IPHV8I1027X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

AN IMPROVED ALGORITHM FOR MINING ASSOCIATION RULES IN LARGE DATABASES

[#1]S.SUSHMITHA, M.Tech Student,
[#2]K.SADANANDAM,Assistant Professor,
Dept of CSE,
SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR, T.S., INDIA.

Abstract: The main aim of privacy is to get the global result without affecting on security. Security and privacy (confidentiality) is of utmost importance in any kind of large scale data-mining, especially where the corporate are involved as parties. Here we overview & introduce a privacy-preserving algorithm for horizontally partitioned data distributed over two or more parties. Privacy concerns may prevent the parties from directly sharing the data, and some types of information about the data. This paper addresses secure mining of association rules over horizontally partitioned data. The methods incorporate cryptographic techniques to minimize the information shared, while adding little overhead to the mining task. The proposed is simple, yet powerful, methods to generate SQL code to return aggregated columns in a horizontal tabular layout, returning a set of numbers instead of one number per row. This new class of functions is called horizontal aggregations. Horizontal aggregations build data sets with a horizontal de normalized layout (e.g. point-dimension, observation-variable, instance-feature), which is the standard layout required by most data mining algorithms. The proposed method used three categories to evaluate horizontal aggregations: CASE: Exploiting the programming CASE construct; SPJ: Based on standard relational algebra operators (SPJ queries); PIVOT: Using the PIVOT operator, which is offered by some DBMSs. Experiments with large tables compare the proposed query evaluation methods. A CASE method has similar speed to the PIVOT operator and it is much faster than the SPJ method. In general, the CASE and PIVOT methods exhibit linear scalability, whereas the SPJ method does not.

Keywords: SPJ Queries, PIVOT, SQL Aggregations, CASE Method, Horizontal Aggregation.

I.INTRODUCITON

We study here the problem of secure mining of association rules in horizontally partitioned databases. In that setting, there are several sites (or players) 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, x1, . . . , xM, and they wish to securely compute y = f(x1, . . . , xM) 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 [32] was the first to propose a generic solution for this problem in the case of two players.

- The main part of the protocol is a sub-protocol for the secure computation of the union of private subsets that are held by the different players.
- The most costly part of the protocol and its implementation relies upon cryptographic primitives such as commutative encryption, oblivious transfer, and hash functions.
- In particular, our protocol does not depend on commutative encryption and oblivious transfer (what simplifies it significantly and contributes towards much reduced communication and computational costs). While our solution is still not perfectly secure, it leaks excess information only to a small number (three) of possible coalitions.

This paper addresses the problem of computing association rules within such a scenario. We assume 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 straight forward. In a relational database, especially with normalized tables, a significant effort is required to prepare a summary data set that can be used as input for a data mining or statistical algorithm. Most algorithms require as input a data set with a horizontal layout, with several Records and one variable or dimension per column. That is the case with models like clustering, classification, regression and PCA; consult. Each research discipline uses different terminology to describe the data set. In data mining the common terms are point-dimension. Statistics literature generally uses observation-variable. Machine learning research uses instance-feature. This paper introduces a new class of aggregate functions that can be used to build data sets in a horizontal layout (de normalized with aggregations), automating SQL query writing and extending SQL capabilities. We show evaluating horizontal aggregations is a challenging and interesting problem and we introduced alternative methods and optimizations for their efficient evaluation.

II.LITERATURE SURVEY

We study here the problem of secure mining of association rules in horizontally partitioned databases. In that setting, there are several sites (or players) 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 given minimal support and confidence levels that hold in the unified database, while minimizing the information disclosed about the private databases held by those players.

