International Journal of Pure and Applied Mathematics
Volume 118 No. 24 2018
ISSN: 1314-3395 (on-line version)
url: http://www.acadpubl.eu/hub/
Special Issue

http://www.acadpubl.eu/hub/

Privacy Preservation of Association Rules in Transaction Databases

Geeta S. Navale
Smt. Kashibai Navale College of Engineering, Pune
Savitribai Phule Pune University, Pune
Pune, India
geetanavale@gmail.com
Dr. Suresh N. Mali
Dr. D. Y. Patil Institute of Technology,
Pimpri, Pune
Savitribai Phule Pune University, Pune
Pune, India
snmali@rediffmail.com

May 26, 2018

Abstract

Abstract Association rule mining (market basket analysis) is a popular data mining technique employed on transaction databases to obtain insights from the transaction databases. These insights can be used for providing smart recommendations to the user, shelf configuration management, creating new discount schemes etc. and can be particularly helpful for organizations to gain an edge over their competitors. Over the years, businesses have seen increase in sharing association rules with collaborators. However, one may want to conceal certain discovered patterns from competitors and collaborators, called sensitive rules. Companies may share only the discovered knowledge with competitors, but occasionally, companies prefer sharing their data with collaborators. The latter comes as a challenge to

hide the sensitive rules, without affecting the other rules and due to the value of the data and the knowledge that might be discovered from it, such data may be subject to corporate thefts and espionages. At times, companies might want to share all their data and knowledge to authorized personnel, while hiding it from the unauthorized ones. In this paper we present two algorithms as a part of the proposed framework, it is an attempt to provide a solution in the same context. The first algorithm, RHSAR scrambles the data in such a way that the scrambled data appears genuine, yet it is unintelligible without the framework. Later, the legitimate user can unscramble the data when needed using the second algorithm RSAR.

***Key Words*:**data mining; association rule; information hiding

1 INTRODUCTION

Data Mining is prolific in todays business world. It is used extensively by businesses to improve customer satisfaction, reduce costs and profitability. However, one may mine the data from the database and use the business logic to extract the knowledge. Purpose of the data mining is to discover the unknown information. The discovery of the implicit information also reveals the private or secure information in the data base such as credit card numbers, personal identification numbers, telephone numbers and other confidential data. In business sector, the information retrieval using data mining technique reveals sensitive knowledge about the corporate to the competitors. Information can be retrieved in the form of association rules. These association rules contain important information which needs to be preserved even after mining. This important information in the association rules must be concealed. Many researchers have worked in the field of hiding of association rules. The paper is organized as follows: Important terms are described in Section II. The motivation for conducting the survey is described in section III. Section IV describes the existing approaches to prevent the discovery of association rules. The proposed framework is described in section V and VI. The important findings and results are discussed in section VII. The paper is concluded in section VIII.

2 TERMS

In this section, various terms that have been used throughout the scope of this paper are defined. Consider transaction database shown in Table I.

TABLE I. TRANSACTION DATABASE

T-id	Items
101	x, y
102	x, z, u, v
103	y, z, u, w
104	x, y, z, u
105	x, y, z, w

Itemset [1]: Collection of one or more items from any transaction is called as Itemset. For example x, y, z, u is Itemset. Support Count [1]: It is Frequency of occurrence of an Itemset I in the given transactions. For example: 1. Support Count of (y, x) = 3 and 2. Support Count of (y, x, z) = 2 Confidence Count [1]: It is frequency of how many times the rule found to be true. For example confidence count for the rule x y is (3/4). That means it is the proportion of transactions that contains x also contains y. Frequent Itemset [1]: An itemset is a collection of one or more items, it is called a frequent itemset if the support of an itemset, is greater than or equal to the minimum threshold. While Mining the Associations Rules we have to undergo through two step process: finding the Frequent Itemset and then finding the Association Rule. Therefore the process of discovery of association rules is as shown in Fig. 1.

