BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **13**, Special Issue

Sofia • 2013

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2013-0036

A New Privacy Preserving Association Rule Mining Algorithm Based on Hybrid Partial Hiding Strategy

Jian-Ming Zhu, Ning Zhang, Zhan-Yu Li

School of Information, Central University of Finance and Economics, Beijing 100081, China
zjm@cufe.edu.cn zhangning@cufe.edu.cn lizhanyu813@163.com

Abstract: *Data mining is the progress of automatically discovering high level data and trends in large amounts of data that would otherwise remain hidden. In order to improve the privacy preservation of association rule mining, a hybrid partial hiding algorithm (HPH) is proposed. The original data set can be interfered and transformed by different random parameters. Then, the algorithm of generating frequent items based on HPH is presented. Finally, it can be proved that the privacy of HPH algorithm is better than that of the original algorithm.*

Keywords: *Data mining, association rule, privacy preservation.*

1. Introduction

Today, with the development of e-commerce and e-Government and more and more personal data exchanged online, data privacy has become one of the most important issues. Protection of privacy from unauthorized access is one of the primary concerns in data use, from national security to business transactions. Data mining and knowledge discovery in databases are important areas that investigate the automatic extraction of previously unknown patterns from large amounts of data. The power of data mining tools to extract hidden information from large collections of data lead to increased data collection efforts by companies and government agencies. Naturally this raised privacy concerns about collected data. Therefore, after the data miners collect large amounts of private data from data providers, the data might be perturbed in different ways in order to avoid the privacy disclosure, as well as to keep some useful patterns for further data mining.

At the same time, since the needs to protect privacy information continue to strengthen, privacy protection issues in data mining become the hotspot in research. In the original data, there is some private information we do not want to divulge. Privacy Preserving Data Mining (PPDM) is a method which can obtain more accurate data mining results in case of imprecise access to the original data.

We can classify the representative privacy preserving data mining techniques into two categories, data perturbation and Secure Multi-party Computation (SMC). Secure multi-party computation is a privacy protection technology for distributed data mining; it has accurate results, but requires a large amount of calculation. The aim of data perturbation is to preserve privacy information by perturbing the data values. Based on the different noise addition techniques, this technique can be categorized as additive perturbation method, multiplicative perturbation, data micro aggregation, data anonymization, data swapping and other randomization techniques [1].While divided in accordance with the basic strategy, privacy preserving data mining techniques can be divided into two types, data interference and query restriction [2]. Data interference is to interfere with the original data by some methods (data conversion, noise, etc.), and mining the interference data. Query restriction is the use of certain technologies (data hiding, sampling, etc.) to avoid all raw data presented to the data miners.

2. Problem description

Privacy preserving association rule mining is to find frequent itemsets in case of imprecise access to the original dataset and provide the association rules meeting the given support and confidence. The most famous algorithm is Mining Associations with Secrecy Constraints (MASK) proposed by R i z v i and H a r i t s a [3]. The main idea of this algorithm is to map the original dataset into two-dimensional Boolean matrix, then transform the data with the Bernoulli probability model. Data miners can get the transformed Boolean matrix and estimate the original support by the reconstruction algorithm to discover frequent itemsets. MASK algorithm protects privacy through the method of data interference, but has certain limitations. The transformed data and the original data are relevant, privacy protection effect is not very ideal, and the value of the random parameter is subject to certain restrictions.

Another privacy preserving association rule mining algorithm is proposed by Z h a n g et al. [2], called Randomized Response with Partial Hiding (RRPH). This algorithm uses three randomized parameters to interfere with the data, has better properties and efficiency compared to MASK algorithm. However, RRPH still has limitations for the data corresponding to the first random parameter has not been disturbed, which makes the data privacy not well protected. There are some other improved algorithms, like Privacy Association Rules Mining-Related Technology [4], multi-parameters randomized disturb algorithm [5], Partial Hiding Transition Probability Matrix [6]. For the inadequacies of the above algorithms, we proposed a Hybrid Partial Hiding algorithm (HPH) to interfere with the original data, and also

given the frequent itemset generation algorithm, better realize the privacy protection in association rule mining.

