International Journal of Mathematical Archive-2(2), Feb. - 2010, Page: 289-293
Available online through www.ijma.info ISSN 2229 – 5046

DESIGN AND DEVELOPMENT OF SANITIZATION ALGORITHM FOR MINING PRIVACY - PRESERVING FREQUENT ITEMSETS

Bharat Solanki*, Rashmi Awasthy and Rajesh Shrivastava

Shri Ram Institute of Technology, Computer Science Department Jabalpur, Madhya Pradesh, India
E-mail: Jabalpur_bharat@yahoo.co.in, Rashmi.8sept@gmail.com

(Received on: 03-12-10; Accepted on: 17-12-10)

ABSTRACT

Data mining services require accurate input data for their results to be meaningful, but privacy concerns may influence users to provide spurious information. In order to preserve the privacy of the client in data mining process, a variety of techniques based on random perturbation of data records have been proposed recently. In this paper we concentrate on Data sanitization problem by providing a Non-uniform Randomized sanitization algorithm for sanitizing the original database to transform it into a sanitized database devoid of any sensitive patterns specified by the data owner. The Uniform randomized sanitization algorithm considers any item in a restricted itemset as a victim item to be removed from sensitive transactions with an equal probability. But the Non-uniform Randomized sanitization algorithm prefers items with high support as victim items, thereby minimizing the effect on non-sensitive patterns. As a result accuracy will be increased.

Keywords-*frequent patterns, sensitive patterns, non-sensitive patterns, legitimate patterns, sanitization, privacy preserving*

---

1. INTRODUCTION:

The goal of the sanitization process is to hide some restrictive patterns that contain highly sensitive knowledge. This process is composed of four steps. In the first step, the set P of all patterns from D is identified. The second step distinguishes Restricted patterns R_p from non-restrictive patterns $\sim R_p$ by applying some security policies. It should be noted that what constitute as restrictive pattern depends on the application and the importance of these patterns in a decision process. In third step, sensitive transactions are identified within D. In this approach the authors have used an efficient retrieval mechanism called the transaction retrieval engine to speed up the process of finding the sensitive transactions. Finally, Step 4 is dedicated to the alteration of these sensitive transactions to produce the sanitized database. The process of modifying such transactions satisfies a risk of disclosure threshold controlled by the user. This threshold basically expresses how relaxed the privacy preserving mechanisms should be. When $\psi = 0\%$, no restrictive patterns are allowed to be discovered. When $\psi = 100\%$, there are no restrictions on the restrictive patterns.

A. Some Definitions:

Let D be a transactional database, P be a set of all frequent patterns that can be mined from D, and $Rules_H$ be a set of decision support rules that need to be hidden according to some security policies. A set of patterns, denoted by R_P, is said to be restrictive if $R_p \subset P$ and if and only if R_P would derive the set $Rules_H$. $\sim R_P$ is the set of non-restrictive patterns such that $\sim R_P \cup R_P = P$.

Corresponding author:
Bharat Solanki*
E-mail: Jabalpur_bharat@yahoo.co.in
Shri Ram Institute of Technology, Jabalpur, (M.P.)

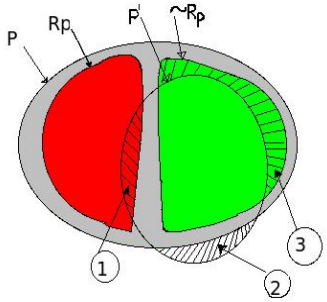

Figure 1: Visual representation of restrictive and non-restrictive patterns and the patterns effectively discovered after transaction sanitization

Figure illustrates the relationship between the set P of all frequent patterns in the database D, the restrictive and non-restrictive patterns, as well as the set P of frequent patterns discovered from the sanitized database D. 1, 2, and 3 are potential problems that represent the restrictive patterns that were failed to be hidden, the artificial patterns created by the sanitization process and the legitimate patterns accidentally missed.

A group of restrictive patterns is mined from a database D based on a special group of transactions. We refer to these transactions as sensitive transactions and define them as follows.

Let T be a set of all transactions in a transactional database D and R_P be a set of restrictive patterns mined from D. A set of transactions is said to be sensitive, as denoted by S_T, if $S_T \subset T$ and if and only if all restrictive patterns can be mined from S_T and only from S_T.

