expressed as PPA, are recorded. Then compute the information dependency

© 2011 ACADEMY PUBLISHER

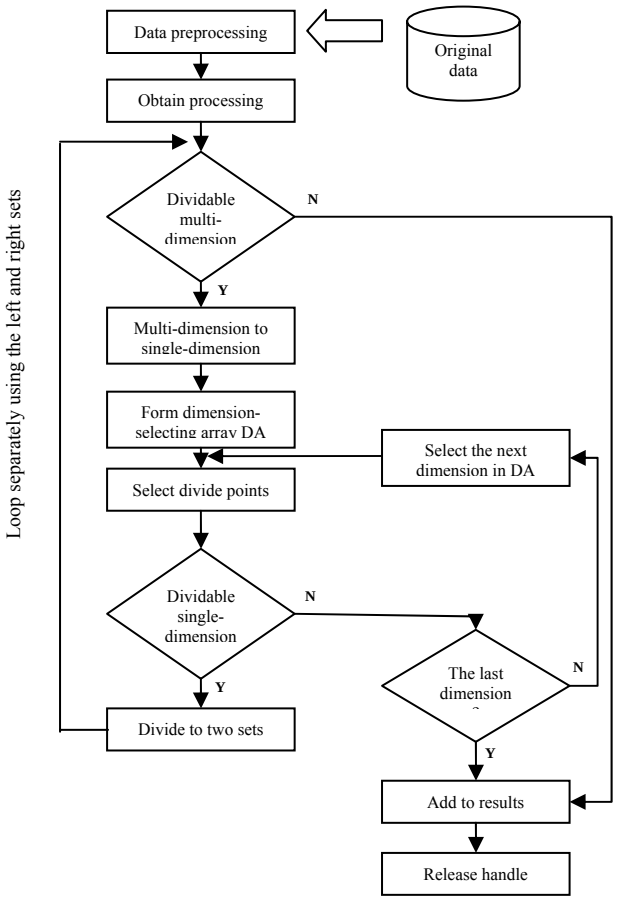


Figure 1. Work flow of the proposed algorithm

of each dimension based on Pro and PPA. After that, sort the information dependency in ascending order to form an array DA. Finally, using DA, k-anonymity can be realized by the divided points, which are selected by divide-and-conquer strategy. Recursively partition the subset, until that the subset is not dividable.

B. Work flow of the proposed algorithm

a. Data preprocessing

Since the original data from database is often noisy, missing and inconsistent, data preprocessing becomes very important. Data preprocessing aims to clear the data, fill or ignore the missing data, remove the noisy data, and correct the inconsistency of data.

Before applying k-anonymization on data table T ($A_1,A_2,\ldots,A_m$), every record in table should be transformed to a point p($x_1,x_2,\ldots,x_n$) in n-dimensional space, according to quasi-identifier attributes QI($X_1,X_2,\ldots,X_n$). For the non-numeric value, transforms it to numeric type. For example, gender can be valued by 0 and 1. Avoid using uncertain data so as to reduce the loss of information. If the attribute values are disperse, a min-max normalization transformation is needed to keep the relative value of original data, otherwise, attribute with larger initial value domain will get greater weight than the one with smaller initial value domain. Assuming that $\min_{Ai}$ and $\max_{Ai}$ are the minimum and maximum value of attribute Ai, the formula for getting a mapping value v' in interval [$\min_{Ai}$',$\max_{Ai}$'] from value of attribute Ai is as follows:

$$v' = \frac{v - \min_{Ai}}{\max_{Ai} - \min_{Ai}} (\max_{Ai}' - \min_{Ai}') + \min_{Ai}' \quad (1)$$

Data preprocessing is needed in order to map from multi-dimension to single-dimension.

After preprocessing, multi-attribute dataset turns to a n $\times$d matrix D as follows. Here n is the size of dataset, while d stands for the dimension count

$$D_{n\times d} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1d} \\ x_{21} & x_{22} & \ldots & x_{2d} \\ \ldots & \ldots & \ldots & \ldots \\ x_{n1} & x_{n2} & \ldots & x_{nd} \end{bmatrix}.$$

b. Dividable multi-dimension

For the point set D in n-dimensional space $\{X_1,X_2,\ldots,X_n\}$, if and only if exist C_{ij}, which satisfies $\sum_{m=1}^{n} f_{JUD}(x_{mj} > C_{ij}) \geq k$ and $\sum_{m=1}^{n} f_{JUD}(x_{mj} \leq C_{ij}) \geq k$, the n-dimensional space is dividable at the axis of $X_i = C_{ij}$. To each dimension of the space it is vertical to the axis. If there is no repeated point in multi-dimensional set D' and $|D'| \in [k, 2k-1]$, D' is minimum-partitioned in multi-dimensional space. The following gives proof of the statement.

