Hiding Sensitive Association Rules without Altering the Support of Sensitive Item(s)

Dhyanendra Jain[1,*], Pallavi Khatri[1], Rishi Soni[1], and Brijesh Kumar Chaurasia[2]

[1] Institute of Technology and Management-Gwalior, (M.P.), India
[2] ITM University, Gwalior, (M.P.), India
{dhyanendra.jain,rishisoni17,bkchaurasia_itm}@gmail.com,
pallavi_magic@yahoo.com

Abstract. Association rule mining is an important data-mining technique that finds interesting association among a large set of data items. Since it may disclose patterns and various kinds of sensitive knowledge that are difficult to find otherwise, it may pose a threat to the privacy of discovered confidential information. Such information is to be protected against unauthorized access. Many strategies had been proposed to hide the information. Some use distributed databases over several sites, data perturbation, clustering, and data distortion techniques. Hiding sensitive rules problem, and still not sufficiently investigated, is the requirement to balance the confidentiality of the disclosed data with the legitimate needs of the user. The proposed approach uses the data distortion technique where the position of the sensitive items is altered but its support is never changed. The size of the database remains the same. It uses the idea of representative rules to prune the rules first and then hides the sensitive rules. Advantage of this approach is that it hides maximum number of rules however, the existing approaches fail to hide all the desired rules, which are supposed to be hidden in minimum number of passes. The paper also compares of the proposed approach with existing ones.

Keywords: Privacy preserving data mining, Association rule, Association rule hiding.

1 Introduction

Data mining is the knowledge discovery process of finding the useful information and patterns out of large database. In recent times data mining has gained immense importance as it paves way for the management to obtain hidden information and use them in decision-making. While dealing with sensitive information it becomes very important to protect data against unauthorized access [1], [2]. A key problem faced is the need to balance the confidentiality of the disclosed data with the legitimate needs of the data users. In doing this it becomes necessary to modify the data value(s) and relationships (Association Rules). Obtaining a true balance between the disclosure and hiding is a tricky issue [2], [4]. This can be achieved largely by implementing hiding of rules that

[*] Corresponding author.

N. Meghanathan et al. (Eds.): CCSIT 2012, Part I, LNICST 84, pp. 500–509, 2012.

expose the sensitive part of the data. One such method is hiding of association rule because association amongst the data is what is understood by most of the data users.

Such vulnerability of association rule posses' a great threat to the data if the data is in hands of a malicious user.

To prevent data from being misused two common strategies exist. First strategy alters the date before delivering it to the data miner [3]. Second strategy releases only a subset of the complete data using distributed databases approach.

Algorithms have been proposed in the literature for hiding of rules. The proposed algorithms are based on modifying the database transactions so that the confidence of the rules can be reduced. Hiding association rule by using support and confidence is discussed in [3], basically this approach algorithm hides a specific rule while association rule mining [4] algorithm hide rules with respect to sensitive item(s) either on the left or on the right of the rule. However, these approaches fail to hide all the desired rules, which are supposed to be hidden in minimum number of passes.

In this paper, we propose strategies and a suit of algorithms for privacy preserving and hiding knowledge from data by minimal perturbing values. The proposed approach uses the data distortion technique where the position of the sensitive item(s) is altered but its support is never changed however the size of the database remains the same. The proposed heuristics use the idea of representative rules to prune the rules first and then hides the sensitive rules. This approach results in a significant reduction of the number of rules generated, while maintaining the minimum set of relevant association rules and retaining the ability to generate the entire set of association rules with respect to the given constraints [1], [2]. Advantages of the proposed approach is that the support of the sensitive item(s) is neither increased nor decreased as done in existing approaches and the size of the database is kept same while the previous approaches either increase or decrease the size of the database. Support of the sensitive item(s) is kept same and simply its position have been changed i.e. it is being deleted from one transaction and added to some other transaction in which it does not exist. Another advantage of this approach is that it hides maximum number of rules in minimum number of alterations in the database. An algorithm is also proposed for this work and demonstrated by example. The proposed approach is compared with previously existing approaches [5].

The rest of the paper is organized as follows. Section 2 describes the association rule mining. In section 3, the association rule is described. Problem definition is given in section 4. Proposed scheme is presented in section 5. The scheme is evaluated through simulation and results are in section 6; section 7 concludes the work.