B. Framework:

It encompasses a transactional database, an inverted file, a set of sanitizing algorithms used for hiding restrictive patterns from the database, and a transaction retrieval engine for fast retrieval of transactions.

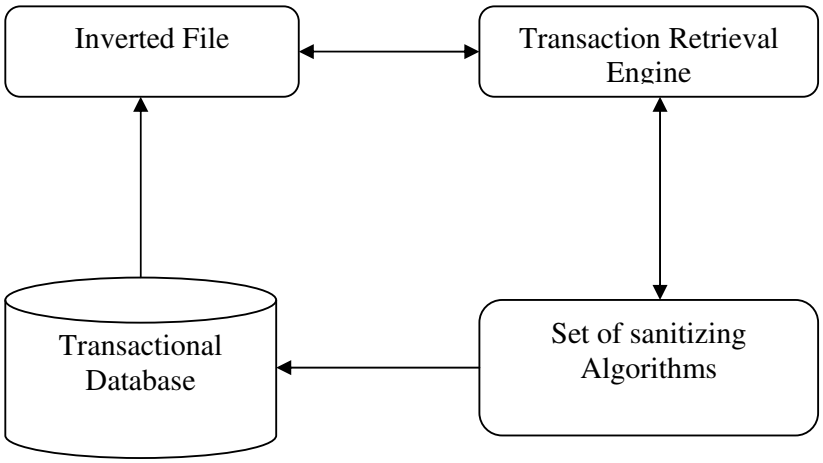

Figure 2: Privacy Preservation Framework

Sanitizing a transactional database consists of identifying the sensitive transactions and adjusting them. To speed up this process, we model transactions into documents in which the items simply become terms. This model preserves all the information and provides the basis for our indexing, borrowing from the information retrieval domain.

C. The Inverted File Index:

One very efficient strategy for indexing a text database is an inverted file. An inverted file, a structure comprising the vocabulary and the occurrences, is a word-oriented mechanism for indexing a text collection with the purpose of speeding up the searching task.

In this framework the inverted file's vocabulary is composed of all different items in the transaction database, and for each item there is a corresponding list of transaction Ids in which the item is present. The figure shown below is an example of inverted file corresponding to the sample transaction database shown in the figure.

Docs	Items/Terms
T1	A B C D
T2	A B C
T3	A B D
T4	A C D
T5	A B C
T6	B D

Items	Frequencies
A	5
B	5
C	4
D	4
Vocabulary	

Transaction IDs
T1, T2, T3, T4, T5
T1, T2, T3, T5, T6
T1, T2, T4, T5
T1, T3, T4, T6

Figure 3: An Example of transactions modeled by documents and the corresponding index file

For a given item, one access suffices to find the list of all transaction IDs that contain the item. The occurrences with transaction IDs are created and simultaneously sorted in ascending order of transaction IDs. Thus, to search for the transaction ID of a particular item, we use a binary search in which, in the worst case, the access time is O(log N), where N is the number of transaction IDs in the occurrences.

D. The Transaction Retrieval Engine:

To search for sensitive transactions in the transactional database, it is necessary to access, manipulate, and query transaction IDs. The transaction retrieval engine performs these tasks. It accepts requests for transactions from a sanitizing algorithm, determines how these requests can be filled (consulting the inverted file), processes the queries using a query language based on Boolean model, and returns the results to the sanitizing algorithm. The process of searching for sensitive transactions through the transactional database works on the inverted file. In general, this process follows three steps: (1) Vocabulary search: each restrictive pattern is split into single items. Isolated items are transformed into basic queries to the inverted index; (2) Retrieval of transactions: The lists of all transaction IDs of transactions containing each individual item respectively are retrieved; and (3) Intersections of

transaction lists: The lists of transactions of all individual items in each restrictive pattern are intersected using a conjunctive Boolean operator on the query tree to find the sensitive transactions containing a given restrictive pattern.

E. Sanitization Algorithms:

Sanitizing algorithms for transactional databases can be classified into two classes as shown in Figure 4, the algorithms that solely remove information from the transactional database and those that modify existing information.

