IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 14, Issue 4 (Sep. - Oct. 2013), PP 01-05
www.iosrjournals.org

An improved Item-based Maxcover Algorithm to protect Sensitive Patterns in Large Databases

Mrs. P.Cynthia Selvi[1], Dr.A.R.Mohamed Shanavas[2]
1. Associate Professor, Dept. of Computer Science, KNGA College(W), Thanjavur 613007 /Affiliated to Bharathidasan University, Tiruchirapalli, TamilNadu, India.
2. Associate Professor, Dept. of Comuter Science, Jamal Mohamed College, Tiruchirapalli 620 020/ Affiliated to Bharathidasan University, Tiruchirapalli, TamilNadu, India.

Abstract: *Privacy Preserving Data Mining(PPDM) is a rising field of research in Data Mining and various approaches are being introduced by the researchers. One of the approaches is a sanitization process, that transforms the source database into a modified one by removing selective items so that the counterparts or adversaries cannot extract the hidden patterns from. This study address this concept and proposes a revised Item-based Maxcover Algorithm(IMA) which is aimed at less information loss in the large databases with minimal removal of items.*

Keywords: *Privacy Preserving Data Mining, Restrictive Patterns, Sensitive Transactions, Maxcover, Sanitized database.*

I. Introduction

PPDM is a novel research direction in DM with the main objective to develop algorithms for modifying the original data in some way, so that the private data and private knowledge remain private even after the mining process[1]. This modification is done by deleting one or more items from source database or even adding noise to the data by turning some items from 0 to 1 in some transactions. The procedure of transforming the source database into a new database that hides some sensitive patterns or rules is called the *sanitization process*[2] and the released database is called the *sanitized database.* The approach of modifying some data is perfectly acceptable in some real applications[3, 4] provided the impact on the source database is kept minimal.

This study mainly focus on the task of minimizing the impact on the source database by reducing the number of removed items from the source database with only one scan. Section-2 briefly summarizes the previous work done by various researchers; In Section-3 preliminaries and basic definitions are discussed. In Section-4 the proposed algorithm is presented and Section-5 shows the experimental results.

II. Related work

The idea behind data sanitization to reduce the support values of restrictive patterns was first introduced by Atallah et.al[1] and they have proved that the optimal sanitization process is NP-hard problem. In [4], the authors generalized the problem and proposed algorithms that ensure privacy preservation; but they require multiple scans over a transactional database. In the same direction, Saygin [5] introduced a method for selectively removing individual values from a database and proposed some algorithms to obscure a given set of sensitive rules by replacing known values with unknowns which also require various scans to sanitize a database depending on the number of association rules to be hidden.

Oliveira introduced algorithms, IGA[6] & SWA[7] that aims at multiple rule hiding in which IGA has low misses cost; It groups restrictive patterns and assigns a victim item to each group but the clustering is not performed optimally and it can be improved further by reducing the number of deleted items. Whereas, SWA improves the balance between protection of sensitive knowledge and pattern discovery but some rules are removed inadvertently.

In [8] heuristic-based approach is proposed which perform the sanitization process with minimum number of removal of restricted items. However, when the sensitive patterns to be hidden are mutually exclusive, this approach has more hiding failure. Hence, in this work the algorithm proposed in [8] is revised and it is tested with large database which ensure no hiding failure and very low information loss.

III. Preliminaries & Definitions

Transactional Database: A transactional database is a relation consisting of transactions in which each transaction *t* is characterized by an ordered pair, defined as *t* = <*Tid, list-of-elements*>, where *Tid* is a unique transaction identifier number and *list-of-elements* represents a list of items making up the transactions.

Basics of Association Rules: One of the most studied problems in data mining is the process of discovering association rules from large databases. Most of the existing algorithms for association rules rely on the support-confidence framework introduced in [9].