2 Association Rule Mining

Let $I = \{i_1, i_2, ...i_m\}$ be a set of *m* distinct literals, called items. Given a set of transactions *D*, where each transaction *T* is a set of items such that $T \subseteq i$. An association rule is an implication of the form $X -> Y$ where $X \subset I, Y \subset I$ and

$X \cap Y = \phi$ X and Y are called antecedent/body and consequent /head of the rule respectively [6].

Strength of a rule whether it is strong or not is measured by two parameters called support and confidence of the rule. These two parameters help in deciding the interestingness of a rule [5], [7].

For a given rule $X \Rightarrow Y$

Support is the percentage of transaction that contains both ***X*** and ***Y*** or $X \cup Y$ is the proportion of transactions jointly covered by the LHS and RHS and is calculated as:

$$S = |X \cup Y| / |N|$$

Where, ***N*** is the number of transactions.

Confidence is the percentage for a transaction that contains ***X*** also contains ***Y*** or is the proportion of transactions covered by the LHS that are also covered by the RHS and is calculated as

$$C = |X \cup Y| / |X|$$

For the database given in Table1, with a minimum support of 33% and minimum confidence 70% following nine association rules could be found:

$$C \Rightarrow A(66.667\%,100\%), A, B \Rightarrow C(50\%,75\%),$$

$$B \Rightarrow C, A(50\%,75\%), C, B \Rightarrow A(50\%,100\%),$$

$$\mathrm{C => A,B\ (50\%,75\%),C,A => B\ (50\%,75\%),}$$

$$\mathrm{B => C\ \ (50\%,75\%),C => B\ (50\%,75\%),}$$

$$\mathrm{B => A\ (66.667\%,100\%)}$$

Table 1. Set of transactional data

TID	ITEMS
T1	ABC
T2	ABC
T3	ABC
T4	AB
T5	A
T6	AC

3 Representative Association Rule

Generally, number of association rules discovered in a given database is very large. It is observed that a considerable percentage of these rules are redundant and useless. A user should be presented with all of them, which are original, novel and interesting. To address this issue, [6] introduced a notion for concise (loss less) representation of association rules, called representative rules (RR). RR is a least set of rules that allow deducing all association rules without accessing a database. In a notion of cover operator was introduced for driving a set of association rules from a given association rule. The cover ***C*** of the rule $\mathrm{X} => \mathrm{Y}$, $Y \neq \phi$ is defined as follows:

$$\mathrm{C}(\mathrm{X} => \mathrm{Y}) = \{X \cup Y => \mathrm{V/Z}, \mathrm{V} \subseteq \mathrm{Y} \;\; \boldsymbol{and}\, \mathrm{Z} \cap \mathrm{V} = \phi \; \boldsymbol{and} \; V \neq \phi\}$$

Each rule in $\mathrm{C}(\mathrm{X} => \mathrm{Y})$ consists of a subset of items occurring in the rule $\mathrm{X} => \mathrm{Y}$. The number of different rules in the cover of the association $\mathrm{X} => \mathrm{Y}$ is equal to $3^{\mathrm{m}} - 2^{\mathrm{m}}, m = |Y|$.

In general, the process of generating representative rules may be decomposed in to two sub-processes: frequent item-sets generations and generation of RR from frequent item-sets. Let ***Z*** be a frequent itemset and $\phi \neq X \subset Z$. The association rule $\mathrm{X} => \mathrm{Z/X}$ is representative rule if there is no association rule $(\mathrm{X} => \mathrm{Z}'/\mathrm{X})'$ where $\mathrm{Z} \subset \mathrm{Z}'$, and there is no association rule $(\mathrm{X}' => \mathrm{Z/X}')$ such that $\mathrm{X} \supset X'$. Formally, a set of representative rules (RR) for a given association rules (AR) can be defined as follows:

$$RR = \{r \in AR / \neg\exists r' \in AR, r \neq r' \,\mathrm{and}\; \mathrm{r} \in \mathrm{C}(\mathrm{r}')\}$$

Each rule in RR is called representative association rule and no representative rule may belong in the cover of another association rule [8], [9].

4 Problem Definition

The expression ‘Data Mining’ indicates a wide range of tools and techniques to extract useful information, which can be sensitive (interesting rules) from a large collection of data. Objective of this work is to propose a new strategy to avoid extraction of sensitive data. Data should be manipulated /distorted in such a way that sensitive information cannot be discovered through data mining techniques. While dealing with sensitive information it becomes very important to protect data against unauthorized access. The key problem faced is the need to balance the confidentiality of the disclosed data with the legitimate needs of the data users. The proposed algorithm are based on modifying the data base transaction, so that, the confidence of the rules can be reduced for this both the approaches either pros or cons the support of the item.