3. Hybrid partial hiding algorithm

HPH is a data perturbation algorithm to transform and hide raw data. Here we are dealing with Boolean data, which means all items are mapped from to 1 or 0. Let $I=\{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items, such that $T \subseteq I$. The algorithm is as follows:

There are four random parameters p_1, p_2, p_3, p_b, where $0 \leq p_1, p_2, p_3, p_b \leq 1$ and $p_1+p_2+p_3=1$. For $x \in \{0, 1\}$, the random function is $r(x)$. Let $r_1 = 1$, $r_2 = 0$, $r_3 = d(x)$, the random function takes the value r_i with probability p_i. Function $d(x)$ means the value of x is kept the same with probability p and is flipped with probability $1-p$. Table 1 shows the correspondence relationship of the function value and the probability. We use the same random parameters for all the items, and the dataset transformation process is independent for each column.

Table 1. $r(x)$ function value and the probability

$r(x)$	1	0	x	$1-x$
Probability	p_1	p_2	p_3p_b	$p_3(1-p_b)$

That is to say, x takes the value 1 with probability p_1, takes the value 0 with probability p_2; the value of x is kept the same with probability $p_3 p_b$ and is flipped with probability $p_3(1 - p_b)$. The algorithm achieved data interference strategy through four random parameters, and the items are hidden when parameter p_2 changed the value of the function to 0. We can hide the data that needs to be protected by parameter p_2. In this way the two privacy preserving strategies, data perturbation and query restriction are combined to transform and hide the original data. The following is the specific algorithm for data processing using HPH method.

Algorithm 1. Hybrid Partial Hiding algorithm

- Input: the original transaction set D, random parameters p_1, p_2, p_b.
- Output: the transaction set D' after the processing of HPH algorithm.
- Method:

(1) Scan the transaction set D, for each transaction $t \in D$ {

(2) for (k=0; k<N; k++) // N is the number of items contained in each transaction

(3) for each item $i \in I$ {

(4) Generate random number θ_1; //$0 \leq \theta_1 \leq 1$

(5) if ($\theta_1 \leq p_1$) i=1; //item i takes the value 1 with probability p_1

(6) else if ($p_1 \leq \theta_1 \leq p_1+p_2$) i=0; //item i takes the value 0 with probability p_2

(7) else {

(8) Generate random number θ_2; //$0 \le \theta_2 \le 1$

(9) if ($\theta_2 \le p_b$) $i=i$; //the value of i is kept the same with probability $(1-p_1-p_2)p_b$

(10) else $i=1-i$; // the value of i is flipped with probability $(1-p_1-p_2)(1-p_b)$

(11) }

(12) }

(13) }

(14) Output the transaction set D' after the processing of HPH algorithm.

4. Privacy preserving association rule mining algorithm

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps:

- First, minimum support is applied to find all frequent itemsets in a database;
- Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

As it can be seen, the generation of the association rules is based on frequent itemsets, has nothing to do with the original transaction set, so our study is to protect privacy in the process of discovering frequent itemsets, that is to find the frequent itemsets as accurately as possible after transform and hide the raw data. First, we describe how to reconstruct the support of the set, and then give the complete association rule data mining algorithm.

4.1. Reconstructing the support of singleton

For convenience, assuming the original dataset is S, the dataset after HPH algorithm is D. Now we consider the i-th column in the dataset, let C_1^S and C_0^S represent the number of 1's and 0's in the i-th column of S, while C_1^D and C_0^D represent the number of 1 and 0 in the i-th column of D. Let M be the transformation matrix, $C^D = MC^S$, that is $C^S = M^{-1}C^D$, where $C^S = \begin{bmatrix} C_1^S \\ C_0^S \end{bmatrix}$, $C^D = \begin{bmatrix} C_1^D \\ C_0^D \end{bmatrix}$. Since D is the processed dataset, the value of C^D can be derived by scanning, so it is possible to calculate the value of C^S as long as we know the form of M. The item S_i in dataset S is converted into the item D_i in dataset D after HPH algorithm; Table 2 shows the probability of data mapping.

