A privacy protection algorithm based on hierarchical multiple sensitive attributes allowed by least mean square criterion

Tao Cui[1, a], Jijia Yang[2, b], Nan Meng[3, c], Wei Xie[4, d]

[1]China Academy of Telecommunication Research (CATR), Beijing, 100191, China

[2]China Academy of Telecommunication Research (CATR), Beijing, 100191, China

[3]China Academy of Telecommunication Research (CATR), Beijing, 100191, China

[4]China Academy of Telecommunication Research (CATR), Beijing, 100191, China

[a]email: cuitao@caict.ac.cn, [b]email: yangjijia@caict.ac.cn,

[c]email: mengnan@caict.ac.cn, [d]email: xiewei@caict.ac.cn

Keywords: Information Security; L-diversity; privacy Protection

Abstract. Based on L-diversity multiple sensitive modules, a hierarchical multiple sensitive attributes algorithm is proposed according to least mean square criterion --- L-LMSU (L-Least Mean Square Uniqueness). The algorithm makes a hierarchical strategy according to the frequencies of the whole attributes firstly. Beyond the hierarchical strategy, the algorithm could decrease the hidden loss because of non-uniform distribution of attributes when releasing privacy data in groups. Analysis and experiments show that L-LMSU is with linear time complexity and could improve the availability of released privacy data and time performances effectively.

Introduction

With the development of network technology, there is an increasing requirement of shared information for various organizations and individuals. How to make privacy information in shared data not leaked becomes a focus of all parties. At present, there are various data release module and algorithms in order to decrease the risk because of information leakage. Among all these research achievements, the original module is k-anonymity [1-3]. The main idea of k-anonymity is to generalize Quasi Identifier property Table (QIT) of every data of each released group and make sure that there are at least k same attribute values in each group. Accordingly, it is difficult for attackers to deduce individuals' privacy information by QIT. However, k-anonymity could not resist from Homogeneity attack and Background Knowledge Attack [4]. Beyond k-anonymity module, L-diversity [5-6] module, (a, k)-anonymity [7] module and t-closeness [8] module are proposed in succession. However, these modules are aiming to achieve single sensitive attribute problems but not multiple sensitive attributes, which could not satisfy the requirement of multiple sensitive attributes. Aiming to protect privacy information with multiple sensitive attributes, Yang Xiaochun [9] proposed a multiple sensitive attributes technology based on multi-dimensions buckets and described 3 corresponding algorithms according to greed method. Document [10] proposes an algorithm to classify multiple sensitive attributes which is to satisfy privacy protection by assign different sensitivity coefficients to different sensitive attributes. Document [11] proposes (k, l) module which assigns every degree of anonymity and the diversity of privacy attribute values of data items and achieves degree of anonymity and diversity with personalization. However, most of current algorithms are of high information hide loss. When releasing data contains various privacy attributes, it is inevitable to destroy data integrity because of high hide loss and low data usability at last.

L-LMSU proposed in this paper, beyond the module called (l_1, l_2, …, l_d)-uniqueness [12], uses the idea of controlling frequencies of attribute values in equivalence class in (a, k)-anonymity. It puts the items with the same attribute value of an attribute into a group bucket. These group buckets are merged together according to the square mean of attribute frequencies. In this return, these merged group buckets frequencies are tend to be equal and decrease information loss while

© 2016. The authors - Published by Atlantis Press

grouping equivalence classes. Experiments show that L-LMSU is of favorable time performances which information loss is tend to zero under ideal conditions. It can also protect from homogeneity attack and background knowledge attack.

Definition and Relative Problem Analysis

Currently, most of multiple sensitive attributes releasing method uses L-diversity module which could protect from homogeneity attack by insuring at least L different attribute values in an equivalence class. However, the drawback of the module is obvious. When partitioning equivalence classes of the data being published, it is inevitable to hide data items because these items could not meet L-diversity grouping criteria and in turn undermine the integrity of the original data. Therefore, in order to minimize the loss of information of released data, and improve the time performance and resist homogeneity attack at the same time, it is necessary to promote an algorithm to solve these problems.

