Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

SECURED FREQUENT ITEMSET DISCOVERY IN MULTI PARTY DATA ENVIRONMENT

FREQUENT ITEMSET WITH MULTIPARTY DATA ENVIRONMENT

M.P.Anitha[1], A.Sathyapriya[2], T.Balasubramaniam[3]
[1]PG Scholar Computer Science and Engineering, Vivekanandha College Of Engineering For Women
[2]Assistant professor Computer Science and Engineering, Vivekanandha College Of Engineering For Women
[3]Assistant professor Computer Science and Engineering, Vivekanandha College Of Engineering For Women

Abstract - Security and privacy methods are used to protect the data values. Private data values are secured with confidentiality and integrity methods. Privacy model hides the individual identity over the public data values. Sensitive attributes are protected using anonymity methods. Two or more parties have their own private data under the distributed environment. The parties can collaborate to calculate any function on the union of their data. Secure Multiparty Computation (SMC) protocols are used in privacy preserving data mining in distributed environments. Association rule mining techniques are used to fetch frequent patterns.Apriori algorithm is used to mine association rules in databases. Homogeneous databases share the same schema but hold information on different entities. Horizontal partition refers the collection of homogeneous databases that are maintained in different parties. Fast Distributed Mining (FDM) algorithm is an unsecured distributed version of the Apriori algorithm. Kantarcioglu and Clifton protocol is used for secure mining of association rules in horizontally distributed databases. Unifying lists of locally Frequent Itemsets Kantarcioglu and Clifton (UniFI-KC) protocol is used for the rule mining process in partitioned database environment. UniFI-KC protocol is enhanced in two methods for security enhancement. Secure computation of threshold function algorithm is used to compute the union of private subsets in each of the interacting players. Set inclusion computation algorithm is used to test the inclusion of an element held by one player in a subset held by another.The system is improved to support secure rule mining under vertical partitioned database environment. The subgroup discovery process is adapted for partitioned database environment. The system can be improved to support generalized association rule mining process. The system is enhanced to control security leakages in the rule mining process.

Keywords-Secure rule mining process,Secure multiparty computation protocol,Fast distributed mining algorithm,Association rule mining.

I. INTRODUCTION

1.1. Privacy Preserving and Data mining

Data mining and knowledge discovery in databases are two new research areas that investigate the automatic extraction of previously unknown patterns from large amounts of data. Recent advances in data collection, data dissemination and related technologies have inaugurated a new era of research where existing data mining algorithms should be reconsidered from a different point of view, this of privacy preservation.

Privacy preserving data mining, is a novel research direction in data mining and statistical databases, where data mining algorithms are analyzed for the side-effects they incur in data privacy. The main consideration in privacy preserving data mining is two fold. First, sensitive raw data like identifiers, names, addresses and the like, should be modified or trimmed out from the original database, in order for the recipient of the data not to be able to compromise another person’s privacy. Second, sensitive knowledge which can be mined from a database by using data mining algorithms, should also be excluded, because such a knowledge can equally well compromise data privacy. The main objective in privacy preserving data mining is 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. The problem that arises when confidential unauthorized users can derive information from released data is also commonly called the "database inference" problem.

1.2. Classification of Privacy Preserving Techniques

There are many approaches, which have been adopted for privacy preserving data mining. They are classified on the basis of the following dimensions:
data distribution
data modification
data mining algorithm
data or rule hiding
privacy preservation
The first dimension refers to the distribution of data. Some of the approaches have been developed for centralized data, while others refer to a distributed data scenario. Distributed data scenarios can also be classified as horizontal data distribution and vertical data distribution. Horizontal distribution refers to these cases where different database records reside in different places, while vertical data distribution, refers to the cases where all the values for different attributes reside in different places. The second dimension refers to the data modification scheme. In general, data modification is used in order to modify the original values of a database that needs to be released to the public and in this way to ensure high privacy protection. It is important that a data modification technique should be in concert with the privacy policy adopted by an organization. Methods of modification include:

- perturbation, which is accomplished by the alteration of an attribute value by a new value,
- blocking, which is the replacement of an existing attribute value with a "?",
- aggregation or merging which is the combination of several values into a coarser category,
- swapping that refers to interchanging values of individual records, and
- sampling, which refers to releasing data for only a sample of a population.

