ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)

Survey on Association Rule Mining for Horizontally Distributed Databases

[I]Dr.P.Sumitra, [II]S.Kokila
[I]Assistant Professor, [II]M.Phil Research Scholar
[I,II]Dept. of Computer Science and Applications, Vivekanandha College of Arts and Sciences for Women (Autonomous), Elayampalayam, Tiruchengode, Tamil Nadu, India.

Abstract
Data mining is the automatic extraction of previously unknown patterns from the database. In parliamentary law to better service the demands of privacy preserving in mining association rule given.In a horizontally distributed database, the transactions redistributed amongnsites. The global support count of anitem set is the sum of all the local support counts. Anitem setX is globally supported if the global support count of X is bigger thans% of the total transaction database size. Ak -item set is called a globally largek -item set if it is globally supported. In this paper analysis the protocols for secure association rule on the allotted database. The current leading protocol is that of Kantarcioglu and Clifton. It is grounded on the Fast Distributed Mining (FDM) algorithm of Cheung et al., which is an unsecured distributed version of the Apriori algorithm.In summation, it is simpler and is significantly more effective in terms of communication rounds, communication cost and computational cost

Keywords
Privacy Preserving Data Mining; Distributed Computation; Frequent Item sets; Association Rules.

I. Introduction

Data mining concepts have been brought in successfully to retrieve knowledge in order to sustain a diversity of domains, marketing, weather prediction, medical diagnosis, and national security. Merely it is even a challenge to mine the data by protecting the private database of users.

Most organizations want information about individuals for their own specific demands.We consider here the problem of safe mining of association rules in vertically partitioned databases. Association rule mining is an active data mining research area. Nevertheless, most ARM algorithms cater to a centralized environment.Current technology for mining data typically applies to information stored centrally .

A fundamental component of many algorithms for mining association rules in large data sets is a subroutine that is to find so called frequent item sets.The frequent item sets are very heavy due to transactions data increasing. The existing approach Fast Distributed Mining (FDM) algorithm is an unsecured distributed version of the Apriorialgorithm.

The distributed mining algorithms can be practiced in distributed databases, as considerably as for mining large databases by partitioning them between sites and processing them in a diffused way. The high flexibility, the scalability, the small cost/performance ratio and the connectivity of a distributed system make them an ideal platform for data mining .

An association rule is a formula which involves a certain association relationship among a lot of objects in a database.Since finding interesting association rules in a database may reveal some useful rules for decision support, selective marketing, financial forecast, medical diagnosis and many other applications it has drawn a great deal of attention in recent data mining research[14].

With the existence of many large transaction databases, huge amount of data, high scalability of distributed system, and the easy partition and distribution of a centralized database, it is significant to investigate effective methods for distributed mining of association rule.Association rule mining finds interesting association or correlation relationships among a large band of information points.Sensitive data are a section of every large organization's normal business pattern.

Allowing sensitive data from production applications to be replicated and used for development and testing environments increases the potential for theft, loss or exposure -- hence increasing the organization's risk.Data masking is emerging as a best practice for obfuscating real data then it can be safely applied in non-production environments.This helps organizations meet compliance requirements for PCI, HIPAA, GLBA and other data privacy rules .

II. Related Work

D. Beaver, S. Micali, and P. Rogaway has presented a protocol for safe mining of association rules in horizontally distributed databases.The protocol is planned based on Fast Distributed Mining (FDM) Algorithm and Secure MultiParty Algorithm.The Protocol offers enhanced privacy with regard to the protocol in [4].In accession to that, it is simpler and is significantly more effective in terms of communication rounds, computational cost and communication cost.Among the DM algorithms of interest to us is the discovery of association principles.The trouble of discovering association rules was introduced by R. Agrawal et.al. [1]

R. Srikant and R. Agrawal. Considered the problem of applying the Apriori algorithm to a more general class of attributes which could be either quantitative (e.g. Age, income) or flat (e.g.Zip code, make of car) when dealing with distributed data sets, one significant topic is the heterogeneity of the data.There has been considerable study in thedatabase field dealing with heterogeneous databases and while we realize this as an important issue, we are not looking into it right away. D. Agrawal and A. El Abaddi .