Step 1: Assuming | D'|<k, obviously it doesn't meet the condition of k- anonymity, so | D'| must greater than k.

Step 2: Assuming $|D'| \geq 2k$. If there exists k points selected from D', which can partition D' to a couple of sets R_1 and R_2, $|R_1| = k$, $|R_2| = |D'| - |R_1| \geq 2k - k = k$, obviously, the partitioning performs better, which contradicts to the original statement of minimum partitioning.

Otherwise, $|D'| \leq 2k-1$, for any R_1 partitioned from D', if $|R_1| \geq k$, then $|R_2| = |D'| - |R_1| \leq 2k-1-k = k-1 < k$, It doesn't meet the condition of k-anonymity. Thus, $|D'| \leq 2k-1$.

c. Dividable single-dimension

For a point set S in single-dimension mapping from point set D in multi-dimension, if and only if there exists C_i, which satisfies $\sum_{p_j \in S} f_{JUD}(p_j.X_i > C_i) \geq k$ and $\sum_{p_j \in S} f_{JUD}(p_j.X_i \leq C_i) \geq k$, S can be partitioned at the axis of $X_i = C_i$. Single-dimensional k-anonymity partitioning is a process of continuous partitioning operation on set in single-dimensional space.

d. Mapping from multi-dimension to single-dimension

Define D_X as the domain of attribute X in dataset D. Using the function $D_{X_i} = mapping(D)$, data point in multi-dimension is mapped to each single-dimension $\{X_1,X_2,\ldots,X_d\}$. Map each point in multi-dimensional space to single-dimension Ai, and get d mapping domain D_X, expressed as:

© 2011 ACADEMY PUBLISHER

$$D_{X_i}=\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix} \times C_i = \begin{bmatrix} d_1 \\ d_2 \\ \dots \\ d_n \end{bmatrix}$$, C_i is a d×1 matrix,

and

$$C_i=\begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_d \end{bmatrix}$$ (c_i =1 while the rest = 0 in the matrix.)

In the process of mapping, the number of data points that each dimension is mapped to single-dimensional set, Pro, and number of multi-dimensional data points that each single-dimensional data point is mapped to, PPA, should be recorded.

For the process matrix $$P_{n\times n}=\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ d_1-d_2 & 1 & 1 & \dots & 1 \\ d_1-d_3 & d_2-d_3 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 1 \\ d_1-d_n & d_2-d_n & \dots & d_{n-1}-d_n & 1 \end{bmatrix},$$

define the product of each column as $MC_i=\prod_{j=1}^{n} p_{ji}$.

Define

$$Pro=\sum_{i=1}^{n-1} V_i \text{, among which } V_i=\begin{cases} 0, (MC_i=0) \\ 1, (otherwise) \end{cases} \quad (2)$$

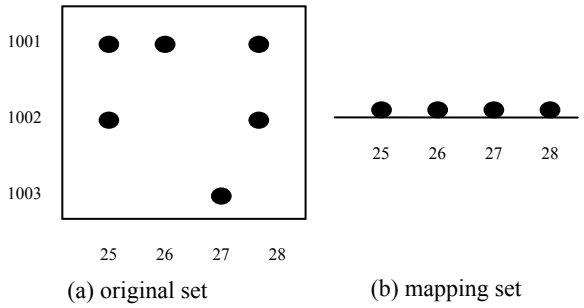
and

$$PPA=\{y_m \mid y_m=\sum_{i=1}^{n}\sum_{j=1}^{n} f_{JUD}(x_{ij}=d_m), 1\le m\le Pro, d_m\in D_{X_i}\} \quad (3)$$

Pro is used to work out the dimension-selecting array DA, single-dimension with less information changes during the mapping work takes priority to start the k-anonymity partitioning. PPA is used to select the partition points in divide-and-conquer strategy. For example, in Fig. 2, points in two-dimension are mapped to single-dimension, and it obtains four single-dimensional points numbered 25, 26, 27 and 28. Thus, pro gets the value of 4, while PPA is recorded as {2, 1, 1, 2}.

e. Getting dimension-selecting array

During the process of k-anonymity, the generalization to attribute value is needed. The closer that the values are, the less the information loses. To reduce the degree of information loss, while keeping the availability at the most, tuples with closer value should be partitioned into the same set, which depends on dimension selecting. The following gives the definition of information dependency to measure information change, which severs as the reference of dimension selecting.



