Privacy Preservation using Association Rule Mining with Limited Side Effect

Deepti Gatne
Department of Computer Engineering KKWIEER,Nasik

Prof. S. S. Sane
Department of Computer Engineering KKWIEER,Nasik

Prof. Manoj Jhade
Department of Computer Engineering KKWIEER,Nasik

ABSTRACT

The privacy preservation using association rule mining is the base of this research. The concept of privacy preserving data mining has been proposed in response to the concerns of preserving personal information from data mining algorithms. The proposed method focuses on minimizing side effects caused by privacy preservation techniques. Side effects are loss of rules and generation of the false rules. One of the techniques in privacy preservation selectively modifies individual values from a database to prevent the discovery of a set of rules. There are two known algorithms for it, ISL (Increase Support of Left) and DSR (Decrease Support of Right). Since ISL & DSR techniques aim at hiding all sensitive rules, they cannot avoid the undesired side effects. ISL algorithm results in false rules generation where DSR results in loss of rules. The propose system suggest modification to both of these algorithms in such a way that output is generated with limited side effects. Also it takes the decision about which algorithm to be used to hide a specific rule.

General Terms

Data mining: Data mining is the process of extracting patterns from data. Data mining is seen as an increasingly important tool by modern business to transform data into business intelligence giving an informational advantage.

Association rules: Association rules are statements of the form {X1,X2,….Xn }–>Y, meaning that if we find all of X1;X2; : : :;Xn in the market basket, then we have a good chance of finding Y.

Support of the rule: The support supp(*X*) of an item set *X* is defined as the proportion of transactions in the data set which contain the item set.

Confidence of the association rule: Confidence is the ratio of the number of transactions that include all items in the consequent as well as the antecedent (namely, the support) to the number of transactions that include all items in the antecedent:

Keywords

Privacy preservation, limited side effects, ISL, DSR

1. INTRODUCTION

Privacy preserving data mining is a novel research direction in data mining and statistical databases where data mining algorithms are analyzed for the side-effects which incur in process of privacy preservation. Privacy here means the logical security of data not the traditional security of data e.g. access control, theft, hacking etc. Aim is to publish data in such way that information remains practically useful but at the same time identity of an individual cannot be determined [4]

1.1 Motivation

The basic objective of this project is providing privacy to database with limited side effects. ISL & DSR are the algorithm used for providing privacy preservation using association rule mining, both of these algorithms causes side effect like false rule & lost rule. False rule means the spurious rules those are falsely generated and lost rules means non sensitive strong rules get falsely hidden. Is there any way using which it is possible to minimize the side effect caused by ISL & DSR algorithms?

1.2Existing system*s*

The common idea to modify the database for rule hiding is as follows: For a sensitive rule r: X → Y, deleting item i ϵ X U Y from transactions that contain X U Y will decrease both Sup_{XUY} and Conf_r. Moreover, inserting item i ϵ X into transactions that contain X but {i} and do not contain Y will decrease Conf_r. The first strategy, called ISL, decreases the confidence of a rule by increasing the support of the item sets in its LHS (left-hand-side). The second approach, called DSR, reduces the confidence of the rule by decreasing the support of the item sets in its RHS (right-hand-side) [2]. Both of these algorithms sequential start modifying (i.e. either deleting or inserting) all the records that contain rule X→ Y and whose confidence value is greater than MCT. This results into side effects as lost rule & false rule generation.

1.3Concept or seed idea

Concept is to minimize the side effect caused by privacy preservation process (i.e. ISL & DSR algorithms). For minimize the side effect there are two things that can be modified.

i. Rather than modifying all the records why not to modify only selected number of record so that confidence of the rule will not fall down to 0.
ii. Rather than modifying all records sequential we can modify selected records.

Now decision has to be made properly while hiding each and every rule about how many and which records to be modified so, there will be limited side effects.

2. LITERATURE SURVEY

2.1 Existing Algorithms

There are two different algorithms ISL & DSR. Here we are going to discuss DSR algorithm in detail. As we have already discussed it reduces the confidence of the rule by decreasing the support of the item sets in its RHS (right-hand-side).

2.1.1 DSR algorithm [3]

Input: source database D, MCT & MST values, set of a rule that needs to be hidden.

Output: A transformed database D, where rules containing X on Right Hand Side (RHS) will be hidden

Algorithm:

```
1. Find all possible rules from given items X;
2. Compute confidence of all the rules.
3. For each rule containing h, compute confidence of rule U
4. For each rule U in which h is in RHS
4.1. If confidence (U) < min conf, then Go to next large 2-itemset;
Else go to step 5
```

```
5. Decrease Support of RHS i.e. item h.
        5.1. Find T = t in D | t fully support U;
        5.2. While (T is not empty)
            5.2.1. Choose the first transaction t from T;
            5.2.2. Modify t by putting 0 instead of 1 for
                   RHS item;
            5.2.3. Remove and save the first transaction t
                   from T;
        End While
        5.3. Compute confidence of U;
        5.4. If T is empty, then h cannot be hidden;
 End For
```

Where ISL decreases the confidence of a rule by increasing the support of the item sets in its LHS (left-hand-side). So it considers set of records (t) which not support a rule that needs to be hidden. It will modify t by putting 1 instead of 0 for LHS item

2.1.2 ISL algorithm [1]

Input: source database D, MCT & MST values, set of a rule that needs to be hidden.

