Distributed Database using Randomized Response Technique in FP Tree Algorithm

Jyotirmayee Rautaray, Raghvendra Kumar

Abstract— Data mining is broadly used in all walks of existence. Through the association rule mining, the practical storage space location of commodities can be found. This paper mainly uses FP-tree algorithm and combines with distributed secure sum protocol as well as secure multi party computation using randomized response technique to provide the more privacy to the distributed database in the homogeneous horizontal partitioned distributed environment.

Keywords— Distributed Secure Sum Protocol, FP Tree Algorithm, Randomized response technique, Secure Multi Party Computation.

I. INTRODUCTION

Data mining knowledge has been broadly used in business, finance, telecommunication and other fields and also achieved acceptable results. However, with the development of research in data mining application and the uses of distributed database that desire to increase the privacy in database, there should present more and more privacy preserving protocol to provide the security to the database. Database is divided into four main partition like horizontal partitioning, vertical partitioning, derived horizontal partition and hybrid partition.

A. Horizontal Partitioning

Horizontal partition [10] [11] [12] [13] defines the partitioning of global database into a number of small partitioned databases but there each partition will share their common database schema or structure.

B. Vertical Partitioning

Vertical partitioning [10] [11] [12] [13] defines the portioning of global database into number of small portioned databases but each partition is obtained by projecting the global relation.

C. Derived Horizontal Partitioning

Derived horizontal partition [7] can be defined as a property of distributed database such that two different relations are joining to find out derived horizontal partition.

D. Hybrid partitioning

Hybrid partitioning [11] [12] [13] first divide the global database into horizontal partitioned and then vertical portioning or vice versa.

E. Association Rules Mining Algorithm

Discovering association rules are grand significance of data mining. Association rules mining [1] [2] [3] is primarily digging out hidden and interesting relevance and rules from big data set. By means of quantified Numbers, its purpose is to illustrate the influence on Class B while Class A appears. Through the association rule mining analysis we can get the straightforward rules about which produce to acquire together and decide the location of the two kinds of produce on the shelf to make considered arrangement in the position and promote crossover sales and confident trading modes.

F. FP - tree Algorithm

Due to the defects of Apriori algorithm [1] [2] [3] [4] put onward a frequent mode emergent algorithm, which is FP -tree mining algorithm to produce frequent set. FP – tree mining algorithm does not need to generate alternatives. It adopts the following divide-and-conquer strategy, the entire process is separated into two steps: first, compress the database that provides frequent item sets into a particular frequent pattern tree, but still retain the significance information connecting to item sets. Then, divide this kind of dense database into a set of restricted databases, each linking a frequent database. FP growth technique convert extended frequent mode problem into recursively found some short mode, then join the suffix. It only needs to construct FP – tree model and conditional FP – tree model, recursively visit FP – tree model. Then produce frequent item sets. It only needs to traverse the business database twice: the first time to compute the frequent 1-items, then construct frequent items database table; the second time to scan the database and build the tree. FP tree algorithm greatly reduces the number of database transaction to produce the frequent item sets and the searching cost, at the same time it uses most infrequent items as suffixes and provides superior selectivity. FP tree algorithm is more effective and retractable, and it is even about an order of magnitude faster than Apriori association rule mining algorithm.

G. Secure Multi Party Computation

It is frequently the case that mutually distributed parties need to perform a joint calculation but cannot want to disclose their inputs to each other. This can happen, for example, during data mining, distributed data mining, voting, negotiations and business analytics. Secure multi-party computation [6] allows a set of parties, each with a private input to securely and jointly execute any computation over their inputs.

Manuscript published on 28 February 2013.
***Correspondence Author(s)**

Jyotirmayee Rautaray School of Computer Engineering, KIIT University, Odisha, India.
Raghvendra Kumar School of Computer Engineering, KIIT University, Odisha, India.