The third dimension refers to the data mining algorithm, for which the data modification is taking place. This is actually something that is not known beforehand, but it facilitates the analysis and design of the data hiding algorithm. The problem of hiding data as included for a combination of data mining algorithms. For the time being, various data mining algorithms have been considered in isolation of each other. Among them, the most important ideas have been developed for classification data mining algorithms, like decision tree inducers, association rule mining algorithms, clustering algorithms, rough sets and Bayesian networks.

The fourth dimension refers to whether raw data or aggregated data should be hidden. The complexity for hiding aggregated data in the form of rules is of course higher, and for this reason, mostly heuristics have been developed. The lessening of the amount of public information causes the data miner to produce weaker inference rules that will not allow the inference of confidential values. This process is also known as "rule confusion".

1.3. Privacy Preserving Algorithms

1.3.1. Heuristic-Based Techniques

A number of techniques have been developed for a number of data mining techniques like classification, association rule discovery and clustering, based on the premise that selective data modification or sanitization is an NP-Hard problem, and for this reason, heuristics can be used to address the complexity issues.

1.3.2. Cryptography-Based Techniques

A number of cryptography-based approaches have been developed in the context of privacy preserving data mining algorithms, to solve problems of the following nature. Two or more parties want to conduct a computation based on their private inputs, but neither party is willing to disclose its own output to anybody else. The issue here is how to conduct such a computation while preserving the privacy of the inputs.

1.3.3. Reconstruction-Based Techniques

A number of recently proposed techniques address the issue of privacy preservation by perturbing the data and reconstructing the distributions at an aggregate level in order to perform the mining. Some of these techniques are listed and classified. The work presented addresses the problem of building a decision tree classifier from training data in which the values of individual records have been perturbed. While it is not possible to accurately estimate original values in individual data records, the authors propose a reconstruction procedure to accurately estimate the distribution of original data values. By using the reconstructed distributions, they are able to build classifiers whose accuracy is comparable to the accuracy of classifiers built with the original data.

1.3.4. Association Rule Mining Algorithm

A number of data mining algorithms have been recently developed that greatly facilitate the processing and interpreting of large stores of data. One example is the association rule-mining algorithm, which discovers correlations between items in transactional databases. Apriori algorithm is an example of association rule mining algorithm.

Given a set of transactions, each described by an unordered set of items, an association rule $X \cup Y$ may be discovered in the data, where X and Y are conjunctions of items. The intuitive meaning of such a rule is that transactions in the database, which contain the items in X, tend to also contain the items in Y. An example of such a rule might be many observed customers who purchase tires and auto accessories also buy some automotive services.

In this case, X = {tires, auto accessories} and Y = {automotive services}. Two numbers are associated with each rule that indicates the support and confidence of the rule. The support of the rule $X \cup Y$ represents the percentage of transactions from the original database that contain both X and Y. The confidence of rule $X \cup Y$ represents the percentage of transactions containing items in X that also contain items in Y. Applications of association rule mining include cross marketing, attached mailing, catalog design and customer segmentation. An association rule discovery algorithm searches the space of all possible patterns for rules that meet the user-specified support and confidence thresholds. The problem of discovering association rules can be divided into two steps:

1. Find all item sets whose support is greater than the specified threshold. Item sets with minimum support are called frequent item sets.
2. Generate association rules from the frequent item sets. To do this, consider all partitioning of the item set into rule left-hand and right-hand sides. Confidence of a candidate rule $X \cup Y$ is calculated as support (XY) / support (X). All rules that meet the confidence threshold are reported as discoveries of the algorithm.

```
$L_1$: = {frequent 1-itemsets};
        k:= 2; // k represents the pass number
        While ($L_{k-1}$)
        $C_k$ = New candidates of size k generated from $L_{k-1}$
        For all transactions t
        D       Increment count of all candidates in $C_k$
                Those are contained in t
        $L_k$ = All candidates in Ck with minimum support
        k = k+1
Report $U_k$ $L_k$ as the discovered frequent item sets
```