Output: A transformed database D, where rules containing X on Left hand Side (RHS) will be hidden

Algorithm:

```
1. Find all possible rules from given items X;
2. Compute confidence of all the rules.
3. For each rule containing h, compute confidence of rule U
4. For each rule U in which h is in RHS
4.1. If confidence (U) < min conf, then Go to next large 2-
itemset;
Else go to step 5
5. Increase Support of LHS i.e. item h.
        5.1. Find T = t in D | t does not support U;
        5.2. While (T is not empty)
            5.2.1. Choose the first transaction t from T;
            5.2.2. Modify t by putting 1 instead of 0 for
                   LHS item;
            5.2.3. Remove and save the first transaction t
                   from T;
        End While
        5.3. Compute confidence of U;
        5.4. If T is empty, then h cannot be hidden;
 End For
Output updated D, as the transformed D
```

2.2 Analysis of Existing System

We will try to analyze the system by considering one example. Table1 [3] shows the input database which is in the form of Bit vector. Table2 specifies all sensitive rules for input database. These are the sensitive whose confidence value is greater than MCT here it is 70%.

Table1. Original Database

TID	a	b	c	d	e
1	1	1	0	0	1
2	0	0	1	0	1
3	1	0	1	0	1
4	1	1	0	1	1
5	0	1	0	0	0
6	1	1	0	0	0

Table2. Set of sensitive rules with confidence value

Precedent	Consequent	Confidence
A	B	3/4
A	E	3/4
B	A	3/4
C	E	2/2
D	A	1/1
D	B	1/1
D	E	1/1

Now apply DSR algorithm to given database. It will start with the first rule that is a→b. Transaction 1, 4, 6 fully support this rule. So it will modify value of b from 1 to 0 for these three transactions sequentially. Accordingly it will consider all the remaining sensitive rules and go on modifying the database. Table 3 shows the output Database after processing all sensitive rules.

Table3. Output of DSR algorithm

TID	A	B	C	D	E
1	0	0	0	0	1
2	0	0	1	0	0
3	1	0	1	0	0
4	0	0	0	1	0
5	0	1	0	0	0
6	1	0	0	0	0

Table 4 shows the confidence value of all sensitive rule which is now 0.

Table4. Confidence value after applying DSR algorithm

Precedent	Consequent	Confidence
A	B	0/2
A	E	0/2
B	A	0/1
C	E	0/2
D	A	0/1
D	B	0/1
D	E	0/1

3. SYSTEM ARCHITECTURE

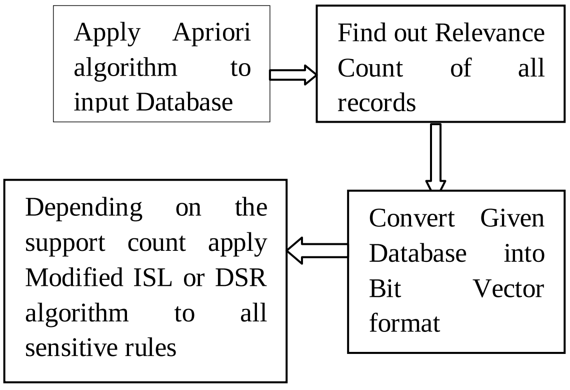
Apply Apriori algorithm to input Database
Find out Relevance Count of all records
Convert Given Database into Bit Vector format
Depending on the support count apply Modified ISL or DSR algorithm to all sensitive rules

Fig 1: System Architecture

3.1 Apriori algorithm

As is common in association rule mining, given a set of itemsets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found [5].

3.2 Relevance Count

It is the number of association rule that are fully supported by specific record. It will be calculated for all the records and count is stored with every record. Once calculation is done all records are sorted in descending order of the Relevance Count. Consider the same database given in Table I and the set of sensitive rule given in Table II then Relevance count will be as follows.

Table5. Relevance count for all transaction

TID	A	b	c	d	e	Relevance Count
1	1	1	0	0	1	2
2	0	0	1	0	1	1
3	1	0	1	0	1	3
4	1	1	0	1	1	5
5	0	1	0	0	0	0
6	1	1	0	0	0	1

3.3 Bit Vector Format

The Database is represented in the form of 0 & 1. That is if particular item exist in particular record then it is marked as 1 otherwise it is marked as 0. After converting database into Bit vector format it will be easy to perform further operation.

3.4 Applying ISL & DSR algorithm

Suppose we want to hide rule X → Y then first it will check Sup(X). If support value of X is less than 30% it will apply ISL algorithm otherwise it will apply DSR algorithm. Reason behind it is if the count of specific item is low then it would not be possible to remove it as it may result in loss of rule.

