www.ijiarec.com

ISSN: 2348-2079

MAY-2015

International Journal of Intellectual Advancements and Research in Engineering Computations

MULTI PARTY DATA DISTRIBUTION AND RULE MINING WITH PRIVACY

[1]K. K. Kavishree, [2]A. N. Karthikeyan.

ABSTRACT

Attribute behavior are identified using the rule mining techniques. 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 distributed mining model is used to fetch attribute behavior under the 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.

INTRODUCTION

Data mining technology has emerged as a means of identifying patterns and trends from large quantities of data. Data mining and data warehousing go hand-in-hand: Most tools operate by gathering all data into a central site, then running an algorithm against that data. Privacy concerns can prevent building a centralized warehouse data may be distributed among several custodians, none of which are allowed to transfer their data to another site.

In homogeneous databases, all sites have the same schema, but each site has information on different entities. The goal is to produce association rules that hold globally while limiting the information shared about each site. Computing association rules without disclosing individual transactions is straightforward. We can compute the global support and confidence of an association rule AB→ C knowing only the local supports of AB and ABC and the size of each database:

The privacy preserved distributed mining scheme does not require sharing any individual transactions. We can easily extend an algorithm such as a priori to the distributed case using the following lemma: If a rule has support > k% globally, it must have support > k% on at least one of the individual sites. A distributed algorithm for this would work as follows: Request that each site send all rules with support at least k. For each rule returned, request that all sites send the count of their transactions that support the rule and the total count of all transactions at the site. From this, we can compute the global support of each rule and be certain that all rules with support at least k have been found. More thorough studies of distributed association rule mining can be found.

Author for correspondence:

[1]Final year ME CSE, Mahendra Institute of Technology, Mahendhrapuri, Tamilnadu, India.

[2]Assistant Professor, Mahendra Institute of Technology, Mahendhrapuri, Tamilnadu, India.

RELATED WORK

Privacy-preserving data mining has emerged to address the situation when the use of data mining techniques is desired, but the data to be mined may not be disclosed or brought together for privacy considerations. The work presented in this paper falls into the category of approaches based on secure multi-party computation. Here, the premise is that the data is distributed over different parties, and the goal is to obtain data mining results without revealing the (private) data of one party to another. Data publishing issues, which are concerned with the amount of private information which can be deduced from a published set of patterns or data records (e.g. [2]), are not considered here.

Secure multi-party computation originates in Yao' famous millionaires problem: two millionaires want to find out who is richer without revealing the precise amount of their wealth. Yao has presented a generic solution that allows the secure evaluation of any two party functionality. The solution is based on the encryption and secure evaluation of circuits. This technique has shown to be very useful to securely compute some sub functionality and has been applied for that purpose in several settings. While in principle, this generic solution could be used to securely compute any kind of data mining task simply by using an appropriate circuit, this naive approach would result in huge circuits whose inputs would be entire databases – an approach which is not feasible in practice. Another limitation of the approach is that it only considers two-party problems. While there also exist generic constructions for the multi-party case, these have additional drawbacks which prevent their use in most applications.

For these reasons, custom secure multi-party solutions have been developed for many different data-mining problems. These include privacy-preserving protocols for decision tree induction, for nearest neighbor search [1] or for support vector machine classification [9]. The work most similar to ours is clearly the privacy-preserving association rule mining algorithm presented. The distributed global subgroup mining protocol of Wurst and Scholz, does not consider issues like privacy or secure multi-party computation and hence there are no obvious guarantees concerning the amount of information disclosed during the computation. In fact, sophisticated pruning strategies like pruning based on partially available counts rely on the disclosure of private information and thus violate the concept of secure computation. But even if such pruning strategies are disabled, the standard top-k subgroup discovery approaches cannot be applied in the context of secure multi-party computation. The problem is that in order to collect only the top-k subgroups, they make use of an increasing quality threshold which changes in view of the best subgroups collected so far. This approach reveals information about the quality of the subgroups visited, which again violates the concept of secure computation.

ANONYMITY MODELS FOR PRIVACY

Given a data set T, T[c][r] refers to the value of column c, row r of T. T[C] refers to the projection of set of columns C on T and T[.][r] refers to selection of row r on T. Although there are many ways to generalize a given data value, in this paper, we stick to generalizations according to domain generalization hierarchies (DGH). We also abuse notation and write $\Delta^{-1}(v^*)$ to indicate the children of v^* at the leaf nodes. For example, given DGH structures Δ_1(USA) = AM,

Δ_2 (Canada) =*; $\Delta_{0.1}$ (<M, USA>) = <M, AM>,

Δ(USA) = {USA, AM,*},Δ^{-1}(AM)

= {USA, Canada, Peru, Brazil}.