The first algorithms only reduce the support of some items, while the second may increase the support of some items. The following taxonomy of sanitizing algorithms, depicted in Figure 4, relies on the first category.

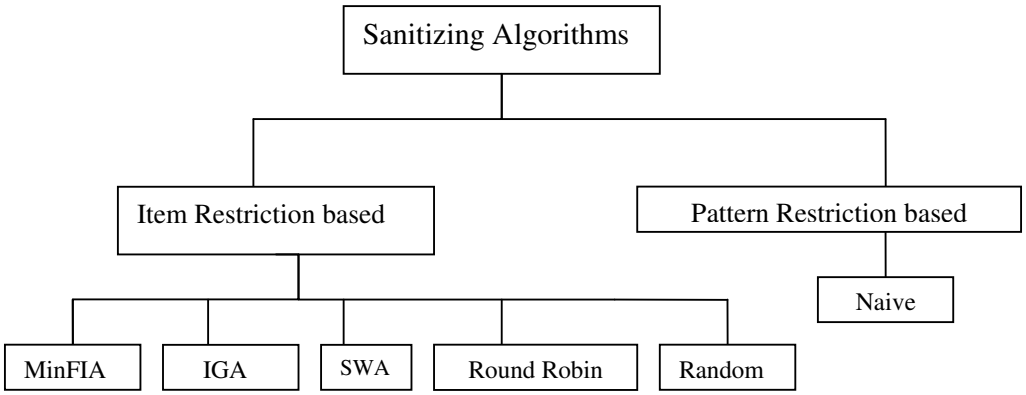

Figure 4: Taxonomy of sanitizing algorithms

Algorithms that solely remove information create a smaller impact on the database since they do not generate artifacts such as illegal association rules that would not exist had the sanitizing not happened. Among the approaches that remove information only we can distinguish the pattern restriction based methods that remove complete restrictive patterns from the sensitive transactions and the item restriction based methods that selectively remove some items from sensitive transactions. The pattern restrictive-based approaches have a bigger impact on the database as more legal patterns end up hidden along with the restricted patterns.

To sanitize a database, each sanitizing algorithm requires an additional scan over the original database D in order to alter some sensitive transactions while keeping the other transactions intact. An initial scan is necessary to build the inverted index.

In most cases, a sensitive transaction contains more than one restrictive pattern. We refer to these transactions as conflicting transactions since modifying one of them causes an impact on other restrictive patterns or even on non-restrictive ones. The degree of conflict of a sensitive transaction is defined as the number of restrictive patterns that can be mined from the sensitive transaction.

To illustrate the presented concepts, let us consider the sample transactional database in Figure 5. Suppose that we have a set of restrictive patterns RP = {ABD, ACD}. This example yields the following results. The sensitive transactions ST containing the restrictive patterns are {T1, T3, T4}. The degrees of conflict for the transactions T1, T3 and T4 are 2, 1 and 1 respectively. Thus, the only conflicting transaction is T1, which covers both restrictive patterns at the same time. An important observation here is that any pattern that contains a restrictive pattern is also a restrictive pattern. Hence, if ABD is a restricted pattern but not ACD as above, the pattern ABCD will also be restrictive since it contains ABD. This is because if ABCD is discovered to be a frequent pattern, it is straight forward to conclude that ABD is also frequent, which should not be disclosed.

All the item restriction-based algorithms have essentially four major steps: (1) Identify sensitive transactions for each restrictive pattern; (2) For each restrictive pattern, identify a candidate item that should be eliminated from the sensitive transactions. This candidate item is called the victim item; (3) Based on the disclosure threshold ψ, calculate for each restrictive pattern the number of sensitive transactions that should be sanitized; and (4) Based on the number found in step 3, identify for each restrictive pattern the sensitive transactions that have to be sanitized and remove the victim item from them.

These sanitizing algorithms mainly differ in step 2 in the way they identify a victim item to remove from the sensitive transactions for each restrictive pattern, and in step 4 where the sensitive transactions to be sanitized are selected. Steps 1 and 3 remain essentially the same for all approaches.

The complexity of these sanitization algorithms in main memory is $O(n_1 N \log N)$, where n_1 is the number of restrictive patterns and N the number of transactions in the database. This is considering the number of items per restrictive pattern relatively small compared to the size of the database. The proof of this is given in (Oliveira & Zaïane 2002).

