IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 3, Ver. III (May-Jun. 2016), PP 38-43
www.iosrjournals.org

Study of Hiding Sensitive Data in Data Mining Using Association Rules

Amol L. Deokate, Amruta D. Pawar, Harshal S. Sangle
(IT Department, Sanjivani K.B.P. Polytechnic College, India)

Abstract: *This paper describes Apriori algorithm for association rules for hiding sensitive data in data mining if Large data contain sensitive information that data must be protected from the unauthorized users. Here, we are going to hide this sensitive information in data mining using association rules, when we are going to apply rules for data that time it will falsely hidden information and fake rules falsely generated. So here, we examine confidentiality issues of a broad category of rules, which are called association rules. If the disclosure risk of some of these rules are above a certain privacy threshold, those rules must be characterized as sensitive information in some cases sensitive rules should not be disclose to the public since, other things, they may be used for inference of sensitive data, or they may provided these sensitive data to business competitors with an advantages.*

Keywords: *Data mining, Association rules, mining methods and algorithms, Support, Confidence.*

I. Introduction

A various types of data mining problems have been studied to help people get an insight into the huge amount of data. One of them is association rule mining, which was first introduced by Agrawal et al. Agrawal and Srikant extend and define the problem as follows: An item set is a set of products (items) and a transaction keeps a set of items bought at the same time. The support of an item set I (denoted as Sup I) in a transaction database is the percentage of transactions that contain I in the entire database. An item set is frequent if its support is not lower than a minimum support threshold (denoted as MST). For two item sets X and Y where X ∩Y = ø, The confidence of an association rule X → Y (denoted as Conf X→Y) is the probability that Y occurs given that X occurs, and is equal to Sup XỤY divided by Sup X. We say that X → Y holds in the database. if X Ụ Y is frequent and its confidence is not lower than a minimum confidence threshold (denoted as MCT). Such a rule is called the strong association rule. Association rule mining is to discover all the strong rules in the database. However, the misuse of them may bring undesired effects to people.

II. Background

Data Mining is the process of semi automatically analyzing large databases to find useful patterns. It attempts to discover rules and patterns from databases broad application of data mining are: Prediction and Association. We concentrate on association rule. Association rule mining finds interesting associations or correlations relationships among large set of data items. An example of such rule might be that 98% of the customers that purchases keyboard also tend to buy mouse at the same time.

Association rules having certain measures such as: support and confidence

1. Support –It is measure of frequency of a rule.
2. Confidence- It is a measure of strength of the relation.

Table 3.3 (a) Sample database D, (b) Large item sets from obtained from D.

TID	Items	Item set	Support
T1	ABC	**A**	66%
T2	ABC	B	66%
T3	ABC	C	66%
T4	AB	AB	66%
T5	A	BC	50%
T6	AC	AC	66%
		ABC	50%

Fig(a) Fig.(b)

Table 3.4: The rules derived from the large item sets of Table 3.3

Rules	Confidence	Support
B → A	100%	66%
B → C	75%	50%
C → A	100%	66%
C →B	75%	50%

B → AC	75%	50%
C → AB	75%	50%
AB → C	75%	50%
AC → B	75%	50%
BC → A	100%	50%

Assumptions:
For the simplicity of presentation and without loss of generality, we make the following assumptions in the development of the algorithms:

- We hide association rules by decreasing either their support or their confidence
- We select to decrease either the support or the confidence based on the side effects on the information that is not sensitive.
- We hide one rule at a time.
- We decrease either the support or the confidence one unit at a time
- We hide only rules that are supported by disjoint large item sets.

According to the first assumption we can choose to hide rule by changing either its confidence or its support. The second assumption means that, in order to decrease the confidence or the support of a rule, either we turn to 0 the value of a nonzero item in a specific transaction, or we turn to 1 all the zero items in a transaction that partially support an item set. The third assumptions states that hiding one rule must be considered as an atomic operation. This implies that the hiding of two different rules should take place in a sequential manner, by hiding one rule after the other. The fourth assumptions are based on the minimality of changes in the original database. By changing the confidence or the support of each rule, one step at a time, we act proactively in minimizing the side effects of the hiding schemes. The fifth assumption sates that we hide only rules that involve disjoint set of items. In a different situation, interactions among the rules (i.e. common subsets of items) should be considered beforehand.

