S-KACA Anonymity Privacy Protection Based on Clustering Algorithm

MAO Qingyang, HU Yan

***Abstract*—In order to prevent data disclosure in the privacy of individuals, privacy protection technology continues to improve, the S-KACA algorithm can protect the sensitive privacy attributes and make the published data available, but it sets a privacy protection parameter when protecting sensitive privacy information, which results in the low efficiency of the algorithm and does not apply to the large-scale data set. In order to solve this problem, an algorithm called K-Prototypes-S-KACA is proposed, which combines the efficient K-Prototypes clustering algorithm with S-KACA algorithm. Firstly, the algorithm divides the microdata set into several slightly larger clusters by clustering algorithm K-Prototypes, and then using S-KACA algorithm anonymizes these microdata set. Experiments show that the algorithm is similar to S-KACA algorithm in terms of privacy protection and data availability, but the efficiency of the algorithm is greatly improved.**

Index Terms—K-anonymity, privacy protection, clustering algorithm, KACA algorithm, K-Prototypes-S-KACA algorithm

Ⅰ. INTRODUCTION

In recent years, with the rapid development of Internet, computing technology and data storage technology, people can more easily extract unknown, hidden and potentially valuable information from the data, but it leads to the issue of privacy disclosure during data publishing. In the process of data release, privacy disclosure mainly refers to the leakage of sensitive information in individual privacy information. Traditional encryption technology and access restriction technology can not achieve the purpose of individual information privacy protection. Therefore, in order to reduce the sensitive information leakage in the process of data release, we can do some pretreatment of the data items before the release of the data, by using anonymization and add random noise and data perturbation and so on. Samarati and L.Sweeney proposed a basic K-anonymous model which applies anonymization to the data, mainly takes generalization and concealment technology to prevent the attackers through the linking attacks to obtain the correspondence between sensitive attribute value and individual identity, and achieves the purpose of sensitive information protection. On the basis of the K-anonymous model, Reference [1] proposes a l-diversity anonymous model, which requires at least 1 different sensitive attribute values in one equivalence class and improves the privacy protection of K-anonymity. Through improving the l-diversity model, Reference [2] proposes a (L, α)-diversity model, which is not only required to satisfy l-diversity, but also requires that the sum of the weights of the sensitive attribute values in each equivalence class is at least α; Reference [3] proposes an improved K-anonymity which is named (K, e) -Anonymity for numerical sensitive attributes. The model requires that the range of sensitive attribute values in the equivalence class is at least e (e> 0). In Reference [4], an improved K-anonymity model which is called P-Sensitive-K-Anonymity model is proposed for sensitive attributes of classification. It limits the frequency of occurrence of sensitive attribute values in equivalence classes. Reference [5] proposes a probabilistic K-anonymous model, which does not require at least K identical quasi-identifier attribute values in equivalence classes. Reference [6] proposes a K-anonymous model based on clustering, in order to improve the efficiency of the algorithm.

Manuscript received December 15, 2016.

Mao Qingyang is a master student of the Department of Software Engineering, School of Computer Science and Technology, Wuhan University of Technology (Email: maoqingyanghao@whut.edu.cn or 1214637835@qq.com; Telephone: 15972083052)

Hu Yan is a Professor of Department of Software Engineering, School of Computer Science and Technology, Wuhan University of Technology (Email: huyan@whut.edu.cn; Telephone: 13329706007)

