2012 International Conference on Software and Computer Applications (ICSCA 2012)
IPCSIT vol. 41 (2012) © (2012) IACSIT Press, Singapore

Privacy Preserving Mining of Association Rules on Horizontally Distributed Databases

Seyed Ziaeddin Alborzi[1+], Fatemeh Raji[2] and Mohammad H. Saraee[3]

[1]Computer Engineering School, Nanyang Technological University, Singapore

[2, 3] Dept. of Elec. and Computer Engineering, Isfahan University of Technology, Isfahan, Iran.

Abstract. Many algorithms have been proposed to provide privacy preserving in data mining. These protocols are based on two main approaches named as: the Randomization approach and the Cryptographic approach. The first one is based on perturbation of the valuable information while the second one uses cryptographic techniques. The randomization approach is much more efficient with reduced accuracy while the cryptographic approach can provide solutions with perfect accuracy. However, the cryptographic approach is a much slower method and requires considerable computation and communication overhead. In this paper, a new protocol is proposed which combines the advantages of the two previous approaches to perform privacy preserving in distributed mining of association rules. Both the privacy and performance characteristics of the proposed protocol are studied and compared with the randomization and cryptographic approaches. The approach introduced in this paper has great advantages, such as higher flexibility and resistance against conspiracy, over the similar methods.

Keywords: Association rules mining, Distributed Data Mining, Privacy, Cryptography

1. Introduction

A common approach for privacy preserving Data Mining is Association rules mining [1]. Association rules are used in order to discover the elements of a dataset which usually happen together, and in one sense their occurrence somehow related together. If it is assumed that a database includes a set of transitions and each transition includes an item set, then each rule that is achieved from this set has the total form of $X \rightarrow Y$ that $X \cap Y = \phi$ (X or Y includes a set of items). The rule shows that if the pattern X happens, then the pattern Y will most likely happen too. This rule has the *support* of s quantity, if s percent of the transitions of the database includes X and Y. Also this rule has the *confidence* of c quantity, if c percent of the database transitions that includes X, include Y as well. An itemset with the quantity of k items is called a k-item set. The purpose of a mining association rules algorithm is to find all of the rules that contain *support* with a larger quantity than *support_min* and *confidence* with a larger quantity than *confidence_min*. This mining takes place in 2 steps:

1. All of the large itemsets (*L-itemset*) are found. An itemset is large if its *support* is larger than *support_min.*
2. Using the found itemsets, association rules are found.

Considering the fact that association rules are easily achieved from *L-itemsets*, the efficiency of an association mining rule depends on the first step. For this reason, different algorithms have been concentrated on the presenting of an efficient method for it, that the most famous of them is Apriori [2]. Apriori does several searches on data to find the *L-itemsets*. This algorithm, in the (*k*)th search, finds all of the element sets that contain element *k* which is called the candidate sets of *k-itemset*. This candidate set is

[+] Corresponding author. Tel.: +65-90380348;
E-mail address: seyed1@e.ntu.edu.sg

achieved from all of the *(k-l)-itemset* elements that has been found in (*k-l*)th search. Apriori uses the quality that each subunit of an *L-itemset* is certainly an *L-itemset*.

This quality is useful in deleting the unlarge members of the candidate set. The order of the production of candidates is in a way that the candidate set of *k-itemset* is a *superset* for the elements of *k-itemset.* After the achievement of the candidate sets of *k-itemset*, the largeness of each itemset is checked. It means that if the *support* of a member of candidates is larger than *support-min*, this itemset is considered as the *L-itemset* of the (*k*)th search.

In many situations, database is saved in the form of distributed between two or more sites.In a concentrated data search model, it is supposed that all needed data for data search algorithm exists or at least could be sent to a central site. A simple way of doing data search in a distributed form is that all data be transformed to one site so that data search algorithm is applied on the collection of data. Apart from the protection of the privacy of data, this method will be so inefficient because there is need for the transformation of a large volume of data that is the transformation of the whole database to a central site.