Table 2.2.2.3.1. summarizes the Apriori algorithm. The first pass of the algorithm calculates single item frequencies to determine the frequent 1-itemsets. Each subsequent pass *k* discovers frequent item sets of size *k*. To do this, the frequent item sets L_{k-1} found in the previous iteration are joined to generate the candidate item sets C_k. Next; the support for candidates in C_k is calculated through one sweep of the transaction list.

k-item set	An item set containing k items
L_k	Set of frequent k-item sets(k-item sets with minimum support)
C_k	Set of candidate k-item sets (potentially frequent item sets)
U_k L_k	Set of generated item sets

Table 1.3.4.1. Apriori Algorithms

From L_{k-1}, the set of all frequent (k-1) item sets; the set of candidate k-item sets is created. The intuition behind this Apriori candidate generation procedure is that if an item set X has minimum support, so do all the subsets of X. Thus new item sets are created from (k-1) item sets p and q by listing p.item1, p.item2, p.item (k-1), q.item (k-1). Items p and q are selected if items 1 through k-2 are equivalent for p and q and item k-1 is not equivalent. Once candidates are generated, items etcs are removed from consideration if any (k-1) subset of the candidate is not in L_{k-1}.

II. LITERATURE SURVEY

2.1Secure Distributed Subgroup Discovery in Horizontally Partitioned Data

The question of the privacy of data can be an important aspect in the real-world application of data mining. In privacy-sensitive scenarios, in particular those with distributed data, a failure to guarantee certain privacy-preserving constraints means that data mining can not be applied at all. As an example, consider the case of competing mail order companies. To a large part, these companies make money by knowing their customers better than their competitors do. On the other hand, they lose money due to fraud. Typically, the risk of disclosing sensitive customer information by far outweighs the chances of reducing expenses by a joint fraud detection effort. Only privacy-preserving data mining techniques will allow an analysis of fraud patterns over all companies.

In applications like the above, descriptive techniques like rule mining are very popular, as they have the potential to provide more insight than numerical methods like SVMs or neural networks. Actually, protocols have been proposed that allow secure association rule mining over distributed databases. These, rely on the classical support/ confidence framework, which has been observed to have effects undesired in some settings: In particular, there is a danger to come up with huge amounts of rules which are not significant or do not express a correlation. In this paper, we present secure protocols for the task of top-k subgroup discovery on horizontally partitioned data.

2.2. A Secure Distributed Framework For Achieving K-anonymity

Privacy is an important issue in our society and has become vulnerable in these technologically advanced times. Legislation has been proposed to protect individual privacy; a key component is the protection of individually identifiable data. Many techniques have been proposed to protect privacy, such as data perturbation, query restriction, data swapping, Secure Multi-party Computation (SMC), etc. One challenge is relating such techniques to a privacy definition that meets legal and societal norms. Anonymous data are generally considered to be exempt from privacy rules – but what does it mean for data to be anonymous? Census agencies, which have long dealt with private data, have generally found that as long as data are aggregated over a group of individuals, release does not violate privacy. k-anonymity provides a formal way of generalizing this concept. A data record is k-anonymous if and only if it is indistinguishable in its identifying information from at least k specific records or entities. The key step in making data anonymous is to generalize a specific value. For example, the ages 18 and 21 could be generalized to an interval. Details of the concept of k-anonymity and ways to generate k-anonymous data are provided.

2.3.An Efficient Approximate Protocol for Privacy-Preserving Association Rule Mining

Privacy-preserving data mining has made major advances in the recent years. Many protocols have been proposed for different data mining algorithms such as classification, association rule mining, clustering and outlier detection, etc. provides a comprehensive survey. There are two main types of technique – perturbation based methods and secures multiparty computation techniques. In the perturbation methods the data is locally perturbed before delivering it to the data miner. Special techniques are used to reconstruct the original distribution and the mining algorithm needs to be modified to take this into consideration. The seminal paper by Agrawal and Srikant introduced this approach in the form of a procedure to build a decision tree classifier from perturbed data.