H. Randomized Response Technique

The Randomized Response Technique [11] [12] was developed by statistics community to protect the privacy to the surveyed data base. The main work of Randomized Response Technique is to collect the data from the previous parties and send to the next party that presents in the database environment with adding user defined random number. If consider a datasets I= {I1, I2……..In} and random number R= {R1, R2…….Rn} so that the new number is I1+R1, I2+R2…….In+Rn. That's why the partial support is

$$Pij=I+R$$

$$I=Pij-R$$

I. Distributed Secure Sum Protocol

Let, P1, P2 , …, Pn are n parties involved in accommodating secure sum computation where each party is accomplished of breaking its data block into n number of segments such that the sum of all the segments is equal to the value of the data block of with the intention of parties. In distributed secure sum protocol [8] number of segments in a data block is kept equal to the number of parties. The values of the segments are randomly selected by the party and it is a secret of the party. If n is the number of segments (same as the number of parties) then in this scheme every party holds any one segment with it and n-1 segments are sent to n-1 parties, one to all of the parties. Thus at the end of this reorganization each of the parties holds k segments in which simply one segment belongs to the party and other segments belong to remaining parties, one from each. Now, k-Secure Sum Protocol [11] can be applied to get the sum of all the segments. In this protocol, one of the parties is unanimously selected as the protocol initiator party which starts the computation by sending the data segment to the next party in the ring. The receiving party adds its data segment to the conventional partial sum and transmits its result to the next party in the ring. This process is repetitive until all the segments of all the parties are added and the sum is announced by the protocol inventor party. Now even if two adjacent parties unkindly cooperate to know the data of a middle party they will be able to know only those k segments of a party which belong to every party. The sum of these segments is a garbage value or random number and thus that information is valueless for the hacker party.

II. IMPLEMENTATION AND EXPERIMENTAL RESULT

This algorithm is applicable when the number of sites greater than or equal to 3 (n≥3) and every party divides their partial support into number of segments and each party divides the random number into number of different number of segments. And every time each party will send their own data segments and own random number to the next party presents in the distributed environments (the number of segments is equal to the number of parties).

Randomized response technique [10] is used to improve the privacy making the partial support in more masked form. When sites try to find the global frequent item sets from its local frequent item sets then the sites also include some of the infrequent item sets to its local frequents items and send to the subsequent sites. So that frequent item set of a site is not revealed to the subsequent site. So it makes the subsequent site confused. Hence it provides high security to the database. Proposed algorithm shown in Fig: 1 and Fig: 2, the whole partitioned database is shown in below given tables.

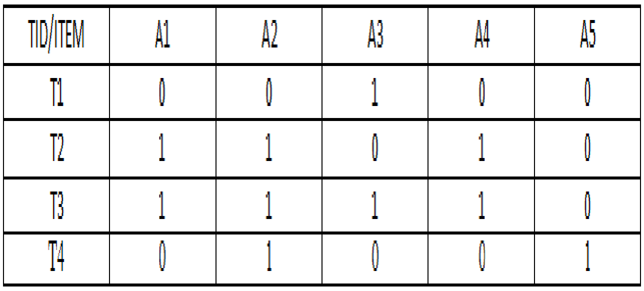
Algorithm1: Distributed Secure Sum protocol

```
Step1. Define P1, P2 , …, Pn as n parties.
Step2. Assume these parties have random number r1, r2… rn.
Step3. Every party Pi breaks its data ri interested in n segments di, di2,…, din such that Σ dij = ri for j =1 to n.
Step4. Every party keeps any one segment with it and distributes k-1 segments to other parties such that one segment is distributed to one party.
Step5. Each party reshuffles the received segments randomly.
Step6. Suppose rc =n and Sij = 0, //Sij is partial sum and rc is round counter
Step7. while rc!=0
            begin
              for j = 0 to n-1
                for i = 0 to n-1
                  Pi sends Sij = dij + Sij to P (i+1) mod n
                  rc = rc – 1
            end
Step8. The protocol inventor party broadcasts sum Sij to all the parties.
Step9. End of algorithm.
```

Algorithm2:-

```
Step1:- Reproduce on Parties P1, P2, P3…………Pn.
Step2:-All Party will generate their own random number R1, R2……….RN
Step3:-Join the Parties in the ring (P1, P2, P3………PN) and let P1 is a procedure initiator.
Step4:-Allow RC=N, and Pij=0 /*RC is round counter and Pij is partial support*/
Step5:-Partial support P1 party calculate through using subsequent formula
Psij = Xij.support – Min support * |DB| + RN1 - RNn
Step6:-Party P2 computes the PSj for each item received the list using the formula
PSij= PSij + Xij. Support – minimum support * |DB| +Rn1-Rn (i-1)
Step7:-While RC! =0begin for j=1 to N do
Begin for I=1 to N do
Begin each Party will calculate their partial support Pij sends to the next Party that is neighbor to the current Party.
Step8:- Party P1 swap over its position to P (j+1) mod N
        End
        RC=RC-1
        End
Step9:-Party P1 allowance the result Pij
Step10:-End of algorithm.
```