Following section proposes a algorithm for hiding the sensitive rules (sensitive rules are those rules that contain sensitive item(s)).

5 Proposed Algorithm

In order to hide an association rule a new concept of ‘*not altering the support*’ of the sensitive item(s) has been proposed in this work. Based on this strategy an algorithm has been proposed .in this assumed that a set of sensitive item(s) is passed and the proposed algorithm distorts the original database such that sensitive rules cannot be discovered through Association Rule Mining algorithms.

Input to proposed algorithm is a database, min_supp, min_conf, and a sensitive item(s) ***H***(each sensitive item in ***H*** is represented by *h*) to be hidden and the goal is to distort the database ***D*** such that no association rules containing $h \in H$ either on the left or on the right can be discovered.

ALGORITHM:

```
Input
      (1) A source database D
      (2) A min_support.
      (3) A min_confidence.
      (4) A set of sensitive items H.

Output
   A transformed database D′ where rules containing H
 on RHS/LHS will be hidden
 1.  Find all large itemsets from D;
 2.  For each sensitive item $h \in H$  {
 3.  If h is not a large itemset then H=H-{$h$};
 4.  If H is empty then EXIT;
 5.  Select all the rules containing h and store in U
                         //$h$ can either be on LHS or RHS
 6.  Select all the rules  from U with $h$ alone on LHS
 7.  Join RHS of selected rules and store in R;
                             //make representative rules
 8.  Sort R in descending order by the number of supported items;
 9.  Select a rule $r$ from R
 10. Compute confidence of rule $r$.
 11. If conf>min_conf then   {
               //change the position of sensitive item h.
 12. Find $T_1$={t in D|t completely supports $r$ ;
 13. If t contains x and h then
 14. Delete h from t
 15. Else
 16. Go to step 19
 17. Find $T_1$ ={t in D|t does not support LHS($r$) and
     partially supports x;
 18.     Add h to t
 19.     Repeat
 20.     {
 21.             Choose the first rule from R;
 22.             Compute confidence of $r$ ;
 23.     } Until(R is empty);
 24.      }                  //end of if conf>min_conf
 25.      Update D with new transaction t;
 26.      Remove h from H;
 27.      Store U [i] in R;   //if LHS (U) is not same
 28.      i++, j++;
 29.      Go to step 7;
 30.      }//end of for each h∈H
```

Output updated as the transformed database Proposed algorithm selects all the rules containing sensitive item(s) either in the left or right. Then these rules are represent in representative rules (RR) format. After this a rule from the set of RR's, which has sensitive item on the left of the RR is selected.

Now delete the sensitive item(s) from the transaction that completely supports the RR i.e. it contained all the items in RR selected and add the same sensitive item to a transaction which partially supports RR i.e. where items in RR are absent or only one of them is present.

For example in Table1 at a *min_supp of 33% and a min_conf* of 70 % and sensitive item ***H***={***C***}, choose all the rules containing '***C***' either in RHS or LHS

$$C \Rightarrow A(66.667\%,100\%), A,B \Rightarrow C(50\%,75\%),$$

$$B \Rightarrow C,A(50\%,75\%), C,B \Rightarrow A(50\%,100\%),$$

$$\mathrm{C} => \mathrm{A,B}\ (50\%,75\%), \mathrm{C,A} => \mathrm{B}\ (50\%,75\%),$$

$$\mathrm{B} => \mathrm{C}\ \ (50\%,75\%), \mathrm{C} => \mathrm{B}\ (50\%,75\%),$$

and represent them in representative rule format.

Like $C \longrightarrow A$ and $C \longrightarrow B$ can be represented as $C \longrightarrow AB$

Now delete ***C*** from a transaction where ***A, B*** and ***C*** are present and add ***C*** to a transaction where both of them (***A*** and ***B***) are either absent or only one of them is present. If transaction $\boldsymbol{T_1}$ is modified to ***AB*** and transaction $\boldsymbol{T_5}$ is modified to ***AC*** then the rules that will be hidden are:

$$C \longrightarrow B,\ C \to A,\ C \to AB,\ B \to C,\ AB \to C,\ B \to AC, \text{and } AC \longrightarrow B$$

i.e. seven rules out of eight rules containing sensitive item(s) are hidden.

