Completely Hide Sensitive Association Rules Using EMO by Deleting Transactions

Peng Cheng, Jeng-Shyang Pan

Shenzhen Graduate School, Harbin Institute of Technology,
HIT Campus of University Town of Shenzhen, Shenzhen, China (518055)
{chengp.mail, jengshyangpan}@gmail.com

ABSTRACT

Data mining techniques enable efficient extraction of useful knowledge from a large data repository. However, it also can disclose sensitive information if used inappropriately. A feasible way to address this problem is to sanitize the database to conceal sensitive information. In this paper, we focus on privacy preserving in association rule mining. In light of the tradeoff between hiding sensitive rules and disclosing non-sensitive ones during the hiding process, a novel association rule hiding approach is proposed based on evolutionary multi-objective optimization (EMO). It modifies the original database by deleting identified transactions/tuples to hide sensitive rules. Experiment results are reported to show the effectiveness of the proposed approach.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications --- *Data mining*; I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search --- *Heuristic methods*

Keywords

Association rule hiding; evolutionary multi-objective optimization; EMO

1. INTRODUCTION

Data mining technologies have been widely used in many fields. They can discover useful information in a large data repository and provide valuable knowledge for users to make decisions. However, data mining technology could be misused and lead to the disclosure risk of user's privacy. Association rule mining is a common used technique in data mining. In a like manner, it also can pose a threat to privacy if not used properly. Oftentimes users hope to protect some sensitive rules not to be mined out after the database is released or shared, because of privacy policy enforcement or possibility of providing competitors with a business advantage.

Association rule hiding refers to modification on the database in some ways so that certain sensitive rules existent in the original database cannot be mined out in the modified database. There are at least three side effects which can be used to evaluate the hiding performance:

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2598466

1) How many sensitive rules which fail to be hidden?
2) How many non-sensitive rules are lost in the modified database?
3) How many new spurious rules are generated after the hiding process?

Some exact algorithms were proposed to solve it in [1]. They tried to find the optimal solutions by sorting the transactions with the transaction sizes and select the shortest transactions to modify firstly. So far, there is no related work to take it as a multi-objective optimization problem (MOP) to solve although this problem beholds the characteristic of a MOP. In this paper, we solved the rules hiding problem as a multi-objective optimization process and formulated the above three side effects as optimization goals to be minimized. NSGA II [4] was utilized to hide rules by removing transactions.

2. THE METHOD BASED ON EMO

The task is to search the database and find out appropriate subset of candidate transactions to remove in such a way that sensitive rules escape mining in the modified database at some predefined thresholds, while the three side effects are minimized to the maximal extent. Different transactions removing choice will incur different side effects. So the three side effects can act as the optimization goals of this problem. Before introducing the hiding method, some notations are defined as follows:

- $|DB|$: The size of the database DB.
- $Supp(X)$: The relative support of the itemset X
- $Conf(X \rightarrow Y)$: The confidence of the rule $X \rightarrow Y$.
- MST: The minimum support threshold.
- MCT: The minimum confidence threshold.

The proposed hiding approach, named as "EMO-RH-DT", consists of two main phases: in its initial phase, an improved version [2] of the Apriori algorithm is used to find all frequent patterns and association rules under given MST and MCT. The output of the first phase is a set of frequent item sets and association rules. In the second phase, user need select some rules as sensitive ones from this set. Then the EMO algorithm is performed to find the optimal subset of transactions to remove.

The method hides a sensitive rule by reducing its support below MST or its confidence below MCT. Figure 1 gives an example to demonstrate how to hide rules by deleting transactions and the side effects. Assuming MST=50% and MCT=80%, the sensitive rules $A \rightarrow C$ is hidden by deleting the 2nd and 4th transactions because its confidence is below threshold after modification. Meanwhile, non-sensitive rule $A \rightarrow D$ is lost mistakenly due to its confidence also below MCT in modified database.

However, if we select the 1st (indicated by a dotted line) and 4th transactions to remove, the sensitive rule $A \rightarrow C$ still can be hidden but the non-sensitive rule $A \rightarrow D$ will not be missing. By this way,

Supp(*A*→*D*)=66.7% and *Conf*(*A*→*D*) = 100%. Alternately, we also may choose 1st and 2nd to delete.