Table 1:- Data Set of Party 1 and Minimum Support is 40%



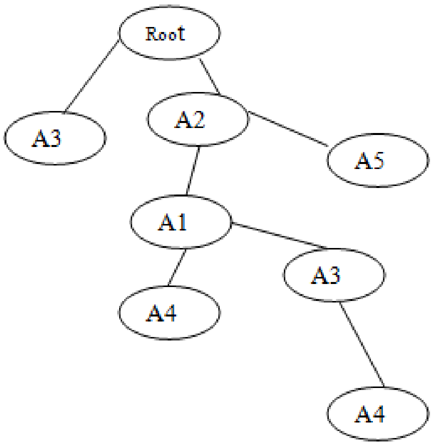
TID/ITEM	A1	A2	A3	A4	A5
T1	0	0	1	0	0
T2	1	1	0	1	0
T3	1	1	1	1	0
T4	0	1	0	0	1

Step1:-

TID	List
T1	A3
T2	A1, A2, A4
T3	A1, A2, A3, A4
T4	A2, A5

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

Step2:- A1: 2, A2:3, A3:2, A4:2, A5:1
Step3:-Arranging in descending order
A2:3, A1:2, A3:2, A4: 2, A5:1
Step4:-



Step5:-
A1 = {(A2:2)}
A2= {(A2:3)}
A3= {(A1:1, A2:1), A3:1}
A4 = {(A1:1, A2:1), (A3:1, A1:1, A2:1)}
A5= {(A5:1)}
Step6:- A4= {(A1:2) (A2:2)}
Step7:- A4A1:2, A4A2:2
Step8:- Support (A4A1) =Count (A4A1)/Total number of Transaction= 2/4=50%
Support (A4A2) =Count (A4A2)/Total number of Transaction= 2/4=50%
Candidate is selected = (A1, A2, A4)

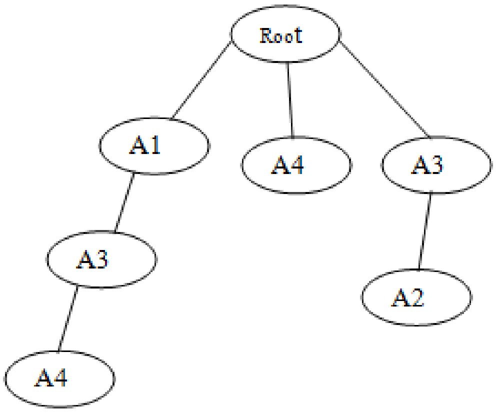
Table2:- Data Set of Party 2 and Minimum Support is 40%

TID/ITEM	A1	A2	A3	A4
T1	1	0	1	1
T2	0	0	0	1
T3	1	1	1	0
T4	1	0	0	0
T5	0	1	1	0

Step1:-

TID	List
T1	A1, A3, A4
T2	A4
T3	A1, A2, A3
T4	A1
T5	A2, A3

Step2:-
A1: 3, A2:2, A3:3, A4:2
Step3:-
Arranging in descending order
A1:3, A3:3, A2:2, A4: 2
Step4:-



Step5:-
A1 = {(A1:3)}
A2= {(A3:1, A1:1) (A3:1)}
A3= {(A1:2, (A3:1)}
A4 = {(A3:1, A1:1), (A4:1)}
Step6:- A2= {(A3:2)}
Step7:- A2A3:2
Step8:- Support (A2A3) =Count (A2A3)/Total number of Transaction= 2/5=40%
Candidate is selected = (A2, A3)

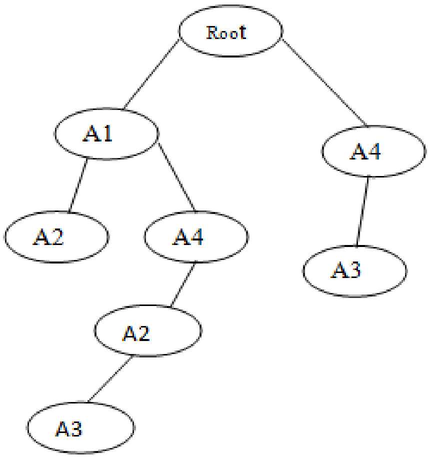
Table3:- Data Set of Party 1 and Minimum Support is 40%