To simplify the discussing issues, we start with the discussion of releasing the single attribute sensitive data. For example, table 1 is a set of data to be released in which *Age* and *Zip Code* is quasi-identifier while *Country* is the privacy data to be released. According to the literature [12], the published rank list of sensitive attribute of *Country* is shown as Table 2. In Table 2, there are 6 group buckets shown as S_1*={Brazil}*, S_2*={Canada}*, S_3*={France}*, S_4*={China}*, S_5*={Japan}*, S_6*={America}*. According to L-diversity, *L* is assigned as 4. After grouped as $G=\{t_5, t_6, t_7, t_8\}$, the remnant items are t_1 to t_4 which contain only 2 sensitive ranks as rank 4 and rank 5. Because there are only 2 sensitive ranks, it cannot meet the L-diversity condition which calls for at least *L=4* sensitive ranks. Even if item t_1 and t_3 could be added to group *G* to generate another group $G'=\{t_1, t_3, t_5, t_6, t_7, t_8\}$, there are still 2 items (t_2 and t_4) loss. So at last t_1 and t_3 would be discarded and the hide loss of *G* is 1/4. So it is likely to result high hide rate of information form designating sensitivity level manually because of nonuniform distribution of sensitivity grade.

Tab.1. raw data

ID	Age	Zip code	Country
t1	25	14111	America
t2	26	14112	America
t3	27	14113	Japan
t4	28	14114	Japan
t5	25	19822	China
t6	43	19823	France
t7	21	19824	Canada
t8	14	19825	Brazil

Tab.2. sensitive grades

Sensitive Grade	0	1	2	3	4	5
Country	Brazil	Canada	France	China	Japan	America

In order to solve the issue of hide loss, we propose the algorithm of L-LMSU. It can be assumed that Table $T\{A_1, A_2, ..., A_p, S_1, S_2, ..., S_d\}$ is the raw table that will be released. A_i $(1 \le i \le p)$ is the Quasi-identifier attribute and S_j $(1 \le j \le d)$ *i*s the sensitive attribute. There are *n* items in Table *T* in which *t[X]* is assigned as the *X* attribute of Table *T*. There are m different attributes in S_j. For the k^{th} attribute in S_j, its frequency is:

$$S_j[k], \text{ as } \sum_{k=1}^{|Sj|} S_j[k] = |T|. \quad (1)$$

Small mean square error of a set of positive real numbers will lead to the uniform distribution closer to the average numbers. The thought of minimum variance classification is right based on this principle and the ideal condition is that the mean square error is equal to zero and any two numbers in the group of these positive real numbers are equal with each other. According to the

information theory, if a record appears frequently, it will carry less information and vice versa. Thus the sensitivity of a data attribute expressed as $S_j(1 \leq j \leq d)$ in a group of data to be released will be on the contrary side to its appearance frequency expressed as $S_j[k]/|S_j$. We can specify appearance frequency $S_j[k]/|S_j|$ as weight for highly sensitive attributes to merge the whole sensitivity level into a lower degree. Minimum mean square deviation can be designed as the sensitive classification policy which could effectively reduce the hidden rate caused by nonuniform distribution of attribute value. Then we will interpret the merging process.

First of all some definitions are proposed as follows.

Definition 1: Initial Sensitivity Level Set

The k^{th} value of the j^{th} sensitive attribute S_j could be denoted as $V=\{v_1, v_2, \ldots, v_k\}$. According to the value of $S_j[k]/|S_j|$, all the items in V would be sorted as v' with ascend order corresponding with the sensitive grade form 0 to k. The initial sensitivity level set will be composed of k sensitive buckets corresponding with k sensitive grades. As shown in Table 2, the sensitive levels in this table are raw data without being handled and each contains only one sensitive attribute.

Definition 2: Merge

For the j^{th} sensitive attribute S_j, two sensitive buckets are merged into a new one whose appearance frequency is the sum of two buckets frequencies before merged.