(a) original set (b) mapping set

Figure 2 mapping from two-dimension to single-dimension

$$DEP(x,y)=w_1\times\frac{1}{k}\times\sum_{i=1}^{n}\frac{y_i}{x}+w_2\times\frac{1}{k}\times\sqrt{\frac{\sum_{i=1}^{n}(y_i-\bar{y})^2}{n-1}} \quad (4)$$

Here x represents Pro, y stands for PPA, and y_i is the tuple of PPA. w_1 is the weight of the impact that Pro takes on the information change, while w_2 is weigh of the impact that dispersion degree of PPA takes on information change. They are defined as:

$$w_1=\max_{i\in[1,d]}\{\sum_{j=1}^{n}\frac{y_{ij}}{x_i},\sqrt{\frac{\sum_{j=1}^{n}(y_{ij}-\bar{y}_i)^2}{n-1}}\} \quad (5)$$

$$w_2=\frac{1}{w_1} \quad (6)$$

Compute the information dependency of each dimension, and sort them in ascending order. The sorted array is now dimension-selecting array DA, and k-anonymity begins prior from dimension with minimum DEP. DA can help to select a targeted partition dimension. After a successful division, the only thing to do is to insert value $DEP_i(x_i,y_i)$ of the changed dimension into the array, no need to sort the array again, which significantly reduces the compute complexity.

f. Divide point selection in the divide-and-conquer strategy

There are two kinds of methods for selection of divided point in divide-and conquer strategy. One is to select the median point which is frequently used, and the other is to select pre-k points according to the requirement of k-anonymity.

Selecting the median point is a more loose way. It selects the midpoint of number of single-dimension points to partition. After the partitioning, the left and right sets have the same size, that is lRegion = rRegion (lRegion = rRegion + 1 when size of singe-dimension set is odd) . Therefore, in the mapping domain DX = $\{d_1,d_2,\dots,d_n\}$, the position for dividing is given :

$$L_m=\begin{cases} \dfrac{d_{\frac{n+1}{2}}+d_{\frac{n+1}{2}+1}}{2} & (\text{ if n is odd }) \\ \dfrac{d_{\frac{n}{2}}+d_{\frac{n}{2}+1}}{2} & (\text{ otherwise }) \end{cases} \quad (7)$$

Another way for divide point selection is based on k-anonymity property, and it applies the recorded PPA, PPA = $\{x_1,x_2,\dots,x_m\}$. In the mapping domain D_X, the location for dividing is defined as $L_k=\dfrac{d_{close\_k}+d_{close\_k+1}}{2}$, in which, the divide point d_{close_k} is given like:

© 2011 ACADEMY PUBLISHER

$$d_{close\_k}=\begin{cases} d_{pre},(\sum_{i=1}^{pre}x_i \le \sum_{j=fin}^{m}x_j,|D|>3k-1) \\ d_{fin},(\sum_{i=1}^{pre}x_i > \sum_{j=fin}^{m}x_j,|D|>3k-1) \\ d_{entropy},(|D|\le 3k-1) \end{cases} \quad (8)$$

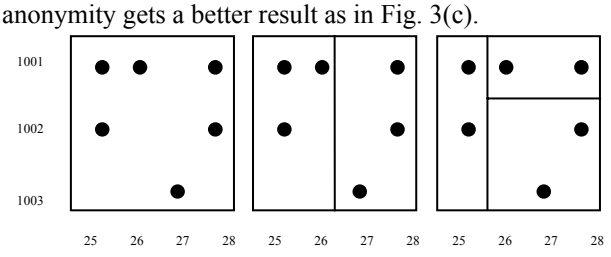
When the set D can still be divide for many times(|D|>3k-1), select the point from either d_{pre} or d_{fin} which can result smaller subsets as divide point, so as to reduce information loss. d_{pre} represents the divide point that x_i corresponds to, and i (≤ m)takes the smallest value while satisfies $\frac{x_1}{k}+\frac{x_2}{k}+\cdots\frac{x_i}{k}=\frac{1}{k}\times\sum_{p=1}^{i}x_p \ge 1$. While d_{fin} signifies the divide point that x_j corresponds to, and j (≤ m)takes the greatest value while satisfies inequality $\frac{x_j}{k}+\frac{x_{j+1}}{k}+\cdots\frac{x_m}{k}=\frac{1}{k}\times\sum_{p=j}^{m}x_p \ge 1$.