The second approach assumes that data is distributed between two or more sites that cooperate to learn the global data mining results without revealing the data at individual sites. This approach was introduced by Lindell and Pinkas, with a method that enabled two parties to build a decision tree without either party learning anything about the other party's data, except what might be revealed through the final decision tree. Typically, the techniques used are cryptographic. While the first approach has known security problems and cannot lead to provably secure solutions, the second approach has typically been too computationally intensive. This is especially the case for vertically partitioned data, where unlike horizontally partitioned data little data summarization can be carried out before engaging in the distributed protocol.

We now give some background on Bloom Filters. Bloom filters have been extensively used for various application domains ranging from networking to databases. Basically, a bloom filter represents a set $S = \{x_1, x_2, \ldots, x_n\}$ of n elements using array of m bits ($m \leq n$), initially all set to 0. For each element $x \in S$, we use k independent random hash functions $h_1(), h_2(), \ldots, h_k()$ with range $\{1, \ldots, m\}$ such that the bits $h_i(x)$ of the array are set to 1 for $1 \leq i \leq k$. In this basic version, a location can be set to 1 multiple times but only the first change has an effect. To check whether an item $t \in S$, we need to check whether all $h_i(t)$ for $1 \leq i \leq k$ are set to 1. If they are not all set to 1, we can conclude that $t \in S$. On the other hand, if all $hi(t)$ for $1 \leq i \leq k$ set to 1, we can assume that $t \in S$, with some nonzero probability.

Bloom filters can be used to approximate the intersection size between two sets. Given two bloom filters with the same m and k values that represent two sets S_1 and S_2, we can approximate $|S_1 \cap S_2|$ by getting the dot product of the two filters. More precisely, let $Z1$ be the number of 0s in the filter S_1 and Z_{12} be the number of 0s in the inner product, then we can approximate $|S1 \cap S2|$ using the following formula:

$$\frac{1}{m}\left(1-\frac{1}{m}\right)^{-k|S1\cap S2|} - \approx \frac{Z1+Z2-Z12}{Z1Z2} \qquad (1)$$

$$= |S1 \cap S2| \approx \frac{\ln\left(m(Z1+Z2-Z12)\right) - \lambda v(Z1) - \lambda v(Z2)}{-k\ln\left(1-\frac{1}{m}\right)} \qquad (2)$$

2.4. Secure Two-Party Differentially Private Data Release for Vertically Partitioned Data

Huge databases exist today due to the rapid advances in communication and storing systems. Each database is owned by a particular autonomous entity, for example, medical data by hospitals, income data by tax agencies, financial data by banks and census data by statistical agencies. Moreover, the emergence of new paradigms such as cloud computing increases the amount of data distributed between multiple entities. These distributed data can be integrated to enable better data analysis for making better decisions and providing high-quality services. For example, data can be integrated to improve medical research, customer service, or homeland security.

This research problem was discovered in a collaborative project with the financial industry. We generalize their problem as follows: A bank A and a loan company B have different sets of attributes about the same set of individuals identified by the common identifier attribute (ID), such that bank A owns DA, while loan company B owns DB. These parties want to integrate their data to support better decision making such as loan or credit limit approvals. In addition to parties A and B, their partnered credit card company C also has access to the integrated data, so all three parties A, B and C are data recipients of the final integrated data. Parties A and B have two concerns. First, simply joining DA and DB would reveal sensitive information to the other party. Second, even if DA and DB individually do not contain person-specific or sensitive information, the integrated data can increase the possibility of identifying the record of an individual.

2.5.Privacy-Preserving Updates to Anonymous and Confidential Databases

It is today well understood that databases represent an important asset for many applications and thus their security is crucial. Data confidentiality is particularly relevant because of the value, often not only monetary, that data have. For example, medical data collected by following the history of patients over several years may represent an invaluable asset that needs to be adequately protected. Such a requirement has motivated a large variety of approaches aiming at better protecting data confidentiality and data ownership. Relevant approaches include query processing techniques for encrypted data and data watermarking techniques. Data confidentiality is not, however, the only requirement that needs to be addressed.

