# Clustering-anonymity method for data-publishing privacy preservation

Jiang Huowen[1, a]

[1]School of Mathematics & Computer Science, Jiangxi Science & Technology Normal University, Nanchang, 330038, China

[a]jhw_604@163.com

**Keywords:** Clustering-anonymity, privacy preservation, table data

**Abstract.** Data-publishing generally need to be treated by anonymity to protect its privacy information from disclosure. Existing anonymity methods have little distincation between different types of Quasi-identifiers in investigating generalization.Aimed to privacy preservation for pulblishing data from table, A clustering-anonymity data publishing method is proposed by using the ideas of clustering algorithm. The method makes generalization into Quasi-identifiers according to its different type, It gives the reasonable definition of the distance between one tuple and the other or one equvialance class; Dueing to partitioning cluster one by one controlled by the value of $k$ ,it achieves partition with the approximate same size of every equvialance class, So it reduces the amount of calculation of distances,and saves the running time accordingly.Experimental results verify the effectiveness of the method.

## Introduction

Nowadays, large amounts of sharing data grant people convenience, on the other hand, make individuals and groups more vulnerable to private information disclosure. Privacy preserving in data publishing mainly refers to taking protective measures to prevent those who have background knowledge or skills on data-mining, or other technical means from obtaining privacy information from publishing-data. For example, A table with relation schema (“name”, “age”, “gender”, “zipcode”, “disease”) shows some of the medical records of a hospital. if these data are published directly , individual privacy of disease will be exposed. The simplest way of avoiding its privacy disclosure is removing the “Name” field from the table before publishing. However, individual’s identity may be puzzled out by the attribute combination (Age, Sex, Zip),thus it may not prevent privacy from being accessed. The essence of privacy preserving is cutting the connection between sensitive attribute and explicit identity attribute in case figure out the mapping from the former to the latter.

*k*-anonymization, firstly proposed by L. Sweeney, is a classic privacy protection scheme that can prevent the background-knowledge-based attack and link attack[1]. By *k*-anonymity, records in a table are partitioned into some equivalence classes(also called cluster), each of clusters contains at least *k* tuples that share the same value of Quasi-identifiers. So the possibility of re-identifying explicit identifier attribute in virtue of Quasi-identifier attributes is 1/*k* at most. That is the possibility of exposing privacy is not more than 1/*k*. In 2006, G. Aggrawal et al., for the first time, put forward that data anonymization can be implemented using clustering[2]. Since then related researches have sprung up. For instance, J.Y. Li et al. proposed KACA algorithm in 2006[3]; G.Aggarwal et al. elaborates r-gather and r-cellular algorithms in 2010 [4]; Z.H.Wang et al. presented the L-clustering scheme in 2011[5]; J.P. Zhang ,et al. put forward the EBKC algorithm in 2014[6],and so on. Aiming to the privacy preservation problem for relational data publishing, we proposed a novel clustering-anonymity method.

## Relevant concepts of clustering-anonymity

To facilitate discussion, firstly we make some assumptions.Assuming the data table are composed of 1 sensitive attribute *and d* Quasi-identifiers,among which there have $d_1$ numeric type(e.g. the values

 

of "Age" and "Zipcode" are numeric)and $d_2$ category type（e.g. the value of "Sex" is category）,here $d$=$d_1$+$d_2$;contains $n$ tuples. Let $t_j(j=1,2,\cdots n)$ denotes the $j$-th tuple of the table; $A_i^{qi}(i=1,2,d)$ denotes the $i$-th Quasi-identifier attribute; $A^s$ denotes the sensitive attribute； $t_j[A_i^{qi}]$ denotes the value of $A_i^{qi}$ of tuple $t_j$ ..

**Def.1(equivalence class):** An equivalence class is a collection of table tuples, whose values in Quasi-identifiers have a high degree of similarity before generalization, and are identical after generalization, written $C_i(i=1,2,\cdots,m)$.

After clustering-anonymity, a table is partitioned into some equivalence classes,and any two equivalence classes share no mutual tuples,all the equivalence classes form the orginal table. that is, $\bigcup_{i=1}^{m} C_i = T$ , $\forall\ i1,i2 \in N, 1 \le i1 \ne i2 \le m, C_{i1} \bigcap C_{i2} = \phi\ C_{i1} \bigcap C_{i2} = \phi$.

**Def.2(domain generalization)**: Domain generalization (generalization for short) refers to the process that replaces all the values in $A_i^{qi}(i=1,2,d)$ of all tuples contained in an equivalence class with a broader category or a generalized value.

All the tuples in same an equivalence class will have the same value in Quasi-identifiers after generalization. that is, $\forall w \in [1,m], \forall \mathrm{i} \ne \mathrm{j} \in [1,k_w]$ ( $w,i,j \in N$ , $\mathrm{k_w}$ denotes the number of tuples contained in $C_w$ and $k_w \ge k$ ), for $t_i, t_j \in C_w$ , we have $t_i[A^{qi}] = t_j[A^{qi}]$

**Def.3( equivalence tuple):** All the tuples contained in $C_i$ have the same value of Quasi-identifiers after generalization . we call the tuple of having the same value of Quasi-identifiers the equivalence tuple,written $t_{C_i}$ .