Tab.3. merged sensitive grades

Sensitive Grades	0	1	2	3
Country	Brazil，Canada	France，China	Japan	America

As shown in Table 3, the sensitive buckets *{0, 1}* and *{2, 3}* shown in Table 2 are merged into two new buckets. The No.0 bucket contains two attributes and its appearance frequency is 1/4. The No.1 bucket has the same value as No.0 bucket. The appearance frequency of No.3 and No.4 buckets remains 1/4, the same as initial sensitive bucket without merging.

Definition 3: L-uniqueness

As to any sensitive attribute $S_i(1 \leq I \leq d)$ in Table *T*, if each group $S_iG_j(1 \leq j \leq m)$ contains at least L different sensitive attribute values which belong to different sensitive levels, the partition of the sensitive attribute S_i in Table T will meet the law of L-uniqueness in which L is the diversity parameter. The diversity parameter could be assigned manually according to various sensitive attribute values. If a sensitive attribute $S_i(1 \leq I \leq d)$ meets the law of L-uniqueness, the probability for attackers to get an sensitive item would be lower than $1/L$.

Definition 4: Frequency of Sensitive Level

For a set containing k sensitive levels, the ratio of items in each sensitive level to all items is denoted as $f_i(1 \leq i \leq k)$ which is defined as frequency of the i^{th} sensitive level in current sensitive level set and simplified as frequency of sensitive level. The f_i takes the feature as:

$$\sum_{i=1}^{k} f_i = 1 \tag{2}$$

Definition 5: Sensitive Level Variance

For a set containing k sensitive levels, the variance of k frequencies of sensitive level is defined as sensitive level variance denoted as:

$$d(f) = \frac{1}{k}\sum_{i=1}^{k}(fi - \overline{f})^2 \tag{3}$$

Definition 6: Sensitive Level Set with Smallest Sensitive Level Variance

We could merge two arbitrary sensitive levels into a new sensitive level set form initial sensitivity level set. Then there will be a sensitive level set with the smallest sensitive level variance In all possible sensitive level sets.

L-LMSU is processed in two phases. The first step is to find a set of the sensitive levels named S with the smallest sensitive level variance through formulating merging policies based on initial sensitivity level set. Then all the data to be released will be equivalence partitioned and grouped

utilizing sensitivity level set *S* according to L-uniqueness module.

The algorithm of merging policy is shown as follows.

(1) Divide different sensitivity values into different buckets and record the weight of each bucket.

(2) Save current sensitivity level buckets denoted as *B* and weight denoted as *C*.

(3) Calculates the minimum mean square error denoted as *S* of sensitivity level buckets.

(4) While (quantity of current sensitivity level bucket > 1)

(5) Merge two buckets with minimum weight into a new bucket denoted as *B'*.

(6) Update the weight *C* to *C'*.

(7) Calculate the minimum mean square error denoted as *S'* of each bucket.

(8) If (*S'* < *S*)

(9) C' = C; B = B'

(10) End if

(11) End while

In the step of merging sensitive level buckets, each time we will merge two sensitivity level buckets with the minimum frequency. There are two reasons. One reason is to make the frequency of sensitivity level buckets as uniformly as possible. The maximum contribution to frequency variance comes from the ones on the edge. The further buckets with minimum or maximum frequency are apart from the center, the larger frequency variance is. So we will merge the two sensitive buckets with minimum frequencies in order to make the combined frequency close to the center. Another reason is to reduce the sensitivity. The frequency of buckets is inversely proportional to the sensitivity of information. The sensitivity of information can be reduced when their combination as a whole.

The algorithm of equivalence partitioning and group releasing is shown as follows.

(1) While (bucket whose records can be taken out exists)

(2) Sort grouped buckets by descending order according to the total number of records in them.

(3) Take out records randomly form *L* buckets which contain most records in the sensitive level set with the smallest sensitive level variance and form an equivalent class.

(4) End while

(5) For (bucket with remaining records exists)

(6) While (there exist records could be inserted into an equivalence class)

(7) Insert the record to the equivalence class

(8) End while

(9) End for

After step (4) some records will be divided to equivalence classes and each class contains *L* items. After step (9) the remained records will be inserted to equivalence classes as far as possible.