Table 2. Data mapping probability of HPH algorithm

No	S_i	D_i	Mapping probability
1	1	1	$p_1+p_3p_b$
2	1	0	$p_2+p_3(1-p_b)$
3	0	0	$p_2+p_3p_b$
4	0	1	$p_1+p_3(1-p_b)$

We can reach the following conclusion:

$$C_1^D = (p_1 + p_3 p_b) C_1^S,\ C_0^D = (p_2 + p_3(1-p_b)) C_1^S,$$
$$C_0^D = (p_2 + p_3 p_b) C_0^S,\ C_1^D = (p_1 + p_3(1-p_b)) C_0^S,$$
$$C_D = \begin{bmatrix} C_1^D \\ C_0^D \end{bmatrix} = \begin{bmatrix} p_1 + p_3 p_b & p_1 + p_3(1-p_b) \\ p_2 + p_3(1-p_b) & p_2 + p_3 p_b \end{bmatrix} \begin{bmatrix} C_1^S \\ C_0^S \end{bmatrix} =$$
$$= \begin{bmatrix} p_1 + p_3 p_b & p_1 + p_3(1-p_b) \\ p_2 + p_3(1-p_b) & p_2 + p_3 p_b \end{bmatrix} C^S,$$
$$M = \begin{bmatrix} p_1 + p_3 p_b & p_1 + p_3(1-p_b) \\ p_2 + p_3(1-p_b) & p_2 + p_3 p_b \end{bmatrix}$$

(M is reversible).

Thus, after obtaining the inverse matrix of M, the value of C_1^S, i.e., the original support count of attribute i can be calculated from the values of C_1^D and C_0^D.

4.2. Reconstructing the support of k-itemset

A similar method can be used to reconstruct the support of k-itemset. We define the matrices as:

$$C^S = \begin{bmatrix} C_{2^k-1}^S \\ \vdots \\ C_1^S \\ C_0^S \end{bmatrix},\ C^D = \begin{bmatrix} C_{2^k-1}^D \\ \vdots \\ C_1^D \\ C_0^D \end{bmatrix}.$$

For a given k-itemset $A=\{i_1, i_2, \ldots, i_k\}$, C_n^D represents the count of the tuples in D that have the binary form of n. For instance, for a 2-itemset $A = \{i_3, i_6\}$, if the corresponding item of $\{i_3, i_6\}$ is $\{1, 0\}$, then the decimal value is 2, C_2^D refers to the count of this sequence in dataset D. By the same token, C_3^D represents the number of tuples of the sequence 11. The definition of C_n^S is similar.

We define the transform matrix $M_k=[m_{ij}](2^k \times 2^k)$, the value of m_{ij} is the probability that a tuple j of the form corresponding to C_j^S in the original dataset S goes to a tuple i of the form corresponding to C_i^D in dataset D by HPH algorithm. For instance, the value of m_{13} for a 2-itemset is the probability that a 11 tuple transforms to a 01 tuple. As the dataset transformation process is independent for each column, the value of m_{ij} can be calculated according to Table 2, and thereby get the transform matrix M_k. When M_k is reversible, let a_{ij} represents the element in M_k^{-1}, that is $M_k^{-1} = [a_{ij}]$. The support count of the k-itemset can be calculated by $C^S = M^{-1} C^D$, accordingly, $C_{2^k-1}^S = a_{0,0} C_{2^k-1}^D + a_{0,1} C_{2^k-2}^D + \cdots + a_{0,2^k-2} C_1^D + a_{0,2^k-1} C_0^D$.