**Def.4(*k*-anonymity):**Assuming table $T$ is partitioned into $m$ equivalence classes and each equivalence class contains at least $k$ tuples, $T$ is said to have the $k$-anonymity property if and only if all tuples contained in any same an equivalence class have the exact same value of Quasi-identifiers.

To satisfy $k$-anonymity, at first partitioning is needed, then generalization is needed to make the values of Quasi-identifiers identical for same an equivalence.in this paper,we make partition by clustering;make generalization using different way according to two different kinds of Quasi-identifiers. For numeric type Quasi-identifiers,we make generalization with [ $\min C_1(A_i^{num-qi})$ , $\max C_1(A_i^{num-qi})$ ] replacing the orginal value of $A_i^{num-qi}$ to the tuples in $C_1$ .for example,for attribute "age",we replace the orginal value "45", "48" and"52" of three tuples with the same [45,52]. for category type Quasi-identifiers,we make generalization on the basis of building its generalization hierarcy tree.according to the generalization hierarcy tree,The generalization value of two different value for two tuples in same a category attribute lies in the node value which is the nearest common ancestor of the corresponding orginal two nodes in the tree.

**Simility and distance between tuples**

The basic idea of clustering is partitioning a set of data so that data in the same group (called a cluster) are more similar than those in other groups. So we can partition the table $T$ to be anonymized into some equivalence classes(corresponding with cluster) by clustering.here we define the distance between one tuple and the other to measure the simility between them. The smaller the distance means the greater similarity.

For any two tuples in $T$ , say $t_p$ and $t_q$ , the distance between them refers to the average value of the degree of difference of the two values in each Quasi-identifier attribute held by $t_p$ and $t_q$ , written $dist(t_p, t_q)$.

Let $A_i^{num-qi}$ and $A_i^{cat-qi}$ represent the $i$-th numeric attribute and the $j$-th category attribute of Quasi-identifiers respectively; $A^{num-qi}$ and $A^{cat-qi}$ represent all the numeric types and all the category types of Quasi-identifiers; $\max T(A_i^{num-qi})$ and $\min T(A_i^{num-qi})$ respectively represent the maximum

value and the minimum value of $A_i^{num-qi}$ of the table ; $\max C_1(A_i^{num-qi})$ and $\min C_1(A_i^{num-qi})$ respectively represent the maximum and the minimum value of $A_i^{num-qi}$ of the equivalence class $C_1$.

The distance between two tuples is defined as follows:

$$dist(t_p, t_q) = \frac{1}{d}\left(\sum_{i=1}^{d_1}\left(\frac{\left|t_p(A_i^{num-qi}) - t_q(A_i^{num-qi})\right|}{\max T(A_i^{num-qi}) - \min T(A_i^{num-qi})}\right)\right.$$

$$\left. + \frac{1}{2}\sum_{i=1}^{d_2}\left(\frac{h(t_p[A_i^{cat-qi}], t_{p\wedge q}^{*}[A_i^{cat-qi}])}{h(t_p[A_i^{cat-qi}], root_i)} + \frac{h(t_q[A_i^{cat-qi}], t_{p\wedge q}^{*}[A_i^{cat-qi}])}{h(t_q[A_i^{cat-qi}], root_i)}\right)\right)$$

The distance between tuple $t_p$ and equivalence class $C_i$ $(t_p \notin C_i)$ refers to the distance between $t_p$ and equivalence tuple $t_{C_i}$ of $C_i$, written $dist(t_p, C_i)$. So $dist(t_p, C_i) = dist(t_p, t_{C_i})$,it means $dist(t_p, C_i)$ also can be computed according to the formula above.

**The clustering-anonymity approach**

Based on the definition of the distance above in section 3, this section we put forward a clustering-anonymity algorithm-CAAT,so as to protect privacy for table data-publishing. The basic idea of the algorithm is as follow: given a table $T$ containing *n* tuples,it can be treated as *n* points in a *d*-dimensional space, classify the *n* points into *m* clusters (corresponding with equivalence classes) with a size of greater than or equal to *k* and less than 2*k*, and conduct generalization on each cluste. Detailed description of the algorithm is given as follow:

**Algorithm 1:** The clustering-anonymity algorithm for table data-publishing privacy preservation---called CAAT for short.

Inputs: The table $T$ to be released, anonymity parameter *k*.

Outputs: The table $T^*$ that satisfies k-anonymity.

