Use EMO to Protect Sensitive Knowledge in Association Rule Mining by Adding Items

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

When data is released or shared among different organizations, some sensitive or confidential information may be subject to be exposed by using data mining tools. Thus, a question arises: how can we protect sensitive knowledge while allowing other parties to extract the knowledge behind the shared data. In this paper, we address the problem of privacy preserving in association rule mining from the perspective of multi-objective optimization. A sensitive rule can be hidden by adding items into the dataset to make the support of the antecedent part of the sensitive rule increase and accordingly the confidence of the sensitive rule decrease. The evolutionary multi-objective optimization (EMO) algorithm is utilized to find suitable transactions (or tuples) to be modified so as the side effects to be minimized. Experiments on real datasets demonstrated the effectiveness of the proposed method.

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; EMO; evolutionary multi-objective optimization

1. INTRODUCTION

The rapid developing data mining techniques enable people to extract useful knowledge from a large data collection. However, data mining also can pose the threat of disclosing sensitive knowledge when data is shared or released to other parties inappropriately. Verikios et al. [1] presented the scenario where a supermarket needs to hide sensitive rules before releasing the database of customer purchases to a paper company. In this paper, we focus on sensitive knowledge hiding in association rule mining. In order to avoid exposing the privacy, the released data can be modified in some way so that the sensitive rules cannot be mined out at specified thresholds.

However, database modification could lead to non-sensitive rules also to be lost or new spurious rules to be generated. So the challenge is how to protect sensitive rules while minimizing these side effects.

Most algorithms proposed so far on association rule hiding are deterministic [1, 2]. Intentionally or not intentionally, they made some limitation or assumption on the problem to be solved. For instance, the methods in [1] suppose implicitly that the transaction size in a database varies greatly. However, not all databases behold this characteristic. In this paper, we solved the rules hiding problem from the point view of multi-objective optimization. A new hiding method based on evolutionary multi-objective optimization (EMO) is proposed. It is robust to the database used.

2. THE PROPOSED SOLUTION

The hiding process is as follows. First, for each sensitive rule, the transactions which partially support the antecedent but not support the whole rule are filtered out. Secondly, from the filtered transactions, the EMO algorithm is utilized to find suitable candidates to modify by inserting new items so as to hide sensitive rules.

The three possible side effects produced by the hiding process can be formulated as optimization goals for EMO. They are respectively:

|S-N-H|: the number of sensitive rules not to be hidden.
|N-S-L|: the number of non-sensitive lost rules.
|S-F-G|: the number of spurious rules newly generated.

For instance, assuming there are 8000 transactions in the database, after calculation we need to modify at least 200 transactions by adding items to hide all sensitive rules. The task of EMO is to find out 200 transactions so that the modification on these identified transactions can produce the minimal side effects.

Generally, for this problem, the decision space is very huge but the objective space is sparse. For the above instance, there are 200 dimensions in the decision space, in contrast to 3 dimensions in the objective space. Furthermore, each dimension in the decision space takes the value from the large set {1, 2, …, 8000}.

Before introducing the algorithm, the meanings of some notations are defined as following:

- D: The database to be sanitized.
- R_S: The sensitive rules set.
- $Supp(X)$: 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.

2.1 The hiding strategy

Given a sensitive rule $X \rightarrow Y$, let $\Sigma_{X'}$ be the set of all transactions which partially (or not) support rule's antecedent X and not (fully) support Y. The model we adopted to hide it is to insert new

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2598465

items into some transaction in Σ_X to make them fully support the antecedent X. The support of the antecedent is increased while the support of the generating itemset of the rule remains unchanged. Thus the rule's confidence descends. When the confidence falls below *MCT*, the rule is hidden. Assume we need to modify at least NUM_{add} transactions in $\Sigma_{X'}$ to reduce the confidence of the rule below *MCT*, then