A.Ben-David, N. Nisan and B. Pinkas, have proposed a system for Multi Party Computation. Secure computation is one of the big achievements of modern cryptography, enabling a set of untrusted parties to compute any function of their private inputs while disclosing nothing but the result of the function. They have presented FairplayMP, a generic system for Secure Multiparty Computation. This is an extension of the Fair play system which supported secure computation by two parties. The reference to the multi-party case is required for cryptographic protocols for the multi-party scenario are entirely different from protocols for

the two-party case [3].

M. Kantarcioglu , have presented a protocol for Privacy-Preserving distributed mining of Association Rules on Horizontally Partioneddata [18]. The report addresses the problem of computing association rules where the data may be spread among various custodians, none of which are permitted to transmit their data to some other situation[6].Databases are homogeneous where all websites deliver the same scheme but each site receives information on different entities.Association Rules have been computed based on Fast Distributed Algorithm (FDM) and Secure Multiparty computation [20].

JaideepVaidya,have suggested a protocol for privacy preserving Association Rule Mining in Vertically Partitioned Data. [19] The protocol is carried out through a two-party algorithm for efficiently discovering frequent item sets with minimum support levels, without either site communicating individual transaction values. They have shown that it is potential to achieve good individual security with communication cost comparable to that required to build a centralized data warehouse.

III. Preliminaries

Definitions and notations

Let D be a transaction database. As in [18], we view D as abinary matrix of N rows and L columns, where each row isa transaction over some set of items,$A = \{a1, \ldots, aL\}$, and each column represents one of the items in A. (In other words,the (i, j)th entry of D equals 1 if the i^{th} transaction includesthe item a_j, and 0 otherwise.) The database D is partitioned horizontally between M players, denoted $P1, \ldots, PM$. Player $P_m$holds the partial database Dmthat contains $N_m = |D_m|$ of the transactions in D, $1 \le m \le M$. The unified database is $D = D1 \cup \cdots \cup DM$, and it includes $N := \Sigma M\, m=1\, Nm$ transactions. Anitem 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, $suppm(X)$, is the number of transactions in Dmthat contain it. Clearly, $supp(X) = \Sigma Mm=1\, suppm(X)$. Let $s$be a real number between 0 and 1 that stands for a required support threshold. An itemset$X$is called s-frequent if $supp(X) \ge sN$. It is called locally s-frequent at $D_m$if $suppm(X) \ge sNm$.For each $1 \le k \le L$, let $Fk\,s$ denote the set of all k-itemsets (namely, itemsets of size k) that are s-frequent, and $Fk,m\,s$ be the set of all k-itemsets that are locally s-frequent at Dm,$1 \le m \le M$. Our main computational goal is to find, for a given threshold support $0 < s \le 1$, the set of all s-frequent itemsets, $Fs := \cup Lk=1\, Fk\,s$. 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. (Recall that if X and Y are two disjoint subsets of A, the support of the corresponding association rule $X \Rightarrow Y$ is $supp(X \cup Y)$ and its confidence is $supp(X \cup Y)/supp(X)$.)

The Fast Distributed Mining algorithm

The protocol of [19], as well as ours, are based on theFast Distributed Mining (FDM) algorithm of Cheung et al. [8], which is an unsecured distributed version of the Apriorialgorithm. Its main idea is that any s-frequent itemset mustbe also locally s-frequent in at least one of the sites. Hence, in order to find all globally s-frequent item sets, each playerreveals 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 alreadyjointly calculated $Fk-1s$. The goal is to proceed andcalculateFks.

(2) Candidate Sets Generation: Each player Pm computesthe set of all $(k - 1)$-itemsets that are locally frequent inhis site and also globally frequent; namely, Pm computesthe set $Fk-1,ms \cap Fk-1s$. He then applies on that set theApriori algorithm in order to generate the set Bk,msofcandidate$k$-itemsets [24].

(3) Local Pruning: For each $X \in Bk,ms$, Pm computes$suppm(X)$. He then retains only those itemsets that arelocally$s$-frequent. We denote this collection of itemsetsbyCk,ms.

(4) Unifying the candidate itemsets: Each player broadcastshis$Ck,ms$and then all players compute $Ck\,s := Mm=1\, Ck,ms$.