4. DETAIL DESIG

4.1 Modified DSR

```
1. Sort the given database according to Relevance count in descending order
2. Calculate dsr_count = $C_U – C_x × MCT + 1$[2]
3. Find T = t in D | t fully support U;
4. Choose the first transaction t from T;
5. While (dsr_count > 0)
       5.1 Modify t by putting 0 instead of 1 for RHS item;
       5.2 Check for loss of rule if yes then go to step 5.4
       5.3 Remove and save the transaction t from T. Change the relevance count accordingly and decrease the value of dsr_count by 1
       5.4 Consider next transaction t
   End While
6. Compute the Confidence of U;
7. If dsr_count is not equal to 0, then h cannot be hidden;
```

4.2 Modified ISL

```
1. Sort the given database according to Relevance count in ascending order
2. Calculate isl_count = $C_U / MCT - C_x + 1$[2]
3. Find T = t in D | t fully support U;
4. Choose the first transaction t from T;
5. While (isl_count > 0)
       5.1 Modify t by putting 1 instead of 0 for LHS item;
       5.2 Check whether there is any losses of rule if yes then go to step 5.4
       5.3 Remove and save the transaction t from T. Change the relevance count accordingly and decrease the value of isl_count by 1
       5.4 Consider next transaction t
   End While
6. Compute the Confidence of U;
7. If dsr_count is not equal to 0, then h cannot be hidden;
```

5. EXPECTED RESULT

We will consider the same example of Table I and will apply proposed framework on it. So it will start with the first rule a → b as value of Support a is greater than 30% we will apply Modified DSR. In Modified DSR it will first sort the record according to Relevance count in descending order as shown in Table VIII. According to formula algorithm will first calculate the **dsr_count =** $\mathbf{C_U – [C_x × MCT] + 1}$ so for rule a → b it will be **3 – [3] + 1 = 1.** It means to hide rule a → b it is necessary to modify only one transaction. It will try to modify transaction 4 as it supports rule a → b but if we put the value of b as 0 the rule d → b is getting loss. Algorithm will not modify the transaction 4. Then it will check for next transaction that is 1 and it will modify the value of b from 1 to 0.

Table6. Transaction sorted according to Relevance count

TID	A	B	c	d	E	Relevance Count
4	1	1	0	1	1	5
3	1	0	1	0	1	3
1	1	1	0	0	1	2
2	0	0	1	0	1	1
6	1	1	0	0	0	1
5	0	1	0	0	0	0

Accordingly it will try to hide all sensitive rules one by one. Final Expected output is shown in Table7.

Table7. Output after applying proposed system

TID	A	b	C	D	E
1	1	1	0	0	1
2	0	0	1	0	1
3	1	0	1	0	0
4	1	0	0	1	1
5	0	1	0	1	0
6	1	1	0	0	0

Confidences of all sensitive rules are shown in the Table8. According to result we can say that rules are hidden but at the same time they are preserved as well. Also there is no false rule generation is been observed in output.

Table8. Confidence value

Precedent	Consequent	Confidence
A	B	2/4
A	E	2/4
B	A	2/3
C	E	1/2
D	A	1/2
D	B	1/2
D	E	1/2

6. CONCLUSION

In this paper, we present a novel approach that modifies the database to hide sensitive rules with limited side effects. Propose method classify all the valid modifications such that every class of modifications is related with the sensitive rules, non-sensitive rules that can be affected after the modifications. It modifies the transactions in an order so that both the numbers of hidden sensitive rules and modified entries are considered. In most cases, all the sensitive rules are hidden without false rules generated or lost rule. In addition, it is observed that the common items and the overlapping degrees among sensitive rules have a great impact on the performance of rule hiding. Efficient mechanisms are required to speed up the rule hiding process for large databases. Another issue is the fast recognition of sensitive rules that cannot be hidden according to the user-specified constraint

7. REFERENCES

[1] Tinghuai Ma, Sainan “Privacy Preserving Based on Association Rule Mining” Wang School of Computer & Software Nanjing University of Information Science & Technology Nanjing, China thma@nuist.edu.cn

[2] Yi-Hung Wu, Chia-Ming Chiang, and Arbee L.P. Chen, “Hiding Sensitive Association Rules with Limited Side Effects” Senior Member, IEEE Computer Society

[3] Ila Chandrakar, Yelipe Usha Rani, Mortha Manasa and Kondabala Renuk “Hybrid Algorithm for Privacy Preserving Association Rule Mining” Department of Information Technology,VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India.

[4] Chris Clifton, Murat Kantarcioglou ,Xiadong Lin. “Tools for privacy preserving distributed data mining.” SIGKDD Explorations, 2003, 4(2):28-3

[5] PEI-JI WANG “MINING ASSOCIATION RULES BASED ON APRIORI ALGORITHM AND APPLICATION”