Steps:

(1) Pick a tuple randomly from table $T$, classify it into a new cluster $C$ as initial tuple;

(2) move the tuple that has the shortest distance from $C$ into $C$ from $T$(if there exist more than one tuples that meet the requirement, then move them all) ,until the size of $C$ is greater than or equal to *k*. Add $C$ into the table $T^*$;

(3) Pick the tuple ,which have the longest distance from $C$, from $T$ and treat it as the initial tuple of a new cluster $C$;

(4) repeating step 2~step 3, until there only exist less than *k* tuples in $T$;

(5) Classify each remaining tuple from $T$ into one of clusters existing in $T^*$ with the shortest distance between the tuple and the cluster;

(6) Conduct generalizations on each $A^{qi}$ of every cluster in $T^*$;

(7) finally output $T^*$.

**Experiment and its analysis**

In this chapter, we conduct experiments to analyze the practical efficiency of CAAT and compare it with KACA proposed in the Ref.[3].The design of data used refer to Ref.[3]. Hardware environment of experiment is: Intel Pentium double-core E2140 @1.60GHz CPU,2GB (DDR ) memory. The algorithm is implemented on Microsoft Visual C++ 7.0. Two experiments aimed to respectively reveal the runtime how to be changed with increasing the number of Quasi-identifiers when *k*=3 and *k*=6. In view of that picking different initial tuple to initialize the first new cluster may lead to different result, each experiment runs 10 times, and the final result of each experiment is the average of the 10 outcomes.The final results of two experiments are shown in fig.1 and fig.2 respectively.

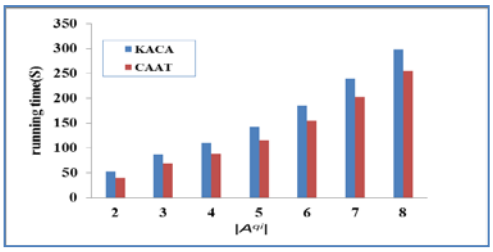


Fig.1 runtime changes with $|A^{qi}|$ when k=3

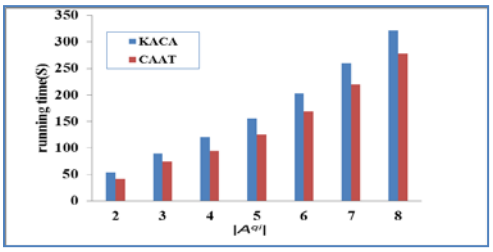


Fig.2 runtime changes with $|A^{qi}|$ when k=6

The results above reveal: ① With the increase of *k* value,the runtime of two algorithm is Significantly increasing.This can be explained that computation quantity grows with the increase of $|A^{qi}|$ dueing to much more data need to be generalized much more data need to be generalize , which will certainly lead to a rise in handling time.② when *k*=3 or 6, the value of runtime of CAAT is less than KACA,this show the CAAT is better than KACA in the aspect of runtime. Obviously the experiment show that the algorithm of CAAT is feasible.

## Conclusions

*k* -anonymizing the corresponding table data is an important way to achieve privacy preservation for data publishing. Private information hidden in table data may be accessed by attackers who have certain background knowledge. To cope with this issue, we propose CAAT, a novel clustering-anonymity approach, which transfers a table to be released into a table satisfying *k*-anonymity,can effectively prevent the privacy disclosure. It is undeniable that our method change orginal value of Quasi-identifiers to some extent and brings up information loss. Therefore, in our future research,we will focus on how much information loss is caused by clustering-anonymity, and how to balance the information loss and the degree of privacy preservation.

## References

[1] L. Sweeney. *k*-anonymity: a model for protecting privacy. Int'l Journal on Uncertainty, Fuzziness, and Knowledge-Based Systems, 2002, 10(5):557-570. [1] L. Sweeney. *k*-anonymity: a model for protecting privacy. Int'l Journal on Uncertainty, Fuzziness, and Knowledge-Based Systems, 2002, 10(5):557-570.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, A. Zhu. Achieving anonymity via clustering[C]. In: Vansummeren S, ed. Proc. of the 25th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems. New York:ACM, 2006, 153-162.

[3] J.Y. Li, C.W. W. Raymond,W.C. F. Ada,P. Jian.Achieving *k*-anonymity by clustering in attribute hierarchical structures[J].Data Warehousing and Knowledge Discovery Lecture Notes in Computer Science Volume 4081,2006:405-416.

[4] G. Aggarwal, R. Panigrahy,Tom,et al.Achieving anonymity via clustering[J].ACM Transactions on Algorithms,2010,6(3): 49.1- 49.19.

[5] Z.H. Wang,J. Xu,W. Wang,B.L. Shi. Clustering-based approach for data anonymization[J]. Journal of software(in chinese), 2010,21(4):680-693. [5] Z.H. Wang,J. Xu,W. Wang,B.L. Shi. Clustering-based approach for data anonymization[J]. Journal of software(in chinese), 2010,21(4):680-693.

[6] J.P. Zhang,Y. Zhao,Y. Yang,et al.A *K*-anonymity clustering algorithm based on the information entropy[C].Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design,2014:319-324.