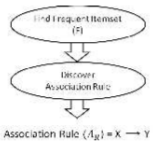

Figure 1Use of Frequent Itemset to find Association Rule

Figure 1Use of Frequent Itemset to find Association Rule Once we get the Frequent Itemset (F), one can easily discover any interesting Association Rules [1] such as X Y. Finding Frequent Itemset generally is pretty easy but very costly as size of the data base increases. The simple approach would be to count all Itemsets that appear in any transaction. Given a set of items of size m, there are 2m subsets that are possible. As we are not interested in the empty set, the possible number of itemsets is 2m - 1. With the help of database transactions, items, frequent itemsets and threshold values of Confidence (C_min) and Support (S_min) one can develop an algorithm to extract all possible association rules as shown in Fig. 2. As the minimum threshold support S_min decreases, the number of frequent itemsets increases. This gives a real challenge for the data mining process.

3 MOTIVATION

Consider the scenario as shown in Fig. 3. If owner of one of the organizations get an access to the database of the other organization, one can mine the database and able to get the various business secrets (sensitive association rules) of that owner and may result in loss of business [2]. Thus, there is a need of privacy of association rules.

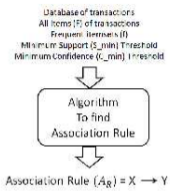

Figure 2 Use of Frequent Itemset, Support and Confidence to find Association Rule

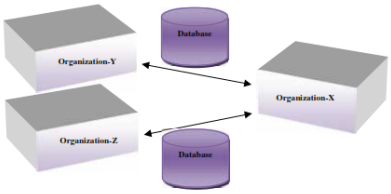

Figure 3Sharing of Database

4 LITERATURE SURVEY

In this section, we discuss various approaches that have been made to hide association rules by modifying the databases. Mining association rules is one of the methods for discovering relationships among data items in data mining, which was first proposed by Agrawal et al [1]. Association rule mining is a powerful model of data mining used for finding hidden patterns in large databases. One of the great challenges of data mining is to protect the confidentiality of sensitive patterns when releasing database to third parties. Association rule hiding algorithms sanitize database such that certain sensitive association rules cannot be discovered through association rule mining techniques.

Wang et al. [3] proposed two data mining algorithms of ISL and DSR for hiding sensitive predictive association rules, which none of them need for data mining and selecting of the sensitive rules. In

the former, a rule is hidden through reducing its confidence by an increase in the support of item sets on the left hand side of the rule; while, in the latter, the rule is hidden through reducing the confidence by a decrease in the support of item sets on the right hand side. To specify hidden rules, entire data mining process needs to be executed. For some applications, we are only interested in hiding certain sensitive predicative rules that contain given items. In this work, they assumed that only sensitive items are given and proposed two algorithms, ISL (Increase Support of LHS) and DSR (Decrease Support of RHS), to modify data in database so that sensitive predicative rules containing specified items on the left hand side of rule cannot be inferred through association rule mining Domadiya et al. [4] presented the MDSRRC algorithm for overcoming the limitations of the DSRRC algorithm, which can also hide rules that have several items in their left or right hand side. This algorithm calculates the sensitivity of each item and each transaction at first. Afterwards, the number of occurrences for each item in the right side of the sensitive rules is obtained and the item with the most number of repetitions is removed from the transaction with the most sensitivity. MDSRRC (Modified Decrease Support of R.H.S. item of Rule Clusters) hides the sensitive association rules with multiple items in consequent (R.H.S) and antecedent (L.H.S). This algorithm overcomes the limitation of existing rule hiding algorithm DSRRC. Proposed algorithm selects the items and transactions based on certain criteria which modify transactions to hide the sensitive information. Experimental result shows that proposed algorithm is highly efficient and maintains database quality. Jain et al. [5] proposed an algorithm that uses the data distortion technique where the position of the sensitive items is altered but their supports and the size of the database remains same. It uses the idea of representative rules to prune the rules first and then hides the sensitive rules. This was achieved using the concept of representative association rule, which was introduced by Kryszkiewicz in 1998. Representative association rule is a small subset of association rules using which all sensitive rules could be deduced without accessing the original database. Advantage of this approach is that it hides maximum number of rules. Modi et Al. [6] proposed DSRRC algorithm. DSRRC (Decrease Support of R.H.S. item of Rule Clusters), provides privacy for sensitive rules at certain level while ensuring

data quality. This algorithm clusters the sensitive association rules based on certain criteria and hides as many as possible rules at a time by modifying fewer transactions. Because of less modification in database it helps maintaining data quality.