TID/ITEM	A1	A2	A3	A4
T1	1	1	1	1
T2	0	0	1	1
T3	1	1	0	0
T4	1	0	0	1

Step1:-

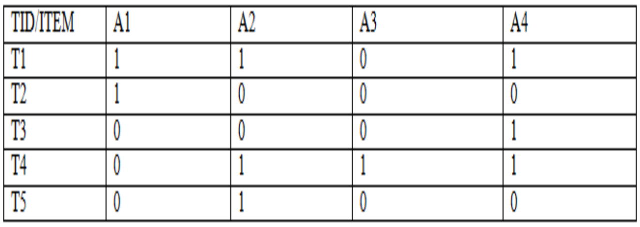
TID	List
T1	A1, A2, A3, A4
T2	A3, A4
T3	A1, A2
T4	A1, A4

Step2:-
A1: 3, A2:2, A3:3, A4:3
Step3:-Arranging in descending order
A1:3, A4:3, A2:2, A3: 2
Step4:-



Step5:-
A1 = {(A1:3)}
A2= {(A4:1, A1:1) (A1:1)}
A3= {(A2:1, A4:1, A1:1) (A4:1)}
A4 = {(A1:2), (A4:1)}
Step6:- A2= {(A1:2)}, A3= {(A4:2)}
Step7:- A2A1:2, A3A4:2
Step8:- Support (A2A1) =Count (A2A1)/Total number of Transaction= 2/4=50%
Support (A3A4) =Count (A3A4)/Total number of Transaction= 2/4=50%
Candidate is selected = (A1, A2, A3, A4)

Table4:-Data Set of Party 1 and Minimum Support is 40%

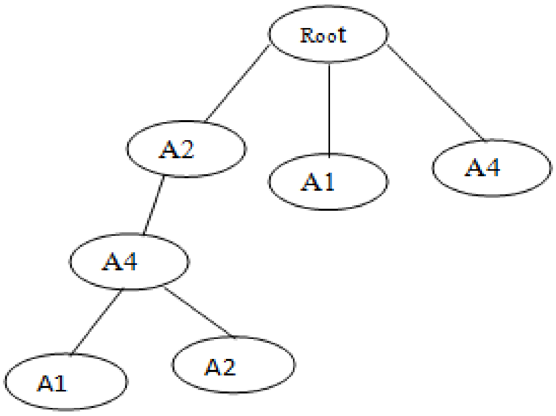


TID/ITEM	A1	A2	A3	A4
T1	1	1	0	1
T2	1	0	0	0
T3	0	0	0	1
T4	0	1	1	1
T5	0	1	0	0

Step1:-

TID	List
T1	A1, A2, A4
T2	A1
T3	A4
T4	A2, A3, A4
T5	A2

Step2:-
A1: 2, A2:3, A3:1, A4:3
Step3:-Arranging in descending order
A2:3, A4:3, A1:2, A3:1
Step4:-



Step5:-
A1 = {(A4:1, A2:1) (A1:1)}
A2= {(A2:3)}
A3= {(A4:1, A2:1)}
A4 = {(A2:2), (A4:1)}
Step6:- As here minimum support is 40% so no candidates is selected
Candidates selected at site 1:- {A1, A2, A4}
Candidates selected at site 2:- {A2, A3}
Candidates selected at site 3 :-{ A1, A2, A3, A4}
Candidates selected at site 4:- As here minimum support is 40% so no candidates is selected in site 4.
Believe the item set {A1,(A1,A2)}
Consider the item set I = {A1}
PS=I1Support- Minimum support*|Size of Database|
First calculate the partial support of every party
Party P1 will calculate the partial support by suing the following the formula
PS1=I1Support- Minimum support*DB=2-.4*4=.4
PS2=I2Support- Minimum support*DB=3-.4*5=1
PS3=I3Support- Minimum support*DB=3-.4*4=1.4
PS4=I4Support- Minimum support*DB=2-.4*5=0
Divides the partial support of each party into number of segments let party P1 divides the partial support into n number of segments
PS11=.1, PS12=.1, PS13=0, PS14=.2
PS21=.1, PS22=.1, PS23=.4, PS24=.4
PS31=.2, PS32=.5, PS33=.5, PS34=.2
PS41=0, PS42=0, PS43=0, PS44=0
Every party selects their random number
Let the party P1 have the random number RN11=1, RN12=1, RN13=1, RN14=2
Let the party P2 have the random number RN21=1, RN22=1, RN23=2, RN24=1
Let the party P3 have the random number RN31=1, RN32=2, RN33=1, RN34=1
Let the party P4 have the random number RN41=2, RN42=1, RN43=1, RN44=1
Every party calculated their partial support by using the following formula
PS=I1Support- Minimum support*|Size of the Database| + (RN i -RN (i-1))
Calculation for Round 1-
PS11= 0.1+ (1-2) = -0.9
PS12=0.1+ (1-1) -.9= -0.8
PS13=0.2+ (1-1) -.8= -0.6
PS14=0.0+ (2-1) -.6=0.4
Calculation for Round 2 –
PS11= 0.1+ (1-1) + 0.4 =0.5
PS12=0.1+ (1-1) +0.5= 0.6
PS13=0.5+ (2-1) +0.6=2.1
PS14=0.0+ (1-2) + 2.1=1.1
Calculation for Round 3 –
PS11= 0.0+ (1-1) +1.1=1.1
PS12=0.4+ (2-1) +1.1=2.5
PS13=0.5+ (1-2) +2.5=2
PS14=0.0+ (1-1) +2=2
Calculation for Round 4 –
PS11= 0.2+ (2-1) +2 =3.2
PS12=0.4+ (1-2) +3.2=2.6
PS13=0.2+ (1-1) +2.6=2.8
PS14=0.0+ (1-1) +2.8=2.8
After applying the following formula that given below initiator party will allowance the global excess support
Global excess support (GES) = Partial support
GES=2.8

