Knowledge Protection by Subjective Measure

Cynthia Selvi P
Associate Professor, Dept. of Computer Science
K.N.Govt.Arts College for Women(Autonomous)
Thanjavur, TamilNadu, India

Mohamed Shanavas A R, Ph.D
Associate Professor, Dept. of Computer Science
Jamal Mohamed College(Autonomous)
Trichirapalli, TamilNadu, India

ABSTRACT
Basically a Data Mining system would generate thousands or even millions of patterns or rules. However all the generated patterns would not actually be interesting to any given user; In fact the interestingness of the patterns would be assessed only on the users' beliefs and expectations which is rather termed as subjective measure. When such interesting patterns are to be shared in a collaborative business environment, it would be more meaningful to restrict them based on the significance of individual items in the patterns to be protected. Hence, this work attempts to hide interesting patterns on the subjective measure and propose an algorithm which is tested for its effectiveness.

Keywords
Subjective measure, Restrictive patterns, Sensitive transactions, Maxcover, Sanitization.

1. INTRODUCTION
A data mining system can uncover thousands of patterns of which interesting patterns represent knowledge. A pattern is interesting to a user if it is potentially useful and novel, or if it validates a hypothesis that the user sought to confirm. Measures of pattern interestingness are essential for the efficient discovery of patterns of value to the given user[1]. Whether or not a pattern or rule is interesting can be assessed either objectively or subjectively.

Several objective measures of pattern interestingness exist. These are based on the structure of discovered patterns and the statistics underlying them. Basically used objective measure for association rules of the form x $\Longrightarrow$ y are *support* representing the percentage of transactions from a transaction database that the given rule satisfies and *confidence* which assess the degree of certainty of the detected association. In general, each objective measure is associated with a threshold, which may be controlled by the user and the rules that satisfy the threshold can be considered interesting; the rest are treated as uninteresting and are probably of less value.

Although objective measures help identify interesting patterns, they are insufficient unless combined with subjective measures that reflect the needs and interests of a particular user. Furthermore, many patterns that are interesting by objective standards may represent common knowledge and therefore are actually uninteresting. Subjective measures of interestingness are based on user beliefs in the data. These measures find patterns interesting if they offer strategic information on which the user can act or if they confirm a hypothesis that the user wished to validate or resemble a user's hunch[1]. This study makes an attempt to introduce an algorithm for knowledge protection by combining both subjective measure and objective measure.

In this article, section-2 shortly covers the previous work; section-3 states basic concepts and definitions(original contribution on which the proposed algorithm is designed) section-4 illustrates the proposed algorithm and section-5 visually presents the experimental results.

2. LITERATURE SURVEY
Recently, researchers have been concentrating on the concept of protecting sensitive knowledge in association rule mining; Many approaches have been proposed so far that includes randomization, data partition, and data sanitization. The underlying principle of data sanitization that reduce the support values of restrictive rules was initially introduced by Atallah et.al[2] and they proved that the optimality in sanitization process is NP-hard problem. In [3], the authors proposed some generalized approach that hides both sensitive frequent itemsets and sensitive rules; but they are CPU-intensive due to the requirement of multiple scans over a transactional source database. Similarly, Saygin [4] proposed an approach for selective removal of individual values by replacing the known values with unknowns that reduces the side effects on non-sensitive rules but need multiple scans over source database depending on the number of association rules to be protected.

The algorithms IGA and SWA introduced in [5&6] respectively are aimed at multiple rule hiding. However, IGA has low misses cost but it assigns a victim item to every cluster of restrictive rules and this clustering is not optimally handled which leads to overlapping. Whereas, SWA is aimed to improve the balance between privacy protection and accuracy of pattern discovery but with an extra cost, as it involve inadvertent removal of some rules. A heuristic based frequent itemset hiding algorithm is proposed by [7] which eliminate pre-mining; but it is restricted for smaller databases and sensitive itemsets that are mutually exclusive.