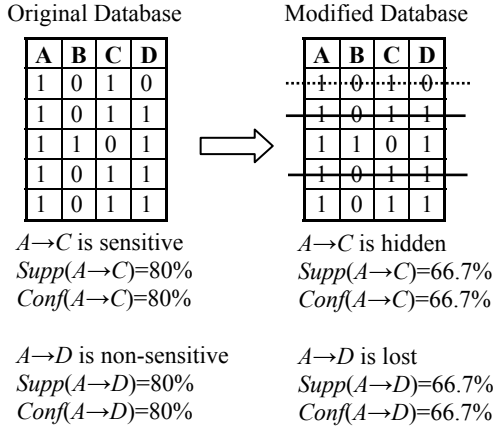

Figure 1. Hiding by deleting transactions and its side effects (*MST*=50%, *MCT*=80%)

2.1 The encoding scheme

The chromosome represents the set of *IDs* of selected transactions to be removed. Each gene represents an *ID* of one selected transaction. The chromosome is divided into several segments. Assuming there are *n-1* sensitive rules to be hidden, and then the chromosome includes *n* segments. Each of the front *n-1* segments is related to a sensitive rule and it selects genes only from the transactions group which supports the corresponding sensitive rule. In addition, the last one segment is related to spurious rules and it picks genes only from the transactions group which support itemsets with the absolute support between *MST**(|*DB*|-*MAX_DEL_LEN*) and *MST**|*DB*|. Here *MAX_DEL_LEN* is the maximum number of transactions allowed to be removed. This kind of itemsets could easily become spurious rules after database reduction.

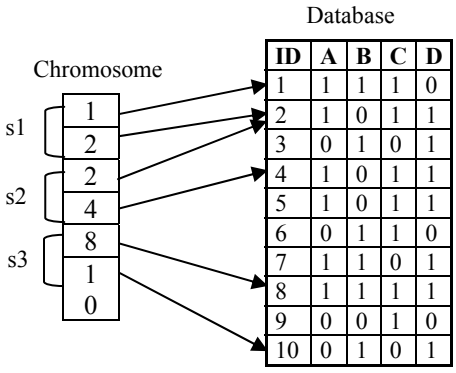

Figure 2. The encoding mechanism

Figure 2 illustrates the principle of encoding. According to the Figure 2, the collection of *IDs* contained in the whole chromosome is {1, 2, 4, 8, 10}. Thus, the transactions with *ID* in {1, 2, 4, 8, 10} need to be removed. Assume *MST*=50% and *MCT*=75%. The chromosome is divided into 3 segments: s1, s2 and s3. Part s1 is related to the sensitive rule *A*→*C* and part s2 is devised for the sensitive rule *A*→*D*. Before modification, *Supp*(*A*→*C*)=50%, *Conf*(*A*→*C*)=83.3%, *Supp*(*A*→*D*)=50% and *Conf*(*A*→*D*)=83.3%; After modification, *Supp*(*A*→*C*)=20% *Conf*(*A*→*C*)=50%, *Supp*(*A*→*D*)=40%, *Conf*(*A*→*D*)=100%. Hence the rule *A*→*C* and the rule *A*→*D* are hidden. Part s3 is aimed to keep from generating the spurious rule *B*→*D*.

The length of each segment can be determined by calculating how many transactions are need to be removed to ensure that the corresponding sensitive rule can be hidden. Here we adopt the strategy of reducing the support of sensitive rule below *MST*. For the sensitive rule *X*→*Y*, it can be hidden if the following amount of transactions is removed from its supporting transactions set.

$$\lceil Supp(X \cup Y)^* \mid DB \mid -MST(\mid DB \mid -MAX\_DEL\_LEN) \rceil + 1$$

For the front *n-1* segments in the chromosome, each segment's length can be calculated by the above formula. Thus the encoding can ensure all sensitive rules to be hidden if the overall length of the front *n-1* segments is not beyond *MAX_DEL_LEN*.

3. PERFORMANCE EVALUATION

We tested the proposed algorithm on the mushroom, BMS-WebView-1 and BMS-WebView-2 datasets. The proposed algorithm was implemented in C++ based on the PISA platform [3].The population size is 30 and the maximal evolution is 100. Figure 3 shows one of the running results on the mushroom dataset. Because all sensitive rules could be hidden, the outcome is only shown in two dimensions. The outcome in the final generation often only consists of several different solutions. This phenomenon comes from the very sparse objective space. We will attempt to design different objective functions in order to take full advantage of the selection mechanism of EMO.

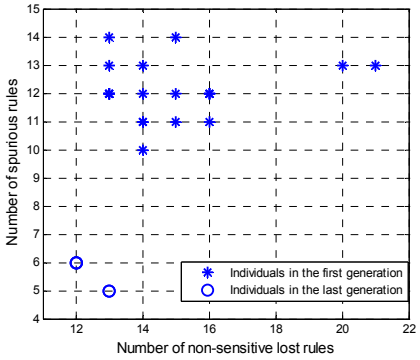

Figure 3. One of the running results on the mushroom dataset

References

[1] V.S. Verykios, A.K. Elmagarmid, and et al. Association rule hiding. IEEE Transactions Knowledge and Data Engineering 16(4): 434-447, 2004.

[2] F. Bodon. Surprising results of trie-based FIM algorithms. In Proceedings of IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, 2004.

[3] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA - A platform and programming language independent interface for search algorithms. In Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), volume 2632 of LNCS, Berlin, pp. 494--508, 2003, Springer.

[4] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2): 182–197, 2002.