ADSRRC [7] overcomes limitation of multiple sorting in database as well as it selects transaction to be modified based on different criteria than DSRRC algorithm. ADSRRC is based on concept of sensitivities i.e. 1. Item Sensitivity is the frequency of data item exists in the number of the sensitive association rule containing this item. It is used to measure rule sensitivity. 2. Rule Sensitivity is the sum of the sensitivities of all items containing that association rule. 3. Cluster Sensitivity is the sum of the sensitivities of all association rules in cluster. Cluster sensitivity defines the rule cluster which is most affecting to the privacy. 4. Sensitive Transaction is the transaction in given database which contains sensitive item. 5. Transaction sensitivity is the sum of sensitivities of sensitive items contained in the transaction Initially association rules are mined from the source database by using association rule mining algorithms e.g. Apriori algorithm. Then sensitive rules are specified from mined rules. Selected rules are clustered based on common R.H.S. item of the rules. Then transactions are indexed by sensitivities. In DSRRC transaction sensitivity is different for each cluster but ADSRRC calculates transaction sensitivity irrespective of clusters. It means for all clusters transaction sensitivity is same. After transaction indexing, ADSRRC sorts the transactions based on sensitivity. In DSRRC, each time item is removed, transactions are sorted. But in ADSRRC, first transactions are sorted in decreasing order of their sensitivity. Then transactions having same sensitivity are sorted in decreasing order of their length. When an item is removed from any transaction, sensitivity is not modified. Thus transactions are sorted only two times. Thus we are avoiding here multiple sorting of transactions which will significantly reduce execution time of algorithm for large database. After sorting process, rule hiding process, starts by selecting highest sensitive transaction for deleting R.H.S. item. If two transactions have same sensitivity then lengthiest transaction is chosen to be modified. This process continues until all the sensitive rules in all clusters are not hidden. Finally modified transactions are updated in original database and produced database is called sanitized database

which ensures certain privacy for specified rules and maintains data quality. RRLR proposed by Shah et al. [7] also uses concept of sensitivity. This algorithm hides sensitive association rules having multiple RHS items. In algorithm RRLR, to hide sensitive association rule we are decreasing support and confidence both. In LHS insertion, confidence of rule can be decreased by inserting LHS of rule in transaction not supporting RHS of rule. Initially association rules are mined from the source database by using association rule mining algorithms e.g. Apriori algorithm. Then sensitive rules are specified from mined rules. Then transaction sensitivity and item sensitivity is found. Transactions are sorted in decreasing order of their sensitivity and length. These all steps are similar to ADSRRC algorithm except that in RRLR we are not creating clusters. After sorting process, rule hiding process hides all the sensitive rules in sorted transactions by using LHS insertion and Deletion Process. It will not update the sensitivity of transactions during rule hiding. Hiding process starts from highest sensitive transaction and continues until all the sensitive rules are not hidden. Finally modified transactions are updated in original database and produced database is called sanitized database which ensures certain privacy for specified rules and maintains data quality. Algorithm RRLR overcomes limitation of hiding rules having multiple R.H.S. items. Both ADSSRC and RRLR outperform DSRRC in terms of side effects generated and data quality in most cases. Weng et al. [8] implemented FHSAR algorithm, which completely hides sensitive association rules with limited side effects. In FHSAR, a strategy is developed for avoiding hidden failures. The algorithm can completely hide any given SAR by scanning database only once. FHA algorithm is introduced for hiding sensitive patterns by Fouladfar [9]. This algorithm is used to reduce the overload of ordering transactions by decreasing database scans. Also, they have reduced the side effects by selecting the appropriate item for performing the modifications. In this algorithm, the transactions are modified in a manner that the confidence of sensitive rules would be reduced. They decrease database scans and calculate the amount of changes before starting the hiding process in order to reduce temporal and computational overloads. Also by selecting the appropriate item for performing the changes, they have reduced the amount of lost rules and ghost rules and by inserting deleted items in suitable transac-

tions, they give back the number of sensitive items to the initial state. Three assessment parameters which are used to compare above algorithms are as shown in Table II. The result of comparison of various algorithms is shown in Table III.

TABLE II. ASSESSMENT PARAMETERS

Hiding Failure [10]	Amount of sensitive information that can be accessed even after applying hiding process.
Lost Rules [10]	Number of non-sensitive rules mistakenly hidden after applying hiding process.
Ghost Rules [10]	Number of rules generated in modified database which are not there in original database.