Almost all the proposed approaches are aimed at sanitizing multiple rules but with the consideration of objective measures(support-confidence framework). In the proposed work, a heuristic based approach is focused which protect sensitive knowledge using subjective measure by associating a sensitivity cost for the individual items that are to be restricted.

3. PRELIMINARIES & DEFINITIONS
Transactional Database : A transactional database consists of a file where each record represents a transaction that typically includes a unique identity number (*trans_id*) and a list of items that make up the transaction.

Association Rule : It is an expression of the form $X \Longrightarrow$ Y, where X and Y contain one or more itemsets(categorical values) without common elements ($X \cap Y = \phi$).

Frequent Pattern : An itemset or pattern that forms an association rule is said to be frequent if it satisfies a prespecified minimum support threshold(*min_sup*).

Transactional Database : Let D be a source database which is a transactional database containing a set of transactions T, where each transaction *t* contain an itemset $X \in D$. Also, every $X \subseteq I$ has an associated set of transactions $T \subseteq D$, where $X \subseteq t$ and $t \in T$.

Restrictive Patterns : Let P be a set of significant patterns that can be mined from transactional source database D, and R_H be a set of rules to be hidden according to some privacy policies. A set of all patterns rp_i 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 R_H. $\sim R_P$ is the set of *non-restrictive patterns* such that $\sim R_P \cup R_P = P$ [4].

Sensitive Transactions : A set of transactions is said to be *sensitive*, denoted by S_T, if every $t \in S_T$ contain atleast one restrictive pattern rp_i . ie $S_T = \{ t \in T \mid \exists\, rp_i \in R_P, rp_i \subseteq t \}$.

Null Transactions : A set of transactions is said to be *null transactions* if they do not contain any of the patterns being examined[1].

Transaction Size : The number of items that make up a transaction is the size of the transaction.

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[5].

Definition 1: Cover: The *Cover*[8] of an item A_k can be defined as, $C_{Ak} = \{ rp_i \mid A_k \in rp_i \subset R_P, 1 \le i \le |R_P| \}$

i.e., set of all restrictive patterns which contain A_k.
The item that is included in a maximum number of rp_i's is the one with *maximal cover or maxCover;*

i.e., maxCover = max($|C_{A1}|, |C_{A2}|, \ldots |C_{An}|$)

such that $A_k \in rp_i \subset R_P$.

Definintion 2 : Sensitivity Cost - It is a user-defined privacy sensitivity value associated with the individual item of the sensitive patterns or rules. (Only Boolean values are associated in this approach; however values ranging from minimum and maximum range can also be assigned).

4. SANITIZATION ALGORITHM

Given D a transactional *source database*, and R_P the *restrictive patterns* chosen based on some decision making policies, the proposed algorithm removes the restrictive item with sensitivity cost of true value in order to protect it against the mining techniques used to disclose them. The heuristic used in the proposed algorithm is given below:

Heuristic : The individual items in the restrictive patterns are associated with a boolean cost vector; for every transaction *t* of $rp_i \in R_P$, select a *victim item* A_k with cost = 1 and *maximal cover* within *t* such that $A_k \in rp_i \subset t$; In case of tie, choose one in *round robin*.

4.1. Sensitivity Cost Sanitization(SCS) Algorithm

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

Algorithm :
Step 1 : calculate *supCount*(rp_i) $\forall\, rp_i \in R_P$ and sort in decreasing order ;
Step 2 : a) obtain *Sensitive Transactions*(S_T) w.r.t. R_P ;
b) find *deg(t), size(t)* $\forall\, t \in S_T$;
c) sort $t \in S_T$ in decreasing order of *deg & size ;*
Step 3 : filter $\sim S_T \leftarrow D - S_T$; //$\sim S_T$ - null transactions //

Step 4 : // Find S_T' – sanitized transactions //
for each $rp_i \in R_P$ do
{
extract S_{Trpi} ;
find nTs; //nTs - *no. of transactionToSanitize* = |*nonVictimTransactions*|)] //
repeat //initially all *t* are *nonvictim* //
for each $t \in nonVictimTransactions$
{
identify all A_k with cost = 1 such that $A_k \in rp_i \subset t$;
{
find *cover* for each item A_k
delete A_k with *maxCover* (*round robin* in case of tie);
// A_k – *victimItem* //
decrease *supCount* of all rp_i*'s* which contain *victimItem;*
// $A_k \in rp_i \subset t$//
mark *t* as *victimTransaction* w.r.t each rp_i ;
}
}
until (*supCount* = 0) ;
}

Step 5 : D' ← $\sim S_T \cup S_T'$

4.2. Illustration

The following example illustrate the principle used in the proposed algorithm.