The distribution of data could be considered in two ways. The first way is the way in which existing data in each database has different features which is called vertically partitioned. In the second way, data is distributed similarly among different bases. In other words, the features of all of the databases are the same. This condition is called horizontally partitioned which is used in this paper.

In horizontally partitioned condition, transitions are distributed among n sites and global support for an itemset like X is achieved from the total sum of local support in each site:

$$\mathrm{support}_x = \sum_{i=1}^{n} \mathrm{support}_x(i) = \frac{\sum_{i=1}^{n} \mathrm{support-count}_x(i)}{\sum_{i=1}^{n} \mathrm{database-size}(i)} \tag{1}$$

In[3], it has been proved that an item set in global form is large if it is large at least in one site in the local form, and the total sum of *support* in all sites is more than *support-min.*

$$\mathrm{support-count}_x \geq \mathrm{support}_{\min} * \mathrm{database-size} \tag{2}$$

The phrase (2) could be written as follows:

$$\begin{aligned} &\Rightarrow \mathrm{support-count}_x \geq \mathrm{support}_{\min} * \sum_{i=1}^{n} \mathrm{database-size}(i) \\ &\Rightarrow \sum_{i=1}^{n} \mathrm{support-count}_x(i) \geq \mathrm{support}_{\min} * \sum_{i=1}^{n} \mathrm{database-size}(i) \\ &\Rightarrow \sum_{i=1}^{n} (\mathrm{support-count}_x(i) - \mathrm{support}_{\min} * \mathrm{database-size}(i)) \geq 0 \end{aligned} \tag{3}$$

Therefore, the aim of a distributed association mining rules is to find all of the global large itemsets that is the itemsets that have global support with a quantity larger than *support-min*, and global confidence with a quantity larger than confidence-*min.* One of the best algorithms of distributed association mining rules is the FDM algorithm [4] which is based on Apriori. This algorithm has been designed for horizontally partitioned database.

FDM like Apriori dose several passes on data. In the (*k*)th search in each site, candidate set of *k-itemset* is made based on a collection of global large itemsets in (*k-l*)th search. It is supposed that *G.L-itemset* is a collection of global large itemsets, and *L.L-itemset* is a collection of local large itemsets. Then each site calculates the quantity of *support* so that it locally checks for each of its candidate members, the largeness of that itemset. After that each site spreads its *L.L-itemsets* so that thereafter, other sites can achieve the collection of *L.L-itemsets* of all of the sites. At the end, each site spreads the quantity of all the existing itemsets in the collection of *L.L-itemsets* so that each site can calculate the *G.L-itemset.*

2. The related works

In the field of encrypting, designing of a protocol is performed through considering behavior of participants while performing the protocol [5]. If one of the participants correctly performs the protocol however the percentage of analyses of interchanged message are during the performance so that through this method other information could be obtained therefore, this participant is semi-honest. On the other hand if a participant infringes correct performance of the protocol, it is called malicious. This type of participant gives incorrect inputs to the algorithm so it can obtain other valuable data [6,[**7**]. In the procedures that will be explained, the model of semi-honest participant is used. Additionally in all of these methods is assumed that to each site, a ascending number is allocated so that the site before and after any site is defined.

In the method concerning [3], several pass is performed on the data. In this method broadcasting (dispersing) L-itemset, as well as their supports, the commutative encryption has been employed through assistance of which it would be possible to compare data without showing their contents.

This procedure, due to employment of the commutative encryption in a large database in which the number of L.L.item sites is too large, needs a high line and space expenses and would be very inefficient. On the other hand this method cannot resist against conspiracy of the sites with each other. In [8] Two methods have been proposed. The first one is based on the trusted third party for improvement of which, the second method has been proposed. The second method has two phases in the first of which to find the *L.L-itemset*, two procedures have been employed. The procedure [9] is almost similar to the procedure [8] however it has been trusted from the third side and for the same reason, will not be considered in this section. In [8], after each L.L. site identifying itself, from which it finds the maximum frequent (large) item site. Then, similar to [3], using the commutative encryption, all members of the resultant set are decoded and the result is sent to the next site so that this set is coded by all the sites; then performs the decoding stage so that the set of all the maximum frequent (large) itemset relevant to all sites is obtained.