When the set D is on the last divide step (|D|≤3k-1), entropy is introduced to measure the uncertainty of information, so as to get a better distributed divide. For a sample set S, assuming that quasi-identifier attributes take m different values, then define class C_i (i=1,…,m). If s_i is sample size of different C_i, the entropy for this sample can be computed from the expression below:

$$I(s_1,s_2,...,s_m)=-\sum_{i=1}^{m}p(s_i)\log_2(p(s_i)) \quad (9)$$

Here $p(s_i)$ stands for the probability that a random sample belongs to C_i , obtained from $p(s_i)=\frac{s_i}{|S|}$. The greater entropy values, the more uncertainty the sample is of, and the better that divide result distributes. Compute the different entropy of the divided set by taking each single-dimension point as divide one, thus, the point results the greatest entropy is $d_{entropy}$, the selected divide point.

The upper two methods can both divide k-anonymity problem into sub-ones, and then recursively using the divide-and-conquer till the sub-problem is soluble. However, the latter selection performs better in some circumstances. For example, in Fig. 3(b), 2-anonymization partition the original set in Fig. 3(a), employing median selection, and it cannot be further partitioned just after the first partitioning. While selecting Close-k points according to the requirement of k-anonymity gets a better result as in Fig. 3(c).



(a) original set (b) Median (c) Close-k

Figure 3 Selection for divide point

C. Algorithm and time analysis

The following gives the description of the proposed algorithm:

Input: original dataset pointSet, multi-dimensional space regIn, anonymization degree k.

Output: dimensional sets after anonymization, named regOut

```
Anonymize ( pointSet , regIn , k )
If ( regIn not diviable multiple dimenbsion )
  Return regIn → results regOut
Else
  For ( int i = 0 ; i < d ; i++ )
    Mapping to single dimension DA ← mapping( D )
  End for
  DA ← chooseDimArray ( Pro , PPA )
  For ( int i = 0 ; i < DA.length ; i++ )
    splitPoint ← findSplit ( pointSet , DA[i] , k )
    If ( dividable single dimension )
      lRegion ← { t □ regIn : t. DA[i] □ splitPoint }
      rRegion ← { t □ regIn : t. DA[i] > splitPoint }
      Break  out of loop
    Else if ( i = DA.length – 1 )
      Return regIn → results ragout
    End if
  End for
  Return Anonymize (pointSet , lRegion , k)□Anonymize (pointSet , rRegion , k)
End if
```

The algorithm is of a recursive process, and the time for left and right sets iteration are $T(lRegion)$ and $T(rRegion)$. The time complexity for mapping multi-dimension to single-dimension is $O(n\times d)$ (n is size of dataset, and d is dimension count). Dimension is sorted by quick sort, and its time complexity is $O(d\times\log d)$. Looking for divide points in the loop takes time of $O(n\times d)$. Thus, the time complexity of the whole algorithm can be obtained from the equation below:

$$T(n)=O(n\times d)+O(d\times\log d)+O(n\times d)+T(lRegion)+T(rRegion) \quad (10)$$

In general, n is much greater than d, then

$$T(n)=T(lRegion)+T(rRegion)+O(n) \quad (11)$$

Consider two extreme cases: size of the left set is the same as the right one after division; size of the left set is k, and the right one is n-k.

Case 1: size of the left set is the same as the right one after division. Then the time complexity is: $T_1(n)=2\times T(\frac{n}{2})+O(n)=2\times(2\times T(\frac{n}{2^2})+O(\frac{n}{2}))+O(n)$, By simplification, it is $T_1(n)=2^x\times T(\frac{n}{2^x})+x\times O(n)$. For $T(k)=1$, $\frac{n}{2^x}=k\Leftrightarrow x=\log_2\frac{n}{k}$, then the time complexity is computed as $T_1(n)=2^{\log_2\frac{n}{k}}\times T(\frac{n}{2^{\log_2\frac{n}{k}}})+\log_2\frac{n}{k}\times O(n)=O(n\times\log n)$.