6 Results and Analysis

We performed our experiments on a Intel PIV workstation with 1.8 GHz processorand with 512 MB RAM, under Microsoft Windows-xp operating system using α-minor tool. To demonstrate the working of the above algorithm for hiding sensitive items the following section shows some results.

6.1 For a given database in Table1 with a minimum support of 33% and minimum confidence of 70% we hide 7 rules out of 8 rules containing sensitive item(s), if transaction T_1 is changed to ***AB*** and transaction T_5 is changed to ***AC***. This has been discussed in above section of the paper.

6.2 For database in Table1. if ***H***={***B***} i.e. if sensitive item is ***B*** and the rule which is to be hidden is $B \Rightarrow C$ $(B \Rightarrow A$ *and* $B \Rightarrow C$ represented as $B \Rightarrow AC)$ then change transaction T_1 to ***AC*** and transaction T_5 to ***AB***.

6.3 For a database given in Table2 with a minimum support of 33% and minimum confidence of 70% following association rules are mined:

$$A \Rightarrow B,\ A \Rightarrow C,\ B \Rightarrow C,\ B \Rightarrow A,\ C \Rightarrow A,\ C \Rightarrow B,$$

$$AD \Rightarrow C,\ CD \Rightarrow A$$

Select the rules containing sensitive items either in the LHS or in the RHS

$$A \Rightarrow C,\ B \Rightarrow C,\ B \Rightarrow A,\ C \Rightarrow A,\ C \Rightarrow B,\ AD \Rightarrow C,\ CD \Rightarrow A$$

Representation of rules in representative rule format is:

$$C \Rightarrow A, \textit{and}\, C \Rightarrow B \textit{ can be represented as}\, C \Rightarrow AB$$

Delete ***C*** from a transaction in which ***A*** and ***B*** are present and add it in a transaction where both ***A*** and ***B*** are absent or only one of them is present

This results in modification of the database by changing the transaction T_2 to ***ABD*** and transaction T_5 to ***CDE***.

Out of 6 rules containing sensitive items all of them are hidden.

6.4 Similarly for database in Table 2. if ***H***={***B***} *i.e.* and the rule which is to be hidden is $B \Rightarrow C\,(B \Rightarrow A\, \textit{and}\, B \Rightarrow C$ represented as $B \Rightarrow AC)$ if sensitive item is ***B*** then change transaction T_2 to ***ACD*** and transaction T_5 to ***BDE***.

Out of 4 rules containing sensitive items all of them are hidden.

6.5 Analysis: This section analyzes some of the characteristics of the proposed algorithm and it is compared with the existing algorithms.

First characteristic of proposed algorithm as described in previous section of the paper is that support of the sensitive item is not changed. The position of the sensitive item is changed only. This characteristic is demonstrated with examples in previous section and is summarized in Table 3 and 4.

The second characteristic we analyze is the efficiency of the proposed algorithm with previous approaches.

For proposed algorithm, the number of DB scans required for Table1 are 4 and number of rules pruned are 7.

Table 2. Set of transactional data

TID	ITEMS
T1	ABC
T2	ABCD
T3	BCE
T4	ACDE
T5	DE
T6	AB

Table 3. Database before and after hiding *C* and *D*

TID	***D***	D_1(***C*** sensitive)	D_2(***B*** sensitive)
T1	ABC	AB	AC
T2	ABC	ABC	ABC
T3	ABC	ABC	ABC
T4	AB	AB	AB
T5	A	AC	AB
T6	AC	AC	AC

Table 4. Database before and after hiding *C* and *B*

TID	***D***	D_1(***C*** sensitive)	D_2(***B*** sensitive)
T_1	ABC	ABC	ABC
T_2	ABC D	ABD	ACD
T_3	BCE	BCE	BCE
T_4	ACDE	ACDE	ACDE
T_5	DE	CDE	BDE
T_6	AB	AB	AB

In [3] algorithm DB scans are 4 and number of rules pruned are 0. For Wang's approach DB scans are 3 and number of rules pruned are 2. This characteristic is summarized in Table5 and the same characteristics for the database of Table2 is summarized in Table6 and it is clear from both the tables that the proposed algorithm prunes more number of rules in the same number of DB scans[3].

One of the reasons that the existing approaches fail is that the approach in tries to hide every single rule from a given set of rules without checking if some of the rules could be pruned after changing some transactions of all.