Table 1. Source Database-D

Tid	Itemset
T1	A,B,C,E,F
T2	B,C,D
T3	A,C,D,F
T4	A,E
T5	B,C,D,E
T6	A,C,D,E

Table 2. Restrictive Patterns-R_P

Pid	Pattern	SupCount
P1	A,C	3
P2	C,D	4

Table 3. Sensitivity Cost of Restrictive Items

Item	Patterns	Cost
C	P1, P2	1
D	P2	1
A	P1	0

Here, T1, T3 & T6 are transactions that contain the pattern P1 and T2, T3, T5 & T6 contain P2.

For the pattern P1, |*cover*| of both A & C in T1 is 1; However C would be the victim item as the *sensitivity cost* of C=1. So the *SupCount*(P1) is reduced by 1. In T3 & T6, |*cover*(A)|=1 and |*cover*(C)|=2; so C (with *cost*=1) is *victim* and *SupCount* of both P1 & P2 are reduced.

For the pattern P2, transactions T3 & T6 are already visited in the previous iterations; In the remaining transactions, T2 & T5 |*cover*| and *sensitivity cost* of both C & D is 1; Using round robin, C & D is *victim* in T2 & T5 respectively.

The set of modified sensitive transactions are denoted as S_T' and the Sanitized database D' is formed by merging $\sim S_T$ and S_T'.

Table 4. Sanitized Database-D'

Tid	Itemset
T1	A,B,E,F
T2	B,D
T3	A,D,F
T4	A,E
T5	B,C,E
T6	A,D,E

5. EXPERIMENTAL RESULTS

The algorithm was executed for the restrictive patterns chosen at random(5 nos. with their support ranging between 0.6 and 5, confidence between 32.5 and 85.7 and length between 2 and 6) and by varying the number of transactions using the real dataset *T10I4D100K*[9]; The test run was made on 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 6.9.0 - SQL 2005. The performance issues are studied based on the metrics suggested in [5].

The frequent patterns were obtained based on the logic proposed in [10] 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; the transactions visited in the previous iterations are dynamically updated; so that redundancy is completely avoided.

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, which is given by,

HF = $\frac{|RP(D')|}{|RP(D)|}$. This approach has 0% HF (Fig.1).

Misses Cost(MC) : This measure deals with the legitimate patterns(non restrictive patterns) that were accidently missed. MC = $\frac{|\sim RP(D)| - |\sim RP(D')|}{|\sim RP(D)|}$. In this approach MC is obtained to be very minimum ranging between 0% and 0.35% (Fig.2).

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 this approach does not introduce any false drops, the AP is 0%.

Sanitization Rate(SR) : It is defined as the ratio of the 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 = $\frac{|\text{victim items}|}{\text{total supCount(rpi)}}$ and it is found to be less than 65% (Fig.3), 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*. This approach has very low percentage of dissimilarity that ranges between 0.23% and 0.31% (Fig.4). This shows that information loss is very low and so the utility is well preserved.

CPU Time : The execution time is shown in the graph (Fig.5) and it can be observed that this approach has very good scalability. This time requirement can further be reduced by adapting parallelism. However, time is not a significant criteria as the sanitization is done offline.