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Traditional K-anonymization algorithms are generally divided into two categories: global generalization algorithm and local generalization algorithm [7]. The K-anonymity algorithm of global generalization will over-generalize the micro-database. It is not conducive to the effective use of the published data, and the amount of information loss is higher than the local generalization algorithm; the local generalization algorithm is based on the idea of clustering, and avoids the over-generalization of information. The basic idea is that according to the similarity of tuples on quasi-identifiers in microdata, they are divided into several clusters, and the number of tuples in each cluster is greater than or equal to K. In order to minimize the amount of information loss after generalization, we set a weight for the generalized path of each quasi-identification attribute, and then use the generalized value of the tuples within the cluster and replace the value on the quasi-identifier of the tuples within the cluster, and finally set the different sensitive privacy protection degree according to the different sensitive information attributes to realize the K-anonymization of the microdata. Although the K-anonymity algorithm considering the weight and the sensitive privacy protection degree can better anonymize data, and more pertinently protect the sensitive privacy information. But when the size of the microdata set becomes larger, the efficiency of the algorithm will be significantly reduced. In order to improve the efficiency of the algorithm and not to reduce the degree of privacy protection and the availability of published data, This paper introduces an efficient clustering algorithm K-Prototypes-S-KACA algorithm, which combines the K-Prototypes and S-KACA algorithms to play the advantages of the two algorithms. The new algorithm first divides large-scale micro data sets into K clusters by clustering algorithm, and then uses S-KACA algorithm to anonymize the tuples in these K clusters. It is both efficient and targeted to achieve the protection of sensitive privacy information.

Ⅱ. KACA TECHNOLOGY

KACA (K-anonymization by clustering in attribute hierarchies algorithm) is a K-anonymization method based on local generalization [9]. It reduces the information loss of anonymized micro data by generalizing the weighted hierarchical distance and generalization distortion, and improves the usability of post-release data. The following is a measure of the distance of generalization level and the degree of deformation in the KACA method:

The Generalized Weighted Hierarchical Distance: Let h be the highest level of possible generalization of attribute A, D1 is the Range, D2 ,…, Dh are the Generalization Domain, $W_{j,j-1}$is the Generalization Weight between Dj and Dj-1 (2≤j≤h) . The distance, which is generalized from Dm to Dn (m> n), is called the generalized weighted hierarchical distance. Such as Formula 1.

$$WHD(m,n) = \sum_{j=n+1}^{m} w_{j,j-1} / \sum_{j=2}^{h} w_{j,j-1} \quad (1)$$

There are several methods to define generalization weight $W_{j,j-1}$in generalization hierarchy. Data publishers can choose different methods according to different needs. The following two methods are used to calculate the generalization weight:

（1）$w_{j,j-1}$*=1,2≤j≤h.* This definition method is relatively simple and easy to calculate. However, this definition does not reflect the variability of the deformation that is produced by the generalizations of different generalization hierarchy.

（2）$w_{j,j-1}$*=1/(j-1)*$^{\beta}$，*2≤j≤h*，β is defined by the data publisher, such as β = 1. The method can reflect the degree of variance of the data deformation that is caused by different levels of generalization.

The Tuple Generalization Distortion Degree: Let t = $\{v1, v2, \ldots, vm\}$ be a tuple, t′ = $\{v1', v2', \ldots, vm'\}$ is the generalization tuple of t, Levels(vj) is the level of the generalized domain where Vj is located. The Generalization Distortion Degree from t to t′ is defined as Formula 2.

$$Distortion(t,t') = \sum_{j=1}^{m} WHD(levels(vj), levels(vj')) \quad (2)$$

The Data Table Generalization Distortion: Let T′ be the generalization table of the entity data table T, ti is the tuple in T, ti′ is the generalized tuple of ti（ti′ ∈ T′）, the data table T that is generalized to T′ generates the degree of distortion Such as Formula 3:

$$Distortion(T,T') = \sum_{i=1}^{|T|} Distortion(ti, ti') \quad (3)$$

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

KACA Algorithm flow is as follows：

Input: Data set D with N records, Parameter K value (show K-anonymity restriction)

Output: K- anonymity table

1） The initial equivalence classes are generated from the microdata set D, in which each tuple in the equivalence class has the same value on the quasi-identifier;

2） **The loop processing**, there are no equivalence classes whose number of tuples is less than K:

① Randomly select an equivalence class C whose size is less than K;

② Calculate the distance of privacy protection between C and all other equivalence classes according to formula 1, formula 2 and formula 3;

③ On the basis of minimizing information loss, the equivalent class C is merged into the equivalence class nearest to its distance and form a new equivalence class C';

④ Generalize equivalence class C';

The loop ends

3） Return the K-anonymity table.