© 2011 ACADEMY PUBLISHER

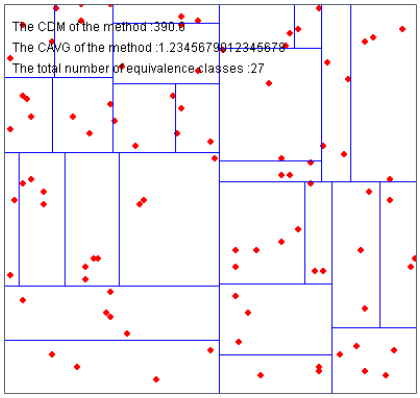


Figure 4 Median

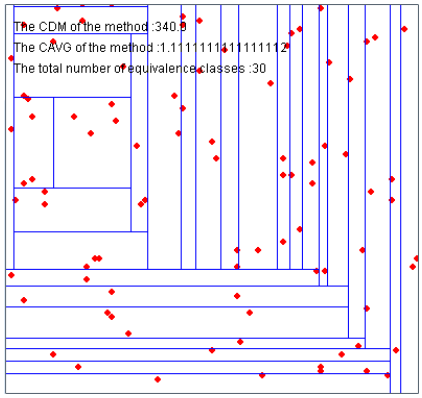


Figure 5 Close-k

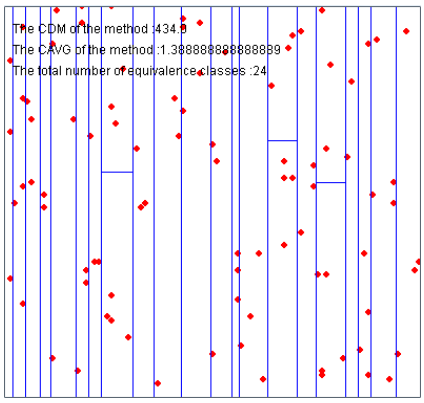


Figure 6 Single

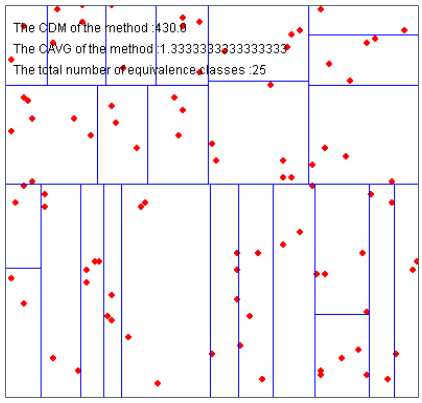


Figure7 Mondrian

Case 2: size of the left set is k, and the right one is n-k. Then the time complexity is:

$$T_2(n) = T(k) + T(n-k) + O(n) = x \times T(k) + T(n - x \cdot k) + x \times O(n)$$

Make $n - x \cdot k = k$, then

$$T_2(n) = \frac{n-k}{k} \times T(k) + T(n - \frac{n-k}{k} \cdot k) + \frac{n-k}{k} \times O(n),$$

equals to $T_2(n) = n \times T(k) + \frac{n-k}{k} \times O(n) = O(n^2)$.

From the upper two cases, we can get $T_1(n) = O(n \times \log n)$, and $T_2(n) = O(n^2)$. The practical time complexity is surely between these two extreme cases. Since $T_1(n)$ and $T_2(n)$ is acceptable, the time complexity of the proposed algorithm is satisfactory.

IV. Experiments and Results

To give the algorithm a better measure and minimize the deviation from data, the experiment is done separately using experiment data and practical data. With experiment data, a visual plane graph is shown as result in 2-dimension case, A comparison of results by four different methods is made in multi-dimension. In the second approach, we use the Adults database from the UC Irvine Machine Learning Repository [17].

A. Measurements

There mare many measurements for k-anonymity depending on the needs. This paper takes visual and widely used discernability metric (C_{DM}) and average size of equivalence class (C_{AVG})[10][11], as the measurements.

Discernability metric (C_{DM}) assigns a penalty to each tuple based on how many tuples in the transformed dataset are indistinguishable from the quasi-identifiers. The following is the definition:

$$C_{DM} = \sum_{\forall E \text{ s.t. } |E| \geq k} |E|^2 + \sum_{\forall E \text{ s.t. } |E| < k} |D| \cdot |E| \qquad (12)$$

In this expression, the set E refers to the equivalence classes of tuples in D induced by the anonymization. The first sum computes penalties for each non-suppressed tuple, the second for suppressed tuples. |E| is the size of equivalence classes, and |D| is the size of the input dataset.