TABLE III. COMPARISON OF VARIOUS ALGORITHMS

Algorithm	Hiding Failure	Lost	Ghost Rules
ISL [3]	Y	Y	Y
DSR [3]		Y	
MDSRRC [4]		Y	
DSRRC [6]		Y	
ADSRRC [7]		Y	
RRLR [7]	Y	Y	
FHSAR [8]		Y	Y

5 RHSAR ALGORITHM

Recoverable Hiding of Sensitive Association Rules (RHSAR) is proposed in this section. The frequent itemsets and set of association rules based on the support and confidence are computed. Each item from the database is marked as frequent or non-frequent based on the support. The inverted index of the transaction database is computed. Non Frequent Item (NFI) is randomly selected. Replace Frequent Item to be Hidden (FIH) by NFI and encode the FIH.

Input:

A. A transaction database.

B. A set of sensitive association rules to hide.
C. The minimum support of the rules
D. The minimum confidence of the rules
E. A Key, which can be used to restore the original database
Output: A Sanitized Database with hidden sensitive association rules Algorithm to hide is given in Figure 4.

6 RSAR ALGORITHM

Recovering Sensitive Association Rules (RSAR) algorithm to recover the original database from sanitized database is as below:
Input: A. Sanitized Database
B. Key
C. Minimum Support
D. Minimum Confidence
Output: A Recovered Database with original association rules

```
Compute the frequent itemsets and set of association rules based on the support and confidence.
Mark each item from the database as frequent or non-frequent based on the support.
Compute the inverted index of the transaction database.
For each sensitive rule from the set of association rules to hide:
  Select First Antecedent Element as FAE
      For each antecedent element of the association rule:
          If (supp (FAE)> supp (Antecedent Element)) then
              Select Antecedent element as FAE.
      For each Consequent element of the association rule:
          If (supp (FAE)> supp (Consequent Element)) then
              Select Consequent element as FAE
       While (supp (FAE)>=minSupport)
          Increment the counter for iteration of NFI.
          Randomly select a transaction.
          If (if transaction contains all elements of the association rule)
                  Remove FAE from transaction.
                  If (supp (NFI) < minSupport)
                     Add NFI in the Transaction.
                  Else
                      Add NFI in the Transaction.
                      Randomly select a NFI.
                    Update the support of FAE
              Else
                      Remove the NFI from the transaction
      Initialize a storage transaction containing FAE and counts of each NFI.
      Convert the FAE and count of the iteration of each NFI into binary add stores them in Storage
          transition.
      Store the count of elements present in Storage transaction as the first element of the same in binary.
      Initialize the demarcation transaction for demarks the boundaries between the different fields stored in
          storage transaction.
      Encode the Storage and Demarcation transaction, and add the transaction to the appropriate position,
          subject to key.
Encode the elementremovecount in the last transaction.
```

Figure 4RHSAR

```
Compute the frequent itemsets and set of association rules based on the support and confidence.
Mark each item from the database as frequent or non-frequent based on the support
Load the inverted index of the transaction database.
Decode the itemcount from the last transaction.
Calculate the extraTransCount from itemcount.
For each extraTransCount:
    Decode the Storage and Demarcation transaction
    Extract all the elements present in the Storage transaction w.r.t. Demarcation Transaction and convert
    them to decimal.
    For (i from 1 to countOfElements)
        Get the iterations for NFI
        Randomly select the NFI (Non frequent Item which will replace the FAE), selection is subject to
        input key.
        While (iterations)
            Randomly select a transaction, the random selection is subject to the NFI.
            If (NFI present in Transaction)
                Add FAE in the transaction.
                iteration--
Remove all the extra transactions.
```

Figure 4RHSAR

7 EXPERIMENTAL RESULT AND ANALYSIS

This section presents experimental results of three assessment parameters shown in Table 2. Experimentation was carried out on an ASUS N56 Laptop with Intel Processor Core i5 3230M at 2.67 GHz, RAM 4GB and Operating System Windows 7. Experimental Results on Mushroom Dataset [11], Average Results for 3, 5 and 7 sensitive association rules of Support=0.6 Confidence=0.8 is shown in Table IV. We have compared our results with six algorithms as shown in Table 4. Experimental Results on Chess Dataset [12], Average Results for 3, 5 and 7 sensitive association rules of Support=0.85 Confidence=0.95 is shown in Table V.

