# Protecting Privacy by Multi-dimensional K-anonymity[+]

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*—Privacy protection for incremental data has a great effect on data availability and practicality. K-anonymity is an important approach to protect data privacy in data publishing scenario. 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. Now research on k-anonymity mainly focuses on getting high quality anonymity while reducing the time complexity, and new method of realization of k-anonymity properties according to the requirement of published data. Although most k-anonymity algorithms perform well on static data, their effects decrease when they are on the changing data of real world. This paper proposes a multi-dimensional k-anonymity algorithm based on mapping and divide-and-conquer strategy that is feasible and performs much better in k-anonymity. The second main contribution of this paper is an effective k-anonymity method based on incremental local update on large dataset. It incrementally updates the changing dataset, and a threshold is set to assure the stability of update. Neighbor equivalence sets and similar equivalence sets are computed by their position, which not only avoids the cost of recalculation aroused by little change of dataset, but also improves the practical application performance since the dataset satisfies k-anonymity properties. The experiment shows that the proposed algorithm has a better performance in both time cost and anonymity quality, compared to the methods at present.**

***Index Terms*—security, k-anonymity, multi-dimension, incremental update**

## I. INTRODUCTION

Privacy protection is now a great challenge in the modern information era, since more and more micro datasets, which includes tuples of different individuals, have been published for various purposes. The changes of dataset may result in some security breaches for attackers, who spend their energy maliciously to collect some privacy information and recognize some certain individuals. And update for dataset may increase the computation work, while the practicality and anonymity of dataset should be guaranteed.

There have been some technologies for privacy protection [1] [18], among which, k-anonymity is widely used. In k-anonymity, published dataset must satisfy the properties requirement that each tuple of the dataset cannot be distinguished with the other k-1 tuples. 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 [23]. How to improve the anonymity and minimize information loss, meanwhile reduce the time and space complexity, is now the focus of k-anonymity. At present k-anonymity properties has been introduced and researched a lot [9] [19], and 1-diversity [15] [20] becomes one trend of its development.

There are kinds of k-anonymity methods for static dataset, and they make the anonymity through generalization and concealing technologies [3][4][5]. The anonymity quality is a measurement for k-anonymity, besides, the information loss can also be taken into consideration, since the measurement of information loss is a significant standard for degree of anonymity [17] [21] [22]. Some clustering methods are adopted to form a clustering model in k-anonymity [8] [24]. Many researchers do research on k-anonymity and have proposed various ways to implement k-anonymity. Methods for k-anonymity can be divided into two groups: bottom-up and top-down.

The representative heuristic algorithm Datafly[5] implements k-anonymity by full-domain generalization. Incognito [6] is also based on full-domain generalization. Reference [7] uses searching approach based on genetic algorithm, which do single-dimensional generalization by full-subtree recoding, yet its runtime is not acceptable. Most partitions in k-anonymity at present are based on single-dimensional, for instance, in reference [9]. Single-dimensional doesn't perform well in preventing privacy attacks. Reference [10] proposes a multi-dimensional k-anonymity algorithm Mondrian that can do the partition on multiple attributes at the same time. Reference [11]

[+] Supported by Chongqing Municipal Natural Science Foundation (CSTC 2009BB2046)
Project No.CDJXS11180018 Supported by the Fundamental Research Funds for the Central Universities


doi:10.4304/jsw.7.8.1873-1880

makes a comparison of approximation algorithms. Reference [12] and [13] give a proof that optimal k-anonymity on multiple attributes set is a NP-hard problem. Reference [14] employs the information entropy also proves the optimal k-anonymity on multiple attributes set to be a NP-hard problem.

Although most k-anonymity algorithms perform well on static data, their effects decrease when they are on the changing data of real world. When some new data are added to the published dataset, the privacy information it contains will change. To overcome this problem, these two methods are applied. One is global update, which recalculates the all new dataset, and gives up the calculated dataset before. It can obtain a better result of k-anonymity, while cost too much time. Particularly when the dataset changes frequently, the algorithm will suffer from a long time cost. The other one is local update. Reference [25] [26] firstly position the new data into equivalence group, and then do the k-anonymity work on the equivalence group. However, they neglect the relation that the new data have with the around equivalence groups, which will lower the quality of anonymity, and also bring much more information loss.