II. RELATED WORK:

A. Limiting Disclosure of Sensitive Rules [4]:

The authors of paper [4] devised an approach that addresses the security needs in the context of specific type of knowledge, known as association rules, consists of a set of statements of the form " *90% of air-force bases having a super secret plane A, also have helicopters of type B* ". An association rule is

usually characterized by two measures the support, and the confidence. In general association rule mining algorithms discover rules whose support is higher than a minimum threshold value. We refer to such rules as "significant rules". The problem that is discussed in this paper is how to modify a given database so that the support of a given set of sensitive rules, mined from the database, decreases below the minimum support value. The authors have also proved that optimal sanitization is **NP-Hard**. So, a heuristic approach has been proposed to solve the optimal sanitization problem.

B. Sanitization Matrix Method [9]:

This method works by defining a sanitization matrix by observing the relationship between sensitive patterns and non-sensitive patterns. By setting the entries in sanitization matrix to appropriate values and multiplying the original transaction database with the sanitization matrix, we get a sanitized database. The sanitized database is the database which has been modified for hiding sensitive patterns.

A transaction database D is represented as a matrix in which the rows represent transactions and the columns represent the items. If D contains *m* transactions and *n* items D is represented by an $m \times n$ matrix. The entry D_{ti} is set to 1 if item *i* is purchased in transaction *t*. otherwise set to 0.

Let D be transaction database, P be the set of frequent patterns that can be mined from D. Let P_h denote a set of sensitive patterns that need to be hidden according to some security policies, and $Ph \subset P$. $\sim P_h$ is the set of non-sensitive patterns. $\sim P_h \cup P_h = P$. The problem is to transform D into D′ such that only patterns belong to P_h can be mined from D′.

C. Integer Programming Approach [10]:

In this work the authors proposed an exact technique for association rule hiding based on the notion of distance between the original database and its sanitized version, where all sensitive rules have been hidden. By quantifying distance, we gain knowledge of the minimum modification that needs to be made in the original dataset in order to hide sensitive, while minimally affecting non-sensitive, itemsets. An algorithm is formulated based on integer programming, in which distance is the optimization criterion that needs to be minimized. The itemset hiding process is captured as *border revision* operation.

III. OUR CONTRIBUTION:

We propose an algorithm called *non-uniform randomized victim selection* which fits into the category of the item restriction based algorithms shown in Figure 4. In *Uniform random algorithm,* for each sensitive transaction and for restrictive pattern an item is selected as victim randomly. So each item in the restrictive pattern can be selected as a victim with equal probability. We can call it as a uniform randomized victim selection. In non-uniform randomized victim item selection procedure, items with higher support are preferred over items with smaller supports as victim items. The sketch of the *non-uniform randomized victim selection* algorithm is given as follows

Algorithm 1: Non-uniform Random:

```
Input     D, Rp, ψ
Output    D´
Step 1    For each restrictive pattern rpi ∈ Rp do
          T[rpi] = Find_sensitive_transactions(rpi,D)
Step 2    For each restrictive pattern rpi ∈ Rp do
          Victims_rpi = itemv such that itemv ∈ rpi and if there are k
          items in rpi, the item assigned to itemk is biased_random(k)
Step 3    For each restrictive pattern rpi ∈ Rp do
          NumTrans_rpi = |T[rpi]| x (1−ψ)
Step 4    D´ = D
          For each restrictive pattern rpi ∈ Rp do
          a. Sort transactions(T[rpi])
          b. TransToSanitize = Select first NumTrans_rpi
             transactions from T[rpi]
          c. in D´ for each transaction t ∈ TransToSanitize
               t = t − victims_rpi
```

Figure 5

IV. EXPERIMENTAL RESULTS :

To assess the effectiveness of our algorithms, the experiments are conducted on three popular real time datasets *Retail, BMS-Webview-1, BMS-Webview-2* [7]. For each of the three datasets, a set of patterns are randomly chosen to be hidden, and the three algorithms are applied on the original database to hide the given set of restrictive patterns. Accuracy values are calculated for each of the database by applying the three algorithms.