The time complexity of L-LMSU could be calculated separately in two steps. In the step of formulating merging policy, the running time of algorithm depends on the attribute values number denoted as *m*. In the process of merging buckets, it will cost *m-1* times for merging into one bucket from *m* buckets and therefore the time complexity is *o(m)*. In the step of partitioning and group releasing, the running time of algorithm depends on the number of records waiting to be released. The number of records is denoted as *n*. From step (1) to (4), it is necessary to traverse all records and the time complexity is *o(n)*. The records needed to traverse during step (5) to (9) are the ones not divided into any equivalence classes formerly. The number of records is denoted as *k*. Because *k* is smaller than *n*, the time complexity of this phase is *o(n+k)=o(n)*. The time complexity of L-LMSU consequently is *o(m+n)* in which *m* is the number of attribute values and *n* is the amount of records to be released.

The L-LMSU algorithm discussed above mainly aims to solve problems with single sensitive attribute. Also this algorithm can be expanded to support multiple sensitive attributes through calculating merging policy and equivalent class partition method for each sensitive attribute separately. L-LMSU will calculate average value of the hiding rate and additional information loss rate of each attribute when counting the whole statistics.

The algorithm of L-LMSU with multiple sensitive attributes is shown as follows. Table $T\{A_1, A_2, ..., A_p, S_1, S_2, ..., S_d\}$ will be the data to be released.

(1) Assign hide loss ratio $H=0$, additional information loss ratio $A=0$, equivalence partitioning set $K=\{\}$.

(2) For ($i = 1$ to d)

(3) As to attribute S_i, calculate the hide loss ratio H_i , additional information loss ratio A_i and equivalence partitioning K_i.