Formally, association rules are defined as follows: Let $I = \{i_1,...,i_n\}$ be a set of literals, called items. Let D be a database of transactions, where each transaction t is an itemset such that $t \subseteq I$. A transaction t supports X, a set of items in I, if $X \subset t$. An association rule is an implication of the form $X \Longrightarrow Y$, where $X \subset I$, $Y \subset I$ and $X \cap Y = \phi$. Thus, we say that a rule $X \Longrightarrow Y$ holds in the database D with *support* (σ) if $\frac{|X \cup Y|}{N} \geq \sigma$, where N is the number of transactions in D. Similarly, we say that a rule $X \Longrightarrow Y$ holds in the database D with *confidence* (φ) if $\frac{|X \cup Y|}{|X|} \geq \varphi$, where $|A|$ is the number of occurrences of the set of items A in the set of transactions D. While the support is a measure of the frequency of a rule, the confidence is a measure of the strength of the relation between sets of items. Association rule mining algorithms rely on the two attributes, *minimum Support(minSup σ)* and *minimum Confidence(minConf φ)*.

***Frequent Pattern*:** A pattern X is called a frequent pattern if *Sup(X)* ≥ *minSup* or if the *absolute support* of X satisfies the corresponding *minimum support count* threshold.

***Privacy Preservation in Frequent Patterns*:** The task of privacy preserving data processing deals with removing/modifying sensitive entries in the data which are basically decided by the user, who may either be the owner or the contributor of the data.

Definitions :

Definition 1: Let D be a source database, containing a set of all transactions. T denotes a set of transactions, each transaction containing itemset $X \in D$. In addition, each k-itemset $X \subseteq I$ has an associated set of transactions $T \subseteq D$, where $X \subseteq t$ and $t \in T$.

Definition 2: *Restrictive Patterns* : Let D be a source 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 $\sim R_P \cup R_P = P$.

Definition 3 : *Sensitive Transactions* : Let T be a set of all transactions in a source database D, and R_P be a set of restrictive patterns mined from D. A set of transactions is said to be *sensitive*, denoted by S_T, if every $t \in S_T$ contain atleast one restrictive pattern, ie $S_T = \{ t \in T \mid \exists X \in R_P, X \subseteq t \}$. Moreover, if $S_T \subset T$ then all restrictive patterns can be mined one and only from S_T.

Definition 4 : *(i) Transaction Size :* The number of items which make up a transaction is the size of the transaction.

***(ii) Transaction Degree* :** Let D be a source database and S_T be a set of all sensitive transactions in D. The *degree of a sensitive transaction t*, denoted as *deg(t),* such that $t \in S_T$ is defined as the number of restrictive patterns that *t* contains.

Definition 5: *Cover* : The *Cover[8]* of an item A_k can be defined as, $C_{Ak} = \{ rp_i \mid A_k \in rp_i \subset R_P, 1 \leq i \leq |R_P| \}$ *i.e.,* set of all restrictive patterns(rp_i) which contain A_k. The item that is included in a maximum number of rp_i is the one with *maximal cover or maxCover; i.e., maxCover = max*($|C_{A1}|, |C_{A2}|, \ldots |C_{An}|$) such that $A_k \in rp_i \subset R_P$.

IV. Sanitization Algorithm

Given the source *database (D),* and the *restrictive patterns(R_P)*, the goal of the sanitization process is to protect R_P against the mining techniques used to disclose them. The sanitization process decreases the support values of restrictive patterns by removing items from sensitive transactions using some heuristics. The heuristics used in this work is given below:

Heuristic : Sensitive Transactions(S_T) in source database(D) are identified and sorted in decreasing order of (*deg + size)*, which enable multiple patterns to be sanitized in a single iteration. Then the victim item is the one which is selected based on *maxcover* (*definition-5*) value of the items in the restrictive patterns.

Algorithm

Input : (i) D – Source Database (ii) R_P – Set of all Restrictive Patterns

Output : D' – Sanitized Database

***Item-based Maxcover Algorithm(IMA)*:** // based on Heuristics 1 & 2 //

Step 1 : (i) find *supCount*(A_k) $\forall A_k \in rp_i \subset R_P$

(ii) find *supCount*(rp_i), $\forall rp_i \in R_P$ and sort in decreasing order ;

Step 2 : (i) find *Sensitive Transactions(S_T)* w.r.t. R_P;

(ii) obtain *deg(t), size(t)* $\forall t \in S_T$;