Average size of equivalence class (C_{AVG}) is based on equivalence class in k-anonymity. The closer that average number of records in equivalence is to k, the lower the information loss is. The definition is:

$$C_{AVG} = \left(\frac{Total\_records}{Total\_equivalence\_classes}\right) / (k) \qquad (13)$$

According to k-anonymity requirements, size of every equivalence class cannot be less than k. So we modify C_{AVG} to C_{MDM}, considering the influence of k. The C_{MDM} is as follows:

$$C_{MDM} = \sum_{EquivclassesE} (|E| - k)^2 \qquad (14)$$

In the best case, size of all the equivalence classes is k, and $C_{MDM} = \frac{n}{k} \times (k-k)^2 = 0$. In the worst case of not considering the repeated points, size of all equivalence class is 2k-1, and $C_{MDM} = \frac{n}{2k-1} \times (2k-1-k)^2 = \frac{(k-1)^2}{2k-1} n$.

© 2011 ACADEMY PUBLISHER

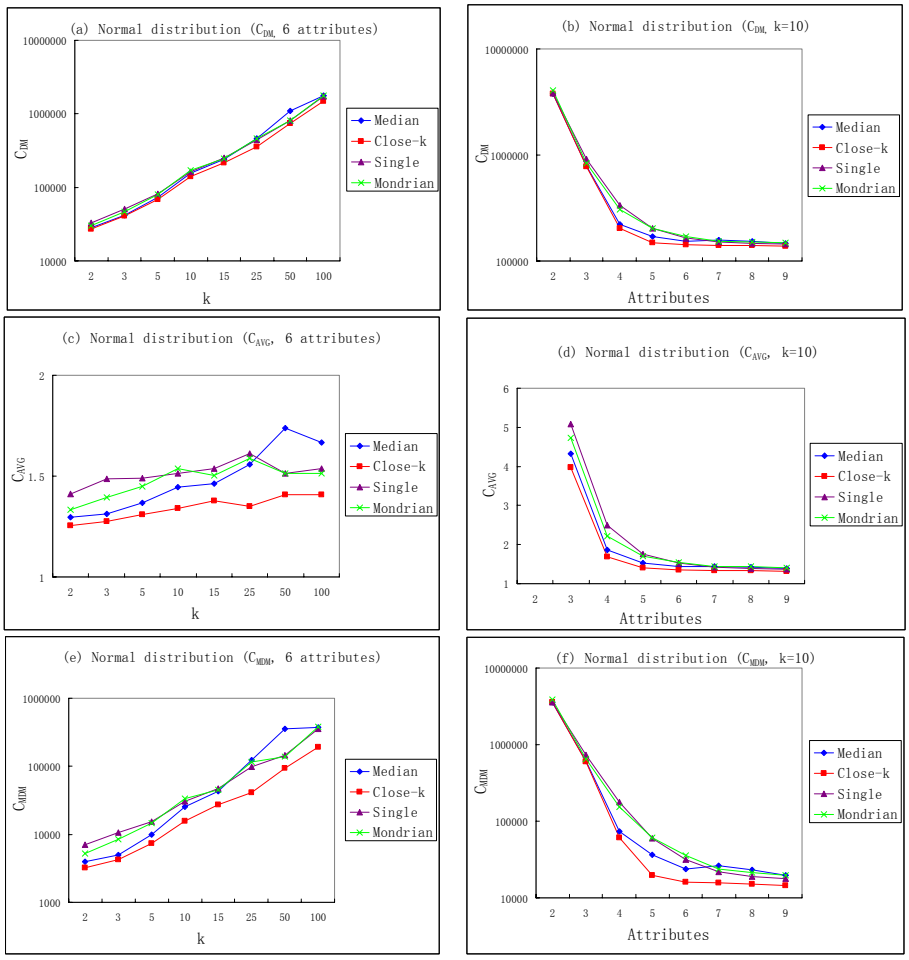


Figure 8 Results comparison in different cases

B. *Results from experiment data*

Since plane graph is visual, the first experiment is on two attributes (2-dimension), and k=3. Four methods of partition are used on 100 evenly discrete distributed tuples, which results as Fig. 4 to 7. Fig. 4 and 5 are separately the results of partition based on median and Close-k points. Fig. 6 is result of single-dimensional partitioning [8][10], and Fig. 7 shows what Mondrian works out[11].