Section 2 gives some definitions, and section 3 introduces the proposed algorithm. Section 4 introduces the incremental local update method to keep the result consistency with data changing. After that, a comparison experiment is done in section 5, and finally a conclusion is drawn in section 6.

## 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-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.

### E. Definition 5: Generalization

Generalization on the quasi-identifiers is widely used in k-anonymity. It generates a generic value to replace the real value of attribute. Generalization is firstly used on categorical attributes, predefined domain and value generalization level, and then it is extended to numerical attribute generalization by using the predefinition level and the predefinition tree model.

## 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. 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 $\sum_{p_j \in R} f_{JUD}(p_j \in R_i) \geq k$, $f_{JUD}(e) = \begin{cases} 1\,(if\ e\ exist) \\ 0\,(otherwise) \end{cases}$. The quality of k-anonymization depends on the size and distribution of subset (also called equivalence class).

### A. About the Proposed Algorithm

Multi-dimensional k-anonymization based on mapping 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 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

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:

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

After preprocessing, dataset turns to a n×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.

*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:

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

$C_i = \begin{bmatrix} c_1 \\ c_2 \\ \ldots \\ c_d \end{bmatrix}$ ($c_i$ =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 $P_{n\times n} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ d_1 - d_2 & 1 & 1 & \ldots & 1 \\ d_1 - d_3 & d_2 - d_3 & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & 1 \\ d_1 - d_n & d_2 - d_n & \ldots & d_{n-1} - d_n & 1 \end{bmatrix}$, 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

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

Pro is used to work out the dimension-selecting array DA. PPA is used to select the partition points in divide-and-conquer strategy.

*e. Getting dimension-selecting array*

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.

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

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.

*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 = $\{d_1,d_2,\ldots,d_n\}$, the position for dividing is given :

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

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

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

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:

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

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.

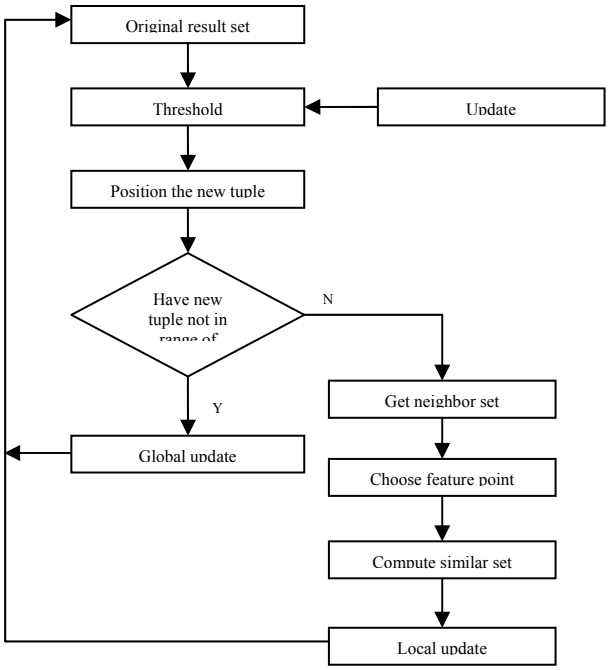


Figure 1 Work flow of incremental update

## IV. INCREMENTAL UPDATE

Since it is costly to update the anonymity result on all the dataset when there are new data added to, we adopt the local update method to keep the result consistency with data changing. The effective local update method in our algorithm can relatively ease the computation work, reduce the time cost, and meanwhile it satisfies k-anonymity properties and is much available in practical use.

### A. Threshold set

The threshold is set to be a reference standard which is related to the properties. It is a reference value which determines whether the system takes data changes into consideration.

A proper threshold value helps to improve the stability of system since it can avoid the big changes aroused by update of small amount of data. In this paper, we set threshold the value k-1 according to the range of equivalence class E, which defined as $T > 2k-1-k = k-1$. Because that when the equivalence group satisfies $|E| \in [k, 2k-1]$ , and there is no repeated point existed, E is minimum undividable. When the times of data update is above the threshold, the incremental update for the dataset is carried out, otherwise, keep the dataset for stability.