In the second phase, each site will generate a random optional number (Ri) and send the resultant random number to its next site. Then, the first site as the beginner, for each *L.L-itemset* resulting of maximum frequent (large) itemset, compute the amount of the following statement for *i = 1* and sends the result to the next site.

$$\text{support}_x(\text{i}) \text{ - } \text{support}_{\min} * |\,\text{database} - \text{size}(\text{i})\,| + \text{R}_\text{i} - \text{R}_{\text{i}-1+\text{n}} \qquad (4)$$

The second site adds amount of the statement (4) with that obtained from the first site and send the result to its next site. This trend is repeated until the last site, throng study of the total result, studies the largeness (frequently) of each itemset. It is clear that the volume of information employed in encoding in method [8] is much less compared with that of the method [3]. Because in method [8], the commutative encryption technique is used one time to transfer the *L.L-itemset*. One of the biggest problems of this approach is that each site knows the random number of its previous site, if the sites *i+1* and *i-1* conspire, in case of knowing size of the database, the of the ith site, support values relevant to different itemset of the *i*th site are revealed and confidentiality is lost [10, 13].

3. The proposed method

In this section, we propose a new and efficient method that the search of the independence rules is obtained while no site can become informed of *L.L-itemset* and *support* values of the other sites. On the other hand, these conditions would be maintained even in case of conspiracy of *n-2* sites with each other. Also, like other proposed methods, in this field, it is assumed that one site can see the set of all private date of the other sites provided that if cannot understand which data belongs to which site. In the proposed method, participants are considered to be partly honest. Although this model, is similar than that of the malicious participants, this state however, is more realistic and a model of semi-honest participants can be changed to a model of malicious participants. In such a case, in each step of the protocol, each participant using the zero knowledge proof can show that it acts according the protocol **[**6].

3.1. The stylish step

In this step each site, chooses a random number (R_i) and divides it to *n-1* section. Then each site sends each part of the random number to one of the other sites. The above said random number is uniformly selected from the (-∞, +∞). It should be noted that here the issue of confidentiality and lack of conspiracy of other sites with each other is raised. Since the model of semi-honest participant is used as basis, the possibility of eavesdropping has not been considered, while to prevent eavesdropping, of sites with each other, a trusted canal such as that based on the SSL protocol can be employed.

3.2. The L.L-itemset computation set

First, all the site using the apriori algorithm compute all their local *L.L-itemset* in an independent from. Regarding the fact that the number of *L-itemset* is high, to decrease its number, like [8], through employment of specification of the maximum frequent (largest) itemset , starts from its largest *L.L-itemset* member and removes all its subsets from the *L.L-itemset* since, as was printed out earlier, each subset is an open *L-itemset* of a *L-itemset*. It is assumed that the *M.L.L-itemset* is the set of the largest itemset.

3.3. The conveyance *M.L.L-itemset*

In this step, all the sites indirectly send their *M.L.L-itemset* to the first site, such that first each site throws a diagonal coin to take decision based on its result to either send this *M. L.L-itemset* to the first site and/or to a site different from the first site. It is assumed that the above said coin, with the probability of *P*, give the result of rotation of *M. L.L-itemset* to the sites (except the first one). On the other hand, each site, for dispatching the received *M.L.L-itemset*, throughs a diagonal coin so that it can choose a desired site (including the first site) and dispatches it. Regarding this mechanism, all *M.L.L itemset* of different sites will be finally dispatched to the first site. For higher flexibility, probably the coin can be considered differently in the sites.