C_{DM} of these four methods are: 390.0, 340.0, 434.0, 430.0, and number of the partitioned equivalence classes are: 27, 30, 24, 25.The less C_{DM} values, the more even the partition is, the better the anonymity performs. Since size of equivalence classes is not less than k, $C_{DM} \geqslant k \times C$, C is number of equivalence classes. The graph shows that multi-dimensional partitioning based on mapping employing divide-and-conquer strategy performs better, while single-dimensional partitioning is easier to fall on one attribute.

For general case, we randomly generate some multi-dimensional data obeys discrete normal distribution for experiment. 10000 tuples are generated at first. Fig. 8 (a)(c)(e)is a comparison when the number of attributes is 6 for all while k varies; Fig. 8 (b)(d)(f) is a comparison when k is 10, and the number of attributes varies; Fig. 8 (a)(b) shows the different C_{DM}, Fig. 8 (c)(d) compares the difference of C_{AVG} , while Fig. 8 (e)(f)is a comparison of C_{MDM}.

Fig. 8 (a)(e) shows that for all the methods on the same quantity of data, C_{DM} and C_{MDM} grows with k. Close-k gets a relative small value of C_{DM}, which means a good quality of anonymity. Fig. 8 (b)(f)shows that with the attributes count increases, C_{DM} and C_{MDM} decrease with dimension gets bigger, original data becomes sparser, and repeated tuples appear in less frequent. The graph also manifests that on the same quantity of data, the more the attributes are, the faster C_{DM} of multi-dimensional k-anonymity based on mapping employing divide-and-conquer decreases than that of single-dimensional method. That is to say, multi-dimensional k-anonymity based on mapping employing divide-and-conquer performs better when count of attributes is greater. Fig. 8 (c)(d)is comparison of C_{AVG} in different cases. Since $C_{AVG} \geqslant 1$, the closer that C_{AVG} is to 1, the better result it gets in anonymity.

C. *Results from practical data*

The Adult database[17] from UCI is used for experiment. Since the database may contain some noisy data, we choose six attributes (Age, Sex, Education, Native country, Salary Class, Occupation) for use. Transfer each record in Adult database to points in multi-dimensional space, and delete the incomplete ones at the

© 2011 ACADEMY PUBLISHER

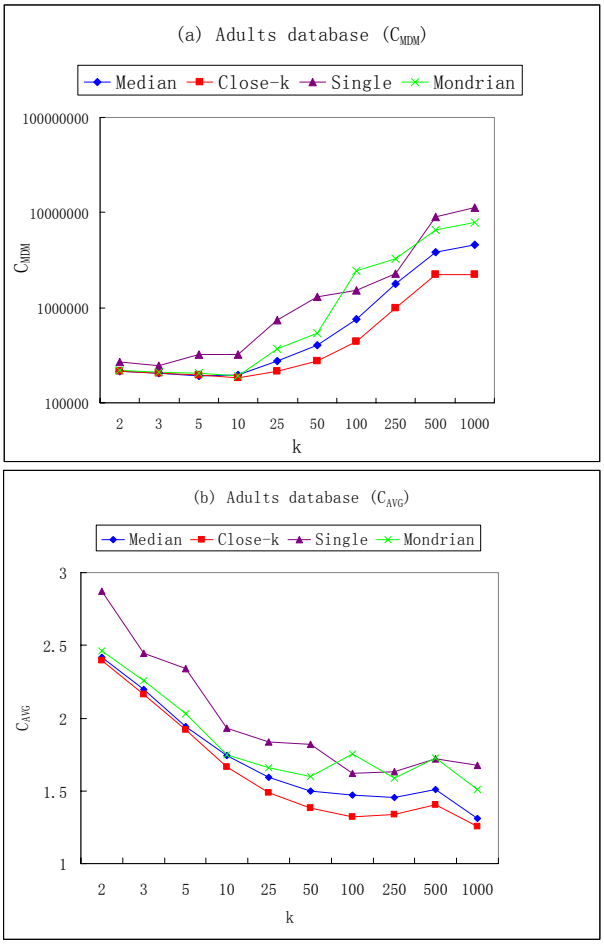


Figure 9 Results on Adult database

same time. After preprocessing, 30162 valid records remain.

