A Hybrid Approach to Privacy Preserving in Association Rules Mining

Narges Jamshidian Ghalehsefidi[1], Mohammad Naderi Dehkordi[2]

[1,2]Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

[1]jamshidian_n@sco.iaun.ac.ir,[2]naderi@iaun.ac.ir

Abstract

Nowadays, data mining is a useful, yet dangerous technology through which useful information and the relationships between items in a database are detected. Today, companies and users need to share information with others for their progress and they should somehow manage this information sharing for preserving sensitive information. Privacy preserving in data mining was introduced for managing information sharing. This paper presents a hybrid algorithm with distortion technique with both support-based and confidence-based approaches for privacy preserving. The proposed algorithm tries to maintain useful association rules and hide sensitive rules from the perspective of the database owner. It also has no limit on the number of items in the left-hand side and the right-hand side of rules. This paper also compares the proposed algorithm with MDSRRC algorithm and 1.b algorithm. The proposed algorithm has less lost rules compared with the MDSRRC and 1.b algorithms and its CPU usage is less then.

Keywords: *privacy preserving, hiding sensitive rules, helpful association rules.*

1. Introduction

The data mining technique aims to detect useful rules and relationships of database items for which standard algorithms are used such as Apriori and Eclat. The detected rules are divided into two groups: sensitive rules and non-sensitive rules. Sensitive rules are those rules the database owner is trying to hide using privacy preserving algorithms in data mining, and non-sensitive rules are useful rules the database owner wants to share with others. Of course, doing anything has its own costs. The cost the database owner pays for hiding sensitive rules is the loss of non-sensitive rules plus other costs we called them side effects due to hiding sensitive rules. These side effects include loss of non-sensitive rules, creating ghost rules, hiding failure, dissimilarity, runtime, etc. The algorithms presented in the context of privacy preserving have tried to reduce these side effects. The proposed algorithm aims to reduce lost rules and reduce hiding failure to zero. Generally there are two approaches for hiding sensitive rules [1]: support-based approach and confidence-based approach. The support-based approach aims to reduce support of the sensitive rule by reducing support of one of element sets composing the sensitive rule. The confidence-based approach aims is reducing confidence of the sensitive rule through increasing support of the consequent of the sensitive rule. This paper uses both approaches for hiding sensitive rules. The proposed algorithm selects either approach to sanitize by calculating the support of left-hand side and right-hand side elements of the sensitive rule. In this algorithm, selecting the item and the transaction to sanitize has a large impact on reducing side effects. This paper compares the proposed algorithm with MDSRRC algorithm on the Chess dataset.

The article is organized as follows. Section 2 describes the framework of association rule. Section 3 reviews the related works. Section 4 explains the proposed algorithm, terms and steps. Section 5 compares and evaluates the proposed algorithm with MDSRRC and 1.b. Finally, conclusions are presented in Section 6.

2. Framework of association rules

Rule extraction in data mining is performed by the level of support and confidence of the rule. The issue of extracting association rule was introduced by [2]. Suppose $I=\{i_1,i_2,\ldots,i_m\}$is a set of elements and the database $D=\{T_1,\ldots,T_n\}$is a set of transactions. Each transaction $T\in D$ contains a subset of I. The general framework of association rules is X⟶Y. If X and Y are subsets of I and if $x\cap y=\emptyset$, then X is called the antecedent or LHS of the rule and Y is called the consequent or RHS.

The support of the rule X⟶Y is defined by calculating the ratio of simultaneous repetition frequencies of X and Y in transactions to the total number of transactions in the database. The support of the rule is calculated by Eq. (1).

$$\text{support}(X\to Y)=\frac{|X\cup Y|}{|D|} \qquad (1)$$

The confidence of the rule X⟶Y is defined by calculating the ratio of simultaneous repetition frequencies of X and Y in transactions to the number of repetition of X alone in transactions of the database. The confidence of the rule is calculated by Eq. (2).

$$\text{confidence } (X\rightarrow Y)=\frac{|X \cup Y|}{|X|} \qquad (2)$$

Minimum Support Threshold (MST) and Minimum Confidence Threshold (MCT) criteria are used to extract useful rules from the database. If Support (x⟶y) >= MST and Confidence (x⟶y) >= MCT, the x⟶y rule becomes important and is extracted from the database at the time of data mining.