Since k-anonymity does not enforce constraints on the sensitive attributes, sensitive information disclosure is still possible in a k-anonymization. This problem has been addressed in [3], [6] enforcing diversity on sensitive attributes within a given equivalence class.. It should be noted that even extensions to k-anonymity have vulnerabilities in the case of external knowledge. As our focus in this paper is the look-ahead process, we do not present further detail. For the sake of

simplicity, from now on we assume data sets contain only QI attributes unless noted otherwise.

Even though k-anonymization of data sets by a single data owner has been studied extensively; in real world, databases may not reside in one source. Data might be horizontally or vertically partitioned over multiple parties all of which may be willing to participate to generate a k-anonymization of the union. The main purpose of the participation is using a larger data set to create a better utilized k-anonymization. As we increase data size [7], fewer tuples need to be suppressed or generalized to satisfy k-anonymity, in other words k-anonymization can be satisfied with lower level mappings. In most cases, there is no trusted party to make a secure local anonymization on the union. SMC protocols are developed among parties to securely compute the anonymization among semihonest parties.

PARALLEL ALGORITHM AND DISTRIBUTED ALGORITHM

PARALLEL ALGORITHM

Parallel ARM algorithms categorized as *data-parallelism* or *task-parallelism* algorithms. The algorithms partition the data sets among different nodes; in the latter, each site performs the task independently but must access the entire data set. Data Distribution is a task-parallelism-based algorithm that partitions the candidate itemsets among the processors. Each processor is responsible for computing the counts of its locally stored subset of the candidate itemsets for all the transactions in the database. Each processor must scan the portions of the transactions assigned to other processors as well as its locally stored portion of the transactions. It thus suffers from high communication overhead and performs poorly when compared with CD.

Candidate Distribution partitions the candidates during iterations, so that each processor can generate disjoint candidates independently. At the same time, it selectively replicates the database so that a processor can generate global counts independently. Candidate Distribution performs worse than CD. The Common Candidate Partitioned Database uses a data-parallel approach in shared-memory architecture [4]. The algorithm partitions the database logically into equal-sized chunks. Each processor generates a disjoint candidate subset, leading to good computational division. The PEAR algorithm is based on the sequential SEAR algorithm. The SEAR algorithm works exactly like the Apriori algorithm but uses a prefix tree rather that a hash tree, improving its performance.

DISTRIBUTED ALGORITHM

DARM discovers rules from various geographically distributed data sets. However, the network connection between those data sets isn't as fast as in a parallel environment, so distributed mining usually aims to minimize communication costs. Researchers proposed the Fast Distributed Mining algorithm to mine rules from distributed data sets partitioned among different sites. In each site, FDM finds the local support counts and prunes all infrequent local support counts. After completing local pruning, each site broadcasts messages containing all the remaining candidate sets to all other sites to request their support counts. It then decides whether large itemsets are globally frequent and generates the candidate itemsets from those globally frequent itemsets.

FDM's main advantage over CD is that it reduces the communication overhead to O ($|C_p| * n$), where $|C_p|$ and n are potentially large candidate itemsets and the number of sites, respectively. FDM generates fewer candidate itemsets compared to CD, when the number of disjoint candidate itemsets among various sites is large. The system can only achieve this when different sites have non homogeneous data sets. FDM's message optimization techniques require some functions to determine the polling site, which could cause extra computational cost when each site has numerous local frequent itemsets. Furthermore, each polling site must send a request to remote sites other than the originator site to find an itemset's global support counts, increasing message size when numerous remote sites exist.

ASSOCIATION RULE MINING WITH SECURITY

The distributed rule mining mechanism is enhanced to solve 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 [8]. 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.

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.

The protocol that we propose here computes a parameterized family of functions, which we call threshold functions, in which the two extreme cases correspond to the problems of computing the union and intersection of private subsets. Those are in fact general-purpose protocols that can be used in other contexts as well. Another problem of secure multiparty computation that we solve here as part of our discussion is the set inclusion problem; namely, the problem where Alice holds a private subset of some ground set, and Bob holds an element in the ground set and they wish to determine whether Bob's element is within Alice's subset, without revealing to either of them information about the other party's input beyond the above described inclusion.

