Asian Journal of Applied Sciences (ISSN: 2321 – 0893)
Volume 02 – Issue 05, October 2014

Optimization of Active Apriori Algorithm Using an Effective Genetic Algorithm for the Identification of Top-l Elements in a Peer to Peer Network

S. Veena[1], P. Rangarajan[2]

[1]Sathyabama University, Chennai
Email: djohnveena {at} gmail.com

[2] R.M.D Engineering College, Chennai.

ABSTRACT-- ***In a distributed system like peer to peer network, there are two ways of storing the data namely homogeneous and heterogeneous. Mining the homogeneous data in a client is less time consuming and fast compared to the mining in the server. Frequent sets play an essential role in many Data Mining tasks that try to find interesting patterns from databases, such as association rules, correlations, sequences, episodes, classifiers and clusters. The mining of association rules is one of the most popular problems of all these. In this paper, Active Apriori algorithm is used to find the frequent items in the data set which reduces the cost. This method compresses the database by removing unnecessary transaction records and data items from the database that are not used for further processing. The speed of algorithm is increased because it needs to scan only the compressed database and not entire database. The results of the Active apriori algorithm can be optimized using an effective genetic algorithm to identify the top l elements or most frequent item sets. In this method, the near distance of rule set are found using equalize distance formula and generate two classes namely, higher class and lower class . The classes are validated by distance weight vector, which maintains a threshold value of rule item set. This Effective genetic algorithm is mainly used for optimization of rule set.***

Keywords--- Data mining, Active Apriori algorithm, Effective genetic algorithm.

---

1. INTRODUCTION

Classical data mining techniques assume that all data is available at a central server. However there exist scenarios in which the data is inherently distributed over a large, dynamic network containing no special servers or clients, for example, peer-to-peer (p2p) networks [1]. In many application scenarios, it is often desirable to know only the top inner products. Such a need is often felt even in emerging large-scale peer-to-peer (P2P) applications such as the formation of interest-based online communities [2]. Association rules are used to identify relationships among a set of items in database. These relationships are not based on inherent properties of the data themselves, but rather based on co-occurrence of the data items. In this paper, an attempt has been made to generate optimized association rule by activet Apriori association rule mining method. These rules are given as input to the Effective Genetic algorithm The Effective Genetic algorithm is used to identify the most frequent item sets of global top l elements (attribute wise) from distributed data.

The rest of this paper is organized as follows. In section (2) the related works are discussed. In section (3) we give a formal definition of association rules and an active Apriori algorithm. Section (4) introduces the Genetic algorithm and an Effective Genetic algorithm. Section (5) contains conclusions.

2. RELATED WORKS

Souptik Datta *et al.* [3] data intensive large-scale distributed systems like peer-to-peer (P2P) networks are becoming increasingly popular where centralization of data is impossible for mining and analysis. Unfortunately, most of the existing data mining algorithms work only when data can be accessed in its entirety. Finding all the network-wide frequent item sets is computationally difficult and usually has large communication overhead in such environment. This paper focuses on developing a communication efficient algorithm for discovering frequent item sets from a P2P network.

A sampling-based approach is adopted to find approximate solution instead of an exact solution with probabilistic guarantee. The benefit of approximation technique is reflected in the low communication overhead in discovering majority of frequent item sets with probabilistic guarantee.

Yiwu Xie, Yutong Li, Chunli Wang, Mingyu Lu [4] had done a study of Apriori algorithm. In this paper, two aspects are discovered that affect the efficiency of the algorithm. One is the frequent scanning database, the other is large scale of the candidate item sets. Therefore, Apriori algorithm is proposed in this paper , that can reduce the times of scanning database, optimize the join procedure of frequent item sets generated in order to reduce the size of the candidate item sets. In this paper, it not only decrease the times of scanning database but also optimize the process that generates candidate item sets.

Gao, Shao-jun Li,[5], have proposed a novel procedure to delete many transactions which need not be scanned repeatedly. It reduced the number of database passes to extract frequent item sets. A method was showed to reduce the number of candidate item sets by optimizing the join procedure of frequent item sets. By a number of experiments, the proposed algorithm outperforms the apriori algorithm in computational time.

Rakesh Agrawal Ramakrishnan Srikant [6] proposed a novel algorithm for optimization of association rule mining, which resolve the problem of negative rule generation and also optimized the process of supremacy of rules. Supremacy of association rule mining is a great challenge for large dataset. In the generation of supremacy of rules association existing algorithm or method generate a series of negative rules, which generated rule affected a performance of association rule mining.