Approach in hides only those rules, which has sensitive item either in the right or in the left. It runs two different algorithms depending on the position of the sensitive item(s) (whether it is antecedent or consequent). This approach also fails to hide more number of rules.

Table 5. Database scans and rules pruned in hiding item *C* using proposed algorithm

	DB scans	Rules Pruned	
		Table1	Table2
Proposed algorithm	4	7	6
ISLF	3	2	3
[3] Dasseni *et al.*	4	0	1

Table 6. Database scans and rules pruned in hiding item *B* using proposed algorithm

	DB scans	Rules Pruned	
		Table1	Table2
Proposed algorithm	4	6	4
ISLF	3	2	2
[3] Dasseni *et al.*	4	1	1

However, proposed approach hides almost all the rules, which contain sensitive item(s) (either on the left or on the right[11], [12],[13], [14] and [15].

7 Conclusion

In this paper, we presented the threats to database privacy and security challenges due to rapid growth of data mining. An algorithm for this is also been proposed which is based on modifying the database transactions so that the confidence of the sensitive rules can be reduced but without altering the support of the sensitive item, which is in contrast with previous algorithms, which either decrease or increase the support of the sensitive item to modify the database transactions. The efficiency of the proposed approach is compared with the existing approaches. It is observed that number of DB scans required by proposed approach is same as in approach in and one more than the approach used in but it prunes almost all the desired rules which previous approaches fail to do.

References

1. Verykios, V.S., Elmagermld, A.K., Bertino, E., Saygin, Y., Dasseni, E.: Association Rule Hiding. IEEE Transactions on Knowledge and Data Engineering 6(4) (2004)
2. Wang, S.-L., Lee, Y.-H., Billis, S., Jafari, A.: Hiding sensitive items in privacy preserving association rule mining. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3239–3244 (2004)
3. Dasseni, E., Verykios, V.S., Elmagarmid, A.K., Hwang, J.: Hiding association rules by using confidence and support. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 369–383. Springer, Heidelberg (2001)
4. Verykios, V.S., Elmagarmid, A.K., Elisa, B., Elena, D., Saygin, Y.: Association Rule Hiding. IEEE Transactions on Knowledge and Data Engineering, 434–447 (2000)
5. Wang, S.-L., Jafari, A.: Using unknowns for hiding sensitive predictive association rules. In: IEEE International Conference on Information Reuse and Integration, pp. 223–228 (2005)
6. Agrawal, A., Thakar, U., Soni, R., Chaurasia, B.K.: Efficiency Enhanced Association Rule Mining Technique. In: Nagamalai, D., Renault, E., Dhanushkodi, M. (eds.) PDCTA 2011. CCIS, vol. 203, pp. 375–384. Springer, Heidelberg (2011)
7. Saygin, Y., Verykios, V.S., Clifton, C.: Using unknowns to prevent discovery of association rules. ACM SIGMOD Record 30(4), 45–54 (2001)
8. Huang, Y., Lu, Z., Hu, H.: A method of security improvement for privacy preserving association rule mining over vertically partitioned data. In: 9th International Database Engineering and Application Symposium, pp. 339–343 (2005)
9. Saygin, Y., Verykios, V.S., Elmagarmid, A.K.: Privacy preserving association rule mining. In: IEEE Proceedings of the 12th Int'l Workshop on Research Issues in Data Engineering, pp. 151–158 (2002)
10. Evfimievski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. In: Proc. of the 8th ACM SIGKDD Int'l. Conference on Knowledge Discovery and Data Mining, Edmonton, Canada (2002)
11. Agrawal, R., Srikant, R.: Privacy preserving data mining. In: ACM SIGMOD Conference on Management of Data, Dallas, Texas, pp. 439–450 (2000)
12. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of ITEMS in Large Databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Washington, D.C., pp. 207–216 (May 1993)
13. Sun, D., Teng, S., Zhang, W., Zhu, H.: An Algorithm to Improve the Effectiveness of Apriori. In: The 6th IEEE Int. Conf. on Cognitive Informatics (ICCI 2007), pp. 385–390 (2007)
14. Ping, D., Yongping, G.: A New Improvement of Apriori Algorithm for Mining Association Rules. In: The International Conference on Computer Application and System Modeling (ICCASM 2010), pp. 529–532 (2010)
15. Lu-Feng, W.: Association Rule Mining Algorithm Study and Improvement. In: The 2nd International Conference on Software Technology and Engineering (ICSTE), pp. 362–364 (2010)