(4) $H = H + H_i$, $A = A + A_i$, $K = K_i$

(5) End for

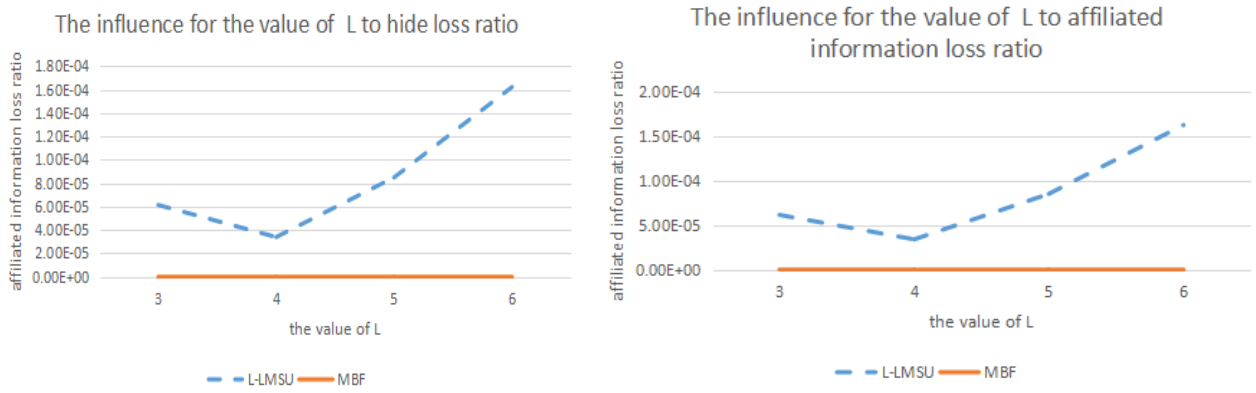
(6) $H = H/d$, $A = A/d$

Experimental Analysis of Algorithms

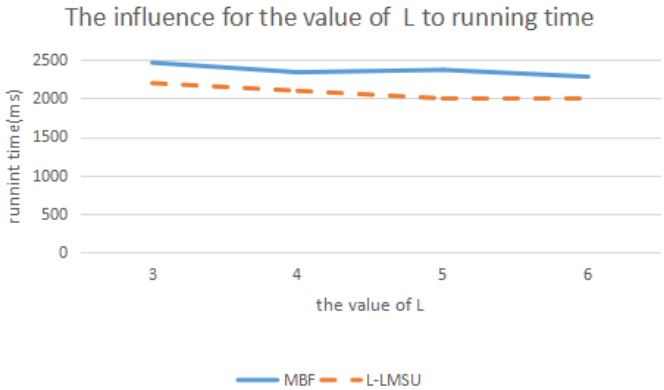
In this paper, the experimental data set is from *Adult Data Set* provided by UCI Machine Learning Repository. *Adult Data Set* is a part of the collection of the United States Census in 1994 and it contains 32561 records which include 14 kinds of attributes such as occupation, relationship and etc.. All the records will be used to compare L-LMSU performance with MBF [9]. The elements for comparison include number of attributes, hide loss ratio, additional information loss and running time. Our experiments will select appropriate attributes and calculate the average value.

The influence of L value to hide loss ratio and additional information loss ratio is shown as follows.

We select *education* and *occupation* as experiment attributes and then compare the hide loss and additional information loss between L-LMSU and MBF.



(a) Influence of L to Hide Loss Ratio (b) Influence of L to Additional Information Loss Ratio



(c) Influence of L to Running Time

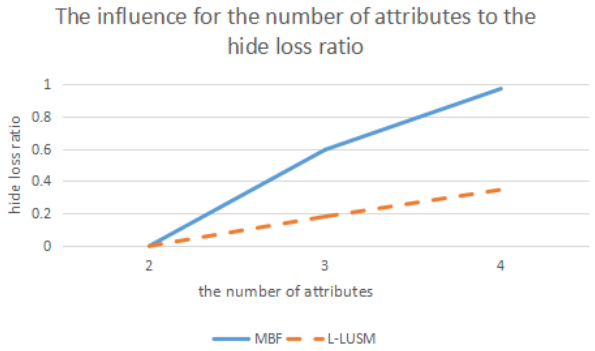
Fig.1. The experimental results-1

Figures above showed the influence of L to hide loss ratio, additional loss ratio and running time. In figure (a) and (c), L-LMSU showed a slow increasing rate of both hide loss ratio and additional

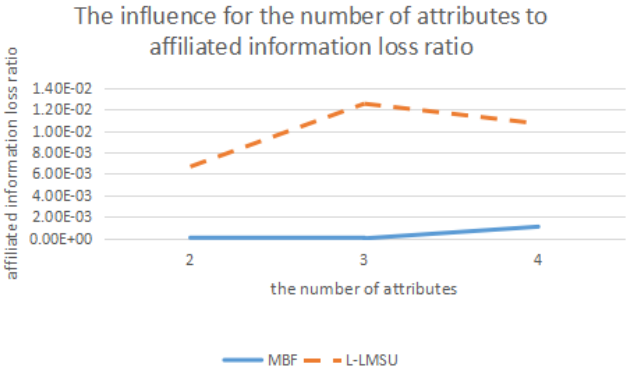
loss ratio with *L* as the horizontal coordinate. L-LMSU also showed a better performance than MBF in running time. Compared with equivalence partitioning through bucket grouping utilizing by MBF, L-LMSU ignores relations between attributes when partitioning equivalence, which reduces the loss because it puts more records into the equivalence class. Moreover the calculation of merging policy before grouping is an important factor to reduce the hide loss. Because of linear complexity in running time, time performance of L-LMSU experimented is better than MBF.

The influence of attribute values number to hide loss ratio and additional information loss ratio is shown as follows.

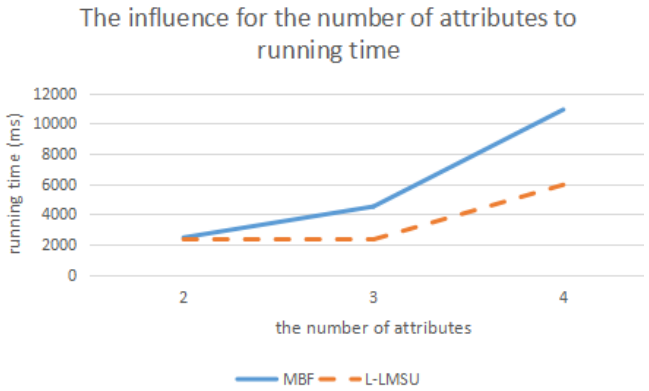
We selected 3 different attribute sets including *{education, occupation}, {education, occupation, workclass}, {education, occupation, workclass, nativecountry}* as experimental samples.



