ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 1, No.13 , January 2015
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

A heuristic algorithm for quick hiding of association rules

Maryam Fouladfar[1], Mohammad Naderi Dehkordi[2]

Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran

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

Abstract

Increasing use of data mining process and extracting of association rules caused the introduction of privacy preserving in data mining. A complete publication of the database is inconsistent with security policies and it would result in disclosure of some sensitive data after performing data mining. Individuals and organizations should secure the database before the publication, because if they neglect this issue they will be harmed. The owners of database consider factors such as database size, precision in immunization and velocity in choosing the right approach in order to hide the association rules. Besides the large volume of data and precision in immunization, we should optimize the time of operation and this is one of the issues that has received a little attention. In this paper, FHA algorithm is introduced for hiding sensitive patterns. In this algorithm, it is being tried to reduce the overload of ordering transactions by decreasing database scans. Also, we have reduced the side effects by selecting the appropriate item for performing the modifications. Conducted experiments indicate the execution of this algorithm in appropriate hiding of sensitive association rules.

Keywords:*Data mining , association rules , privacy in data mining*

1. Introduction

In each business, we need some information in order to further the goals that an important piece of this data would be obtained by exploring databases[1]. Extracting of association rules is one of the useful methods in data mining that it causes extracting useful information from the database in the form of rule (=base)[2]. In this method, the abundant items in the database would be extracted at first and then the regulations will be constituted based on numerous items. Analyzing of obtained association rules will cause us to achieve hidden knowledge among the mass of data that the owner of database considers a part of these rules, sensitive. Database owners are reluctant to release sensitive rules, so the issue of how to protect sensitive knowledge in data mining received a lot of attentions. Privacy preserving in data mining is an approach that explains the way of databases changing before their publication to the extent that after publication, sensitive knowledge among extracted patterns cannot be discovered but still exploiting of the database is possible. Many algorithms have been introduced for extracting association rules, which the most useful is Apriori algorithm[3]. One association rule would be extracted in $X \rightarrow Y$ form. If $I = \{i_1, i_2, i_3 \dots i_n\}$ be a group of items, D be a database of transactions and each transaction be in the form of $T \subset I$, so the results are: $X \subset I$, $Y \subset I$ and $X \cap Y = 0$. For each rule, the two criteria of support and confidence are studied[4,5]. The support and confidence for the rules in the database are respectively calculated from equations (1) and (2):

$$Support(X,Y) = \frac{|X \cup Y|}{|D|} \quad (1)$$

$$Confidance(X \rightarrow Y) = \frac{|X \cup Y|}{|X|} \quad (2)$$

$X \cup Y$ is the number of transactions that support the group of XY items, $|D|$ is the total number of database transactions and $|X|$ is the number of transactions that support the X itemset[6].

Rules that support threshold and confidence of them are more than the user specified threshold, would be extracted from the database as association rules. From extracted rules, some of them would be introduced as sensitive association rules. The database must be secured before the publication in order to prevent the extraction of such rules. Securing a database would be done by using algorithms of privacy preserving in data mining. These algorithms are presented in 3 approaches of heuristic based, border based and exact [7]. One of the most commonly used methods in hiding association rules is heuristic method. The introduced algorithms in this approach with implementing smart policies would extremely restrict the state space of the problem and finding the answer would be done quicker. Many of these algorithms belong to two categories of data blocking and data distortion. [8]. In data blocking, the quantity (=value) of "?" would be replaced with sensitive values and data distorting in binary databases would be done by changing the value of items from 0 to 1 and vice versa. Criteria for evaluating algorithms have been categorized in three groups of failure in hiding, lost rules and ghost rules [9,10]. The purpose of the proposed algorithms is optimizing each of these criteria. The occurrence of any of these factors will cause disharmony in secured database. Failure in hiding indicates the amount of sensitive data which after hiding operation is still extractable. To the best point, this amount is equal to zero and it can be calculated from equation (3). SR(X)

represents the number of sensitive association rules which have been extracted from X database.

$$HF = \frac{\#SR(D')}{\#SR(D)} \quad (3)$$

Lost rules are the side effects of hiding process. This criterion specifies the number of insensitive rules which have been lost during hiding sensitive rules. Ideally, this value is equal to zero and can be calculated from equation (4). ~SR(X) represents the number of insensitive association rules that have been extracted from X database.