### B. Position the New Tuple into the Dataset

The update of tuple $t_u(x_1, x_2, \ldots, x_d)$ is actually for the multi-dimension points in set D on n-dimension space $\{X_1, X_2, \ldots, X_n\}$ . And these multi-dimension points should be positioned into the corresponding equivalence class based on quasi-identifier $\Delta^+ D$ . If there are still some multi-dimension points which have no

corresponding equivalence class, the update should be global, so as to bring all the new data into the algorithm. Otherwise, compute the similar multi-dimension points and local update the dataset can be fine.

### C. Get the Neighbor Set

When the new tuple is positioned into the equivalence class, if the k-anonymity is just on these corresponding equivalence classes, the anonymity result cannot be a satisfactory. Therefore, information relevance is introduced which gives the relevance of new data and neighbor sets a consideration, and it can help to reduce the information loss of dataset.

According to the definition of equivalence class, the range of it can be described as $E_{X_i} = \begin{bmatrix} x_{1_{\min}} & x_{1_{\max}} \\ x_{2_{\min}} & x_{2_{\max}} \\ \cdots & \cdots \\ x_{d_{\min}} & x_{d_{\max}} \end{bmatrix}$, in which, $x_{j_{\min}}$ is the minimum value that the equivalence class $E_{X_i}$ has got in j dimension while $x_{j_{\max}}$ is the maximum value, and here $1 \leqslant j \leqslant d$. If we define the equivalence class that the new tuple position into as $E_Y = \begin{bmatrix} y_{1_{\min}} & y_{1_{\max}} \\ y_{2_{\min}} & y_{2_{\max}} \\ \cdots & \cdots \\ y_{d_{\min}} & y_{d_{\max}} \end{bmatrix}$, then the information relevance between $E_X$ and $E_Y$ can be computed by the expression below.

$$REL(E_X, E_Y) = \begin{cases} \sum_{p_i \in D} f_{IUD}(p_i \in E_X), (\exists j \in [1,d], x_j \in [y_{j_{\min}}, y_{j_{\max}}] \cup y_j \in [x_{j_{\min}}, x_{j_{\max}}]) \\ \frac{\sum_{p_i \in D} f_{IUD}(p_i \in E_X)}{k}, (\exists j \in [1,d], x_{j_{\min}} \le y_{j_{\min}} \le x_{j_{\max}} \le y_{j_{\max}} \cup y_{j_{\min}} \le x_{j_{\min}} \le y_{j_{\max}} \le x_{j_{\max}}) \\ 0, (otherwise) \end{cases} \quad (10)$$

Thus, the neighbor set of EY is given as

$$E_N = \{E_{X_i} \mid REL(E_{X_i}, E_Y) > 0, E_{X_i} \in R\} \quad (11)$$

The neighbor set $E_N$ may have these three kinds of relationship with $E_Y$, which are shown in the Fig. 2.

Here Fig. 2(a) shows $E_N$ is included in $E_Y$, and Fig. 2(b) describes that $E_N$ has intersection with $E_Y$, and Fig. 2(c) indicates that $E_N$ includes $E_Y$.

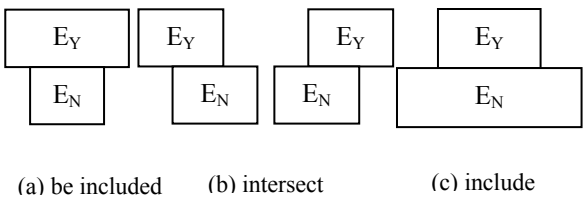


Figure 2 The relationship that $E_N$ may have with $E_Y$

### D. Choose a Feature Point in Each Neighbor Set

In each neighbor set, we need to choose a feature point which is used in similarity computation with new multi-dimension points. Firstly, construct a complete graph from the neighbor sets, then compute the shortest path of each pair of vertices, after that choose the point which has got the minimum sum of paths length to the other vertices, and the chosen one is exactly the feature point. The similarity computation helps to determine the range that local update would function. Assuming that there are m tuples in neighbor set E, then the adjacency matrix of its complete graph is:

$$A_{m \times m} = \begin{bmatrix} a_{[0][0]} & a_{[0][1]} & \cdots & a_{[0][m-1]} \\ a_{[1][0]} & a_{[1][1]} & \cdots & a_{[1][m-1]} \\ \cdots & \cdots & \cdots & \cdots \\ a_{[m-1][0]} & a_{[m-1][1]} & \cdots & a_{[m-1][m-1]} \end{bmatrix}$$, in which $a_{ij} = \sqrt{\sum_{p=1}^{d} (x_{i_p} - x_{j_p})^2}$ stands for the distance between $v_i$ and $v_j$.

Define $path[i][j]$ as a path from $v_i$ to $v_j$, then the path may not be the shortest one, therefore, m times of update are needed to get the shortest path.

For a path $(v_i, v_0, v_j)$, compare the path length of $(v_i, v_j)$ and $(v_i, v_0, v_j)$, and the shorter one will be the shortest path from $v_i$ to $v_j$ and the serial number of mid node is not greater than 0. Assuming that $v_1$ is in the path, then when $(v_i, ..., v_1)$ and $(v_1, ..., v_j)$ are the shortest paths and serial number of their mid node is not greater than 0, $(v_i, ..., v_1, ..., v_j)$ can be the shortest path from $v_i$ to $v_j$ and the serial number of mid node is not greater than 1.Then add $v_2$ into the path to get the shortest path, followed by analogy.

Therefore, it can be concluded that when $(v_i, ..., v_h)$ and $(v_h, ..., v_j)$ are the shortest paths and serial number of their mid node is not greater than h-1, $(v_i, ..., v_h, ..., v_j)$ can be the shortest path from $v_i$ to $v_j$ and the serial number of mid node is not greater than h. The shortest path from $v_i$ to $v_j$ can be obtained by m times of update.

Define an array of n-order matrix as $P^{(-1)}, P^{(0)}, ..., P^{(h)}, ..., P^{(m-1)}$. And in which,

$$P^{(-1)}[i][j] = A.path[i][j] \quad (12)$$

$$P^{(h)}[i][j] = Min\{P^{(h-1)}[i][j], P^{(h-1)}[i][h] + P^{(h-1)}[h][j]\} \ (0 \le h \le m-1) \quad (13)$$

Now the shortest path between each pair of vertices can be computed. $P^{(h)}[i][j]$ is the length of the shortest path from $v_i$ to $v_j$ and the serial number of mid node is not greater than h, while $P^{(m-1)}[i][j]$ stands for the length of the shortest path from $v_i$ to $v_j$.

The following expression describes the feature point pf in a neighbor set, which has got the minimum sum of the shortest path lengths that from $p_f$ to all the other vertices.

$$p_f = \{v_h \mid \forall j \in [0, m-1], \sum_{i=0}^{i=m-1} P^{(m-1)}[h][i] \le \sum_{i=0}^{i=m-1} P^{(m-1)}[j][i]\} \quad (14)$$

### E. Get the Similar Sets

Since the incremental update should make a balance with the quality of k-anonymity, we do the local update work on the attributes with similar value. A modified similarity computation for the sets EX and EY is defined as:

$$sim(E_X, E_Y) = w \times \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}} \quad (15)$$

Where x and y are respectively the feature point in $E_X$ and $E_Y$. $x_i$ and $y_i$ stand for the values they've got in the ith dimension. W is a modified weight for similarity computation, which is defined as:

$$w = \frac{REL(E_X, E_Y)}{\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}} \quad (16)$$

Based on the similarity computation method, the similarity between the feature point in neighbor matrix and the new tuple can be obtained, and get the topN similar ones as the similar sets. The following local update work is on these similar sets, so as to reduce the time cost.