III. PROBLEM DESCRIPTION

We study here the problem of secure mining of association rules in horizontally partitioned databases. In that setting, there are several sites that hold homogeneous databases, i.e., databases that share the same schema but hold information on different entities. The goal is to find all association rules with support at least s and confidence at least c, for some given minimal support size s and confidence level c, that hold in the unified database, while minimizing the information disclosed about the private databases held by those players. The information that we would like to protect in this context is not only individual transactions in the different databases, but also more global information such as what association rules are supported locally in each of those databases.

That goal defines a problem of secure multi-party computation. In such problems, there are M players that hold private inputs, $x_1, \ldots ; x_M$ and they wish to securely compute $y = f(x_1, \ldots ,x_M)$ for some public function f. If there existed a trusted third party, the players could surrender to him their inputs and he would perform the function evaluation and send to them the resulting output. In the absence of such a trusted third party, it is needed to devise a protocol that the players can run on their own in order to arrive at the required output y. Such a protocol is considered perfectly secure if no player can learn from his view of the protocol more than what he would have learnt in the idealized setting where the computation is carried out by a trusted third party. Yao was the first to propose a generic solution for this problem in the case of two players.

In our problem, the inputs are the partial databases and the required output is the list of association rules that hold in the unified database with support and confidence no smaller than the given thresholds s and c, respectively. As the above mentioned generic solutions rely upon a description of the function f as a Boolean circuit, they can be applied only to small inputs and functions which are realizable by simple circuits. In more complex settings, such as ours, other methods are required for carrying out this computation. In such cases, some relaxations of the notion of perfect security might be inevitable when looking for practical protocols, provided that the excess information is deemed benign.

3.1. Preliminaries

Let D be a transaction database. We view D as a binary matrix of N rows and L columns, where each row is a transaction over some set of items, $A = \{ a_1, . . . , a_L \}$ and each column represents one of the items in A. The database D is partitioned horizontally between M players, denoted $P_1, . . . , P_M$. Player Pm holds the partial database Dm that contains Nm = $|D_m|$ of the transactions in D, $1 \le m \le M$. The unified database is $D = D_1 \cup ..... \cup D_M$ and it includes $N := \sum_{m=1}^{M} N_m$ transactions. An item set X is a subset of A. Its global support, supp(X), is the number of transactions in D that contain it. Its local support, $\text{supp}_m(X)$, is the number of transactions in Dm that contain it. Clearly, supp(X) = $\sum_{m=1}^{M} \sup p_m(X)$. Let s be a real number between 0 and 1 that stands for a required support threshold.

An item set X is called s-frequent if supp(X) $\ge$ sN. It is called locally s-frequent at D_m if suppm(X) $\ge sN_m$.

For each $1 \le k \le L$, let F_s^k denote the set of all k-item sets that are s-frequent and $F_s^{k,m}$ be the set of all k-item sets that are locally s-frequent at D_m, $1 \le m \le M$. Our main computational goal is to find, for a given threshold support $0 < s \le 1$, the set of all s-frequent item sets, Fs:= $\bigcup_{k=1}^{L} F_s^k$. We may then continue to find all (s, c)-association rules, i.e., all association rules of support at least sN and confidence at least c. The protocol based on the Fast Distributed Mining (FDM) algorithm of Cheung et al. which is an unsecured distributed version of the Apriori algorithm. Its main idea is that any s-frequent item set must be also locally s-frequent in at least one of the sites. Hence, in order to find all globally s-frequent item sets, each player reveals his locally s-frequent item sets and then the players check each of them to see if they are s-frequent also globally. The FDM algorithm proceeds as follows:

1. Initialization**:** It is assumed that the players have already jointly calculated F_s^{k-1}. The goal is to proceed and calculate F_s^k .

2. Candidate Sets Generation: Each player P_m computes the set of all (k -1)-item sets that are locally frequent in his site and also globally frequent; namely, Pm computes the set $F_s^{k-1,m} \cap F_s^{k-1}$. He then applies on that set the Apriori algorithm in order to generate the set $B_s^{k,m}$ of candidate k-item sets.

3. Local Pruning: For each $X \in B_s^{k,m}$, P_m computes $\text{supp}_m(X)$. He then retains only those item sets that are locally s-frequent. We denote this collection of item sets by $C_s^{k,m}$.