(5) Computing local supports. All players compute the localsupports of all itemsets in Cks.

(6) Broadcast Mining Results: Each player broadcasts thelocal supports that he computed. From that, everyone cancompute the global support of every itemset in Cks.Finally,$Fks$is the subset of Cksthat consists of all globally $s$frequent$k$-itemsets.

In the first iteration, when $k = 1$, the set $C1,ms$that the $m$thplayer computes (Steps 2-3) is just $F1,ms$, namely, the set ofsingle items that are s-frequent in Dm. The complete FDMalgorithm starts by finding all single items that are globally$s$-frequent. It then proceeds to find all 2-itemsets 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. Secure Computation of All Locally Frequent Itemsets

Protocol 1 is the protocol that was suggested by Kantarcioglu [19] for computing the unified list of all locally frequent itemsets, $Ck\,s = \cup Mm=1\, Ck,m\,s$, without disclosing the sizes of the subsets Ck,msnor their contents. The protocol is applied when the players already know $Fk-1\,s$ — the set of all$(k-1)$-item sets that are globally s-frequent, and they wish to proceed and compute Fks. We refer to it hereinafter as Protocol UNIFI-KC (Unifying lists of locally Frequent Item sets — Kantarcioglu and Clifton).

The input that each player Pm has at the beginning of Protocol UNIFI-KC is the collection $Ck,m\,s$, as defined in Steps 2-3 of the FDM algorithm. Let $Ap(Fk-1\,s)$ denote the set of all candidate k-itemsets that the Apriori algorithm generates from $Fk-1\,s$. Then, as implied by the definition of $Ck,m\,s$ (see Section 1.1.2), $Ck,m\,s$, $1 \le m \le M$, are all subsets of $Ap(Fk-1\,s)$. The output of the protocol is the union $Ck\,s = \cup M\, m=1\, Ck,m\,s$. In the first iteration of this computation $k = 1$, and the players compute all s-frequent 1-itemsets (here F_0 s= $\{\}$). In the next iteration they compute all s-frequent 2-itemsets, and so forth, until the first $k \le L$ in which they find no s-frequent k-itemsets.After computing that union, the players proceed to extract from $Ck\,s$ the subset $Fk\,s$ that consists of all k-itemsets that are globally s-frequent; this is done using the protocol that we describe later on in Section 3. Finally, by applying the above described procedure from $k = 1$ until the first value of $k \le L$ for which the resulting set $Fk\,s$ is empty, the players may recover the full set $Fs := \cup L\, k=1\, Fk$of all globally s-frequent itemsets. Protocol UNIFI-KC works as follows: First, each player adds to his private subset $Ck,m\,s$ fake item sets, in order to hide its size. Then, the players jointly compute the encryption of their private subsets by applying on those subsets a commutative

encryption1, where each player adds, in his turn, his own layer of encryption using his private secret key . At the end of that stage, every itemset in each subset is encrypted by all 1. An encryption algorithm is called commutative if $EK1$
$\circ EK2 = EK2 \circ EK1$ for any pair of keys $K1$ and $K2$. 4 of the players; the usage of a commutative encryption scheme ensures that all item sets are, eventually, encrypted in the same manner. Then, they compute the union of those subsets in their encrypted form. Finally, they decrypt the union set and remove from it item sets which are identified as fake. We now proceed to describe the protocol in detail. (Notation agreement: Since all protocols that we present herein involve cyclic communication rounds, the index $M+1$ always means 1, while the index 0

V. Methodology

Kantarcioglu and Clifton studied that problem and got up a protocol for its resolution.The principal component of the protocol is a sub-protocol for the secure computation of the unification of private subsets that are controlled by the different players [19]. The private subset of a given player includes the detail sets that are so-frequent in his partial database.

That is the most costly piece of the protocol and its implementation relies upon cryptographic primitives such as commutative encryption, oblivious transfer, and hash functions. This is also the only portion in the protocol in which the players may pull from their perspective of the protocol information on other databases, beyond what is entailed by the final output and their own input [24].While such leakage of data provides the protocol not perfectly secure, the circumference of the redundant information is explicitly bounded and it is argued there that such information leakage is innocuous, whence acceptable from a pragmatic tip of opinion.Insufficient security, simplicity and efficiency are not well in the databases, not sure in privacy in an existing organization [2].While our answer is yet not perfectly safe, it leaks excess information only to a low number (three) of possible coalitions, unlike the protocol of that discloses information also to some single players .