SECURE PROTOCOL FOR ASSOCIATION RULES

Once the set Fs of all s-frequent item sets is found, we may proceed to look for all (s,c)-association rules (rules with support at least sN and confidence at least c). For X, Y $\in F_s$, where X $\bigcap$ Y = θ, the corresponding association rule X) Y has confidence at least c if and only if supp(X $\cup$ Y)/supp(X) $\geq$ c, or, equivalently,

$$C_{X,Y} := \sum_{m=1}^{M} \left(\sup p_m\left(X \cup Y\right) - c.\sup p_m\left(x\right)\right) \geq 0 \quad (1)$$

Since $|C_{X,Y}| \leq N$, then by taking q = 2N + 1, the players can verify inequality (1) in parallel for all candidate association rules. In order to derive from Fs all (s,c)-association rules in an efficient manner we rely upon the following straightforward lemma. We first find all (s, c)-rules with 1-consequents; namely, all (s, c)-rules X $\Rightarrow$ Y with a consequent Y of size 1. To that end, we scan all item sets Z $\cup$ F_s of size |Z| $\geq$ 2, and for each such item set we scan all |Z| partitions Z = X $\Rightarrow$ Y where |Y| = 1 and X = Z \ Y . The association rule X $\Rightarrow$ Y that corresponds to such a given partition Z = X $\bigcup$ Y is tested to see whether it satisfies inequality (1). We may test all those candidate rules in parallel and at the end we get the full list of all (s, c)-rules with 1-consequents.

We then proceed by induction; assume that we found all (s,c)-rules with j-consequents for all 1 $\geq$ j $\leq$ l - 1. To find all (s, c) -rules with '-consequents, we rely upon Lemma 4.1. Namely, if Z $\in$ Fs and Z = X $\bigcup$ Y where X $\bigcap$ Y = θ and |Y| = l, then X $\Rightarrow$ Y is an (s, c)-rule only if X $\Rightarrow$ Y' were found to be (s,c)-rules for all Y 'Y. Hence, we may create all candidate rules with l-consequents and test them against inequality (1) in parallel. It should be noted that in practice, one usually aims at finding association rules of the form X $\Rightarrow$ Y where

|Y | = 1, or at least |Y| $\leq l$ 'for some small constant'. The above procedure may be continued until all candidate association rules with no upper bounds on the consequent size.

As noted in the players may dispense the local pruning and union computation in the FDM algorithm and instead, test all candidate item sets in $Ap(F_s^{k-1})$ to see which of them are globally s-frequent. Such a protocol is fully secure, as it reveals only the set of globally s-frequent item sets but no further information about the partial databases. Such a protocol would be much more costly since it requires each player to compute the local support of $|Ap(F_s^{k-1})|$ item sets instead of only $|C_s^k|$ item sets. In addition, the players will have to execute the secure comparison protocol to verify inequality for $|Ap(F_s^{k-1})|$ rather than only $|Ck_s^k|$ item sets. Both types of added operations are very costly: the time to compute the support size depends linearly on the size of the database, while the secure comparison protocol entails a costly oblivious transfer sub-protocol. Since, $|Ap(F_s^{k-1})|$ is much larger than $|C_s^k|$, the added computing time in such a protocol is expected to dominate the cost of the secure computation of the union of all locally s-frequent item sets. Hence, the enhanced security offered by such a protocol is accompanied by increased implementation costs.

PROBLEM DEFINITION

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 following problems are identified from the privacy preserved distributed rule mining system.

- Vertical partition based rule mining is not supported
- Subgroup identification is not tuned for partitioned database model
- Communication and computational cost is high
- Security leakage control is not provided

SECURED MULTI PARTY DATA SHARING AND RULE MINING

The system is designed to perform secure rule mining under horizontal and vertical partitioned databases. Data communication process is performed with item set group exchange mechanism. Distributed rule mining is performed under multiparty environment. The system is divided into six major modules. They are partitioned databases, item set generation, local frequent item sets, communication process, HP based rule mining (HPRM) and VP based rule mining (VPRM). Database partitions module is used to manage data in partitioned way. Candidate set and item sets are prepared under item set generation module. Local frequent item set mining is performed to fetch frequent patterns in each partitions. Item set data exchange is performed under the communication process module. HPRM module is used to fetch frequent patterns in horizontal partitioned database environment. The VPRM module is used to fetch frequent rules in vertical partitioned environment.

