Data hiding knowledge by modestly Extending Database

1N.Pandeeswari, 2Dr.P.Ganesh Kumar, 3R.sivakami
1 Assistant Professor
2Professor
3Associate professor
PSNA college of Engineering and Technology, Dindigul
1pandeeswari@psnacet.edu.in
2drpganeshkumar@gmail.com
3rsivakami@psnacet.edu.in

Abstract-**Sharing data among organizations often leads to mutual benefit. Recent technology in data mining has enabled efficient extraction of knowledge from large databases. This, however, increases risks of disclosing the sensitive knowledge when the database is released to other parties. To address this privacy issue, one may sanitize the original database so that the sensitive knowledge is hidden. Sensitive knowledge hiding in large transactional databases is one of the major goals of privacy preserving data mining. Exact solutions for the hiding of knowledge, depicted in the form of sensitive frequent item sets and their association rules, were identified. A novel approach that performs frequent item set, from that sensitive kn is hided, through which the database is extended Knowledge.**

I.INTRODUCTION

Data mining is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories. Data Mining is the nontrivial extraction of implicit previously unknown and potentially useful information from data. Other terms similar to data mining are knowledge mining from databases, knowledge extraction, data / pattern analysis, data archaeology, and data dredging. Data mining is a synonym for Knowledge Discovery in Databases or KDD . Data mining is also viewed as simply an essential step in the process of knowledge discovery in databases. aggregation operations, for instance.

Data mining involves an integration of techniques from multiple disciplines such as database technology, statistics, machine learning , high-performance computing, pattern recognition, neural networks, data visualization, information retrieval , image and signal processing and spatial data analysis. By performing data mining, interesting knowledge, regularities, or high level information can be extracted from databases and viewed or browsed from different angles. The discovered knowledge can be applied to decision making, process control, information management and query processing. Therefore data mining is considered one of the most important frontiers in database systems and one of the most promising interdisciplinary developments in the information industry. In this proposed paper, we are using hybrid approach to frequent item set mining.

II.KNOWLEDGE HIDING

Advances in data collection, processing, and analysis, along with privacy concerns regarding the misuse of the induced knowledge from this data, soon brought into existence the field of privacy preserving data mining[1]. Simple de-identification of the data prior to its mining is insufficient to guarantee a privacy-aware outcome since intelligent analysis of the data, through inference based attacks, may reveal sensitive patterns that were unknown to the database owner before mining the data. Thus, compliance to privacy regulations requires the incorporation of advanced and sophisticated solutions.

A subfield of privacy preserving data mining, known as “knowledge hiding” [6] prevent disclosure of both confidential personal information from summarized data and of sensitive knowledge that can be mined from this data. We present a novel approach that strategically performs sensitive frequent item set hiding based on a new notion of hybrid database generation. In this approach, data sanitization is performed by applying an extension[2] to the original database instead of either modifying existing transactions (directly or through the application of transformations) or rebuilding the data set from scratch to accommodate knowledge hiding. The extended portion of the data set contains a set of carefully crafted transactions that achieve to lower the importance of the sensitive patterns, while minimally affecting the importance of the nonsensitive ones. The released sanitized [4] database, which consists of the initial part (original database) and the extended part (database extension), can guarantee the protection of the sensitive knowledge, when mined at the same or higher support as the one used in the original database.

Given a database D containing transactions T and a minimum support threshold *msup* set by the owner of the data. A subset SI of the frequent items F, discovered in D, are considered as sensitive and must be protected from being disclosed to unauthorized parties. The goal of the hiding algorithm is to create a minimal extension to the original database in a way that the final, sanitized database D protects the sensitive item sets from disclosure.