Ⅲ. S-KACA ALGORITHM BASED CLUSTERING

A. Clustering Algorithm K-Prototypes

K-Prototypes algorithm is a combination of K-Means and K-modes algorithm for mixed attributes. The parameter γ is introduced to control the weight of the numerical attribute and the classification attribute in the clustering process, and the large mixed type data set can be processed efficiently. Let $X=\{X_1, X_2, \ldots, X_n\}$ denote the dataset with n samples, each sample has m attributes, $\{A_1, A_2, \ldots, A_m\}$, $X_i=[x_{i1}, x_{i2}, \ldots, x_{ip}, x_{i(p+1)}, x_{i(p+2)}, \ldots, x_{im}]$denotes the m property values of the i-th sample X_i, where the subscripts are from 1 to p for the numerical attribute, and the subscripts are from p+1 to m for the classification attribute. The value of the attribute Ai is represented by Dom(Ai)= $\{a_{i1}, a_{i2}, \ldots, a_{iq}\}$, and q represents the number of possible values of the classification attribute. Let the initial clustering number of the data set be K, the set of corresponding modules is $V=\{V_1, V_2, \ldots, V_k\}$, the iterative clustering set is $C=\{C_1, C_2, \ldots, C_k\}$, $C_i=[c_{i1}, c_{i2}, \ldots, c_{ik}]$ in the clustering process. K-Prototypes algorithm is a partition-based clustering algorithm. The mean of the clusters is replaced by the modules $(V_1, V_2, \ldots, V_k)$, Each iteration is centered on a module, Calculate the distance from each sampl e$(X_1, X_2, \ldots, X_n)$ to the module, Choose the smallest distance, and divide the sample into clustering set. After each iteration, And Use frequency-based method to carry on module updating in the clustering process, So that the clustering cost function F(X,V) is minimized.

The distance of the Euclidean squared distance is used to define the distance of the numerical attribute. The numerical attribute distance between the sample Xi and modulus Vt is d1(Xi,Vt)=$\sum_{j=1}^{p}(X_{ij}-V_{tj})^2$.The Hamming distance is used to define the distance of the classification attribute. The classification attribute distance between the sample Xi and modulus Vt is d2(Xi,Vt)= $\sum_{j=p+1}^{m}\delta(X_{ij},V_{tj})$,

where $\delta(X_{ij},V_{tj})=\begin{cases}0, & X_{ij}=V_{tj}\\ 1, & X_{ij}\neq V_{tj}\end{cases}$.

The dissimilarity measurement function is used to solve the distance between sample and modulus. The distance d(Xi,Vt) between sample Xi and modulus Vt is shown in Formula 4:

$$d(Xi,Vt)=\sum_{j=1}^{p}(X_{ij}-V_{tj})^2+\gamma\sum_{j=p+1}^{m}\delta(X_{ij},V_{tj})$$
$$=d1(Xi,Vt)+\gamma d2(Xi,Vt) \quad (4)$$

γ is the weight value of the classification attribute, if the classification attribute is important, increase γ; otherwise reduce γ. δ is the dissimilarity of the classification attribute.

The modulus for sample dataset X is defined as V=$(V_1, V_2, \ldots, V_k)$, so that the value of D(X, V)=$\sum(X_i,V_t)$ is minimized, d(Xi,V) is the distance from the sample Xi to the set modulus to which it belongs, in the formula d(Xi,Vt)= d1(Xi,Vt)+d2(Xi,Vt). On the selection of a modulus, for the numerical attribute, the average value of each numerical attribute in the clustering sample is taken; For the classification attribute, the highest probability value that appears in each classification attribute in the clustering sample is taken. Such as Formula 5:

$$\frac{C_{vtj}}{n}\geq\frac{C_{xij}}{n} \quad (5)$$

C_{vtj} represents the modulus Vt, and C_{xij} represents the number of occurrences of the j-th attribute value of the sample Xi in the class.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Definition of cost function of clustering optimization is shown in Formula 6:

$$F(X,V)=\sum_{i=1}^{n}\sum_{j=1}^{k}u_{ij}d(Xi,\ Vj)$$
$$\sum_{i=1}^{k}u_{it}=1(u_{it}\in[0,1]) \quad (6)$$