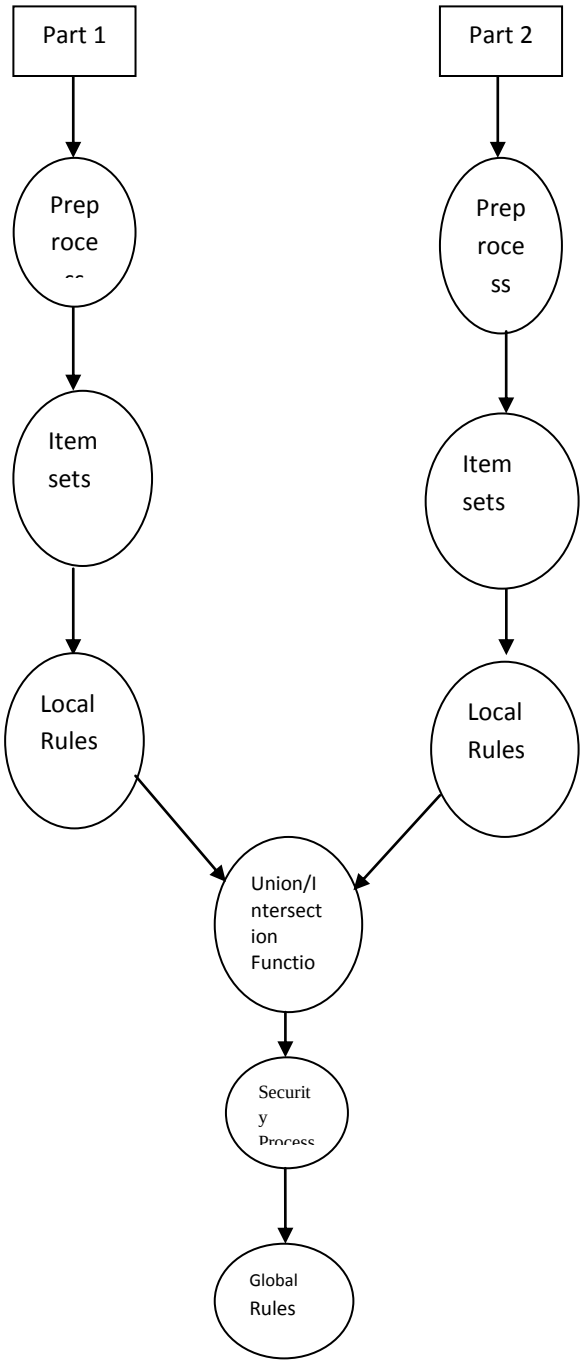

Fig. No: 8.1. Secured Rule Mining Under Multi Party Environment

Databases are maintained in different systems. Databases are partitioned with

homogeneous and heterogeneous structures. Data preprocess is performed to handle noisy transactions. Attribute sensitivity information is also maintained in the databases. Attributes and their values are used to generate candidate sets. Item sets are generated using the candidate sets. Frequency values are estimated for the candidate set and item sets. Support and confidence values are estimated using the frequency values.

Local frequent item set mining process is carried out in each database environment. Minimum support and minimum confidence values are used to filter the rules. Computational functions are used to protect the sensitive attributes. Item sets with sensitive attributes are also protected by the system.

The communication process is designed to manage the data exchange process. Item set groups are exchanged between the parties. Locally frequent item sets are transferred between the parties. Data communication process is tuned for the homogeneous and heterogeneous structure environment. Horizontal Partition based Rule Mining (HPRM) is carried out on homogeneous database environment. Unifying lists of locally Frequent Itemsets Kantarcioglu and Clifton (UniFI-KC) algorithm is used for the rule mining process. Secure computation of threshold functions and set inclusion computation function are used to secure the item sets. The integrated item set collection is used for the global rule mining process. The Vertical Partition based Rule Mining (VPRM) is carried out under heterogeneous database structure environment. Attributes are partitioned and stored in different databases. The UniFI-KC mechanism is tuned for the heterogeneous data environment. Global level item set construction mechanism is used for the rule mining process.

CONCLUSION

Attribute behavior identification process is tuned for the distributed environment. Privacy protection process is integrated with the attribute behavior identification process. 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]. M. Shaneck, Y. Kim, and V. Kumar. Privacy preserving nearest neighbor search. In ICDM Workshops, pages 541–545, 2006.

[2]. M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Anonymity preserving pattern discovery. The VLDB Journal, 17(4): 2008.

[3]. A. Machanavajjhala, J. Gehrke and D. Kifer, "'l-Diversity: Privacy beyond k-Anonymity," Proc. IEEE 22nd Int'l Conf. Data Eng. 2006.

[4]. Dima Alhadidi and Mourad Debbabi, "Secure Two-Party Differentially Private Data Release for Vertically Partitioned Data", IEEE Transactions On Dependable And Secure Computing, January/February 2014

[5]. S. Zhong, Z. Yang, and R.N. Wright, "Privacy-Enhancing KAnonymization of Customer Data," Proc. 24th ACM SIGMODSIGACT- SIGART Symp. Principles of Database Systems pp. 139-147, 2005.

[6]. N. Li and T. Li, "T-Closeness: Privacy Beyond K-Anonymity and L-Diversity," Proc. IEEE 23rd Int'l Conf. Data Eng, Apr. 2007.

[7]. G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast Data Anonymization with Low Information Loss," Proc. 33rd Int'l Conf. Very Large Data Bases, pp. 758-769, 2007.

[8]. L. Shundong, D. Yiqi,W. Daoshun, and L. Ping. Symmetric encryption solutions to millionaire's problem and its extension. In 1st International Conference on Digital Information Management, 2006.

[9]. H. Yu, X. Jiang and J. Vaidya. Privacy-preserving svm using nonlinear kernels on horizontally partitioned data. In SAC '06: Proceedings of the 2006.