Motivation

Let us suppose that we are negotiating a deal with Dedtrees Paper Company, as purchasing directors of BigMart, a large supermarket chain. They offer their products with a reduced price if we agree to give them access to our database of customer purchases. We accept the deal and Dedtrees starts mining our data. By using an association rule mining tool, they find that people who purchase skim milk also purchase Green paper. Dedtrees now runs a coupon marketing campaign saying that you can get 50 cents off skim milk with every purchase of a Dedtrees product. This campaign cuts heavily into the sales of Green paper, which increases the prices to us, based on the lower sales. During our next negotiation with Dedtrees, we find out that with reduced competition, they are unwilling to offer us a low price. Finally, we start to lose business to our competitors, who were able to negotiate a better deal with Green paper. From this aspect, releasing the database is bad for the BigMart. Therefore, for the BigMart, an effective way to release the database with sensitive rules hidden is required. This leads to the research of sensitive rule hiding.

Objectives

Given a transaction database, MST, MCT, a set of sensitive rules, and the user-specified constraint ,no lost rule, no false rule, or both, we have to modify the database such that the user specified constraint is satisfied while the sensitive rules are hidden as many as possible. To solve this problem, we propose a approach that strategically modifies the database to decrease the supports or confidences of the sensitive rules. Our approach classifies the valid modifications for hiding sensitive rules and represents each class of the modifications by three attributes. The first attribute records the modification scheme. In the case of deletion, the second attribute keeps the set of items that must be contained in the transactions to be modified. Among these items, the third attribute designates one as the item to be deleted. In the case of insertion, the second attribute uses two sets of items to describe the transactions to be modified. One is the set of items that must be contained in the transactions, while the other is the set of items that must not appear in the transactions. From the items in the second set, the third attribute specifies one as the item to be inserted. There can be two classes that are the same in the first two attributes, but different in the third attribute. As a result, for each class, a unique way of modifying the associated set of transactions for hiding some sensitive rules is determined.

III. System Analysis

Existing System:

Existing system makes a strong assumption—all the items in a sensitive rule do not appear in any other sensitive rule while inserting or deleting items to/from from transactions for hiding sensitive rules. With this assumption, hiding a sensitive rule will not affect any other sensitive rule and, therefore, hiding them one at a time or all together will not make any difference. Thus, their algorithms hide one rule at a time and decrease the supports or confidences one unit at a time. Since this work aims at hiding all sensitive rules, it cannot avoid the undesired side effects and false rules

Drawback of Existing System:

All the items in a sensitive rule do not appear in any other sensitive rule. With this assumption, hiding a sensitive rule will not affect any other sensitive rule and, therefore, hiding them one at a time or all together will not make any difference. Thus, their algorithms hide one rule at a time and decrease the supports or confidences one unit at a time. Since this work aims at hiding all sensitive rules, it cannot avoid the undesired side effects.

Proposed System

The correlation among rules can make it impossible to hide sensitive rules without violating any constraint. Therefore, we aim at avoiding the side effect in the rule hiding process instead of hiding all sensitive rules. We remove the assumption and allow the user to select sensitive rules from all strong rules such as Minimum support threshold (MST) and Minimum Confidence Threshold (MCT). Therefore, we aim at avoiding the side effects in the rule hiding process instead of hiding all sensitive rules.

IV. System Design

Initially, the original database is converted into the transaction table. Also database is mined to find the sensitive rule table and the non sensitive rule table. Then out of all sensitive rules, you hide one by one all sensitive rules. At a time only one rule is considered. Then we hide the rule that is we select the items and transactions from original database for modification.

Here we apply the modification scheme to decrease the support and confidence of the rule. When we hide the rule, association rules must be updated and the original database is modified. That modified database should be released for rule hiding.