When u_{it} is 1, it means that the sample Xi is in class C_t; When u_{it} is 0, it means that the sample Xi does not belong to class C_t.

B. S-KACA(Sensitivity-KACA) Algorithm

KACA algorithm is one of the local generalization and anonymity algorithms which produces the least information loss. However, this algorithm does not distinguish between different sensitive information, so it can not protect the sensitive attribute more pertinently. Therefore, based on the KACA algorithm, we introduce a numerical measure of sensitive privacy degree x_i [10], whose value is between 0 and 1. In general, The privacy protection degree of the sensitivity attribute is determined flexibly according to the specific application domain or determined by experts in the relevant field.

The calculation of the degree of privacy protection:

The standard deviation of the privacy protection degree of the sensitive attribute in the equivalence class is б, Such as Formula 7:

$$б=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(xi-u)^2}$$
$$\text{where}\quad u=\frac{1}{N}\sum_{i=1}^{N}xi \quad (7)$$

The distance of the standard deviation of the privacy protection degree of the sensitive attribute in the equivalence class is $D_{i,\ j}$, Such as Formula 8:

$$D_{i,\ j}=|бi-бj| \quad (8)$$

The following is an improved KACA algorithm — S-KACA algorithm. The improved algorithm based on KACA algorithm, add a sensitive degreeof privacy protection, to protect more pertinently sensitive privacy information.

Algorithm 1. S-KACA Algorithm flow is as follows:

Input: Data set D with N records, Parameter K value (show K-anonymity restriction), Sensitive privacy protection degree X=$\{x_1,\ x_2,\ \ldots,\ x_n\}$

Output: K- anonymity table

1) The initial equivalence classes are generated from the microdata set D, in which each tuple in the equivalence class has the same value on the quasi-identifier;

2) **The loop processing**, there are no equivalence classes whose number of tuples is less than K:

① Randomly select an equivalence class C whose size is less than K;

② Calculate the standard deviation and the standard deviation distance of privacy protection between C and all other equivalence classes according to formula 7 and formula 8;

③ According to the calculated distance of formula 8 and On the basis of minimizing information loss, the equivalent class C is merged into the equivalence class nearest to its distance and form a new equivalence class C′;

④ Generalize equivalence class C′;

The loop ends

3) Return the K-anonymity table.

Compared with the basic algorithm KACA, the S-KACA algorithm with a sensitive privacy protection degree can protect sensitive privacy attributes more pertinently. However, with the increasing of the micro-data set, the efficiency of S-KACA algorithm is reduced. As the micro-data set increases, the calculation time of sensitive privacy protection degree will increase, resulting that the efficiency of S-KACA algorithm is reduced.

C. S-KACA Algorithm Based Clustering

In order to solve the problem of reducing the efficiency of S-KACA algorithm when dealing with large data sets, an efficient clustering algorithm called K-Prototypes is introduced, Through Combining this clustering algorithm with S-KACA algorithm, we propose a K-Prototypes-S-KACA algorithm. First, the whole data set is divided into some larger clusters by K-Prototypes, then the S-KACA algorithm is applied to the clusters.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Algorithm 2. K-Prototypes-S-KACA

Algorithm flow is as follows:

Input: Data set D with N records, Parameter K value (show K-anonymity restriction), Sensitive privacy protection degree X=$\{x_1, x_2, \ldots, x_n\}$