3. Related Works

Wang et al. presented two algorithms for hiding association rules. The first algorithm, ISL, reduces the rule confidence by increasing support of the left-hand side element set of the sensitive rule. This algorithm has high hiding failure and new-rule creation. The second algorithm, DSR, reduces confidence by reducing support of the right-hand side element set of the rule. The failure of this algorithm is close to zero but many non-sensitive rules are lost [3] [4].
Modi et al. introduced an algorithm called DSRRC which uses clustering of right-hand side common items for hiding. Its disadvantages are as follows: it only hides rules with an element at their right-hand side, it is dependent on the ordering of transactions, gives a different result with reordering transactions in the database, needs sorting of the database after each item is deleted and is not suitable for large databases. Lost rules in this algorithm are high [5].
Komal Shah et al. presented an algorithm called ADSRRC to modify DSRRC. This algorithm also hides the rules with a single RHS, and it sorts transactions only once. In addition, this paper proposed a new algorithm called RRLR which hides rules with a single LHS. It reduces both support and confidence for hiding sensitive rules [6].
To overcome limitations in the left-hand side and right-hand side items, Domadiya et al. proposed an algorithm called MDSRRC. It selects the best item for deletion based on the number of its repetition in the right-hand side of the sensitive rule and its support. This algorithm has less side effects compared to DSRRC. It fails in certain circumstances [7].
Kumar Jain et al. combined the two algorithms ISL and DSR and stated the main goal as reducing the number of database changes and reducing the time for hiding sensitive rules [8].
Vijayarani et al. presented a heuristic algorithm called ABC. In this algorithm, transaction selection is performed randomly according to the behavior of honey bees for finding food. This algorithm uses the support-based approach [9].
Oliveria et al. presented two algorithms called Round Robin and Random. The essence of both algorithms is item selection to sanitize which is done randomly and intermittently [10].
Duraiswamy et al. proposed an algorithm called SRH. It reduces complexity, time and memory by calculating the number of transactions required for hiding sensitive rules [11].
Menon et al. proposed an algorithm with exact approach called Integer programming and also two strategies: Blanket and Intelligent. This algorithm has the best level of accuracy [12].
Verykios et al. proposed two algorithms called WSDA and BA. WSDA hides sensitive rules using the distortion techniques, and BA does the same using the blocking technique [13].
Amiri proposed three algorithms called Aggregate, Disaggregate and Hybrid, which hide sensitive rules using the support-based approach [14].
Dasseni et al. generalized the hiding problem to a combination of sensitive rules hiding and sensitive itemset hiding. Algorithm 1.b selects the best subset from the itemset on right side of the sensitive rule and for hiding the sensitive rule selects the first item as the victim item and eliminates the latter from those transactions fully supporting the sensitive rule. The advantages of this algorithm are in reducing hiding failure and relatively proper CPU usage [15].

4. The proposed algorithm

The proposed approach uses the distortion technique for hiding association rules with both confidence-based and support-based approaches. This algorithm has two main goals:

1. Reduction of non-sensitive lost rules due to hiding sensitive rules.
2. Reduced CPU usage.

We first introduce the terminology used in the proposed algorithm and then describe its steps.

Sensitive item and item sensitivity: Items involved in sensitive rules are called sensitive items, and the number of their repetition in sensitive rules is called sensitive items.

Degree of transaction collision: The number of sensitive rules in the transaction. The transaction in fact contains all items involved in the sensitive rule.

4.1 The proposed algorithm steps

The essence of the proposed algorithm is the selection of sanitizing operation according to the amount of LHS and RHS support of the sensitive rule. Then the sensitive item and the suitable transaction are selected for the sanitizing operation. The sensitive item is selected considering the amount of support and sensitivity of that item which leads to selecting the suitable item for the sanitizing operation.

Before the sanitizing operation, the proposed algorithm first obtains the number of transactions required for hiding sensitive rules using the equation presented in Section [3]. Eq. (3) shows how to calculate mincount.

$$
\begin{aligned}
&\text{MSC } x \longrightarrow y = \text{count }(xUy) - [\,|D| * \text{min support}] + 1.\\
&\text{MCCC } x \longrightarrow y = \text{count }(xUy) - [\text{count}(x) * \text{min conf}] + 1.\\
&\text{MPCC } x \longrightarrow y = [(\text{count }(xUy) - \text{count}(x) * \text{min conf}) / (1 - \text{min conf})] + 1. \qquad (3)\\
&\text{MCC} = \text{minimum (MCCC, MPCC).}\\
&\text{mincount} = \text{minimum (MSC, MCC).}
\end{aligned}
$$