That goal defines a problem of secure multi-party computation. In such problems, there are M players that hold private inputs, x1, . . . , xM, and they wish to securely compute y = f(x1, . . . , xM) 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. Other generic solutions, for the multi-party case, were later proposed in [2, 4, 10].2 T. Tassa In our problem, the inputs are the partial databases, and the required out-put is the list of association rules with given support and confidence. 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 (see examples of such protocols in e.g. [12, 20, 23]). Kantarcioglu and Clifton studied that problem in [12] and devised a protocol for its solution. The main part of the protocol is a sub-protocol for the secure computation of the union of private subsets that are held by the different players.

(Those subsets include candidate item sets, as we explain below.) That is the most costly part of the protocol and its implementation relies upon cryptographic primitives such as commutative encryption, oblivious transfer, and hash functions. This is also the only part in the protocol in which the players may extract from their view of the protocol information on other databases, beyond what is implied by the final output and their own input. While such leakage of information renders the protocol not perfectly secure, the perimeter of the excess information is explicitly bounded in and it is argued that such information leakage is innocuous, whence acceptable from practical point of view. Herein we propose an alternative protocol for the secure computation of the union of private subsets.

The proposed protocol improves upon that in terms of simplicity and efficiency as well as privacy. In particular, our protocol does not depend on commutative encryption and oblivious transfer (what simplifies it significantly and contributes towards reduced communication and computational costs). 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 multi-party computation that we solve here as part of our discussion is the problem of determining whether an element held by one player is included in a subset held by another.

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external suppor. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account

IPHV8I1027X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

for developing the proposed system. As horizontal aggregations are capable of producing data sets that can be used for real world data mining activities.

III.EXISTING MEHODOLOGY

That goal defines a problem of secure multi-party computation. In such problems, there are M players that hold private inputs, x1, . . . , xM, and they wish to securely compute y = f(x1, . . . , xM) 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. Other generic solutions, for the multi-party case, were later proposed in.

IV.PROPOSED WORK

Assumption for the proposed work are taken as the database is horizontally partitioned and distributed among sites and the total number of sites is greater than two. The sites are considered as trusted site and all the site contain their own private data and no other site will be able to know other site data .In this method, basically, hash based secure sum technique [7] has been used. In secure sum each site will determine their own data value and send to predecessor site that near to original site and this goes on till the original site collects all the value of data after that the parent site will determine the global support and global confidence [6] [10] and it also not necessary that the result found is globally frequent or infrequent depending on value which will create after collecting all the value. We have considered four sites s1, s2, s3, s4 where the sites are interchanging its position with another by following the algorithm. The secure sum protocol [9] is based on changing neighbors in each round of segment computation. The number of the site s1 is selected as the protocol initiator site which starts the computation by distributing the first data segment. The site traverses towards sn in each round of the computation. The number of parties for this protocol must be four or more. When all the rounds of segments summation are completed the sum is announced by the protocol initiator site. The steps are as follows.

V. ASSOCIATION RULE MINING

Association rule mining discovers the frequent patterns among the itemsets. It aims to extract interesting associations, frequent patterns, and correlations among sets of items in the data repositories [9]. For Example, In a Laptop store in India, 80% of the customers who are buying Laptop computers also buy Data card for internet and pen drive for data portability.

The formal statement of Association rule mining problem was initially specified by Agrawal [2]. Let I = *I*1, *I*2, … , *Im* be a set of *m* different attributes, T be the transaction that comprises a set of items such that T □ I, D be a database with different transactions Ts. An association rule is an insinuation in the form of X □ Y, where X, Y □ I are sets of items termed itemsets, and X □ Y = □. X is named antecedent. Y is called consequent. The rule means X implies Y.

The two significant basic measures of association rules are *support*(*s*) and *confidence*(*c*). Since the database is enormous in size, users concern about only the frequently bought items. The users can pre-define thresholds of support and confidence to drop the rules which are not so useful. The two thresholds are named

minimal support and *minimal confidence* [20].

Support(*s*) is defined as the proportion of records that contain X □ Y to the overall records in the database. The amount for each item is augmented by one, whenever the item is crossed over in different transaction in database during the course of the scanning.