$$NUM_{add}= \lceil Sup(X \cup Y)*|D|/MCT - Sup(X)*|D| \rceil+1$$

2.2 The algorithm EMO-AddItem

The key of this optimization problem is how to find suitable transactions subset to modify (by adding items) so as to minimizing side effects. The selection mechanism of NSGA II [5] is used to solve it. The encoding mechanism adopted is the integer encoding. The chromosome represents the set of *IDs* of selected transactions to be modified. Each gene on a chromosome represents an *ID* of one selected transaction. The chromosome is divided into several segments. Assuming there are *n* sensitive rules to be hidden, and then the chromosome includes *n* segments. Each of the *n* segments is related to a distinct sensitive rule and it selects genes only from the transactions group which partially supports the antecedent of corresponding sensitive rule.

```
Algorithm: EMO-AddItem
INPUT: Database D, MST, MCT, and the sensitive rules set $R_S$.
OUTPUT: The sanitized database in which sensitive rules cannot
        be mined out.
BEGIN
  Find frequent item sets and association rules using improved
  Apriori algorithm [3]
  For each sensitive rule $r_i$: $X \rightarrow Y$
  {
     $\Sigma_{i'}$ = {t∈D | t partially support the antecedent of $r_i$
                  and not (fully) support the consequent of $r_i$ }
     $NUM_i = \lceil Supp(X \cup Y)*|D|/MCT - Supp(X)*|D| \rceil+1$
     // calculate the minimal number of transactions to
     // be modified in order to hide $X \rightarrow Y$
     Length of the $i^{th}$ chromosome segment = $NUM_i$
  }
  T = EMO_find( )
  // utilize EMO to find transactions set T to modify in order
  // to hide sensitive rules while minimizing side effects.
  // T is divided into s parts, i.e., $T_1, T_2, \ldots, T_s$ , s= $|R_S|$.
  // the $T_i$ part contains selected transactions from $\Sigma_i$.
  // $|T_i|$ is equal to $NUM_i$.
  For each $T_i$ in T
     For each transaction t in $T_i$
        Adding items into t to make it fully support the ante-
        cedent of the rule $r_i$
END
```

3. PERFORMANCE EVALUATIONS

We tested the proposed algorithm on three representative real databases: mushroom, BMS-WebView-1 and BMS-WebView-2. These datasets exhibit varying characteristics with respect to the number of transactions and items that they contain, as well as with respect to the average transaction length. The implementation is based on PISA [4]. Experiment results were measured according to three side effects.

The population size was 40 and the maximal generation for evolution was 100. The crossover probability was 0.95 and the mutation probability was 0.1. The effectiveness of the proposed algorithm under various *MCT*s and on different datasets was tested, as indicated in Table 1. Five sensitive rules were randomly selected from the association rules set. As indicated in Table 1, lower *MCT* values may produce more association rules and possible more side effects. The performance of the EMO-based methods was compared with the performance of the exact method *1.a* proposed in [1]. Both of them hide rules by adding items. In Table 1, each row shows one group of comparison. For each pair of comparison, we adopt the same dataset and sensitive rules set, and the *MCT* and *MST* values used are also the same. The initial outcome indicates that the algorithm EMO-AddItem may achieve as good or better results (not statistically).

Table 1. Comparison of the algorithm EMO-AddItem and *1.a* to hide 5 sensitive rules in three real datasets.

Dataset	***MCT***	**|*R*|**	**Side effects: (|S-N-H|, |N-S-L|, |S-F-G|)**	
			EMO-AddItem	1.a
Mushroom (*MST*=0.05)	0.6	849	(0, 1, 0)	(0, 7, 0)
	0.7	678	(0, 1,0)	(0, 10, 0)
	0.8	560	(0, 3, 0)	(0, 10, 0)
	0.9	461	(0, 4, 0)	(0, 12, 0)
BMS-1 (*MST*=0.001)	0.3	325	(0, 7, 0)	(0, 7, 0)
	0.4	131	(0, 4, 0)	(0, 4, 0)
	0.5	34	(0, 0, 0)	(0, 0, 0)
	0.6	11	(0, 0, 0)	(0, 0, 0)
BMS-2 (*MST*=0.002)	0.3	482	(0, 9, 0)	(0, 9, 0)
	0.4	283	(0, 8, 0)	(0, 8, 0)
	0.5	112	(0, 7, 0)	(0, 8, 0)
	0.6	29	(0, 1, 0)	(0, 2, 0)

We can notice that the performance improvement is more salient on the dataset mushroom. The reason is as following. The algorithm *1.a* takes the transaction size as the basis to select candidate transactions to modify. However, it is not sufficient to make decision only based on the transaction size because most transactions hold the same length in the mushroom dataset.

An important work to be done is that the results need to be further investigated for the statistic significance and more datasets should be used for testing.

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] Ali Amiri: Dare to share: Protecting sensitive knowledge with data sanitization. Decision Support Systems 43(1): 181-191, 2007.

[3] F. Bodon. Surprising results of trie-based FIM algorithms. In: Proceedings of IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04), Brighton, UK, 2004.

[4] 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.

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