Our protocol may leak is less sore than the excess information leaked by the protocol.The protocol that we propose here computes a parameterized family of subroutines, which we call threshold functions, in which the two extreme cases correspond to the problems of calculating the union and intersection of private subsets.Those are in fact general-purpose protocols that can be applied in other settings as well.

Another problem of secure multiparty computation that we work out here as part of our discussion is the set inclusion problem; namely, the problem where Alice has got a secret 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 discovering to either of them information about the other party's input beyond the above described inclusion.We suggested a protocol for safe mining of association rules in horizontally distributed databases that improves significantly upon the current leading protocol in terms of secrecy and efficiency [25].

The primary component in our suggested protocol is a novel secure multi-party protocol for calculating the union (or convergence) of private subsets that each of the interacting players holds.

1. Privacy Preserving Data Mining
2. Distributed Computation
3. Frequent Itemsets
4. Association Rules

Privacy Preserving Data Mining

One, in which the data owner and the data miner are two different entities, and another, in which the information is spread among various parties who propose to jointly perform data mining on the unified corpus of information that they have.In the first context, the goal is to protect the data records from the data miner [22].Hence, the data owner aims at anonym zing the data prior to its expiration. The primary approach in this context is to apply data perturbation. The idea is that aggression breeds aggression.Computation and communication costs versus the number of transactions N the perturbed data can be utilized to understand general trends in the data, without revealing original record data.

In the second setting, the goal is to perform data mining while protecting the data records of each of the data owners of the other data owners[28]. This is a problem of secure multiparty computation. The common plan of attack here is cryptographic rather than probabilities.

Distributed Computation

We likened the operation of two secure implementations of the FDM algorithm, Section In the first implementation (denoted FDM-KC), we did the unification step using Protocol UNIFI-KC, where the commutative cipher was 1024-bit RSA in the second implementation (denoted FDM) we used our Protocol UNIFI, where the keyed-hash function was HMAC. In both implementations, we implemented Step 5 of the FDM algorithm in the most dependable fashion that was traced in later.

We examined the two implementations with regard to three meters:

1) Total computation time of the complete protocols (FDMKC and FDM) over all actors.That amount includes the Apriori computation time, and the time to identify the globally s-frequent item sets, as drawn in later.
2) Total computation time of the unification protocols only (UNIFI-KC and UNIFI) over all actors.
3) Total message size. We ran three experiment sets, where each set tested the dependence of the above measures on a different parameter: • N — the number of minutes in the interconnected database.

Frequent Itemsets

We report here the solution that was proposed by Kantarcioglu and Clifton.They saw two possible mounts.If the required output includes all globally s-frequent item sets, as easily as the sizes of their backups, then the values of Δ (x) can be brought out for all.In such a lawsuit, these values may be calculated using a secure summation protocol, where the private addend of Pm is spam (x) − Sm. The more interesting setting, however, is the one where the support sizes are not part of the required output. We go on to talk about it.

Association Rules

Once the set fees of all then-frequent itemsets is found, we may proceed to look for all (s, c) -association rules (rules to sustain at least sane and confidence at least c).

In order to derive from Fs all (s, c) -association rules in an efficient manner we rely upon the straightforward lemma

VI. Conclusion

We suggested a FDM protocol for safe mining of association rules in vertically distributed databases that improves significantly upon

the current leading protocol in terms of secrecy and efficiency.One of the principal factors in our suggested protocol is a novel secure multi-party protocol for calculating the union (or convergence) of private subsets that each of the interacting players hold.Another factor is a protocol that tests the inclusion of an element contained by one participant in a subset held by some other.Those protocols exploit the fact that the underlying problem is of interest only when the number of participants is greater than two.