3. ASSOCIATION RULE MINING

Association rule mining finds the correlation among items that are grouped into transactions, infers the rules, which define relationships between item sets. The rules have a user-stipulated support, confidence, and length. An association rule is an implication of the form A->B where A and B are the item sets. Support measures the fraction of transactions that contain both A and B. Given a rule A->B and N being the total number of transactions then the support of an association rule is defined as: **Support = A union B / N**
Confidence measures how often item in B, appear in transactions that contain A. Given the rule A ->B, its confidence is defined as follows: **Confidence = A Union B / A**

3.1 Active Apriori algorithm

Formally let V be the set of items. A transaction over V is a pair T = (tid, V) where tid is the transaction identifier and V is the set of items. A database DB over V is a set of transactions over V such that each transaction has a unique identifier.

A transaction T = (tid, V) is said to support a set P, if P ϵ V. The cover of a set P in DB consists of the set of transaction identifiers of transactions in DB that support P. The support of a set P in DB is the number of transactions in the cover of P in DB. The frequency of a set P in DB is the probability that P occurs in a transaction, or in other words, the support of P divided by the total number of transactions in the database. A set is called frequent if its support is not less than a given absolute minimal support threshold min_sup with 0>min_supabs>|DB|. When working with frequencies of sets instead of their support, we use the relative minimal frequency threshold min_suprel, with 0>min_suprel>1. Obviously min_supabs = [min_suprel * |DB|]. In this paper we will mostly use the absolute minimal support

In association rule mining the input given is the database which is homogeneous , same attributes distributed at different sites. Two main steps are there in association rule mining, which uses active apriori algorithm. First, using the minimum support value and the minimum confidence value assigned, the frequent item sets are produced. Second, by using the frequency item sets produced and the minimum hope value allocated to the site or client, the association rules are generated.

Pseudo code for Active Apriori algorithm

Cp: Candidate item set of size p

Lp : frequent item set of size p

Lp = {frequent items};

for (p = 1; Lp !=Æ; p++) do begin

Cp+1 = candidates generated from Lp;

for each transaction T in database do

increment the count of all candidates in Cp+1 that are contained in T

Lp+1 = candidates in Cp+1 with min_support

end

return Cp Lp;

We can increase speed by removing unnecessary transaction records from database. The items that not appear in Lp-1 will no longer need for generation of Lp. So we can delete these items from the transaction database. At the same time, after Lp-1 is generated, delete the transactions where the number of items is less than x from database then the candidate set Cp can be generated by latest DB. The deletion of items and transaction from database will greatly reduce the size of transaction database, which will effectively increase the speed of the algorithm.

Input: T: Database of transactions; *ms*: minimum support threshold

Output: fis: frequent itemsets in D

Method:

1) fis_1=find_frequent_1-itemsets(T);

2) For(x=2;fis_{x-1}; x++){

3) Ck=apriori_gen(fis_x-1, ms);

4) for each transaction t ∈DB{

5) C_t=subset(C_x,t);

6) for each candidate c ∈C_t

7) c.count++;

8) }

9) fis_x={ c ∈ C_x |c.count_ms };

10) if(x>=2){

11) remove_value(DB, fis_x, fis_{x-1});

12) remove_trans (DB, fis_x); }

13) }

14) return fis=U_x fis_x ;

Procedure remove_value (T:Database; fk: frequent(k) -itemsets;

fk-1: frequent(k-1) - itemsets)

for each itemset I ∈ fk-1 and I ∈ |∉fk

{for each transaction t in T

{for each datavalue ∈t

{if (datavalue=i) delete datavalue; } } }

Procedure remove_trans (D: Database; fk: frequent(k) - itemsets)

for each transaction t ∈ T{ if(datarecord.count<k){ delete datarecord;} }

Example :

Transaction ID	**Items**
T1	I1,I3,I4
T2	I2,I3,I5
T3	I1,I2,I3,I5
T4	I2,I5
T5	I5

Candidate – 1 item set

Itemset	**Support**	
I1	2	
I2	3	
I3	3	
I4	**1**	**-→ Delete since minsupport = 2**
I5	4	

Candidate – 1 item set – after Deletion

Itemset	**Support**
I1	2
I2	3
I3	3
I5	4

Delete Transaction T5 since it contains only 1 item < no. of items (2) in candidate -2 item sets

Transaction ID	**Items**
T1	I1,I3
T2	I2,I3,I5
T3	I1,I2,I3,I5
T4	I2,I5

Candidate – 2 item set

Itemset	**Support**	
I1,I2	1	**--→ Delete since minsupport=2**
I1,I3	2	
I1,I5	1	**--→ Delete since minsupport=2**
I2,I3	2	
I2,I5	3	
I3,I5	2	