PSEUDO CODE FOR Proposed Algorithm

Input:Source Database D,MCT,MST,Sensitive rule(R_H)
Output:The Santized database D

```
1. Find sensitivity of each item $\in R_H$ set $I_S$
2. Find support of each item$\in I_S$ in D
3. Find conflict T$\in$D set $T_S$
4. Sort $R_H$ by decreasing order of their support
5. Hiding
6. While(all the sensitive rule hidden$\neq$true){
   a. Foreach r in $R_h$ do{
      i. If support $r_{LHS}$>=Support $r_{RHS}$ {
         1. Sort $I_S$ by sensitive item decreasing , support increasing
         2. Select Victim item where RHS contains it
         3. Mincount =mincount(r)
         4. Sort $T_S$ by conflict decreasing , length increasing
         5. Foreach t in TS do{
            a. If(itemset xyz $\in$t){
               i. Remove itemselected
               ii. If Mincount=0 then break
            b. }
         6. }
      ii. Else{
         1. Sort $I_S$ by support increasing, sensitive item decreasing
         2. Select sensitive item where LHS contains it
         3. Mincount =mincount(r)
         4. Sort $T_S$ by conflict decreasing , length increasing
         5. Foreach t in $T_S$ do{
            a. If(itemset xyz $\in$t){
               i. Remove itemselected
               ii. If Mincount=0 then break
            b. }
         6. }
      iii. }
   b. Start Update support& confidence
   c. }
7. }
```

The first step is to obtain the sensitivity of items in sensitive rules and put them in the I_S set. We then find the level of support for each item in the D database. The third step is to find the degree of transaction collision in the D database and put them in the T_S set. Then we sort sensitive rules in descending order according to their support level. In Step 6, the sanitize operation is performed. Then we select the sensitive rule and calculate its LHS and RHS support, if it is not hidden. If the RHS support is smaller than or equal to the LHS support, the sanitize operation is carried out with the support-based approach. In this stage, the sensitive items of the I_S set are sorted in descending order according to their sensitivity and are sorted in ascending order according to their support. The first sensitive item presented in RHS is selected and the item deletion operation is invoked. If the LHS support is smaller than the RHS support, the sanitizing operation is performed with the confidence-based approach. In this stage, the sensitive items of the I_S set are sorted in ascending order according to their support and are sorted in descending order according to their sensitivity. Then first sensitive item that is presented in LHS is selected and the item deletion and insertion operations are invoked.

5. Comparison and evaluation

We implemented the proposed algorithm with the known MDSRRC algorithm and 1.b algorithm on the Sony F115FM system with CPU Core i7, memory 6GB, HDD 500GB, Windows 7 OS with the C# programming language. For comparison and evaluation purposes, we performed various tests with different rules on the Chess dataset. Figure 1 shows the graph of non-sensitive lost rules and figure 2 shows the graph of CPU usage in the tests conducted on the Chess dataset.