4.3. The complete data mining algorithm

After the reconstruction of support, we can start mining association rules. The algorithm we used is based on the Apriori algorithm. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted by L_1. Next, L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found. The finding of each L_k requires one full scan of the database [7]. The difference of our algorithm is that, in the k-th scan, we need to count all $C_i^D(i=1,2,\cdots,2^k-1)$ to calculate the original support and generate frequent itemsets. The algorithm used to generate frequent itemsets for the HPH algorithm processed data is given below.

Algorithm 2. Find frequent itemsets in HPH algorithm processed data

- Input: dataset D after HPH algorithm processed; minimum support count threshold *minsup*.
- Output: L, frequent itemsets in D.
- Method:

(1) Scan D, for each $i \in I$ count C_0^D, C_1^D;

(2) $L_1=\{\{i\}|i\in I, (C_1^D a_{0,0}+C_0^D a_{0,1})/N\geq \text{minsup}\}$;

(3) for (k=2; $L_{k-1}\neq\phi$; k++) {

(4) C_k=aproiri_gen(L_{k-1});

(5) for each candidate $c\in C_k$ {

(6) for (n=0; $n<2^k$; n++) count $c.C_n^D$;

(7) $c.C_{2^k-1}^S=\sum_{n=0}^{2^k-1} a_{0,n}c.C_{2^k-1-n}^D$;

(8) }

(9) $L_k=\{\{c\}|c\in C_k, c.C_{2^k-1}^S/N\geq \text{minsup}\}$;

(10) }

(11) Return $L=\cup_k L_k$;

5. Algorithm performance evaluation

5.1. Reconstruction probability of the original data

The purpose of privacy preserving association rule mining is to obtain more accurate frequent itemsets in case of imprecise access to the original data, so the reconstruction probability of the original data can be used to evaluate the algorithm. If it has large reconstruction probability, this algorithm does not achieve better privacy protection.

We use the same symbols in 4.1, assuming that the original dataset is S, the dataset after HPH algorithm is D. If the value of an item i in the original dataset S is 1, i.e., $S_i = 1$, then the probability of $S_i = 1$ can be judged from the corresponding item D_i in the dataset D is the reconstruction probability of this item. Let $\sup_i$ represents the original support of item i, i.e., the probability for $S_i = 1$ is $\sup_i$. The reconstruction probability for an item whose original value is 1 can be calculated by the following formula [3]:

$$R_1(p,\sup_i) = P(D_i = 1 \mid S_i = 1) \times P(S_i = 1 \mid D_i = 1) +$$

$$+P(D_i = 0 \mid S_i = 1) \times P(S_i = 1 \mid D_i = 0).$$

We can see from Table 2 that

$$P(D_i = 1 \mid S_i = 1) = p_1 + p_3 p_b, \quad P(D_i = 0 \mid S_i = 1) = p_2 + p_3(1 - p_b).$$

Putting them into the above equation, we can obtain:

$$R_1(p,\sup_i) = (p_1 + p_3 p_b) \times P(S_i = 1 \mid D_i = 1) +$$

$$+P(D_i = 0 \mid S_i = 1) \times (p_2 + p_3(1 - p_b)).$$

We can obtain from the conditional probability formula that:

$$P(S_i = 1 \mid D_i = 1) = \frac{P(S_i = 1, D_i = 1)}{P(D_i = 1)} =$$

$$= \frac{P(S_i = 1) \times P(D_i = 1 \mid S_i = 1)}{P(S_i = 1) \times P(D_i = 1 \mid S_i = 1) + P(S_i = 0) \times P(D_i = 1 \mid S_i = 0)} =$$

$$= \frac{\sup_i \times (p_1 + p_3 p_b)}{\sup_i \times (p_1 + p_3 p_b) + (1 - \sup_i) \times (p_1 + p_3(1 - p_b))}.$$