Note: the K value of K-clusters generated by the clustering algorithm differs from the K value of the K-anonymity restriction

Output: K- anonymity table

1) The initial equivalence classes are generated from the microdata set D, in which each tuple in the equivalence class has the same value on the quasi-identifier, Let m be the number of initial equivalence classes;

2) K1=f(m), f(m) is a function that the data owner defines according to the application scenario, and determines the number of clusters K1 of the initial clustering algorithm;

3) C_results=K-Prototypes(D,K1), where C_results is the result of the clustering algorithm;

4) **The loop processing**: For each sub-dataset Di to be processed, and Perform it K- anonymity:

S-KACA(Di,k), where (show K-anonymity restriction);

The loop ends

5) If the number of tuples in the equivalence class is smaller than k, the tuples in the equivalence class are hidden;

6) Return the K-anonymity table.

Compared with the S-KACA algorithm, the K-Prototypes-S-KACA algorithm can effectively improve the efficiency of the algorithm when dealing with large data sets by preprocessing the efficient clustering algorithm.

Ⅳ. EXPERIMENTAL RESULTS AND ANALYSIS

Experimental operating environment:

1) **Operating System:**
Microsoft Windows 10 64-bit Operating System.

2) **Hardware Environment:**
CPU: Intel(R) Core(TM) i5; RAM: 4.0GB.

3) **Programming Environment:**
Microsoft Visual Studio integrated development environment.

The experiment uses the Adult data set which is provided by the UCI Machine Learning Repository. The data set is from the US Census data, and has a total of 48,842 tuples, in which attribute variables include age, workclass, education, gender, race, occupation and so on, 7 of 14 attribute variables are classification attribute variables. In this experiment, we use the occupation attribute as the sensitive attribute, and set the corresponding sensitive privacy protection degrees for different attribute values. As shown in Table Ⅰ:

TABLE Ⅰ

SETTING SENSITIVE PROPERTY VALUE TABLE

Occupation	Sensitivity
Tech-support	0.6
Craft-repair	0.5
Other-service	0.3
Sales	0.7
Exec-managerial	0.6
Prof-specialty	0.4
Handlers-cleaners	0.8
Machine-op-inspect	0.5
Adm-clerical	0.5
Farming-fishing	0.6
Transport-moving	0.4
Priv-house-serv	0.7
Protective-serv	0.9
Armed-Forces	0.2

40,000 tuples are randomly selected for experiment verification. The K-Prototypes-S-KACA algorithm is compared with KACA algorithm, S-KACA algorithm in the running time of the sensitive privacy protection and in loss of information to prove the superiority of the proposed K-Prototypes-S-KACA algorithm.

A. Runtime Analysis

Fig 1 shows the comparison of the runtime of three algorithms under different k value (K-anonymity restriction) and quasi- identifiers attribute number |QI|=5 and 40,000 randomly selected micro-data tuples.

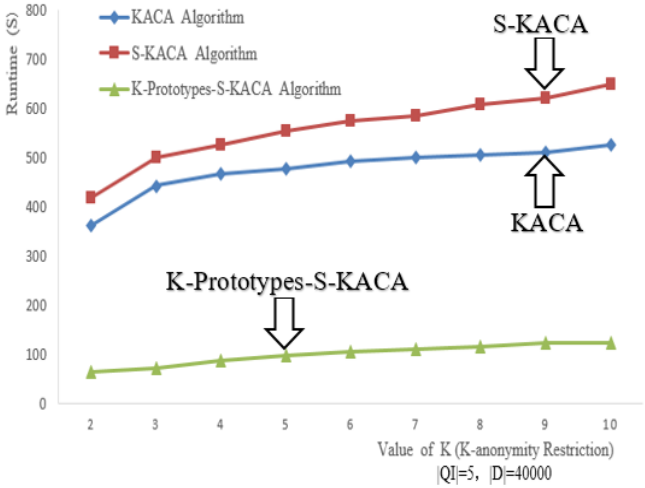

Fig . 1 . the comparison of the runtime of three algorithms under different k value

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Fig . 1 shows that the runtime of three algorithms increases with the increasing of K value, and it is due to the fact that the number of tuples that is less than K in the initial equivalence class will increase with the increasing of K value, this will lead to clustering time increasing, and thereby increasing the K-Prototypes-S-KACA algorithm runtime. The runtime of KACA algorithm and S-KACA algorithm increases relatively fast, and basically the same, but the increase of the K-Prototypes-S-KACA algorithm is relatively slow, so under the same conditions, the algorithm runtime is shortest.