### F. Algorithm Description

The proposed incremental update algorithm can be descripted as:

Input: *OD*,which includes multi-dimension space reg, original dataset pointSet and anonymity degree k;

$\Delta^+ D$ , which includes the new added dataset newPointSet

Output: k-anonymity results

```
IncrementalUpdate ( reg , pointSet , k , newPointSet )
  If ( newPointSet.size < T )
    Return
  End if
  For ( int i = 0 ; i < newPointSet.size ; i ++ )
    If( newPointSet.get(i) out of reg )
      Return Anonymize (pointSet+newPointSet , regIn , k)
    End if
    positionSet ← positioning ( reg , newPointSet.size )
  End for
  neighborSet = obtainNeighbors ( reg , positionSet )
  featurePoints = selectFeature ( neighborSet )
  similarSet = calculationSimilar (positionSet , neighborSet , featurePoints )
  For ( int i = 0 ; i < pointSet.size ; i ++ )
    If ( pointSet.get(i) in similarSet )
      similarPointSet ← pointSet.get(i)
    End if
  End for
  Return Anonymize (similarPointSet , similarSet, k )
```

## V. 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 [16].

### 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 ($C_{AVG}$)[9][10], 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:

$$C_{DM} = \sum_{\forall E \text{ s.t. } |E| \ge k} |E|^2 + \sum_{\forall E \text{ s.t. } |E| < k} |D| \cdot |E| \quad (17)$$

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:

$$C_{AVG} = \left(\frac{Total\_records}{Total\_equivalence\_classes}\right) / (k) \quad (18)$$

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:

$$C_{MDM} = \sum_{EquivclassesE} (|E| - k)^2 \quad (19)$$

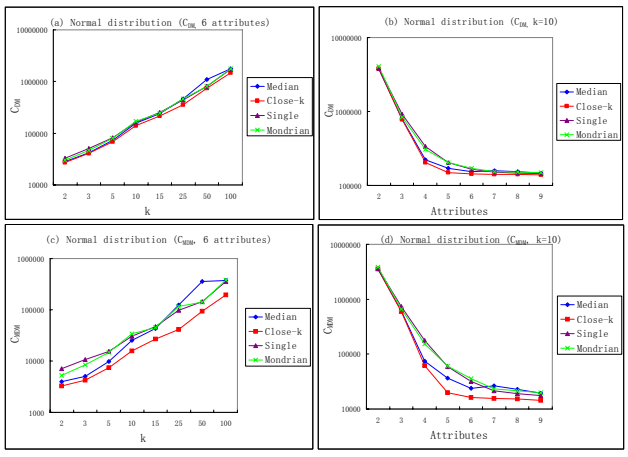


Figure 3 Results comparison in different cases

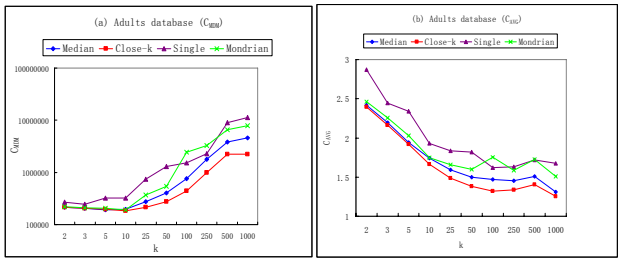


Figure 4 Results on Adult database

### B. Results from Experiment Data

In experiments, Median and Close-k separately stand for the results of partition based on median and Close-k points. Single is result of single-dimensional partitioning [7][9], and Mondrian shows what Mondrian works out[10].

For general case, we randomly generate some multi-dimensional data obeys discrete normal distribution for experiment. 10000 tuples are generated at first. Fig. 3 (a)(c)is a comparison when the number of attributes is 6 for all while k varies; Fig. 3 (b)(d) is a comparison when k is 10, and the number of attributes varies; Fig. 3 (a)(b) shows the different $C_{DM}$, while Fig. 3 (c)(d)is a comparison of $C_{MDM}$.

Fig. 3 (a)(d) 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. 3 (b)(c)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.

### C. Results from Practical Data

The Adult database[16] 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 same time. After preprocessing, 30162 valid records remain.

Fig. 4 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.

### D. Incremental Update Experiments

The experiments are respectively based on experimental dataset and real dataset to test availability, time performance and anonymity quality of the proposed algorithm. And in experiments, Full Update stands for the method of update for all dataset, and Incremental Update is our proposed update method, and I3M algorithm is the method in reference [25].