It is necessary to mentioned that even the first site as well, rotates its *M.L.L-itemset* amongst the sites.So that, this procedure assists in making other sites anonymous. This is done to prevent betrayal of confidential information with each other. Also, the first site through receiving a *M.L.L-itemset* from the other sites, by throwing a diagonal coin (this coin can be similar to one used in dispatching it own *M. L-itemset*), decides to rotate once again the *M. L.L-itemset* amongst the sites. The reason for this action is to prevent from revealing information at the time of conspiracy on behalf of the *n-2* sites, which was fully discussed at the section on analysis of the proposed procedure. Regarding thes explanations, a *M. L.L-itemset* can be several times passed from hand to hand between the first site and the other sites.

3.4. The step of *M.L.L-itemset* computation

In this step, first the first site obtains the association of the *M.L.L-itemset* (the repetitive *M.L.L-itemset* sets are removed). Now the first site can, from the association of *M. L.L-itemset* obtain entirety of candidate *L-itemset.* As it was mentioned, an itemset is a *G.L-itemset* in the event that at least in one site is *L.L-itemset* and that for which the equation (3) holds. Therefore, the first site for each member like x, from the association of *L.L-itemset*, computes the Val_1:

$$\mathrm{Val}_1 = \mathrm{support} - \mathrm{count}_x(1) - \mathrm{support}_{\min} * \mathrm{database} - \mathrm{size}(1) + \sum_{i=2}^{n} \mathrm{R}_{i,1} - \mathrm{R}_1 \tag{5}$$

In (5), R_1 is the detected random number of the first site, and $\sum_{i=2}^{n} R_{i,1}$ is the set of random number of the other sites which the first site has received in the establishment step.

The first site sends the val_1 to its next site (the second site).

The consecutive sites as well, compute the quality of the statement (6) and dispatched the result to their next site.

$$\mathrm{Val}_i = \mathrm{Val}_j + \sup port - count_x(i) - \mathrm{support}_{\min} * database - size(i) + \sum_{k=1,k\neq i}^{n} R_{k,i} - R_i \tag{6}$$
$$\text{where } \mathrm{j} = (\mathrm{i} - 1 + \mathrm{n}) \bmod \mathrm{n}$$

In (6), R_i is the elected number of the ith site and $\sum_{k=1,k\neq i}^{n} \mathrm{R}_{k,i}$ is the set of random number of the other sites which the *i*th site has received at the establishment step.

Up to the *n*th site, all parts of the random numbers of all sites in (6), have been used with a positive sign and the set of all random numbers elected by all sites have been used with a negative sign. Therefore, the value of Val_n of that same statement (3) will be:

$$
\begin{aligned}
&\mathrm{Val}_n = \mathrm{Val}_{n\text{-}1} + \mathrm{support-count}_x(n) - \mathrm{support}_{\min} * \mathrm{database-size}(n) + \sum\nolimits_{k=1,k\neq n}^{n} \mathrm{R}_{k,n} - \mathrm{R}_n \\
&\Rightarrow \mathrm{Val}_n = \sum\nolimits_{i=1}^{n} (\mathrm{support-count}_x(i) - \mathrm{support}_{\min} * \mathrm{database-size}(i)) + \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1,i\neq j}^{n} \mathrm{R}_{i,j} - \sum_{i=1}^{n} \mathrm{R}_i \qquad (7) \\
&\Rightarrow \mathrm{Val}_n = \sum\nolimits_{i=1}^{n} (\mathrm{support-count}_x(i) - \mathrm{support}_{\min} * \mathrm{database-size}(i))
\end{aligned}
$$

4. Analysis of the proposed procedure

Our proposed method without using the third honest group or the third group having only the function of computing, has provided a good security for the participating sites. Further, to better study this claim, security of each step will be discussed.

In the establishment step, it has been estimated that the random number belonging to each site is uniformly selected from the interval of the real numbers. Also, each site in a quite desired form, divides its random number of another site or parts of it will be impossible. The computation step of *M.L.L-itemset* will be done locally as well as a result is quite secure. On the other hand, in this step, due to lack of blow up of the data, finally a precise answer will be obtained.