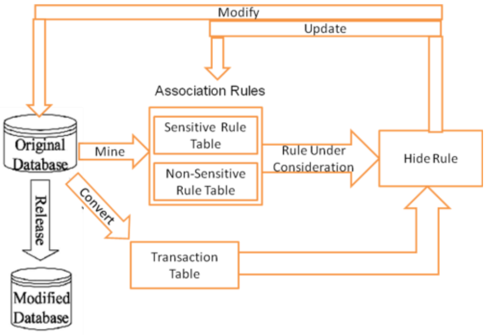


Fig. The Framework of the approach

Proposed Algorithm

An Apriori algorithm for mining frequent item sets for association rules. The key idea of algorithm is to begin by generating frequent item sets with just one item (1-item sets) and to recursively generate frequent item sets with 2 items, then frequent 3-items and so on until we have generated frequent item sets of all sizes. It is easy to generate frequent 1-item sets. All we need to do is to count, for each item, how many transactions in the database include the item. These transaction counts are the supports for the 1-item sets. We drop 1-item sets that have support below the desired cut-off value to create a list of the frequent 1-item sets. The general procedure to obtain k-item sets from (k-1)-item sets for k=2,3…is as follows: Create a candidate list of k-item sets by performing a join operation on pairs of (k-1)-item sets in the list. A pair is combined only if the first(k-2) items are the same in both item sets. If this condition is met the join of pair is a k-item set that contains the common first(k-2) items and the two items that are not in common, one from each member of the pair. All frequent k-item sets must be in this candidate list since every subset of size (k-1) of a frequent k-item set must be a frequent (k-1) item set. However, some k-item sets in the candidate list may not be frequent k-item sets. We need to delete these to create the list of frequent k-item sets. (If any of these subsets of size (k-1) is not present in the frequent (k-1) item set list, we know that the candidate k-item set cannot be a frequent item set.

Implementation steps (Algorithm/code steps) for Modules:

The implementation of system consists of implementation of Apriori Algorithm, ISL and DSR Algorithms as well as techniques to have limited side effects. Rules Generated by Apriori is displayed by the system.

Implementation of Apriori Algorithm

Input – Item sets
Output – Strong Rules
Algorithmic Steps -

The Apriori Algorithm : Pseudo Code

- Join Step: C_k is generated by joining L_{k-1} with itself
- Prune Step: Any(k-1)-item set that is not frequent cannot be a subset of a frequent k-item set
- **Pseudo-code**:

```
$C_k$: Candidate item set of size k
$L_k$: frequent item set of size k
$L_1$= {frequent items};
For (k=1;$L_k$ !=0;k++) do begin
$C_{k+1}$=candidates generated from $L_K$;
For each transaction t in database do
Increment the count of all candidates      in $C_{k+1}$
That are contained in t
            $L_{k+1}$=candidates in $C_{k+1}$ with

            min_support                  End
Return $U_K L_K$;
```

Fig (a)Algorithm for decreasing the support and confidence

INPUT: A set R_h of rules to hide, the source
Database D, the min_conf threshold, the
Min_supp threshold
OUTPUT: the database D transformed so
That the rules in R_h cannot be mined

```
Begin
For each rule U in $R_h$ do
{
Repeat until (conf(U) < min_conf)
{
1. T = {t in D / t supports U}
2. choose the transaction t in T
With the lowest number of items
3. choose the item j in rhs(U)
With the minimum impact on the
(|rhs(U)|-1)-item sets
4. delete j from t
5. decrease the support of U by 1
6. recomputed the confidence of U
}
7. remove U from $R_h$
}
End
```

Distortion Algorithm

Fig. (b)Algorithm for increasing the support and confidence

In order to increase the confidence or the support of a rule, we turn to 1 all the zero items in a transaction that partially support an item set.

This algorithm, for each selected rule, increases the support of the rule's antecedent, until the rule confidence is below the minimum threshold. The compact notation lhs(U) denotes the large item set on the left side of a rule.

INPUT: A set R_h of rules to hide, the source
Database D, the min_conf threshold, the
Min_supp threshold
OUTPUT: the database D transformed so
That the rules in R_h cannot be mined

```
Begin
For each rule U in $R_h$ do
{
Repeat until (conf(U) < min_conf)
{
1. T = { t in D / t  partially supports lhs(U) }
2. count the number of items in each transaction of T
3. sort the transactions in T in descending order of the number of supported items.
4. choose the transaction t E T with the highest number   of items ( the first transaction in T)
5. modify t to support lhs(U)
6. increase the support of lhs(U) by 1
7. recomputed the confidence of U
}
8.  remove U from $R_h$
}
End
```

Fig. (c) Algorithm for increasing support

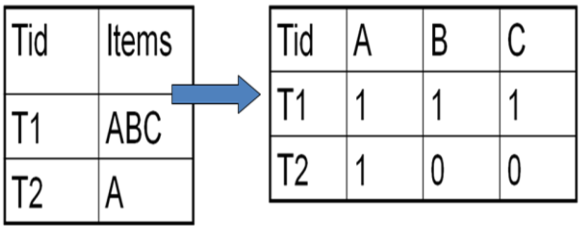
We can decrease the confidence of the rule by increasing the support of the rule antecedent X, through transactions that partially support it.