III.HYBRID SOLUTION METHODOLOGY

A. Frequent Item set

Let I = {i_1, i_2, …………, i_M} be a finite set of literals, called items, where M denotes the cardinality of the set. Any subset $I \subseteq \boldsymbol{I}$ is an item set over *I*. A transaction T over I is a pair T = (tid, I), where I is the item set and tid is a unique identifier, used to distinguish among transactions that correspond to the same item set. A transaction database D = {T_1, $T_{2,}$ ....T_N} over I is an
N x M table consisting of N transactions over I carrying different identifiers, where entry T_{nm} = 1 if and only if the *m*th item (m ∈ [1, M]) appears in the *n*th transaction (n ∈ [1, N]). Otherwise, T_{nm} = 0. A transaction T = (tid, J) supports an item set *I* over ***I***, if I ⊆ J. Let S be a set of items; notation p(S) denotes the power set of S, which is the set of all subsets of S.

Given an item set I over *I* in *D*, sup (I, D) denotes the number of transactions $T \in D$ that support *I* and freq*(I, D)* denotes the fraction of transactions in *D* that support *I*. An item set *I* is called *large or frequent* in database *D* if and only if its frequency in *D* is at least equal to a minimum threshold *mfreq*. A hybrid of Apriori and FP-Tree algorithms are proposed to be used to find the frequent item set.

B. The Apriori Algorithm

- Finding Frequent Itemsets Using Candidate Generation.

Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (K+1) itemsets..

- Find frequent set L_{k-1}.
- Join Step.
  - C_k is generated by joining $L_{k-1}$with itself
- Prune Step.
  - Any $(k - 1)$ -itemset that is not frequent cannot be a subset of a frequent k -itemset, hence should be removed.

where

- (C_k: Candidate itemset of size *k*)
- (L_k: frequent item set of size *k*)

C.FP-Growth Algorithm

-Mining Frequent patterns without candidate generation .

Apriori candidate generate and test method reduces the size of candidate sets significantly and leads to good performance gain. However it may suffer from two nontrivial costs.

- It may need to generate a huge number of candidate sets : For example, if there are 10^4 frequent 1-itemsets, the Apriori algorithm will need to generate more than 10^7 candidate 2-itemsets and accumulate and test their occurrence frequencies. Moreover , to discover a frequent pattern of size 100 such as {a_1 …,a_{100}} , it must generate more than 2^{100} ~10^{30} candidates in total
- It may need to repeatedly scan the database and check a large set of candidates by pattern matching.

Frequent pattern growth or simply FP-growth adopts a divide-and – conquer strategy as follows: compress the database representing frequent items into a frequent pattern tree, or FP tree, but retain the itemset association information and then divide such a compressed database into a set of conditional databases, each associate with one frequent item, and mine each such database separately.

1. Major steps to mine FP-tree

1) Construct conditional pattern base for each node in the FP-tree
2) Construct conditional FP-tree from each conditional pattern-base

3) Recursively mine conditional FP-trees and grow frequent patterns obtained so far,If the conditional FP-tree contains a single path, simply enumerate all the patterns

	a	B	c	d	e	f
Do	1	1	0	0	0	1
	1	1	1	1	0	0
	1	0	1	0	0	1
	1	0	0	0	0	0
	0	1	0	0	1	0
	1	1	1	1	1	0
	0	0	0	1	0	0
	1	1	1	0	1	0
	0	1	1	0	0	0
	1	0	1	1	1	0
D_x	1	0	0	0	0	0
	1	0	1	1	0	0
	1	0	1	1	0	0
	1	1	0	0	0	0

Table1:Sanitized Database *D* as a Mixture of the Original Database D_O and the Applied Extension D_X

Frequent itemset in D_O	Support
{a}	7
{b} {c}	6
{ac}	5
{d}, {e}, {ab}, {bc}	4
{ad}, {ae}, {be}, {cd}, {ce}, {abc}, {acd}, {ace}	3

Table 2 :Frequent Item Sets for D_O at msup = 3 (for table 1)

Frequent itemset in *D*	Support
{a}	11
{c}	8
{b}, {ac}	7
{d}	6
{ab}, {ad}, {cd}, {acd}	5

~~{e},~~ {bc}	4
~~{ac},{be},{ce},{abc},~~{aee}	3

Table 3:Frequent Item Sets for $D(D_0 + D_x)$ at msup = 3 (for table 1)