(a) Influence of attribute values number to Hide Loss Ratio



(b) Influence of attribute values number to Additional Information Loss Ratio



(3) Influence of attribute values number to Running Time

Fig.2. The experimental results-2

Figure 2 illustrated the same feature as figure 1. However in figure 1-(b) and figure 2-(b), the additional information loss ratio of MBF is less than L-LMSU. The reason is that the additional information loss depends on the hide loss. Lower hide loss ratio will produce more equivalence classes and there exists great possibility for those undivided records to be grouped in equivalence classes, which will increase the additional information loss ratio.

Conclusion

Based on the module of L-diversity and (l_1, l_2, ... l_d)-uniqueness, we proposed L-LMSU algorithm. L-LMSU improved time performance and reduced hide loss ratio effectively by calculating the merging policy before partitioning equivalences. Experiments show that L-LMSU has better running time performance and relatively lower hide loss ratio. Because L-LMSU mainly deals with the issue of classification information, there is still no research about issue of numerical information. The next step of our work is to extend the L-LUSM algorithm to the field of numerical data analysis.

Acknowledgement

This paper is sponsored by the National Natural Science Foundation of China (NSFC) under Grant No. 61471129 and the National High Technology Research and Development Program of China under Grant No. 2015AA016106.

References

[1]Sweeney L. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY 1[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2008, 10(5):557-570.

[2]Maheshwarkar N, Pathak K, Chourey V. Performance evaluation of various K- anonymity techniques[C]//International Conference on Machine Vision. International Society for Optics and Photonics, 2011:83501Y-83501Y-8.

[3]Latanya sweeney. Achieving k-anonymity privacy protection using generalization and suppression[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2012, 10(5):571-588.

[4]WANG Shenghe,WANG Jiajun,LIU Tengteng,NI Weiwei. Privacy-preserving data publishing method for dataset with multi-dimensional sensitive attributes[J]. Computer Engineering andApplications, 2012, 48(20):136-141.

[5]Machanavajjhala A, Kifer D, Gehrke J. L -diversity[J]. Acm Transactions on Knowledge Discovery from Data, 2007, 1:3-es.

[6]Liu J, Wang K. On optimal anonymization for l+-diversity[C]// Data Engineering (ICDE), 2010 IEEE 26th International Conference on. IEEE, 2010:213-224.

[7]Raymond Wong, Jiuyong Li, Ada Fu,etc. (α, k)-anonymous data publishing[J]. Journal of Intelligent Information Systems, 2009, 33(2):209-234.

[8]Liang H, Yuan H. On the Complexity of t -Closeness Anonymization and Related Problems[M]// Database Systems for Advanced Applications. Springer Berlin Heidelberg, 2013:331-345.

[9]YANG Xiao-Chun WANG Ya-Zhe WANG Bin YU Ge. Privacy Preserving Approaches for Multiple Sensitive Attributes in Data Publishing[J]. Journal of Computer Science and Technology, 2008, 31(4):574-587.

[10]Liu J, Luo J, Huang J Z. Rating: Privacy Preservation for Multiple Attributes with Different Sensitivity Requirements[C]// 2011 11th IEEE International Conference on Data Mining Workshops. IEEE Computer Society, 2011:666-673.

[11]Zude Li, Guoqiang Zhan, Xiaojun Ye. Towards an Anti-inference (K, ℓ)-Anonymity Model with Value Association Rules[J]. Lecture Notes in Computer Science, 2006:883-893.

[12]ZHANG Xing-lan,LIU Le-wei. Privacy Preserving Methods for Multiple Sensitive Attributes[J]. Computer and Modernization, 2013(8):168-171.