Suppose that we have the database shown in Table4. Given MST=20% and MCT=90% we are interested in hiding the rule A→C, with support =80% and confidence=100%. We select the transaction t=<T3,000,0> and turn to 1 the element of the list of item that corresponds to A. We obtain t=<T3,100,1>. Now the rule A→C has support=80% and confidence=80%, which means that the rule has been hidden since its confidence is below the minimum confidence threshold.

Strategies : Basic Idea

• Transactions viewed as lists One element for each item in DB

Table 3 Transaction Table



Tid	Items
T1	ABC
T2	A

Tid	A	B	C
T1	1	1	1
T2	1	0	0

Decreasing support of S = turning to 0 one item in one transaction supporting S

Increasing support of S = turning to 1 one item in one transaction partially supporting S

Table 4 Example hiding

A	B	C
1	1	1
1	0	1
0	0	0
1	1	1
1	0	1

MST=20%,MCT=80%
EX:Rule A→C has:
Support(A→C)=80%
Confidence(A→C)=100%

Table 5 Example hiding process1

A	B	C
1	1	1
1	0	0
0	0	0
1	1	1
1	0	1

MST=20%,MCT=80%
EX:Rule A→C Now has
Support(A→C)=60%
Confidence(A→C)=75%

Table 6 Example hiding process2

A	B	C
1	1	1
1	0	0
0	0	0
1	1	1
1	0	0

MST=20%,MCT=80%
EX:Rule A→C Now has
Support(A→C)=40%
Confidence(A→C)=50%

V. Conclusion

In this paper, we have proposed the system for hiding sensitive data in data mining Using association rule we propose to classify all the valid modifications such that every class of modifications is related with the sensitive rules, non sensitive rules, and spurious rules that can be affected after the modifications. In most cases, all the sensitive rules are hidden without false rules generated. 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. It can be interesting to discover the full set of rules that will be falsely hidden or generated as the side effects after rule hiding.

Acknowledgements

It is with the greatest pride that we publish this paper. At this moment, it would be unfair to neglect all those who helped me in the successful completion of this paper. I would also like to thank all the faculties who have cleared all the major concepts that were involved in the understanding of techniques behind my paper.

References

[1] R. Agrawal, T. Imielinski, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” Proc. ACM Conf. Management of Data, pp. 207-216, 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo, “Fast Discovery of Association Rules,”Advances in Knowledge Discovery and Data Mining, chapter 12, U.M. Fayyad et al., eds., AAAI/MIT Press, pp. 307-328, 1996.

[3] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proc. Conf. Very Large Data Bases, pp. 487- 499, 1994.

[4] R. Agrawal and R. Srikant, “Privacy-Preserving Data Mining,” Proc. ACM Conf. Management of Data, pp. 14-19, 2000.

[5] M. Atallah et al., “Disclosure Limitation of Sensitive Rules,” Proc. IEEE Workshop Knowledge and Data Eng. Exchange, pp. 45-52, 1999.

[6] C.M. Chiang, “A New Approach for Sensitive Rule Hiding by Considering Side Effects,” master thesis, Dept. of Computer Science, Nat’l Tsing Hua Univ., Republic of China, 2003.

[7] C. Clifton, “Protecting against Data Mining through Samples,” Proc. IFIP Conf. Database Security, pp. 193-207, 1999.

[8] C. Clifton and D. Marks, “Security and Privacy Implications of Data Mining,” Proc. ACM Workshop Research Issues in Data Mining and Knowledge Discovery, 1996.

[9] . D. Agrawal and C.C. Aggarwal, “On the Design and Quantifica- tion of Privacy Preserving Data Mining Algorithms,” Proc. ACM Symp. Principles of Database Systems, pp. 247-255, 2001.

[10] V. Estivill-Castro and L. Brankovic, “Data Swapping: Balancing Privacy against Precision in Mining for Logic Rules,”Proc. Conf. Data Warehousing and Knowledge Discovery, pp. 389-398, 1999.

[11] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy Preserving Mining of Association Rules,”Information Systems, vol. 29, pp. 343-364, 2004.

[12] J. Han, J. Pei, Y. Yin, and R. Mao, “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Ap-proach,” Data Mining and Knowledge Discovery,vol. 8, no. 1, pp. 53-87, 2004.

[13] V.S. Iyengar, “Transforming Data to Satisfy Privacy Constraints,” Proc. ACM Conf. Knowledge Discovery and Data Mining, pp. 279-288, 2002.