Fig . 2 shows the comparison of the runtime of three algorithms with K=6 and 40,000 randomly selected micro-data tuples, as the number of the quasi-identifiers attribute QI increases.

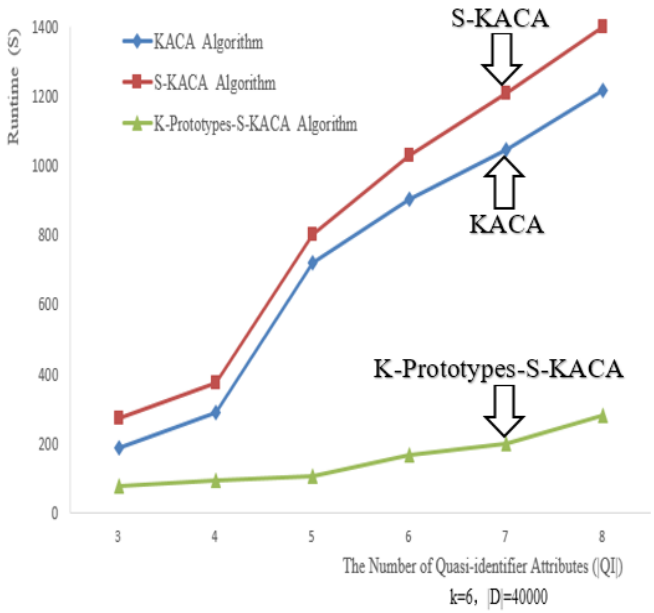

Fig . 2 . the comparison of the runtime of three algorithms under different quasi-identifier attribute QI value

Fig . 2 shows that the runtime of three algorithms increases, as the number of the quasi-identifiers attribute QI increases, and it is due to the fact that with the increase of the number of quasi-identifiers attribute, it will increase computation of K-anonymization, so increases the algorithm's runtime. The runtime of KACA algorithm and S-KACA algorithm increases relatively fast, and basically the same, but the increase of the K-Prototypes-S-KACA algorithm is relatively slow, so under the same conditions, the algorithm runtime is shortest.

Drawn from the comprehensive comparison of **Fig . 1** and **Fig . 2,** it is concluded that the K-Prototypes-S-KACA algorithm has the best overall efficiency under the same conditions.

B. Information Loss Analysis

Fig . 3 shows the comparison of the amount of information loss of three algorithms under different k value (K-anonymity restriction) and quasi-identifiers attribute number |QI|=5 and 40,000 randomly selected micro-data tuples.