Candidate – 2 item set – after Deletion

Itemset	**Support**
I1,I3	2
I2,I3	2
I2,I5	3
I3,I5	2

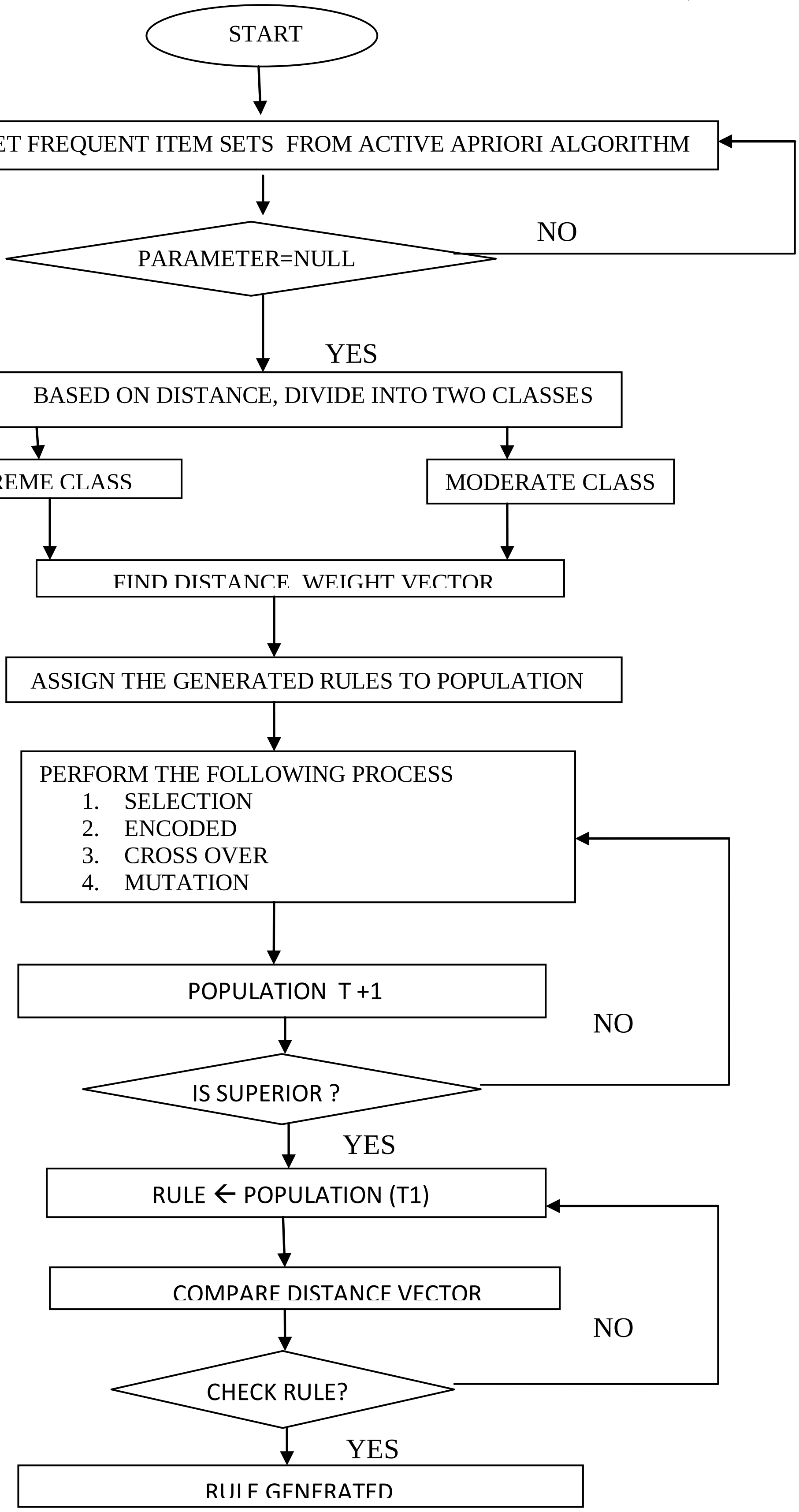
Delete Transaction T1 and T4 since it contains only 2 items < no. of items (3)

Transaction ID	**Items**
T2	I2,I3,I5
T3	I1,I2,I3,I5

Candidate – 3 item set

Itemset	**Support**
I2,I3,I5	2

Figure 1: Architecture of Effective Genetic Algorithm



4. GENETIC ALGORITHM

Genetic algorithm is a family of computational models based on principles of evolution and natural selection. These algorithms convert the problem in a specific domain into a model by using a chromosome-like data structure and evolve the chromosomes using selection, crossover, and mutation operators.

Based on the optimal result, the knowledge can be extracted. The optimized result represent the elements of distributed data set in a peer to peer network, i.e., the top elements in the distributed dataset is identified using these optimization of rules using an Effective Genetic algorithm. From these we received the optimal top *l* elements from the distributed peer networks.