Similarly,

$$P(S_i = 1 \mid D_i = 0) = \frac{\sup_i \times (p_2 + p_3(1 - p_b))}{\sup_i \times (p_2 + p_3(1 - p_b)) + (1 - \sup_i) \times (p_2 + p_3 p_b)}.$$

Thus it can be seen:

$$R_1(p,\sup_i) = \frac{\sup_i \times (p_1 + p_3 p_b)^2}{\sup_i \times (p_1 + p_3 p_b) + (1 - \sup_i) \times (p_1 + p_3(1 - p_b))} +$$

$$+ \frac{\sup_i \times (p_2 + p_3(1 - p_b))^2}{\sup_i \times (p_2 + p_3(1 - p_b)) + (1 - \sup_i) \times (p_2 + p_3 p_b)}.$$

The above expression reflects the reconstruction probability of an item whose original value is 1.To get the total measure of the reconstruction probability, we need to summarize all items:

$$R_1(p) = \frac{\sum_i \sup_i R_1(p,\sup_i)}{\sum_i \sup_i}.$$

When all items use the same original support sup, the total reconstruction probability can be expressed as

$$R_1(p)=\frac{\sup\times(p_1+p_3p_b)^2}{\sup\times(p_1+p_3p_b)+(1-\sup)\times(p_1+p_3(1-p_b))}+$$

$$+\frac{\sup\times(p_2+p_3(1-p_b))^2}{\sup\times(p_2+p_3(1-p_b))+(1-\sup)\times(p_2+p_3p_b)}.$$

Similarly, the total reconstruction probability for an item whose original value is 0 can be calculated as:

$$R_0(p)=\frac{(1-\sup)\times(p_1+p_3(1-p_b))^2}{\sup\times(p_1+p_3p_b)+(1-\sup)\times(p_1+p_3(1-p_b))}+$$

$$+\frac{(1-\sup)\times(p_2+p_3p_b)^2}{\sup\times(p_2+p_3(1-p_b))+(1-\sup)\times(p_2+p_3p_b)}.$$

So the total reconstruction probability is

$$R(p)=\alpha R_1(p)+(1-\alpha)R_0(p),\ \ \alpha\in[0,1],$$

where α represents the weight.

When p_1=0.2, p_2=0.3, p_3=0.5, Fig. 1 shows the reconstruction probability of HPH algorithm and RRPH algorithm for different values of p_b. As can be seen from Fig. 1, no matter how the support changes, the reconstruction probability of RRPH algorithm is always higher than the HPH algorithm, which means the data after using HPH algorithm is not easy to be reconstructed. At the same time, with the rise of the support, the reconstruction probabilities of the two algorithms are on the rise, i.e., the reconstruction probability of these two algorithms are proportional to the support.

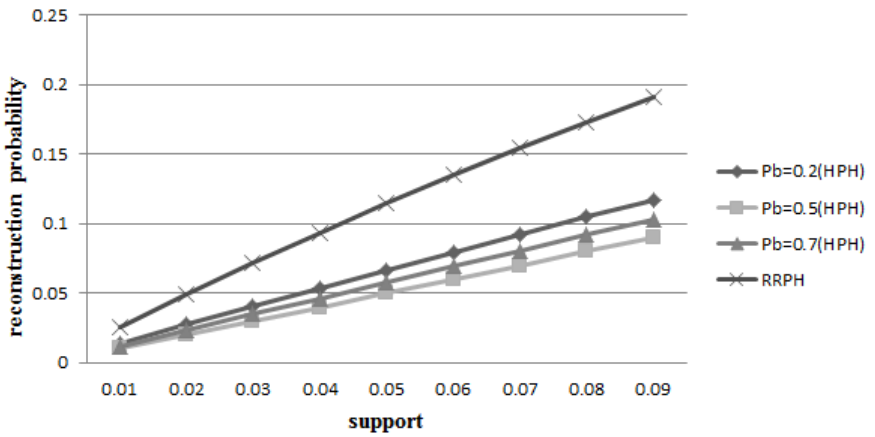

Fig. 1. Reconstruction probability of HPH and RRPH

5.2. Privacy measure

After calculating the reconstruction probability of the dataset, we can simply define the privacy degree [3]:

$$P(p)=(1-R(p))\times 100\%.$$

When p_1=0.2, p_2=0.3, p_3=0.5, Fig. 2 shows the privacy degree of HPH algorithm and RRPH algorithm for different values of p_b. As can be seen from Fig. 2, no matter how the support changes, the privacy degree of HPH algorithm is always higher than the RRPH algorithm, which means HPH algorithm has better privacy protection performance. We can get the same conclusion when the randomization parameters p_1, p_2, p_3 take other values.

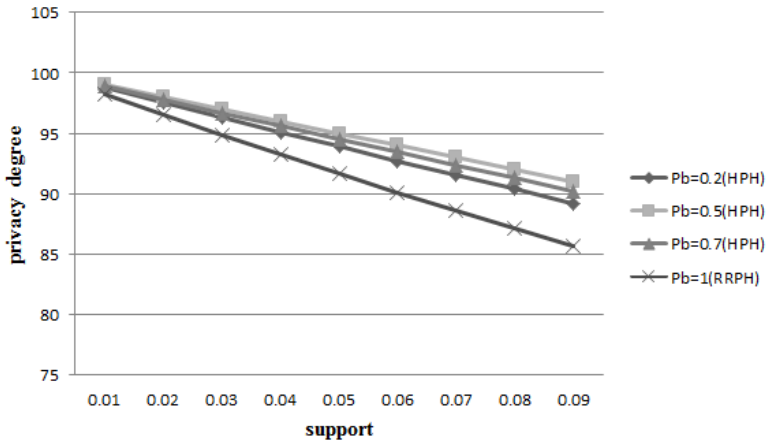

Fig. 2. Privacy degree of HPH and RRPH

6. Conclusion and future work

For the inadequacies of the existing algorithms, we propose a hybrid partial hiding algorithm to improve the privacy preservation. Then, a privacy preserving association rule mining algorithm, based on HPH is presented. Finally, we compare and evaluate the performance of the algorithm which proves that the HPH algorithm outperforms the existing algorithms.

In future work, we hope to improve the operating efficiency of the algorithm and apply the algorithm to more types of data mining.

Acknowledgments: This work is supported by the National Natural Science Foundation of China under Grant 60970143 and 61272398, the National Social Science Foundation of China under Grant 13AXW010, and the Beijing Natural Science Foundation under Grant 4112053.

References

1. Liu, L. Perturbation Based Privacy Preserving Data Mining Techniques for Real-World Data. PhD. Thesis, Unversity of Texas, Dallas, 2008.
2. Zhang, P., et al. An Effective Method for Privacy Preserving Association Rule Mining. – Journal of Software, Vol. **8**, 2006, 1764-1774.
3. Rizvi, S. J., J. R. Haritsa. Maintaining Data Privacy in Association Rule Mining. – In: Proceedings of the 28th International Conference on Very Large Databases, Hong Kong, China, 2002.
4. Zhao, C. H., L. P. Lv. Privacy Association Rules Mining-Related Technology. – In: Proceedings of the 2010 International Conference on Semiconductor Laser and Photonics, Chengdu, China, 2010.
5. Wang, R., J. Liu. Research of Privacy Preserving Association Rules Mining Algorithm. – Computer Engineering and Applications, Vol. **26**, 2009, 126-130.
6. Zheng, L. R., J. Yin. An Association Rule Mining Algorithm Privacy Preserving. – Modern Computer, Vol. **6**, 2009, 10-14.
7. Han, J. W., M. Kamber. Data Mining: Concepts and Techniques. Second Edition. San Francisco, Morgan Kaufmann Publishers, 2006.
8. Yin, Y., et al. Data Mining: Concepts, Methods and Applications in Management and Engineering Design. London, Springer, 2011.