(iii) sort $t \in S_T$ in decreasing order of *deg & size ;*

Step 3 : find $\sim S_T \leftarrow D - S_T$; // $\sim S_T$ - non sensitive transactions //
Step 4 : // Find S_T' //
find *cover* for every item $A_k \in R_P$ and sort in decreasing order of *cover;*
for each item $A_k \in R_P$ do
{
find T = $\bigcap_{i=1}^{|rpi-list|} t$
for each $t \in T$ do
{
delete item A_k in *non VictimTransactions* such that $A_k \in rp_i \subset rp_i$-*list* ; // A_k – *victimItem* //
// initially all *t* are *nonvictim* //
decrease *supCount* of rp_i's for which *t* is *nonVictim*;
mark *t* as *victimTransaction* in each *t-list*(rp_i) $\subset$ rp_i _*list*(A_k) ;
}
}
for each $rp_i \in R_P$ do
{
if (supCount < > 0)
{
for every $t \in$ *nonVictimTransactions* do
{
delete item A_k with *minimum supCount* (*round robin* in case of tie);
decrease *supCount* of every rp_i ; // *ie,* $A_k \in rp_i \subset t$ //
}
}
}
Step 5 : D' ← $\sim S_T \cup S_T'$

V. Implementation

The test run was performed using AMD Turion II N550 Dual core processor with 2.6 GHz speed and 2GB RAM operating on 32 bit OS; The implementation of the proposed algorithm was done with windows 7 - Netbeans 6.9.1 - SQL 2005 for real dataset *T10I4D100K*[10] with characteristics given in table-I. The results are obtained by varying the number of rules and also size of the source database in terms of number of transactions (refer graphs). The restrictive patterns are chosen under various categories like overlapping, mutually exclusive, random, high support, low support with their support ranging between 0.6 and 5, confidence between 32.5 and 85.7 and length between 2 and 6.

The frequent patterns were obtained using Matrix Apriori[11] which uses simpler data structures for implementation. The proposed algorithm makes use of preprocessed lookup(hashed) tables which links the restrictive items and rules with their associated transactions; so that the source database is scanned not more than once. Our algorithm is evaluated based on the performance measures suggested in [6,7].

Performance Analysis

Hiding Failure(HF) : It is measured by the ratio of the number of restrictive patterns in the released sanitized database(D') to the ones in the given source database; HF = $\frac{|RP(D')|}{|RP(D)|}$. In other words, if a hidden restrictive pattern cannot be extracted from the released database D' with an arbitrary *minSup*, then it has no hiding failure occurrence. Our approach has 0% HF for all the strategies under which the rules to be hidden were chosen.

Misses Cost(MC) : This measure deals with the legitimate patterns(non restrictive patterns) that were accidently missed. MC = $\frac{|\sim RP(D)| - |\sim RP(D')|}{|\sim RP(D)|}$. Our approach has very minimum MC which ranges between 0% and 2.43%. It is also observed that MC gets reduced linearly when the size of database is increased.

Artifactual Pattern(AP) : AP occurs when D' is released with some artificially generated patterns after applying the privacy preservation approach and it is given by, AP = $\frac{|P'| - |P \cap P'|}{|P'|}$. As our approach does not introduce any false drops, the AP is 0%.