Graphs :

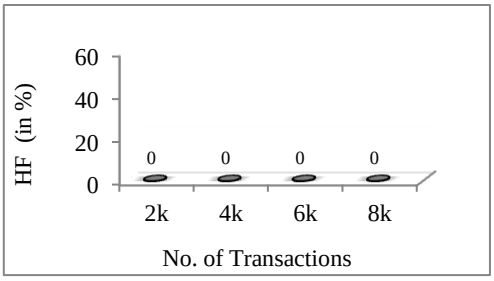

Fig 1: Hiding Failure

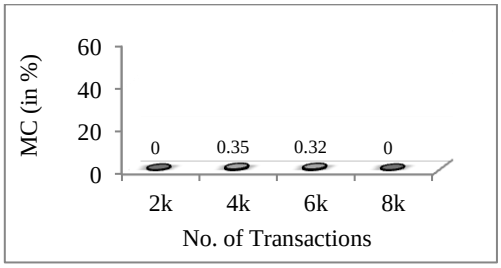

Fig 2: Misses Cost

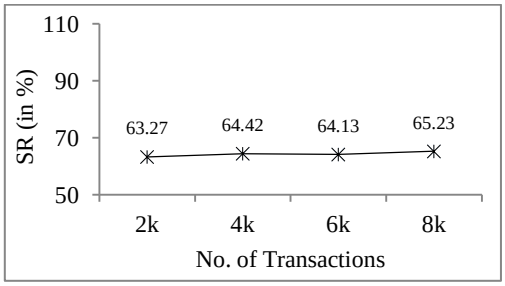

Fig 3: Sanitization Rate

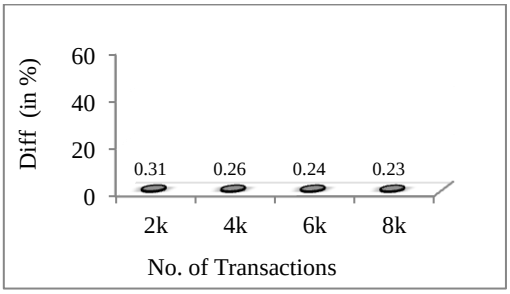

Fig 4: Dissimilarity

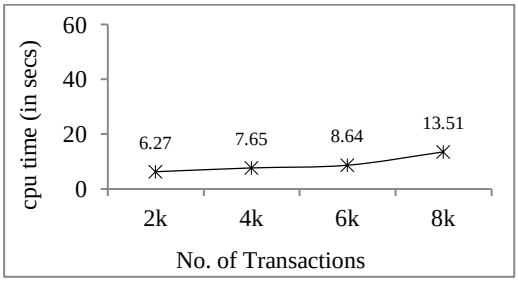

Fig 5: CPU Time

6. CONCLUSION

Many works have been proposed for promoting knowledge protection in data mining. Specifically in this work a heuristic approach using objective measure combined with subjective measure has been initiated to preserve the sensitive items; Moreover this approach also has other promising features that it has no hiding failure and the sanitization process is performed with a minimal removal of items. The proposed algorithm requires only a single scan of the source database for sanitization which is due to the use of preprocessed hashed lookup tables. The dissimilarity of the source and released database is found to be minimal which ensure very low information loss. As no encryption is involved, reconstruction of the source from the released one is not at all possible. Also this approach attempts to hide multiple patterns or rules simultaneously.

7. REFERENCES

[1] Han J, and Kamber M, "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers – Reprint 2011.

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

[3] Dasseni E, Verykios V.S, Elmagarmid A.K & 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.

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

[5] 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.

[6] 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.

[7] Yildz B, and Ergenc B, "Hiding Sensitive Predictive Frequent Itemsets", Proceedings of the International MultiConference of Engineers and Computer Scientists 2011, Vol-I.

[8] Cynthia Selvi P, Mohamed Shanavas A.R, "An Improved Item-based Maxcover Algorithm to Protect Sensitive Patterns in Large Databases", IOSR-Journal on Computer Engineering, Volume 14, Issue 4 (Sep-Oct, 2013), PP 01-05, DOI. 10.9790/0661-1440105.

[9] The Dataset used in this work for experimental analysis was generated using the generator from IBM Almaden Quest research group and is publicly available from http://fimi.ua.ac.be/data/.

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