$$MC = \frac{\#{\sim}SR(D) - \#{\sim}SR(D')}{\#SR(D)} \quad (4)$$

Ghost rules are items and new regulations that unexpectedly are added to the secured database during immunization. Ideally, this value is equal to zero and is calculated from equation (5).

$$AP = \frac{|R'||R \cap R'|}{|R'|} \quad (5)$$

In this paper, FHA (Fast Huristic Approach) heuristic algorithm has been introduced with the distorted approach for hiding sensitive association rules. In this algorithm, the transactions are modified in a manner that the confidence of sensitive rules would be reduced. We 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, we have reduced the amount of lost rules and ghost rules and by inserting deleted items in suitable transactions, we give back the number of sensitive items to the initial state.

The rest of this article is organized in 5 sections: Section (2) describes the tasks that have been done. Section (3) introduces the proposed algorithm. In section (4) an example of the algorithm is discussed and the process of that is examined. Section (5) evaluates and compares the proposed algorithm with the DSRRC, ADSRRC, MDSRRC and RRLR algorithms and in section (6) the conclusion is presented.

2. Accomplished tasks

In recent years, many algorithms have been proposed for hiding association rules and sensitive data that these algorithms do the hiding process of sensitive rules by reducing the amount of support and confidence. In this section we introduce some presented algorithms in heuristic approach with data distortion technique.

In 2005, Wang and Jafari have introduced two ISL (Increase Support of LHS) and DSR (Decrease Support of RHS) algorithms. These two algorithms use the method of reducing the confidence for hiding association rules. In ISL algorithm, the support of LHS[1] set in transactions that do not support the RHS[2] set would increase for reducing the confidence of sensitive rules. DSR algorithm reduces sensitive rule in transactions that completely support the base(=rule) for decreasing the confidence support of RHS. The final result of this algorithm depends on the arrangement of the items in transactions [12].

In 2008, Weng and his colleagues presented an algorithm for hiding association rules that it can hide all the sensitive rules with only one scan. In this algorithm all the information about the transaction and their relationship with sensitive items are collected by using one scan. Then, based on this information, eliminating items from the most appropriate transactions would be done [13].

In 2010, Modi and his colleagues introduced DSRRC (Decrease Support of R.H.S. item of Rule Clusters) algorithm that sensitive rules are clustered on the basis of common RHS with the help of this algorithm. Then the sensitivity of each item in the cluster, each cluster sensibility and sensitivity of each transaction to the clusters are calculated and then, database transactions are arranged according to these sensitivities and after that removing clusters' RHS from arranged transactions would be done. DSRRC algorithm depends on the arrangement of transactions in the main database also in this algorithm, sorting operation would be done after eliminating each item from the database which it increases the algorithm runtime. In addition, DSRRC algorithm can not work with rules that have multiple RHS [14].

In the year 2011, Jain, Yadav and Panday have introduced a mixed algorithm for reducing the side effects of ISL and DSR algorithms. The purpose of this algorithm is to reduce the failure of hiding in ISL. Of course, the number of performed scans in this algorithm is very high which it can increase the running time of the algorithm for large databases [15].

In 2012, Komal Shah and his colleagues introduced two ADSRRC (Advanced Decrease Support of R.H.S items of Rule Cluster) and RRLR (Remove and Reinsert L.H.S of Rule) algorithms for removing limitations of DSRRC algorithm. In ADSRRC algorithm clustering of sensitive rules is done same as DSRRC algorithm, but in running time ADSRCC algorithm is faster than the DSRRC

[1] Left Hand Side
[2] Right Hand Side

algorithm because of performing just two sorting acts. RRLR algorithm is designed to hide association rules with multiple RHS and for hiding sensitive rules, it reduces the confidence of the rules. In this algorithm, only two sorting operations are done and for this reason, the runtime is less than DSRRC algorithm. In addition, RRLR algorithm is superior to DSRRC algorithm in terms of the number of lost rules and the number of database changes [16].

In the year of 2012, Nikunj et al, proposed MDSRRC to hide association rules. MDSRRC can hide rules with different RHS and LHS. At first, sensitivity of items in rules' RHS calculated and then the most sensitive item will be selected to delete. MDSRRC (Modified Decrease Support of R.H.S. item of Rule Clusters), in comparison with DSRRC, reduces databasemodification and side effects with deleting the effective candidate item [17].