IV.BORDER REVISION

The rationale behind this process is that hiding of a set of item sets corresponds to a movement of the original borderline [5] in the lattice that separates the frequent item sets from their infrequent counterparts , such that the sensitive item sets lie below the revised borderline. Since the borders are revised to accommodate for an exact solution, the revised hyperplane is designed to be ideal in the sense that it excludes only the sensitive item sets and their supersets from the set of frequent patterns in D, leaving the rest of the item sets in their previous status as in database DO. The first step in the hiding methodology rests on the identification of the revised borders for D. The hiding algorithm relies on both the revised positive and the negative borders, denoted as Bd^+ $(F^1{}_D)$ and Bd^- $(F^1{}_D)$, respectively. After identifying the new (ideal) borders, the hiding process has to perform all the required minimal adjustments of the transactions in D_x to enforce the existence of the new borderline in the result database.

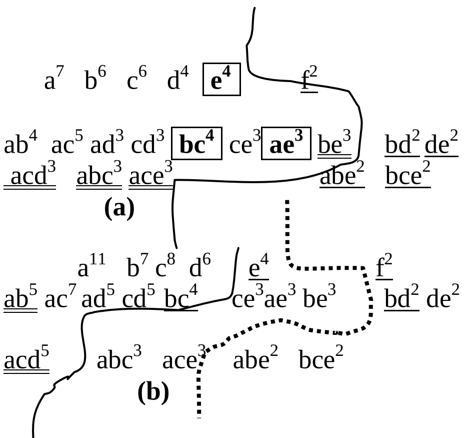
a^7 b^6 c^6 d^4 e^4 f^2
ab^4 ac^5 ad^3 cd^3 bc^4 ce^3 ae^3 be^3 bd^2 de^2
acd^3 abc^3 ace^3 abe^2 bce^2
(a)
a^{11} b^7 c^8 d^6 e^4 f^2
ab^5 ac^7 ad^5 cd^5 bc^4 ce^3 ae^3 be^3 bd^2 de^2
acd^5 abc^3 ace^3 abe^2 bce^2
(b)

Fig. 1 An sample item set lattic demonstration (a) the original border and the sensitive item sets, and (b) the revised border for Table1

V.COMPUTATION OF SIZE OF EXTENDED DATASET

Database D_O is extended by D_X to construct database D. An initial and very important step in the hiding process is the computation of the size of D_X. A lower bound on this value can be established based on the sensitive item set in S, which has the highest support. The rationale here is given as follows: by identifying the sensitive item set with the highest support, one can safely decide upon the minimum number of transactions that must not support this item set in D_X, so that it becomes infrequent in D.

Lower bound Q is

$$Q = \left\lceil \frac{\sup\{I_N, D_O)}{mfreq} - N \right\rceil + 1 \text{.......}\mathbf{1}$$

Equation (1) provides the absolute minimum number of transactions that need to be added in D_X, to allow for the proper hiding of the sensitive item sets of D_O. However, this lower bound can, under certain circumstances, be insuffi-cient to allow for the identification of an exact solution,1 even if one exists. To circumvent this problem, one needs to expand the size Q of D_X as determined by (1), by a certain number of transactions. A threshold, called safety margin (denoted hereon as SM), is incorporated for this purpose. Safety margins can be either predefined or be computed dynamically, based on particular properties of database D_O and/or other parameters regarding the hiding process. In any case, the target of using a safety margin is to ensure that an adequate number of transactions participate in D_X, thus an exact solution (if one exists) will not be lost due to the small size of the extension.

VI. PROBLEM SIZE MINIMIZATION

To enforce the computed revised border and identify the exact hiding solution, a mechanism is needed to regulate the status (frequent versus infrequent) of all the item sets in D. Let *C* be the minimal set of border item sets used to regulate the values of the various uqm variables in D_X. Moreover, suppose that I 2C is an item set, whose behavior we want to regulate in D. Then, item set I will be frequent in D if and only if

sup (I, D_O) + sup (I, D_X) ≥ *mfreq* x (N + Q), or equivalently if

$$\text{Sup (I, } D_O\text{) + } \sum_{q=1}^{Q} \prod_{i_M \in I} U_{qm} \geq mfreq \times (N+Q) \quad \text{-------(2)}$$

and will be infrequent otherwise, when

$$\text{Sup(I,}D_O\text{)+ } \sum_{q=1}^{Q} \prod_{i_M \in I} U_{qm} < mfreq \times (N+Q) \text{ - -------(3)}$$

Inequality (2) corresponds to the minimum number of times that an item set I has to appear in the extension D_X to remain frequent in D. On the other hand, (3) provides the maximum number of times that an item set I has to appear in D_X to be infrequent in database D.