In the *M.L.L-itemset* step, the aim is that the *M.L.L-itemsets* are sent to the first site in such a way that the first site does not understand that each *M.L.L-itemset* belongs to which site. To do so the anonymousness precision of message in the network has been the base of inspiration [11[12] so that some mediate site are placed between the dispatcher and receiver of the message. In the proposed method, to decrease the expense of using intermediate sites, each site, for anonymous dispatching of its *M.L.L-itemset*, employs the other sites. In this case, there would be no relation between the *M.L.L-itemset* and its relevant sites, since each site makes decision about sending a *M.L.L-itemset* based on the result of throwing a coin diagonally. In other words, thorough receiving a *M.L.L-itemset* of another site, there is always this concept that, site is a mediator and the received *M.L.L-itemset* belongs to another site.

In the step of *M.L.L-itemset*, even the first site should pass its *M.L.L-itemset* through some mediator sites. This is of importance when the *n-2* sites have conspired with each other. Since if the first site, does not its *M.L.L-itemset* to other sites, in that case, if the victim site, sends its *M.L.L-itemset* to one of the conspiring sites, due to conspiracy of the *n-2* sites with each other, they will be sure that this *M.L.L-itemset* belongs to the victim site. However, if the first site rotates its *M.L.L-itemset* amongst the sites, in this case the conspiring sites do not know whether the received *M.L.L-itemset* has belonged to the first site or the victim's site.

In addition, the first site, through receiving a *M.L.L-itemset* by throwing a diagonal coin, decides to rotate once again the *M.L.L*-itemset amongst the sites. Since if this is not done, in this case as well, through conspiracy of the *n-2* site, if the first site sends its *M.L.L-itemset* to one of the conspiring sites, that site understands that the received M.L.L-itemset belongs to the first and as a result, anonymity of *M.L.L-itemset* of the first set will be lost.

Due to executing the previous step, *M.L.L-itemset* of all the sites are sent anonymously to the first site. Therefore, the first site with no knowledge of the relation between a *M.L.L-itemset* and its relevant site, obtain the association of al the *M.L.L-itemset*. Following finding the candidate *L.L-itemset* (using the *M.L.L-itemset),* the first site, obtains the statement (5) for each number of the result set and send the result to the second site. Regarding that along side the $\mathrm{support-count}_x(1) - \mathrm{support}_{\min} * \mathrm{database-size}(1)$, a random value is added and deducted, the second site cannot obtain the $\mathrm{support-count}_x(1) - \mathrm{support}_{\min} * \mathrm{database-size}(1)$ value (relevant to the first site). This condition continues up to the *n*th site. Therefore, in brief, none of the sites can obtain the value of the $\mathrm{support-count}_x(\mathrm{i})$ belonging to a *L.L-itemset* in other sites.

Since up to the *n*th site, all the random values has been added to or reduced from the $\sum\nolimits_{i=1}^{n} (\mathrm{support-count}_x(i) - \mathrm{support}_{\min} * \mathrm{database-size}(i))$. Thus, the *n*th site can understand that whether a M.L.L-itemset is a *G.L-itemset* or not. With this description, the proposed method results conspiracy of sites with each other, since a site to obtain the value of *support* in another site needs to know all parts of. It is obvious that resistance of this procedure against conspiracy of sites with each other equals to n-2, which means that if the n-2 sites conspire, they cannot obtain the *support* value of the two remaining sites. However, if the n-1 sites conspire, in that case, through adding up the value parts of the random values that they have received from the victim sites, obtain its random number and using it can compute the $\mathrm{support-count}_x(i) - \mathrm{support}_{\min} * \mathrm{database-size}(i)$ value of that sites. In addition, this n-1 site can easily compromise anonymity of *M.L.L-itemset* of the victim site.