5. CONCLUSION

The proposed Effective Genetic algorithm optimizes by reducing the size of database. The performance of Active Apriori algorithm is improved, so that we can mine association information from massive data faster and better. Based on these knowledge extracted the top *l* items in the P2P network is identified. The optimal set of elements is identified by the Effective genetic algorithm, which is a combination of distance function and genetic algorithm. When we modify the distance weight, it is observed new rules are found in large numbers. This implies that when weight is solely determined through support and confidence, there is a high chance of eliminating interesting rules. With more rules emerging it implies there should be a mechanism for managing their large numbers. The large generated rule is optimized with Effective genetic algorithm. We proofed a relation between locally large and globally large patterns that is used for local pruning at each site to reduce the searched candidates. We derived a locally large threshold using a globally set minimum recall threshold. Local pruning achieves a reduction in the number of searched candidates and this reduction has a proportional impact on the reduction of exchanged messages.

6. REFERENCES

[1] Kanishka Bhaduri, Kamalika Das, Kun Liu, Hillol Kargupta, "Distributed Identification of Top-l Inner Product Elements and its Application in a Peer-to-Peer Network", CIKM , 2006.

[2] K. Liu, H. Kargupta, and J. Ryan, "Random Projection-Based Multiplicative Data Perturbation for Privacy Preserving Distributed Data Mining," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 1, pp. 92-106, Jan. 2006.

[3] Souptik Datta, Hillol Kargupta, "A communication efficient probabilistic algorithm for mining frequent itemsets from a peer-to-peer network," Article first published online: 16 JUN 2009.

[4] Yiwu Xie, Yutong Li, Chunli Wang, Mingyu Lu "The Optimization and Improvement of the Apriori Algorithm", Intelligent Information Technology Application Workshops, 2008. IITAW '08.

[5] Gao, Shao-jun Li,' A method of improvement and optimization on association rules appriori algorithm',proceeding of the 6th congress on intelligent control and automation,2006 pp5901-5905.

[6] By Rakesh Agrawal Ramakrishnan Srikant_ Fast Algorithms for Mining Association RulesVLDB ConferenceSantiago, Chile, 1994.

[7] R. Agrawal and R. Srikant. "Fast algorithms for mining association rules in large databases" Research Report RJ 9839, IBM Almaden Research Center, San Jose, California, June 1994.

[8] By N. Chaiyarataiia and A. M. S. Zalzala Recent Developments in Evolutionary and Genetic Algorithms: Theory and Applications Innovations and Applications, 2-4 September 1997, Conference Publication NO. 4 4 6 ,IEEE , 1997.

[9] By Pengfei Guo Xuezhi Wang Yingshi Han The Enhanced Genetic Algorithms for the Optimization Design 978-1-4244-6498-2/10/$26.00 © IEEE 2010.

[10] By Dieferson Luis Alves de Araujo' , Heitor S. Lopes', Alex A. Freitas2 A Parallel Genetic Algorithm for Rule Discovery in Large Databases 0-7803-5731-0/99$$10.00109 99 IEEE.

[11] By Xiaofeng Yuan, Hualong Xu, and Shuhong Chen "Improvement on the Constrained Association Rule Mining Algorithm of Separate" 1-4244-0682-X/06/$20.00 © IEEE 2006.

[12] By WEI Yong-qing1, YANG Ren-hua2, LIU Pei-yu2 An Improved Apriori Algorithm for Association Rules of Mining 978-1-4244-3930-0/09/$25.00 © IEEE 2009.

[13] By R. Uday Kiran and P. Krishna Reddy An Improved Multiple Minimum Support Based Approach to Mine Rare Association Rules 978-1-4244-2765-9/09/$25.00 © IEEE 2009.

[14] By Peter P. Wakabi ,Waiswa ,Venansius Baryamureeba, Karunakaran and Sarukesi **"**Optimized Association Rule Mining with Genetic Algorithms" Natural Computation 978-1-4244-9953-3/11/$26.00 © IEEE 2011.

[15] By Li-Min Tsai, Shu-Jing Lin, and Don-Lin Yang "Efficient Mining of Generalized Negative Association Rules" Granular Computing 978-0-7695-4161-7/10 $26.00 © IEEE 2010.

[16] Rajneesh K Karan, YK Rana.By Rakesh Agrawal Tomasz Imielinski Arun Swami Mining Association Rules between Sets of Items in Large Databases ACM SIGMOD Conference Washington DC, USA, May 1993.

[17] Badri Patel, Vijay K Chaudhari, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-1, March 2011Optimization of Association Rule Mining Apriori Algorithm Using ACO

[18] "Association rule mining over multiple databases: Partitioned and incremental approaches" by Hima Valli Kona, the University of Texas at Arlington, December 2003.