Table 1: Accuracy values for the three algorithms applied on three Datasets:

Accuracy	BMS-WebView-1	BMS-Webview-2	Retail
Round Robin	94.3	84.64	22.73
Uniform Random	94.6	85.43	24.74
Non-uniform Random	96.3	85.43	60.88

V. CONCLUSION:

Privacy becomes an important factor in data mining so that sensitive information is not revealed after mining. However data quality is important such that no false information is released provided privacy is not jeopardized. In this paper we proposed a novel approach for preserving privacy in frequent itemset mining while maintaining accuracy. The proposed approach applies minimum number of changes to the database and minimal amount of non-sensitive itemsets are missed which is the ultimate aim of data sanitization. The experimental results show that the proposed algorithm provides better accuracy than the previous algorithms. Future work has to be carried over to develop optimal algorithms for data sanitization.

REFERENCES:

[1] Evfimievski, A., Srikant, R., Agrawal, R. & Gehrke, J. (2002), Privacy Preserving Mining of Association Rules, in 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AB, Canada, pp. 217(228).

[2] Rizvi, S.J., and Haritsa, J.R.: Maintaining data privacy in association rule mining. In Proceedings of the 28th Conference on Very Large Data Bases. (2002).

[3] Zhang, N., Wang, S., and Zhao, W. 2004. A new scheme on privacy preserving association rule mining. In Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (Pisa, Italy, September 20 - 24, 2004). Pages: 484-495.

[4] Atallah, M., Elmagarmid, A., Ibrahim, M., Bertino, E., and Verykios, V. 1999. Disclosure Limitation of Sensitive Rules. In Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange (November 07 - 07, 1999). KDEX. IEEE Computer Society, Washington, DC, 45.

[5] Jun-Lin Lin, Julie Yu-Chih Liu, Privacy preserving Itemset mining Through Fake Transactions, In proceedings of the 2007 ACM Symposium on Applied Computing, Seoul, Korea, Pages: 375-379.

[6] Agrawal, Shipra and Krishnan, Vijay and Haritsa, Jayant R (2004) On Addressing Efficiency Concerns in Privacy-Preserving Mining. Proceedings 4th International Conference on Database Systems for Advanced Applications: DASFAA '04 (LNCS), pages Vol.2973, 113-124, Jeju Island, Korea.

[7] Liu, K. and Ryan, J. 2006. Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining. IEEE Trans. on Knowledge and Data Eng. 18, 1 (Jan. 2006), 92-106.

[8] Jin-Lang Wang, Cong-fu Xu, and Yun-He Pan 2006. An Incremental Algorithm for mining privacy preserving frequent Itemsets. Proceedings of Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August, 2006.

[9] Chen, T. 2006. A Novel Method for Protecting Sensitive Knowledge in Association Rules Mining. In Proceedings of the Sixth international Conference on intelligent Systems Design and Applications (ISDA'06) - Volume 01 (October 16 - 18, 2006). ISDA. IEEE Computer Society, Washington, DC, 694-699.

[10] Aris Gkoulalas-Divanis, Vassilios S. Verykios. An integer programming approach for frequent itemset hiding, ACM, CIKM'06, November 5-11 2006, Arlington, Virginia, U.S.A.

[11] S. Oliveira and O. Zaïane. Privacy preserving frequent itemset mining. CRPITS'14: Proceedings of the IEEE International Conference on Privacy, Security, and Data Mining, pages 43–54, 2002.

[12] Oliveira, S. R. and Zaïane, O. R. 2003. Protecting Sensitive Knowledge by Data Sanitization. In Proceedings of the Third IEEE international Conference on Data Mining (November 19 - 22, 2003). ICDM. IEEE Computer Society, Washington, DC, 613.

[13] Oliveira, S. R. and Zaïane, O. R. 2003. Algorithms for balancing privacy and knowledge discovery in association rule mining. In Proceedings of the IEEE seventh International Database Engineering and Applications Symposium, 16-18 July 2003, Hong Kong, China. Pages 54-63.

[14] Y. Saygin, V. S. Verykios, and C. Clifton. Using Unknowns to Prevent Discovery of Association Rules. SIGMOD Record, 30(4):45–54, December 2001.

[15] http://fimi.cs.helsinki.fi/data/.