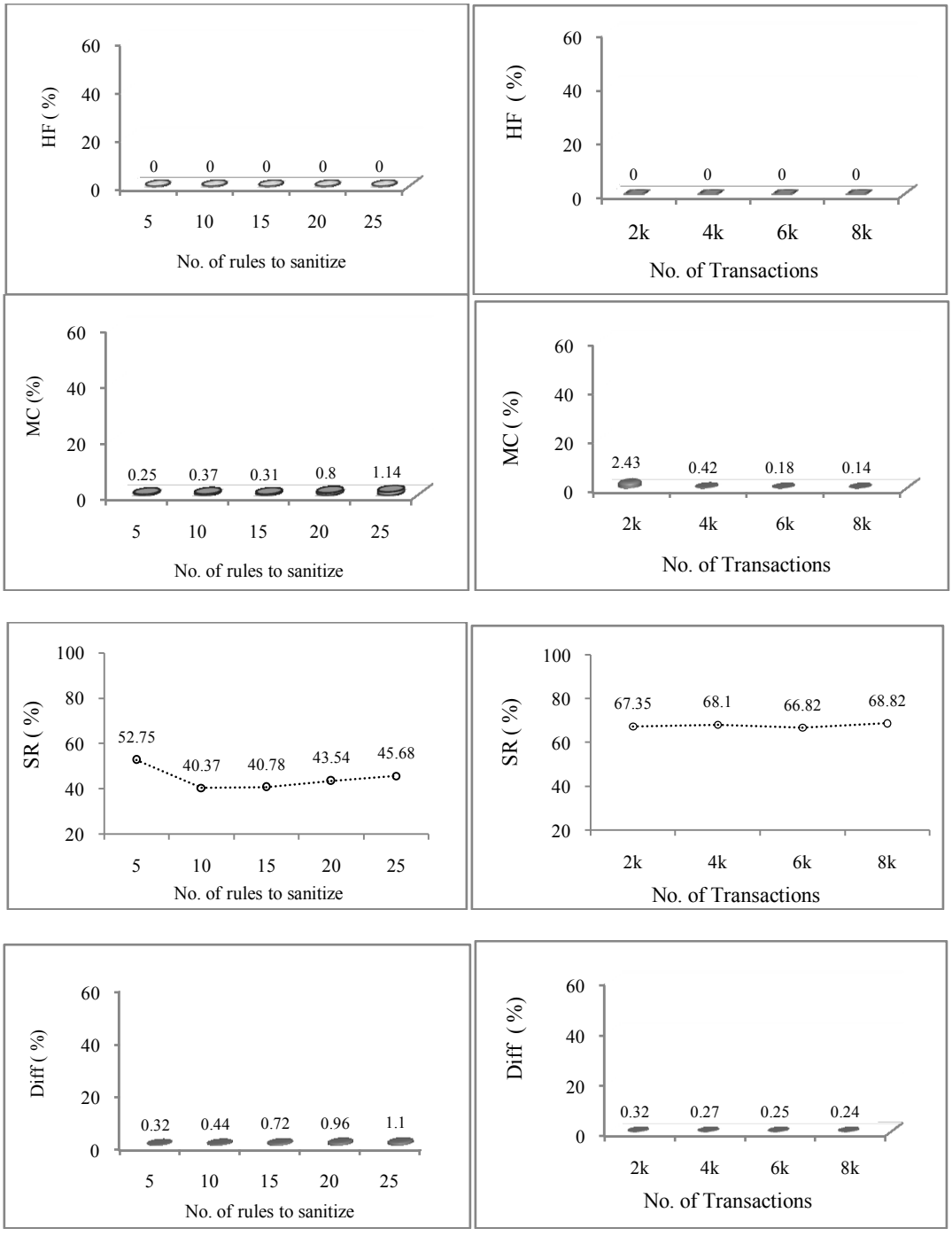
Sanitization Rate(SR) : It is defined as the ratio of the number of selectively deleted items(*victim items*) to the total support count of restrictive patterns(rp_i) in the source database D and is given by, SR = $\frac{|\text{victim items}|}{\text{total supCount (rpi)}}$.

Here the SR is found to be in the range from 40.32% to 68.82%, which shows that the number of restrictive items deleted from the source database is kept minimal.

Dissimilarity(dif) : The dissimilarity between the original(D) and sanitized(D') databases is measured in terms of their contents which can be measured by the formula, $dif(D,D') = \frac{1}{\sum_{i=1}^{n} fD(i)} \times \sum_{i=1}^{n} [fD(i) - fD'(i)]$, where *fx(i)* represents the i^{th} item in the dataset *X*. Our approach has very low percentage of dissimilarity that ranges between 0.24% and 1.1% and this gets reduced when size of the source database is increased. Hence it is observed that information loss is very low and so the utility of the data is well preserved.

CPU Time : The range of time requirement for this algorithm with the rules chosen under various categories is between 1.66 and 17.22 sec. This time requirement can still be reduced when parallelism is adapted with high speed processor. Moreover, time is not a significant criterion as sanitization is performed offline.