References

[1] *R. Agrawal and R. Srikant. Fast algorithms for mining association rulesin large databases. In VLDB, pages 487–499, 1994.*

[2] *R. Agrawal and R. Srikant. Privacy-preserving data mining. In SIGMODConference, pages 439–450, 2000.*

[3] *A. Ben-David, N. Nisan, and B. Pinkas. FairplayMP - A system forsecure multi-party computation. In CCS, pages 257–266, 2008.*

[4] *D. Beaver, S. Micali, and P. Rogaway. The round complexity of secureprotocols. In STOC, pages 503–513, 1990.*

[5] *R. Srikant and R. Agrawal. Mining generalized association rules. InVLDB, pages 407–419, 1995.*

[6] *J.C. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret. In Crypto, pages 251–260, 1986.*

[7] *J. Brickell and V. Shmatikov. Privacy-preserving graph algorithms in the semi-honest model. In ASIACRYPT, pages 236–252, 2005.*

[8] *D.W.L. Cheung, J. Han, V.T.Y. Ng, A.W.C. Fu, and Y. Fu. A fast distributed algorithm for mining association rules. In PDIS, pages 31–42, 1996.*

[9] *D.W.L Cheung, V.T.Y. Ng, A.W.C. Fu, and Y. Fu. Efficient miningof association rules in distributed databases. IEEE Trans. Knowl. Data Eng., 8(6):911–922, 1996.*

[10] *T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31:469–472, 1985.*

[11] *A.V. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In KDD, pages 217–228, 2002.*

[12] *R. Fagin, M. Naor, and P. Winkler. Comparing Information WithoutLeaking It. Communications of the ACM, 39:77–85, 1996.*

[13] *J. Zhan, S. Matwin, and L. Chang. Privacy preserving collaborative association rule mining. In Data and Applications Security, pages 153–165, 2005.*

[14] *M.J. Freedman, K. Nissim, and B. Pinkas. Efficient private matchingand set intersection. In EUROCRYPT, pages 1–19, 2004.*

[15] *O. Goldreich, S. Micali, and A. Wigderson. How to play any mentalgame or A completeness theorem for protocols with honest majority. InSTOC, pages 218–229, 1987.*

[16] *H. Grosskreutz, B. Lemmen, and S. R¨uping. Secure distributed subgroupdiscovery in horizontally partitioned data. Transactions on Data Privacy,4:147–165, 2011.*

[17] *W. Jiang and C. Clifton. A secure distributed framework for achieving k-anonymity. The VLDB Journal, 15:316–273, 2006.*

[18] *M. Kantarcioglu and C. Clifton. Privacy-preserving distributed miningof association rules on horizontally partitioned data. IEEE Transactionson Knowledge and Data Engineering, 16:1026–1037, 2004.*

[19] *M. Kantarcioglu, R. Nix, and J. Vaidya. An efficient approximate protocol for privacy-preserving association rule mining. In PAKDD,pages 515–524, 2009.*

[20] *L. Kissner and D.X. Song. Privacy-preserving set operations. InCRYPTO, pages 241–257, 2005.*

[21] *X. Lin, C. Clifton, and M.Y. Zhu. Privacy-preserving clustering withdistributed EM mixture modeling. Knowl. Inf. Syst., 8:68–81, 2005.*

[22] *Y. Lindell and B. Pinkas. Privacy preserving data mining. In Crypto,pages 36–54, 2000.*

[23] *J.S. Park, M.S. Chen, and P.S. Yu. An effective hash based algorithmfor mining association rules. In SIGMOD Conference, pages 175–186,1995.*

[24] *S.C. Pohlig and M.E. Hellman. An improved algorithm for computing alogarithms over gf(p) and its cryptographic significance. IEEE Transactionson Information Theory, 24:106–110, 1978.*

[25] *A.C. Yao. Protocols for secure computation. In FOCS, pages 160–164,1982.*

[26] *A. Schuster, R. Wolff, and B. Gilburd. Privacy-preserving association rule mining in large-scale distributed systems. In CCGRID, pages 411–418, 2004.*

[27] *J. Zhan, S. Matwin, and L. Chang. Privacy preserving collaborative association rule mining. In Data and Applications Security, pages 153–165, 2005.*

[28] *S. Zhong, Z. Yang, and R.N. Wright. Privacy-enhancing k-anonymization of customer data. In PODS, pages 139–147, 2005.*