Confidence(c) *is* defined as the proportion of the number of transactions that contain X □ Y to the overall records that contain X, where, if the ratio outperforms the threshold of confidence, an association rule X □ Y can be generated.

Confidence is a degree of strength of the association rules, if the confidence of the association rule X □ Y is 80 per cent, it infers that 80 per cent of the transactions that have X also comprise Y together, likewise to confirm the interestingness of the rules specified minimum confidence is also pre-defined by users. Association rule mining is to discover association rules that fulfil the pre-defined minimum support and confidence [1]. The problem is subdivided into two sub problems. The first one is to find the item sets which existences surpass a predefined threshold, usually called frequent item sets. The next one is to generate association rules from large itemsets with the limitations of minimal confidence. If one of the large itemsets is *Lk*, *Lk*= {*I*1, *I*2, …, *I*k-1, *Ik}*, then association rules are generated with those itemsets. Checking the confidence with the rule {*I*1, *I*2, …, *I*k-1} □ {*Ik*}, it can be decided for interestingness. By deleting the last items, the other rules are

created in the antecedent and placing it to the consequent, then the confidences of the new rules are checked to decide the interestingness. The processes iterated till the antecedent becomes empty. The main sub problem can be two folded into *candidate large itemsets* generation process and *frequent itemsets* generation process. Those itemsets whose support exceeds the support threshold called as *large* or *frequent itemsets*, those itemsets that are expected to be large or frequent are known *candidate itemsets*. An efficient model has classification rules with high confidence and large support [28].

VI.DATA MINING TECHNIQUES

The most commonly used techniques in data mining are:

1.Clustering: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.

2.Associations Rule: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.

3. Sequential patterns: Data is mined to anticipate behavior patterns and trends. For example, an outdoor

Equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

4.Artificial neural networks: Non-linear predictive models that learn through training and resemble biological neural networks in structure.

5.Genetic algorithms: Optimization techniques that use processes such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.

6.Decision trees: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome.

7.Nearest neighbor method: A technique that classifies each record in a dataset based on a combination of then classes of the k record(s) most similar to it in a historical dataset (where k 1). Sometimes called the k-nearest neighbor technique.

8.Rule induction: The extraction of useful if-then rules from data based on statistical significance.

9.Data visualization: The visual interpretation of com-plex relationships in multidimensional data. Graphics tools are used to illustrate data relationships. There are three method used as follows:

1 .SPJ Method: The SPJ method is interesting from a theoretical point of view because it is based on relational operators only. The basic idea is to create one table with a Vertical aggregation for each result column, and then join all those tables to produce FH.

2. CASE Method: This method uses the case programming construct available in SQL. The case statement returns a value selected from a set of values based on boolean expressions. From a relational database theo-ry point of view this is equivalent to doing a simple pro-jection/ aggregation query where each non – key value is given by a function t hat returns a number based on some conjunction of conditions.

3. PIVOT Method: The PIVOT Method used PIVOT operator which is a built in operator in a commercial DBMS. Since this operator can perform transposition it can help evaluating horizontal aggregations. The PIVOT method internally needs to determine how many col-umns are needed to store the transposed table and it can be combined with the GROUP BY clause.

VII.CONCLUSION

We proposed a protocol for secure mining of association rules in horizontally distributed databases that improves significantly upon the current leading protocol in terms of privacy and efficiency. One of the main ingredients in our proposed protocol is a novel secure multi-party protocol for computing the union (or intersection) of private subsets that each of the interacting players hold. Another ingredient is a protocol that tests the inclusion of an element held by one player in a subset held by another.