3. The proposed algorithm

The proposed algorithm uses distortion techniques based on reducing the confidence of sensitive rules. In this method, there is no limitation for hiding association rules with each number of items on the left and right hand sides of the base(=rule). Reduction of database scans and calculating the rate of changes before starting the hiding process would significantly reduce the amount of required operations for hiding process that shows the most efficiency on large databases. Also, in order to reduce the lost rules, victim item is calculated in each rule(=base) and according to that the leading rules would be specified for hiding. Eventually, we have reduced the amount of disharmony in the final database by fixing the support of sensitive items. The structure of proposed algorithm is as follows:

Input: Source Database D, Minimum Confidence Threshold (MCT), Minimum Support Threshold (MST).
Output: The Sanitized Database D'.

(Step 1) Select sensitive patterns: First, we choose sensitive patterns and name them SR to perform the hiding process in this method.

(Step 2) Select the victim: Sensitive items, are the participant items in sensitive rules. To identify the victim Item, we measure the amount of each sensitive item repetation in sensitive and insensitive rules according to the presented formula in the article[18]. The item which has the lowest common between sensitive and insensitive rules, will be chosen as the victim. You can see this formula in equation (6).

$$a = sensitivity\ of\ each\ item\ i \in SR$$
$$b = sensitivity\ of\ each\ item\ i \in NSR$$
$$Victim = \frac{a}{b} \quad (6)$$

(Step 3) Create leading rules: We produce the leading rules according to victim item. A leading rule is a combination of rules with the specified victim that by hiding it, all the rules of subset would be hidden.

(Step 4) Calculating N1 and N2: We calculate two N1 and N2 formulas in order to reduce the volume of operations and the running time of the algorithm. We compute the number of needed operations to reduce the confidence from the equation (7) if the victim of a rule is from the right hand side items of the rule, and if it be from the left hand side we use equation (8).

$$\alpha = |D| * MCT$$
$$\beta = \lceil sup(R) - \alpha \rceil$$
$$\gamma = 1 + MCT$$
$$N2 = \beta / \gamma \quad (7)$$

$$\alpha = |D| * MST$$
$$N2 = \lceil sup(R) - \alpha \rceil \quad (8)$$

(Step 5) Remove and insert: We start hiding operation from the first transaction and, if it is possible, we perform one insert and delete operation on it. We continue this work till calculated operations for rules be completed. If all parts of operation were not performed by one passing from database transactions, we would change victim items and return to step 4.

(Step 6) Updating the confidence: We calculate the confidence for all the sensitive rules and we omit the rules that their confidence threshold is less than the user specified minimum confidence threshold. If a sensitive rule had not been hidden, we should resume the operations.

4. Example:

In this section, one example is discussed for a better understanding of the proposed algorithm. Table (1) shows the main database with 10 transactions and 9 items. The extraction of association rules has been done with minimum support threshold of 40 and minimum confidence threshold of 50. The elected sensitive rules and victim item of each rule are shown in table (2). You can see the clustering, creating leading rule and the amount of the required changes for each leading rule in table (3). The secured database by using proposed algorithm is shown in Table (4).

Table 1: main database with 10 transactions and 9 items

Transaction ID	*Items*
1	a, c, e, g, i
2	b, c, d, e, f
3	a, c, e, g
4	d, c, h
5	a, c, e, h
6	c, i
7	g, h
٨	a, c, e, i
9	c, b. e, a, f
10	e, a, d, i

Table 2: Sensitive Rules and victim

Rule	*Victim*
a → e	e
e → a	e
e → ac	e
ac → e	e

Table 3: Clustering, creating a leading rule for each cluster and calculate the required changes

Number of modification	*leading rule*	*rule*
2	ac → e	a → e
		ac → e
3	e → ac	e → a
		e → ac

Table (4): Sanitized Database

Transaction ID	*Items*
1	a, c, g, i
2	b, c, d, e, f
3	a, c, g
4	d, c, h, e
5	a, c, h
6	c, i, e
7	g, h, e
8	a, c, i
9	c, b, e, a, f
10	e, a, d, i

5. Comparison and evaluation of the proposed algorithm:

To demonstrate the efficiency of FHA algorithm, we tested it with DSRRC, ADSRRC, RRLR and MDSRRC algorithms on a system with exhibited characteristics in table (5) which each algorithm has the ability to hide rules with specified format. To test these algorithms, we use the Chess and Mushroom databases. You can observe the specifications of these two databases in table (6).