III. CONCLUSION

In this paper we combined the FP Tree algorithm and Distributed secure sum protocol as well as secure multi party computation with circumstances of homogeneous database and with the assist of randomized response technique. We take for granted that all parties have the same schema, but all party does not have information on dissimilar entities and different random number. The goal is to produce global association rules with the help of FP tree algorithm that hold global rules while limiting the information shared about each party. Many proposals have been partied to apply secure multi party computation.

Secure multi party computation is being used in huge scale databases which extends to protect privacy to the private data segments of different parties. In this paper our focus is based on horizontal partitioned Distributed data through an accepted FP tree Association rule mining technique for providing the highest privacy to the database and data leakage is zero percentage.

REFERENCES

1. Agrawal, R., et al.: Mining association rules between sets of items in large database. In: Proc. of ACM SIGMOD'93, D.C, 1993,pp.207-216 ACM Press, Washington.
2. Agarwal, R., Imielinski, T., Swamy, A.: Mining Association Rules between Sets of Items in Large Databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 1993, pp. 207-210.
3. Srikant, R., Agrawal, R.: Mining generalized association rules. In: VLDB'95,1994, pp.479-488 .
4. Agrawal, R., Srikant, R.: Privacy-Preserving Data Mining. In: proceedings of the 2000 ACM SIGMOD on management of data, 2000, pp. 439-450.
5. Lindell, Y., Pinkas, B.: Privacy preserving Data Mining. In: Proceedings of 20th Annual International Cryptology Conference (CRYPTO) 2000.
6. Kantarcioglu, M., Clifton, C.: Privacy-Preserving distributed mining of association rules on horizontally partitioned data. In IEEE Transactions on Knowledge and Data Engineering Journal, IEEE Press, Vol 16(9), 2004, pp.1026-1037.
7. Han, J. Kamber, M.:Data Mining Concepts and Techniques. Morgan Kaufmann, San Francisco, 2006.
8. Sheikh, R., Kumar, B., Mishra, D, K.: A Distributed k- Secure Sum Technique for Secure Multi-Site Computations. Journal of Computing, Vol 2, 2010, pp.239-243.
9. Sugumar, Jayakumar, R., Rengarajan, C.:Design a Secure Multi Site Computation System for Privacy Preserving Data Mining. In International Journal of Computer Science and Telecommunications, Vol 3, 2012, pp.101-105.
10. Muthu Lakshmi, N. V., Sandhya Rani, K.: Privacy Preserving Association Rule Mining without Trusted Site for Horizontal Partitioned database. In International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.2, 2012, pp.17-29.
11. Muthu lakshmi, N.V., Sandhya Rani, K.: Privacy Preserving Association Rule Mining in Horizontally Partitioned Databases Using Cryptography Techniques. In International Journal of Computer Science and Information Technologies(IJCSIT), Vol. 3 (1),2012 , PP. 3176 – 3182.
12. Goldreich, O., Micali, S. & Wigerson, A.: How to play any mental game. In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, pp.218-229.
13. Franklin, M., Galil, Z. & Yung, M.:An overview of Secured Distributed Computing. Technical Report CUCS- 00892, Department of Computer Science, Columbia University.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)