The latter protocol exploits the fact that the underlying problem is of interest only when the number of players is greater than two. One research problem that this study suggests was described in Section 3 namely, to devise an efficient protocol for set inclusion verification that uses the existence of a semi-honest third party

REFERENCES:

[1]R. Agrawal and R. Srikant, "Fast algorithms for min ing association rules," in Proceedings of the 20th Inter-national Co nference on Very Large Data Bases. Santia-go, Chile: VLDB, Sept. 12-15 1994, pp. 487– 499. [Online]. http://www.vldb.org/dblp/db/conf/vldb/ vldb94-487. html

[2]D. W.-L. Cheung, J. Han, V. Ng, A. W.-C. Fu, and Y. Fu,

"A fast distributed algorithm for mining association rules," in Proceedings of the 1996 International Confer-ence on Parallel and Distributed Information Sys-tems (PDIS'96). Miami Beach, Florida, USA: IEEE, Dec. 1996, pp. 31–42.

[3]D. W.-L. Cheung, V. Ng, A. W.-C. Fu, and Y. Fu, "Ef ficient mining of association rules in distributed data-bases," IEEE T rans. Knowledge Data Eng., vol. 8, no. 6, pp. 911–922, Dec. 1996. R. Agrawal and R. Srikant, "Priva-cypreserving data mining," in Proceedings of the 2000 ACM SIGMOD Conference on Management of Data. Dallas, TX: ACM, May 14-19 2000, pp. 439–4 50. [Online]. Available: http://doi.acm.org/10.1145/342009.335438

[4]D. Agrawal and C. C. Aggarwal, "On the design and q uantification of privacy preserving data mining algo-rithms," in Proc eedings of the Twentieth ACM SIGACT-SIGMODSIGART Symposium on Principles of Database Systems. Santa Barbara, California, USA: ACM, May 21- 23 2001, pp. 247–255. [Online]. Available: h ttp: //doi. acm.org/10.1145/375551.375602

[5]A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, "Privacy preserving mining of association rules," in TheEig hth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Al-berta, Canada, July 23-26 2002, pp. 217–228. [Online]. Available: http://doi.acm.org/10.1145/775047.775080.

[6]S. J. Rizvi and J. R. Haritsa, "Maintaining data pr ivacy in association rule mining," in Proceedings of 28th International Conference on Very Large Data Bas-es. Hong Kong: VLDB, Aug. 20-23 2002, pp. 682–693. [Online]. Available: http://www.vldb.org/conf/2002/ S19P03.pdf

[7]Y. Lindell and B. Pinkas, "Privacy preserving data mining," in Advances in Cryptology – CRYPTO 2000. Springer-Verl ag, Aug. 20-24 2000, pp. 36–54. [On-line]. Available: http://lin k.springer.de/link/ service/ series/0558/bibs/1880/18800036.htm

[8]O. Goldreich, "Secure multi-party computation," Sep t. 1998, (working draft). [Online]. Available: http:// www.wisdom.weizmann.ac.il/oded/ pp.html

[9]. Vaidya and C. Clifton, "Privacy preserving associ ation rule mining in vertically partitioned data," in The Eighth ACM SIG KDD International Conference on Knowledge Discovery and Data Mining, Edmonton,

[10]Alberta, Canada, July 23-26 2002, pp. 639–644. [Onl ine]. Available: http://doi.acm. org/10.1145/775047.775142

[11]A. C. Yao, "How to generate and exchange secrets," in Proceedings of the 27th IEEE Symposium on Founda-tions of Computer Science. IEEE, 1986, pp. 162– 167.

[12]I. Ioannidis and A. Grama, "An efficient protocol f or yao's millionaires' problem," in Hawaii International Con ference on System Sciences (HICSS-36), Waikoloa Village, Hawaii, Jan. 6-9 2003.

[13]O. Goldreich, "Encryption schemes," Mar. 2003, (wor king draft). [Online]. Available: http://www.wis-dom. weizmann.ac.il/oded PSBookFrag/enc.ps.

[14]R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key crypto-systems," C ommunications of the ACM, vol. 21, no. 2, pp. 120–126, 1978. [Online]. Available: http://doi.acm. org/10.1145/359340.359342