To identify an exact solution to the hiding problem, every possible item set in P, according to its position in the lattice—with respect to the revised border—must satisfy either (2) or (3). However, the complexity of solving the entire system of the 2^M – 1 inequalities is well known to be NP-hard. Therefore, one should restrict the problem to capture only a small subset of these inequalities, thus leading to a problem size that is computationally manage-able. The proposed problem formulation achieves this by reducing the number of the participating inequalities that need to be satisfied. Even more, by carefully selecting the item sets of set C, the hiding algorithm ensures that the exact same solution to the one of solving the entire system of inequalities is attained. This is accomplished by exploiting cover relations existing among the item sets in the lattice due to the monotonicity of support.

A. Formulation and Solution of the CSP

A CSP(constraint satisfactory problem) is defined by a set of variables and a set of constraints, where each variable has a nonempty domain of potential values. The constraints involve a subset of the variables and specify the allowable combinations of values that these variables can attain. An assignment that does not violate the set of constraints is called "consistent." A solution of a CSP is a complete assignment of values to the variables that satisfies all the constraints. Since in this work all variables involved are binary in nature, the produced CSP is solved by using a technique called BIP [3] that transforms it to an optimization problem. To avoid the high degree of constraints, the application of a Constraints Degree Reduction(CDR) approach is essential. This approach relies on the binary nature of the variables to linearize all the nonlinear constraints. Linearization does not lead to any information loss; its only side effect is an increase in the number of variables and constraints in the system. On the other hand, the resulting inequalities are simple in nature and allow for fast solutions, thus adhere for an efficient solution of the entire CSP.

VII. CONCLUSION

A novel, exact border-based hybrid approach to sensitive knowledge hiding, through the introduction of a minimal extension to the original database was proposed. A hybrid approach of combining Apriori and FP_Tree was proposed to find the frequent item set. In this phase, frequent item set was computed. In the next phase , a solution to sensitive knowledge hiding is achieved by minimum expansion of the original database. The proposed methodology is capable of identifying an ideal solution whenever one exists, or approximate the exact solution, otherwise. In this work, we provided insight on various topics, such as the minimum expansion of the original database, border revision, validation of the constructed transactions, and the treatment of suboptimality in solutions.

REFERENCES

[1] Agrawal. R and Srikant. R, (2000) "Privacy-Preserving Data Mining," Proc. ACM SIGMOD '00, pp. 439-450,

[2]Aris Gkoulalas – Divanis, Vassilios S. Verykois, (May 2009) "Exact Knowledge Hiding through Database Extension" IEEE Transaction on Knowledge and Data Engineering, Vol. 21, No. 5,

[3]Gkoulalas-Divanis. A and Verykios. V.S, (Nov.2006) "An Integer Programming Approach for Frequent Itemset Hiding," Proc. ACM Conf. Information and Knowledge Management (CIKM '06), pp. 748-757,

[3]Oliveira. S.R.M. and Zaı¨ane. O.R.,, (2003) "Protecting Sensitive Knowledge by Data Sanitization," Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03), pp. 211-218.

[4]Sun. X and Yu. P.S, (2005) "A Border-Based Approach for Hiding Sensitive Frequent Itemsets," Proc. Fifth IEEE Int'l Conf. Data Mining (ICDM '05), pp. 426-433.

[5]Verykios. V.S., Emagarmid. A.K., Bertino. E., Saygin. Y, and Dasseni. E, (Apr. 2004) "Association Rule Hiding," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 4, pp. 434-447,