Graphs :



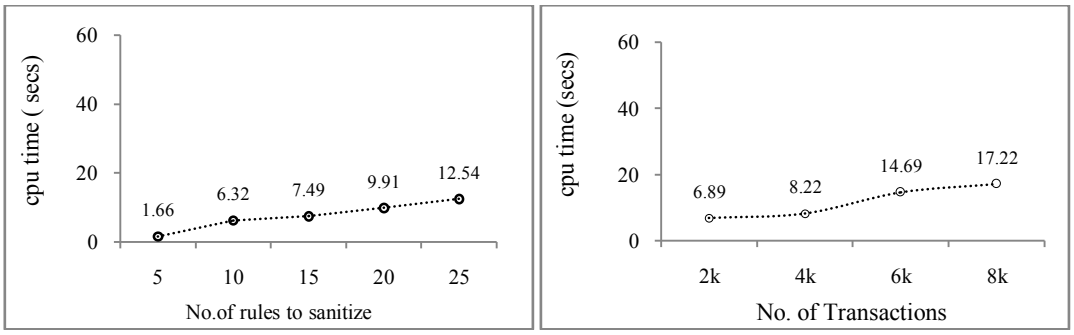


Table – I :Characteristics of Dataset (D & D')

Dataset name T10I4D100K	No. of transactions	No. of distinct Items	Min. Length	Max. Length	Dataset Size (MB)
Source (D)	1K – 8K	795 - 862	1	26	97KB – 650 KB
Sanitized (D')	1K – 8K	795 - 862	1	26	95KB – 648KB

VI. Conclusion

The proposed Item-based Maxcover Algorithm(IMA) is based on the strategy to simultaneously decrease the support count of maximum number of sensitive patterns, with possibly minimal removal of items that results in reduced impact on the source database with no hiding failure. IMA has reduced sanitization rate possibly with very low misses cost. Moreover this algorithm scans the original database only once. It is important to note that the proposed algorithm is robust in the sense that reconstruction of the source database is not at all possible; because the alterations to the original database are not saved anywhere and also encryption techniques are not involved in this approach. The dissimilarity between the source and sanitized database is very minimum that minimizes the information loss and preserves improved data utility. The execution time is also significantly low which can still be reduced to a minimum by parallelism techniques adapted with high speed processors.

References

[1] Verykios,V.S, Bertino.E, Fovino.I.N, ProvenzaL.P, Saygin.Y and Theodoridis.Y, "State-of-the-art in Privacy Preservation Data Mining", New York,ACM SIGMOD Record, vol.33, no.2, pp.50-57,2004.

[2] Atallah.M, Bertino.E, Elmagarmid.A, Ibrahim.M and Verykios.V.S, "Disclosure Limitation of Sensitive Rules", In *Proc. of IEEE Knowledge and Data Engineering Workshop*, pages 45–52, Chicago, Illinois, November 1999.

[3] Clifton.C and Marks.D, " Security and Privacy Implications of Data Mining", In *Workshop on Data Mining and Knowledge Discovery*, pages 15–19, Montreal, Canada, February 1996.

[4] Dasseni.E, Verykios.V.S, Elmagarmid.A.K and Bertino.E, " Hiding Association Rules by Using Confidence and Support", In *Proc. of the 4th Information Hiding Workshop*, pages 369– 383, Pittsburg, PA, April 2001.

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

[6] Oliveira.S.R.M and Zaiane.O.R, "Privacy preserving Frequent Itemset Mining", in the Proc. of the IEEE ICDM Workshop on Privacy, Security, and Data Mining, Pages 43-54, Maebashi City, Japan, December 2002.

[7] Oliveira.S.R.M and Zaiane.O.R, "An Efficient One-Scan Sanitization for Improving the Balance between Privacy and Knowledge Discovery", Technical Report TR 03-15, June 2003.

[8] Cynthia Selvi P, Mohamed Shanavas A.R, "An Effective Heuristic Approach for Hiding Sensitive Patterns in Databases", IOSR-Journal on Computer Engineering, Volume 5, Issue 1(Sep-Oct, 2012), PP 06-11, DOI. 10.9790/0661-0510611.

[9] Agrawal R, Imielinski T, and Swami.A, "Mining association rules between sets of items in large databases", Proceedings of 1993 ACM SIGMOD international conference on management of data, Washington, DC; 1993. p. 207-16.

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

[11] Pavon.J, Viana.S, and Gomez.S, "Matrix Apriori: speeding up the search for frequent patterns," Proc. 24th IASTED International Conference on Databases and Applications, 2006, pp. 75-82.