4. Unifying the candidate item sets: Each player broadcasts his $C_s^{k,m}$ and then all players compute $C_s^k := \bigcup_{m=1} MC \quad C_s^{k,m}$

5. Computing local supports: All players compute the local supports of all item sets in C_s^k.

6. Broadcast mining results: Each player broadcasts the local supports that he computed. From that, everyone can compute the global support of every item set in C_s^k . Finally, F_s^k is the subset of C_s^k that consists of all globally s-frequent k-item sets.

In the first iteration, when k =1, the set $C_s^{l,m}$ that the mth player computes (Steps 2-3) is just $F_s^{l,m}$, namely, the set of single items that are s-frequent in Dm. The complete FDM algorithm starts by finding all single items that are globally s-frequent. It then proceeds to find all 2-item sets that are globally s-frequent and so forth, until it finds the longest globally s-frequent item sets. If the length of such item sets is K, then in the (K + 1)th iteration of the FDM it will find no (K + 1)-item sets that are globally s-frequent, in which case it terminates.

IV.PROPOSED SYSTEM

Fast Distributed Mining (FDM) algorithm is used to fetch frequent rules in Horizontally Distributed Databases. Secure multiparty algorithms are used to mine privacy preserved frequent patterns from different databases. Unifying lists of locally Frequent Itemsets (UNIFI) algorithm is integrated with threshold function and set inclusive functions. The system is enhanced to mine rules under horizontal and vertically distributed environment. The system is improved to support secure rule mining under vertical partitioned database environment. The subgroup discovery process is adapted for partitioned database environment. The system can be improved to support generalized association rule mining process. The system is enhanced to control security leakages in the rule mining process.

V.CONCLUSION

Fast Distributed Mining (FDM) algorithm is used to fetch frequent rules in Horizontally Distributed Databases. Secure multiparty algorithms are used to mine privacy preserved frequent patterns from different databases. Unifying lists of locally Frequent Itemsets (UNIFI) algorithm is integrated with threshold function and set inclusive functions. The system is enhanced to mine rules under horizontal and vertically distributed environment. The system supports Horizontal and Vertical partition based rule mining process. Communication and computational load is reduced in the distributed rule mining process. Sensitive attributes and item sets are protected by the system. The system improves the rule mining accuracy level.

REFERENCES

[1] Alberto Trombetta, Wei Jiang and Lorenzo Bossi, "Privacy-Preserving Updates to Anonymous and Confidential Databases", IEEE Transactions On Dependable And Secure Computing, July/August 2011

[2] H. Grosskreutz, B. Lemmen and S. Ruping, "Secure Distributed Subgroup Discovery in Horizontally Partitioned Data," Trans. Data Privacy, vol. 4, no. 3, pp. 147-165, 2011.

[3] Leopoldo Bertossi and Lechen Li, "Achieving Data Privacy through Secrecy Views and Null-Based Virtual Updates", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 5, May 2013

[4] M. Kantarcioglu, R. Nix and J. Vaidya, "An Efficient Approximate Protocol for Privacy-Preserving Association Rule Mining," Proc. 13th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining, 2009.

[5] Mehmet Ercan Nergiz and Thomas B. Pedersen, "A Look-Ahead Approach to Secure Multiparty Protocols", IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 7, July 2012

[6] Noman Mohammed, Dima Alhadidi and Mourad Debbabi, "Secure Two-Party Differentially Private Data Release for Vertically Partitioned Data"- IEEE Transactions On Dependable And Secure Computing, Vol. 11, No. 1, January/February 2014

[7] W. Jiang and C. Clifton, "A Secure Distributed Framework for Achieving k-Anonymity," The VLDB J., vol. 15, pp. 316-333, 2006.

[8] Sara Hajian and Josep Domingo-Ferrer, "A Methodology for Direct and Indirect Discrimination Prevention in Data Mining", IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 7, July 2013

[9] T. Tassa and E. Gudes, "Secure Distributed Computation of Anonymized Views of Shared Databases," Trans. Database Systems, vol. 37, 2012.

[10] Tamir Tassa, "Secure Mining of Association Rules in Horizontally Distributed Databases", IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 4, April 2014