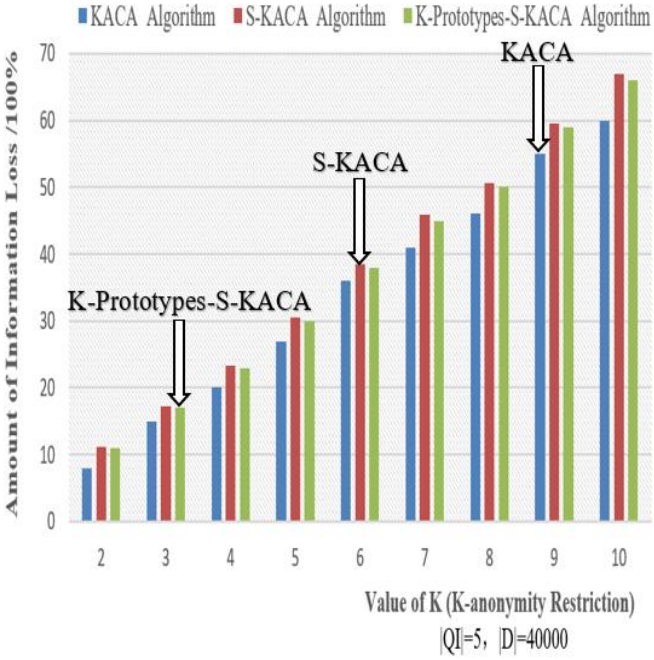

Fig . 3 . the comparison of the amount of information loss of three algorithms under different k value

Fig . 3 shows that with the increase of the K value, the amount of information loss of three algorithms is increasing, because the number of tuples in the equivalence class that can not be differentiated with the increase of the K value is increasing, The degree of generalization of the data will increase, resulting in an increase in the amount of information loss. The difference of the amount of information loss between three algorithms is not too big, but the amount of information loss of the KACA algorithm, which does not set a sensitive privacy protection degree, is slightly lower than the other two algorithms. But the S-KACA algorithm and K-Prototypes-S-KACA algorithm have better sensitive privacy protection than KACA algorithm and are basically the same amount of information loss. The K-Prototypes-S-KACA algorithm, which introduces clustering algorithm, has a slightly lower amount of information loss than S-KACA algorithm. However, it has a large increase in the comparison of runtime, thus the improved algorithm , which is slightly higher cost of the amount of information loss, is acceptable.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Ⅴ. SUMMARY

K-Prototypes-S-KACA algorithm is proposed in this paper to solve the problem of privacy protection of sensitive information in data publishing process. The algorithm can achieve the privacy protection of sensitive information better and more pertinently, and can achieve a good balance between the efficiency of algorithm execution and the amount of information loss. It not only can have an efficient runtime, but also ensure an acceptable cost of the amount of information loss.

REFERENCES

[1] Cheng L, Cheng S, Jiang F. ADKAM: ″A-Diversity K-Anonymity Model via Microaggregation″ [M]Information Security Practice and Experience. 2015:533-547.

[2] Sun Xiaoxun, ″A family of enhanced(L,α)-diversity models for privacy preserving data publishing″ [J].Future Generation Computer Systems · 2011,27(3):348-356.

[3] Yu L, Yang Q. ″An Efficient Local-Recoding k -Anonymization Algorithm Based on Clusterin″ [M]Transactions on Edutainment XI. 2015.

[4] Jing Y, Chao W, Zhang J P. ″ Micro-Aggregation Algorithm Based on Sensitive Attribute Entropy″ [J]. Tien Tzu Hsueh Pao/acta Electronica Sinica, 2014, 42(7):1327-1337.

[5] J Soria-Comas · J Domingo-Ferrer. ″Probabilistic k-anonymity through microaggregation and data swapping″.IEEE International Conference on Fuzzy Systems.2012:1-8.

[6] Xu X, Numao M. ″ An Efficient Generalized Clustering Method for Achieving K-Anonymization[C]International Symposium on Computing & NETWORKING″. IEEE, 2015:499-502.

[7] MRS Aghdamm · N Sonehara. ″ Efficient local recoding anonymization for datasets without attribute hierarchical structure″. International Conference on Cyber Security, 2013.

[8] Hajkacem M A B, N'Cir C E B, Essoussi N. ″ Parallel K-prototypes for Clustering Big Data″ [M]Computational Collective Intelligence. Springer International Publishing, 2015.

[9] Shan-Shan L I, Zhu Y Q, Chen G. ″Clustering-based algorithm for data sensitive attributes anonymous protection″ [J]. Application Research of Computers, 2012, 29(2):469-471.

[10] Sharma V. ″Methods for privacy protection using k-anonymity″ [C]Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on. IEEE, 2014:149-152.

ISBN: 978-988-14047-3-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)