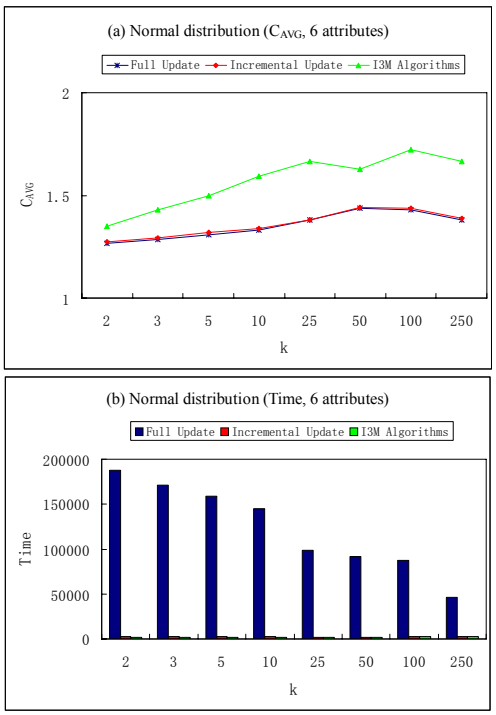


Figure 5 Comparison of different methods on Adult dataset

Figure 5 shows the results on experimental dataset. The experiment data includes 10000 discrete tuples generated randomly by normal distribution, and the tuples are all multi-dimensional. 20 percent of them used as incremental data. Here the dimension is set to be 6, and the experiment compares the time performance and anonymity quality of the three methods with k changes.

Figure 5 (a) indicates that both Full Update and Incremental Update methods are with lower information loss, which means they've got better anonymity quality. Since $C_{AVG} \geqslant 1$, the closer that $C_{AVG}$ is to 1, the better result it gets in anonymity. However, in Figure 5 (b), it shows that Full Update spends much more time in updating all the dataset, and therefore, it is not so available in practice. The proposed incremental update method is not only with better anonymity quality, but also efficient.

Figure 6 gives the results on real dataset, Adult dataset, which is published by UCI. And the results show our proposed k-anonymity algorithm for incremental update can also perform better on real dataset.

## VI. Conclusion

At present most of the k-anonymity algorithms are not available on dynamic publishing of dataset. This paper proposes an effective k-anonymity method based on incremental local update on large dataset, which improves the stability of update. The experiment shows that the proposed algorithm has a better performance in anonymity quality, and besides, it has its advantages of less time cost and better availability. The future work will take the diversity of sensitive attributes into consideration, and make a further research on background knowledge attack.

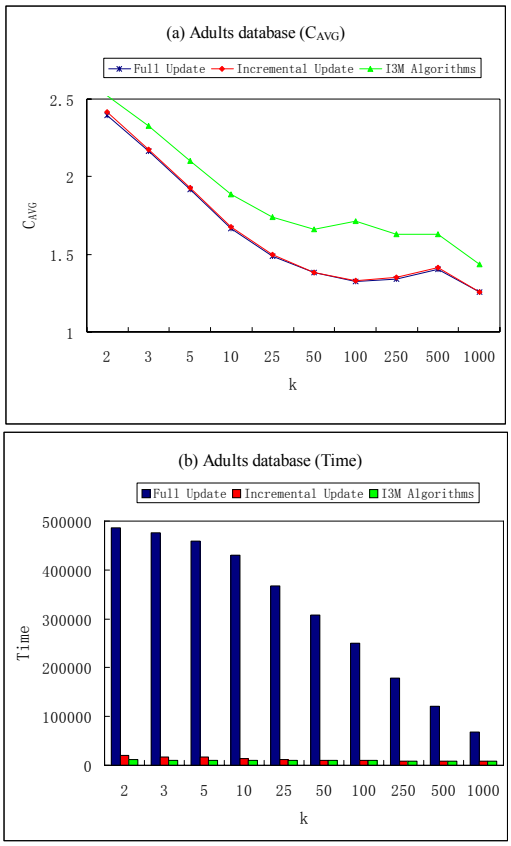


Figure 6 Results Comparison on normal distributed experimental dataset

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, 10 (5), 557-570, 2002.