The performance of our method compared with[8], [3] is highly noticable.If the average time is needed to send each message is equal to *t*, the total time is needed to execute the proposed procedure will be computed as follows:

$$\mathrm{T} = [(m + \mathrm{n}) * (\mathrm{n} - 1) + n * f(P)] * t \tag{8}$$

In (8), *n* is the number of sites, *P* is the probability of rotating a message among *K* sites, *f(P)* is a function which relates *K* and *P*, and *m* is the number of *M.L.L-itemsets* of all sites.

Regarding flexibility of the protocol in changing *P* and as a result modification of *K*, the proposed procedure in addition to enjoying the appropriate efficiency and security, is flexible since the parameter *P* can be regulated based on the relevant security. For example, to reach a higher security, the amount P can be increased so that against higher expense of computations (more rotation of *M.L.L-itemset* between the sites) and of course decrease of efficiency, a higher security would be achieved.

5. Conclusion

We proposed a new efficient method in order to keep confidentionality of data in database. Our algorithm uses three methods of randomizing data (use of random values alongside *support values* of each *L.L-itemset*), anonymous sending of *M.L.L-itemset* and safe computation of *support values* of each *L.L-itemset*.

A virtue of this protocol compared with other protocols is that under an appropriate precision, security, and efficiency of our protocol is considerable without expensive coding mechanism. Through conspiracy of a maximum *n-2* sites, confidentiality of data of other sites is also protected. Furthermore, high flexibility is another advantage of our method, since each site based on its own trust on other sites can regulate the level of confidentiality of its data.

6. References

[1] Ch. Aggarwal; Ph. Yu, “Privacy-Preserving Data Mining: Models and Algorithms”, Kluwer Academic publishers ,2007.

[2] R. Agrawal; R. Srikant, “Fast algorithms for mining association rules,” Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile, September, pp.487-499, 1994.

[3] M. Kantarcioglu; C. Clifton, “Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data”, IEEE Trans. Knowl. Data Eng. 16(9): 1026-1037, 2004.

[4] D. W.-L. Cheung; J. Han, V. Ng; A. W.-C. Fu; Y. Fu, “A fast distributed algorithm for mining association rules" Proceedings of the 1996 International Conference on Parallel and Distributed Information Systems (PDIS'96), Miami Beach, Florida, USA, Dec. 1996.

[5] X. Yi; Y. Zhang, Privacy-preserving distributed association rule mining via semi-trusted mixer, Data and Knowledge Engineering, page 550–567, 2007.

[6] Y. Lindell; B. Pinkas, “Privacy preserving data mining”, Advances in Cryptology, CRYPTO 2000 ,2000.

[7] Y. Lindell; B. Pinkas, “Secure Multiparty Computation for Privacy-Preserving Data Mining”, Journal of Privacy and Confidentiality, 2008.

[8] Ch.Ch Chang; J. Yeh; Y-Ch. Li, “Privacy-Preserving Mining of Association Rules on Distributed Databases”, IJCSNS International Journal of Computer Science and Network Security, VOL.6 No.11, 2006.

[9] A.A., Veloso; Jr.W. Meira; S. Parthasarathy; M.B. de Carvalho, “Efficient, accurate and privacy preserving data mining for frequent itemsets in distributed databases,” Proceedings of the Brazilian Symposium on Databases,

Manaus, Amazonas, Brazil, pp.281-292, 2003.

[10] W. Du; M. J. Atallah, “Secure multi-party computation problems and their applications: A review and open problems”, In Proceedings of the 2001 New Security Paradigms Workshop, Cloudcroft, New Mexico, 2001.

[11] D. Chaum., "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms”. Communications of the ACM, 1981.

[12] M. Reiter; A. Rubin., “Crowd: Anonymity for Web Transaction”, ACM Transactions on Information and System Security, 1998.

[13] A. HajYasien, “Revisiting Protocol for Privacy Preserving Sharing Distributed Data: A Review with Recent Results”, Springer, pp. 542-555, 2011.