TABLE IV. EXPERIMENTAL RESULTS ON MUSHROOM DATASET

Algorithm	Hiding Failure	Lost Rules	Ghost Rules	CPU Time (ms)	CPU Time for Recovery (RSAR) (ms)
ADSRRC	19.04%	51.33%	0.00%	44208	NA
DSRRC	19.04%	57.86%	0.00%	73556.67	NA
MDSRRC	11.43%	52.79%	0.00%	119233.3	NA
RRLR	15.87%	72.58%	24.21%	514497	NA
FHA	0.00%	49.83%	20.66%	16633.33	NA
FHSAR	0.00%	81.75%	0.00%	265	NA
RHSAR	0.00%	18.57%	0.00%	14532	14263

TABLE V. EXPERIMENTAL RESULTS ON CHESS DATASET

Algorithm	Hiding Failure	Lost Rules	Ghost Rules	CPU Time (ms)	CPU Time for recovery (RSAR) (ms)
ADSRRC	4.76%	92.30%	0.00%	1855	NA
DSRRC	4.76%	92.30%	0.00%	3909	NA
MDSRRC	0.00%	65.94%	0.00%	2641	NA
RRLR	0.00%	49.43%	9%	12260	NA
FHA	0.00%	49.40%	10.66%	950	NA
FHSAR	0.00%	34.51%	0.00%	178	NA
RHSAR	0.00%	12.21%	0.00%	3447	3287

8 CONCLUSION

In this paper, we have proposed a framework that will provide enterprises a means to safeguard their valuable insights obtained through association rule mining. Also we have implemented an algorithm RSAR to recover the original database from sanitized database for the intended user. The experimental results shows that proposed framework is effective are effective and achieved significant improvement over other existing methods.

References

[1] Agrawal, R., Imielinski, T., Swami A.: Mining Association Rules between Sets of Items in Large Databases. ACM SIGMOD International Conference on Management of Data (SIGMOD93), Washington D.C., USA, pp. 207216, (1993).

[2] Yucel Saygin, Vassilios S. Verykios, Ahmed K. Elmagarmid: Privacy Preserving Association Rule Mining. Research Issues in Data Engineering: E-Commerece/E-business Systems, 2002. RIDE-2EC, pp. 50-59, (2002).

[3] S.-L. Wang and A. Jafari: Hiding sensitive predictive association rules. In 2005 IEEE International Conference on Systems, Man and Cybernetics, vol. 1, pp. 164169. (2005)

[4] N. H. Domadiya and U. P. Rao: Hiding sensitive association rules to maintain privacy and data quality in database. In Ad-

vance Computing Conference (IACC), 2013 IEEE 3rd International, pp. 13061310 (2013).

[5] Y. K. Jain, V. K. Yadav, and 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, pp. 27922798, (2011).

[6] C. N. Modi, U. P. Rao, and D. R. Patel: Maintaining privacy and data quality in privacy preserving association rule mining. In Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on, pp. 16, (2010).

[7] Komal Shah, Amit Thakkar, Amit Ganatra: Association Rule Hiding by Heuristic Approach to Reduce Side Effects & Hide Multiple R.H.S. Items. International Journal of Computer Applications (0975 8887) Volume 45 No.1, May (2012).

[8] Chih-Chia Weng, Shan-Tai Chen, Hung-Che Lo: A Novel Algorithm for Completely Hiding Sensitive Association Rules. Eighth International Conference on Intelligent Systems Design and Applications, vol. 03, no. , pp. 202-208, IEEE (2008).

[9] Maryam Fouladfar, Mohammad Naderi Dehkordi: A Heuristic algorithm for quick hiding of association rules. ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 1, No.1, January (2015).

[10] Yuhong Guo : Reconstruction-Based Association Rule Hiding. In Proc. of SIGMOD2007 PhD. Workshop on Innovative Database Research 2007 (IDAR2007), pp. 51-56, June (2007)

[11] UCI Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/Mushroom

[12] UCI Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/Chess+(King-Rook+vs.+KingPawn)