Fig. 9 shows the results on Adult database when k gets different values. It indicates that multi-dimensional k-anonymity based on mapping is also of high-quality in anonymity on practical data.

V. CONCLUSION

The method selection for multi-dimensional k-anonymization is significant because it determines the quality and efficiency of algorithm. Multi-dimensional k-anonymity algorithm based on mapping maps multi-dimension to single-dimension, and records some mapping information for partition dimension selection and single-dimensional sets divide. Convert information change to information dependency, and then select the divide dimension based on it to improve the accuracy for divide dimension selection. The paper gives two strategies towards divide point selection. Median is widely used in divide and Close-k satisfies k-anonymity property. Multi-dimensional k-anonymity based on mapping can get a higher quality k-anonymity in polynomial time, which is practical for use. However, the selection for quasi-identifiers in this paper has an effect on information loss by anonymization. The follow-up work is to focus on the minimum quasi-identifier attribute set selection so as to reduce the information loss, and also give privacy protection control of individuals.

REFERENCES

[1] Sweeney L. “K-Anonymity: A model for protecting privacy,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10 (5), pp. 557-570.
[2] Rashid A.H. “Protect privacy of medical informatics using k-anonymization model,” Hegazy A.F. Informatics and Systems (INFOS), the 7th International Conference on, 2010, pp. 1-10.
[3] Sacharidis Dimitris, Mouratidis Kyriakos, Papadias Dimitris. “K-anonymity in the presence of external databases,” IEEE Transactions on Knowledge and Data Engineering, v 22, n 3, 2010, pp. 392-403.
[4] Samarati P. “Protecting respondents’ identities in microdata release.,”Proc of the TKDE’ 01, 2001, pp. 1010-1027.
[5] Samarati P, Sweeney L. “Generalizing data to provide anonymity when disclosing information,” Proc of the 17th ACMSIGMODSIGACT - SIGART Symposium on the Principles of Database Systems, Seattle,WA, USA, 1998, pp. 188.
[6] Sweeney L. “Achieving k-anonymity privacy protection using generalization and suppression,” International Journal of Uncertainty, Fuzziness and Knowledge2Based Systems, 2002, 10 (5), pp. 571-588.
[7] LeFevre K, DeWitt D J, Ramakrishnan R. “Incognito: Efficient full-domain k-anonymity,” ACM SIGMOD International Conference on Management of Data.Baltimore, USA: ACM, 2005, pp. 49-60.
[8] Iyengar V. “Transforming Data to Satisfy Privacy Constraints,” Proc. of the ACM SIGKDD. USA: [s. n.], 2002, pp. 279-287.
[9] Byun Ji-Won, Kamra Ashish, Bertino Elisa, Li Ninghui. “Efficient k-anonymization using clustering techniques,” Lecture Notes in Computer Science, 2007, pp.188-200.
[10] Bayardo R, Agrawal R. “Data privacy through optimal k-anonymization,” In ICDE, 2005.
[11] LeFevre K, DeWitt D J, Ramakrishnan R. “Mondrian multidimensional k-anonymity,” IEEE International Conference on Data Engineering.Atlanta,USA:IEEE,2006.
[12] Park Hyoungmin, Shim Kyuseok. “Approximate algorithms with generalizing attribute values for k-anonymity,” Information Systems, 2010, pp. 933-955.
[13] Meyerson A, Williams R. “On the Complexity of Optimal K-anonymity,” Proceedings of the ACM SIGMOD-SIGACTSIGART Conf. on Principles of Database Systems. New York, USA: ACM Press, 2004, pp. 223-228.
[14] Aggarwal G, Feder T. “Approximation Algorithms for K-anonymity,”Journal of Privacy Technology, 2005, 12(1), pp. 78-94.
[15] Gionis A, Tassa T. “k-Anonymization with Minimal Loss of Information,” Knowledge and Data Engineering, IEEE Transactions on, 2009, pp. 206-219.
[16] Qian Wang, Xiangling Shi. “(a, d)-Diversity: Privacy Protection Based on l-Diversity,” 2009 WRI World Congress on Software Engineering, WCSE 2009, pp. 367-372.
[17] C. Blake and C. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/~mlearn/M1-Repository.html.

© 2011 ACADEMY PUBLISHER