Table (5): Computer System

ASUS N56	
Processor	Intel core i5 with 2.67 GH
RAM	4GB
Operating System	Windows 7

Table (6): Database details

Database Name	*Number of Record*	*Number of Items*	*Number of Items Per record*
Mushroom	8124	119	23
Chess	3196	75	37

For more accurate comparison and evaluation, three tests have been done on dataset with 3, 5 and 7 different laws(=rules). Evaluation criteria of each test are failure in hiding, the number of lost rules, the number of ghost rules and runtime in milliseconds. The average of obtained results for each criterion is provided in tables (7) and (8).

The first set of experiments has been done on the Mushroom database with the support threshold of 60 and confidence threshold of 80 and the second set has been done on Chess database with the support threshold of 85 and confidence threshold of 95.

Table 7: Average results for 3, 5 and 7 rules in the Mushroom database

	Failure	*Lost*	*Ghost*	*CPU Time*
ADSRRC	19.04%	51.33%	0	44208
DSRRC	19.04%	57.86%	0	73556.67
MDSRRC	11.43%	52.79%	0	119233.3
RRLR	15.87%	72.58%	24.21%	514497
FHA	0	49.83%	20.66%	16633.33

Table 8: Average results for 3, 5 and 7 rules in the Chess database

	Failure	*Lost*	*Ghost*	*CPU Time*
ADSRRC	4.76%	92.3%	0	1855
DSRRC	4.76%	92.3%	0	3909
MDSRRC	0	65.94%	0	2641
RRLR	0	49.43%	9	12260
FHA	0	49.4%٠	1٥.66%	950

As you see in the tables (7) and (8), the average failure rate in Chess and Mushroom databases for the proposed algorithm is equal to zero because we control sensitive rules again, even after hiding. Also, due to the selection of suitable item for removal, we have fewer lost rules in both databases for proposed algorithm. Because of insertion operations in FHA algorithm, the number of produced ghost rules is much in comparison with other algorithms. The most power of the proposed algorithm is in optimizing the runtime of the immunization process. We reduced the time of the immunization process by 19.4% for Mushroom database and 30% for Chess database with calculating the amount of required modifications for inserting and deleting of each leading rule. According to the conducted experiments, it can be concluded that the proposed algorithm in dense and sparse databases does not have failure in hiding and by calculating the rate of changes before immunization, the proposed algorithm has better performance rather than DSRRC, ADSRRC, RRLR and MDSRRC algorithms in the number of lost rules and runtime.

6. Conclusion and future works

Until recently, each person was exploring his database and was using from the gained knowledge, but when people decided to expand their business, exploring limited bases were not enough, anymore. Database owners attempted to share their own databases with the aim of gaining mutual benefit which in conjunction with that, the protection of some sensitive data was discussed because database owners were reluctant to disclose the latent sensitive knowledge in their own database. Thus the algorithms of privacy preserving in data mining were used in order to prevent the propagation of sensitive data. Each of these algorithms imposes unintended negative effects on the database during the process of immunization. Generally, the aim of introducing the proposed algorithm is to reduce the side effects of hiding process. In this article we have presented FHA algorithm with distortion technique and with an approach based on confidence. The purpose of this algorithm is to reduce the side effects and specifically, optimizing the time of immunization operations. This algorithm has the ability to hide any kind of sensitive rule with any number of items on the left and right hand sides. Failure in the process of hiding is not desirable and we have solved this problem by controlling the status of sensitive rules at the time of hiding. In this method for hiding each sensitive pattern, we choose an item which has the lowest common with insensitive rules. This action reduces the amount of lost rules. The proposed algorithm has better result on sparse databases compared to dense databases in the field of lost rules. To solve the problem of dense bases, it's enough to perform one sorting action for transactions at the beginning of the immunization process. The calculations indicate that the time of the immunization process in FHA has been decreased to 30% in Chess database and 19.4% in Mushroom database. For this reason, it is expected that the presented algorithm shows a great performance for massive databases at runtime. The weakness of proposed algorithm is in the amount of produced ghost rules. These quantities have been created because of insertion operations to prevent the reduction of the number of sensitive items. Because, the existence of many insensitive items on the left side of leading rule and inserting all of them bring about creating more ghost rule.