[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, 1-10, 2010.

[3] Samarati P.: Protecting respondents' identities in microdata release. Proc of the TKDE' 01, 1010- 1027, 2001.

[4] 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, 188, 1998.

[5] Sweeney L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty, Fuzziness and Knowledge2Based Systems, 10 (5), 571-588, 2002.

[6] LeFevre K, DeWitt D J, Ramakrishnan R.: Incognito: Efficient full-domain k-anonymity. ACM SIGMOD International Conference on Management of Data.Baltimore, USA: ACM, 49-60, 2005.

[7] Iyengar V.: Transforming Data to Satisfy Privacy Constraints. Proc of the ACM SIGKDD, USA [s. n.], 279-287, 2002.

[8] Byun Ji-Won, Kamra Ashish, Bertino Elisa, Li Ninghui.: Efficient k-anonymization using clustering techniques. Lecture Notes in Computer Science, 188-200, 2007.

[9] Bayardo R, Agrawal R.: Data privacy through optimal k-anonymization. In ICDE, 2005.

[10] LeFevre K, DeWitt D J, Ramakrishnan R.: Mondrian multidimensional k-anonymity. IEEE International Conference on Data Engineering, Atlanta, USA, IEEE, 2006.

[11] Park Hyoungmin, Shim Kyuseok.: Approximate algorithms with generalizing attribute values for k-anonymity. Information Systems, 933-955, 2010.

[12] 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, 223-228, 2004.

[13] Aggarwal G, Feder T.: Approximation Algorithms for K-anonymity. Journal of Privacy Technology, 12(1), 78-94, 2005.

[14] Gionis A. Tassa T.: k-Anonymization with Minimal Loss of Information. Knowledge and Data Engineering, IEEE Transactions on, 206-219, 2009.

[15] Qian Wang, Xiangling Shi.: (a, d)-Diversity: Privacy Protection Based on l-Diversity. WRI World Congress on Software Engineering, WCSE, 367-372, 2009.

[16] UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn/M1-Repository.html.

[17] Truta, T.M., Fotouhi, F., and Barth-Jones, D., Privacy and Confidentiality Management for the Microaggregation Disclosure Control Method, Workshop on Privacy and Electronic Society, 10th ACM CCS, 21-30, 2003.

[18] Willemborg, L., and Waal, T. (ed).: Elements of Statistical Disclosure Control. Springer Verlag, 2001.

[19] Truta T.M., and Bindu V.: Privacy Protection.: p-Sensitive k-Anonymity Property. Workshop on Privacy Data Management, 22th IEEE Intl. Conf. of Data Eng, 2006.

[20] Machanavajjhala, A., Gehrke, J., and Kifer, D.: l-diversity: privacy beyond k-anonymity. Proc. of the 22nd IEEE Intl. Conference on Data Eng, 2006.

[21] Domingo-Ferrer, J., Mateo-Sanz, J., and Torra,V.: Comparing SDC Methods for Microdata on the Basis of Information Loss and Disclosure Risk. Pre-proc. of ETKNTTS' 2001 (vol. 2), Luxembourg: Eurostat, 807-826, 2001.

[22] Mateo-Sanz, J.M., Domingo-Ferrer, J., and Sebe, F.: Probabilistic Information Loss Measures in Confidentiality Protection of Continuous Microdata. Data Mining and Knowledge Discovery, Vol. 11, No. 2, 181– 193, 2005.

[23] 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, 392-403, 2010.

[24] Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., and Zhu, A.: Achieving Anonymity via Clustering. Proc. of the 10th Intl. Conf. on Database Theory, 2005.

[25] Truta Traian Marius, Campan Alina.: K-anonymization incremental maintenance and optimization techniques. Proceedings of the ACM Symposium on Applied Computing, p 380-387, 2007.

[26] Byun, J.W., Sohn, Y., Bertino, E., and Li, N.: Secure Anonymization for Incremental Datasets. Proc. of the 3rd VLDB Workshop on Secure Data Management, 2006.