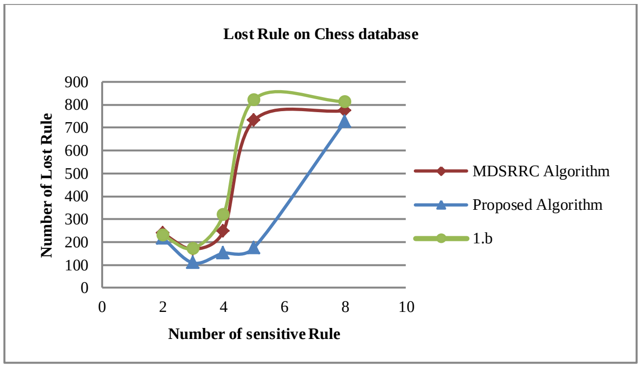

Fig. 1 Lost rule on the Chess dataset

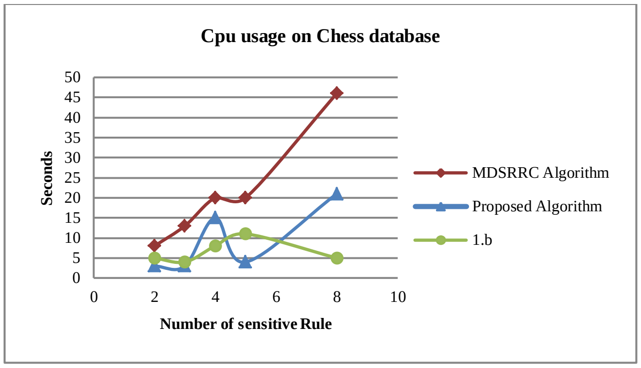

Fig. 2 CPU usage on the Chess dataset

6. Conclusions

This paper proposed a hybrid algorithm with two support-based and confidence-based approaches. The proposed algorithm offers better results in actual compressed databases compared with dummy or actual non-compressed databases. To solve this problem, we can simply change the sorting of sensitive items so that it can also provide good results in non-compressed databases. In the MDSRRC algorithm, there is a possibility of failure in certain circumstances due to uncontrolled hidden rules. We address this problem by controlling sensitive rules even after being hidden. The hiding failure in the proposed algorithm is zero. The proposed algorithm maintains more non-sensitive rules than MDSRRC and 1.b algorithm.

References

[1] V.S. Verykios, E. Bertino, I.N. Fovino, “State-of-the-art in privacy preserving data mining”, SIGMOD Record, Vol. 33, No. 1, March 2004, pp. 50–57.

[2] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, V. Verykios, “Disclosure limitation of sensitive rules”, Knowledge and Data Engineering Exchange, November 1999, pp. 45-52.

[3] S.L. Wang, B. Parikh, A. Jafari, “Hiding informative association rule sets”, Expert Systems with Applications, Vol.33, No.2, June 2007, pp. 316–323.

[4] S.L. Wang and A. Jafari, “Hiding sensitive predictive association rules”, Systems, Man and Cybernetics, October 2005, Vol. 1, pp. 164–169.

[5] C.N. Modi, U.P. Rao, D.R. Patel, “Maintaining privacy and data quality in privacy preserving association rule mining”, Computing Communication and Networking Technologies, July 2010, pp. 1–6.

[6] K. Shah, A. Thakkar, A. Ganatra, “Association rule hiding by heuristic approach to reduce side effects & hide multiple r.h.s. items”, International Journal of Computer Applications, Vol. 45, No. 1, May 2012, pp. 1–7.

[7] N.H. Domadiya and U.P. Rao. “Hiding sensitive association rules to maintain privacy and data quality in database”, Advance Computing Conference, February 2012, pp. 1306–1310.

[8] Y.K. Jain, V.K. Yadav, G.S. Panday, “An efficient association rule hiding algorithm for privacy preserving data mining”, International Journal on Computer Science and Engineering, Vol. 3, No. 7, July 2011, pp. 2792–2798.

[9] S.Vijayarani and M.S. Prabha, “Association rule hiding using artificial bee colony algorithm”, International Journal of Computer Applications, Vol. 33, No. 2, November 2011, pp. 41–47.

[10] S.R.M. Oliveira and O.R. Za¨ıane, “Algorithms for balancing privacy and knowledge discovery in association rule mining”, International Database Engineering and Applications Symposium, July 2003, pp. 1–10.

[11] K. Duraiswamy, D. Manjula, N. Maheswari, “A new approach to sensitive rule hiding”, Computer and Information Science, Vol. 1, No. 3, August 2008, pp. 107–111.

[12] S. Menon, S. Sakar, S. Mukherjee, “Maximizing accuracy of shared databases when concealing sensitive patterns”, information system research, Vol.16, No. 3, September 2005, pp. 256–270.

[13] V.S. Verykios, E.D. Pontikakis, Y. Theodoridis, L. Chang, “Efficient algorithms for distortion and blocking techniques in association rule hiding”, Distributed and Parallel Databases, Vol. 22, No. 1, August 2007, pp. 85–104.

[14] A. Amiri, “Dare to share: Protecting sensitive knowledge with data sanitization”, Decision Support Systems, Vol. 43, No. 1, February 2007, pp. 181–191.

[15] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, E. Bertino, "Hiding association rules by using confidence and support", IHW '01 Proceedings of the 4th International Workshop on Information Hiding, 2001, pp. 369–383.

Narges Jamshidian Ghalehsefidi was born in Isfahan, Iran in 1988. She received a B.S. degree in Computer Engineering from Najafabad Branch, Islamic Azad University, Najafabad, Iran, and is currently an M.S. student in Computer Engineering (major: Software) at this university. Her field of research is Preserving Privacy in Data Mining.