By considering this point that the amount of produced ghost rules in this algorithm depends on the number of insensitive items on the left hand side of leading rule, for future works we can propose reducing the amount of lost rules by offering appropriate formula and clustering.

References

[1] A. Rakesh, and S. Ramakrishanan, " Fast algorithms for mining association rules", Proceedings of the 20th VLDB Conference Santiago, 1994, Vol. 1215, pp. 487-499.

[2] Z. R. Osmar, "Introduction to Data Mining", Citeseer, 1999.

[3] A. Rakesh, and I. Tomasz, and S. Arun, "Mining Association Rules between Sets of Items in Large Databases", ACM SIGMOD Record, Vol. 22, No. 2, 1993, pp. 207-216.

[4] H. P. Tzung, and L. W. Chun, and C. C. Chia, "Hiding sensitive itemsets by inserting dummy transactions", Granular Computing (GrC), 2011 IEEE International Conference on, 2011, pp. 246-249.

[5] D. N. Mohammad, and B. kambiz, and KZ. Ahmad, "A Novel Method for Privacy Preserving in Association Rule Mining Based on Genetic Algorithms", Journal of software, Vol. 4, No. 6, 2009, pp.555-562.

[6] Sh. Komal, and T. Amit, and G. Amit, "A Study on Association Rule Hiding Approaches", International Journal of Engineering and Advanced Technology (IJEAT), Vol. 1, No. 3, 2012, pp. 2249-8958.

[7] R. R. Nilesh, and P. b. Nilesh, S. H. Krupali, "Privacy Preserving in Association Rule mining", international

journal of advanced and innovative research (IJAIR), Vol. 2, No. 4, 2013, pp. 2278-7844.

[8] V. S. Vassilios, and P. D. Emmanuel, and T. Yannis, and C. Liwu, "Efficient algorithms for distortion and blocking techniques in association rule hiding", Distributed and Parallel Databases, Vol. 22, No.1, 2007, pp. 85-104.

[9] O. M. Stanley, and Z. R. Osmar, "Algorithms for balancing privacy and knowledge discovery in association rule mining", Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International, 2003, pp. 54-63.

[10] S. Vijayarani, and M. Sathiya Prabha, "Association Rule Hiding using Artificial Bee Colony Algorithm", International Journal of Computer Applications, Vol. 33, No. 2, 2011, pp. 41-47.

[12] W. L. Shyue, and J. Ayat, "Hiding Sensitive Predictive Association Rules", International Conference on Systems, Man and Cybernetics, 2005 IEEE, Vol. 1, pp. 164-169.

[13] W. C. Chih, and C. T. Shan, and L. C. Hung, "A Novel Algorithm for Completely Hiding Sensitive Association Rules", Eighth International Conference on Intelligent Systems Design and Applications, 2008, Vol. 3, pp. 202-208.

[14] M. N. Chirag, and R. P. Udai, and P. R. Dhiren, "Maintaining Privacy and Data Quality in Privacy Preserving Association Rule Mining", international conference on computing communication and networking technologies (ICCCNT), 2010 , pp. 1-6.

[15] J. K. Yogendra, and Y. K. Vinod, and P. S. Geetika, "An Efficient Association Rule Hiding Algorithm for Privacy Preserving Data Mining", International Journal on computer science and engineering(IJCSE), Vol. 3, No. 7, 2011, pp. 0975-3397.

[16] Sh. Komal, and T, Amit, and G. Amit, "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, 2012, pp. 0975-8887.

[17] D. H. Nikunj, and R. P. Ydai, "Hiding Sensitive Association Rules to Maintain Privacy and Data Quality in Database", Advance Computing Conference (IACC), 2013 IEEE 3rd International, 2013, pp. 1306-1310.

[18] A. Ali. "Dare to share: Protecting sensitive knowledge with data sanitization", Decision Support Systems, Vol. 43, No. 1, 2007, pp. 181-191.