AN EFFICIENT METHOD FOR THE IDENTIFICATION OF TOP – L ELEMENTS BY USING RAPID APRIORI ALGORITHM AND ACO ALGORITHM

[1]S.VEENA, [2]P.RANGARAJAN

[1]Research Scholar, Sathyabama University, Chennai
[2]Professor & Head, R.M.D Engineering College, Chennai.
E-mail: djohnveena@gmail.com

Abstract– A peer-to-peer (P2P) network is one that does not have fixed clients and servers but a number of peer nodes that function as both clients and servers to the other nodes in the network. Association Rule Mining can be used for discovering hidden relationship between items stored in various nodes in the network . By given a user-specified threshold, also known as minimum support, the mining of association rules can discover the complete set of frequent patterns. In this paper, a Rapid Apriori algorithm has been proposed for reducing the cost and it reduces 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 reduced database and not the entire database. Then the top l elements or most frequent item sets are identified based on the effective Ant colony Optimization algorithm.

Keywords- Peer to Peer network, Rapid Apriori Algorithm, Effective Ant Colony Optimization Algorithm.

I. INTRODUCTION

Classical data mining techniques assume that all data is available at a central location. However there exist situations 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. In many application scenarios, it is often desirable to know only the top elements. 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. In the online community formation example, each peer may be associated with a feature vector describing its Web surfing patterns, and the goal is to find peers having similar interest (browsing patterns). This helps in routing queries to peers with relevant interests, resulting in better network-search results. In most cases, each peer may be interested in finding only a few peers with similar interest and not all of them. If the entire data can be conveniently accessed, it is easy to compute the inner product matrix and determine the top ones. However, much of the world's data is distributed over a multitude of systems connected by communication channels of varying capacity. This calls for new techniques to perform data mining in a distributed environment.

Data mining is used to extract the information from any system by analyzing the present in the form of data. In this paper, First step is to focus on the problem of frequent pattern mining. Frequent patterns are the patterns that that occur in database at least user given number of times. Problem of frequent pattern mining can be defined as: given a large database of transactions, each consists of set of items. The main aim of this problem is to find all the frequent item sets i.e. a set of items Y is frequent if greater than min_supp % of all transaction in database contains Y and finding association rules from these frequent itemsets. Association rules was first introduced by Agarwal. Association rules are helpful for analyzing customer behavior in retail trade, banking system etc. Association rule can be defined as {X, Y} => {Z}.In this paper, an attempt has been made to generate optimized association rule by Apriori association rule mining method. These rules are given as input to the Novel genetic algorithm The Second step is to consider the problem of identifying the global top-l inner products (attribute wise) from distributed data. We assume that data is scattered among a large number of peers such that each peer has exactly the same set of attributes (or features). In the data mining literature, this is often referred to as a horizontally partitioned (homogeneously distributed) data scenario. The top l elements or most frequent items are identified by optimizing the results of Rapid Association rule mining algorithm using the Effective Ant Colony Optimization algorithm.

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 the Rapid Apriori algorithm. Section (4) introduces the definition of Ant colony optimization and the Effective Ant colony Optimization algorithm. Section (5) contains conclusions.

II. RELATED WORKS

P2P networks are large, dynamic, asynchronous, and with little central control. It is very difficult, if not impossible, to transfer all the data to a single peer to do the computation since no one would have such extensive storage and computational capabilities, let

alone the enormous communication overhead. Each peer may be associated with a feature vector describing Web surfing patterns and Browsing patterns. The distributed top-k monitoring by Babcock and Olston presents a way of monitoring the answers to continuous queries over data streams produced at physically distributed locations. This helps in routing queries to peers with relevant interests resulting in better network search results. it is easy to compute the inner product matrix and determine the top ones.

Kamalika Das et al. developed a distributed algorithm for efficiently identifying top-l inner products from horizontally partitioned data. To achieve low communication overhead, it use an order statistics-based approach together with cardinal sampling. Ordinal statistics provides a general framework for estimating distribution-free confidence intervals for population percentiles. Cardinal sampling helps to combine the inner product values that are distributed among the peers. Local algorithms can be exact or approximate. However, the class of exact local algorithms that currently exist in the literature work for simple primitives such as average and L2-norm. For solving more complicated distributed problems, researchers have developed approximate solutions. The ordinal analysis technique developed in this paper belongs to this genre of approximate local algorithms.

Shwetha and Kanwal garg uses Apriori to find association rule. The author consider three association rule algorithms: Apriori Association Rule, Predictive Apriori Association Rule and Tertius Association Rule and compares the result of these three algorithms and presents the result. According to the result obtained using data mining tool author find that Apriori Association algorithm performs better than the PredictiveApriori Association Rule and Tertius Association Rule algorithms.

Jaishree Singh*, Hari Ram**, Dr. J.S. Sodhi proposed an Improved Apriori algorithm which reduces the scanning time by deleting the unnecessary transaction as well as reduce the redundant generation of sub-items during pruning the candidate item sets, which can form directly the set of frequent item sets and eliminate candidate having a subset that is not frequent. Sanat Jain et al. present an Apriori-based algorithm which finds all valid positive and negative association rules in a support confidence framework. The algorithm can find all valid association rules quickly and overcome some limitations of the previous mining methods. The complexity and large size of rules generated after mining have motivated researchers and practitioners to optimize the rule, for analysis purpose.

Their optimization done using Genetic Algorithm.

III. ASSOCIATION RULE MINING

The original motivation for searching frequent sets came from the need to analyze so called supermarket transaction data, that is, to examine customer behavior in terms of the purchased products (Agrawal et al., 1993). Frequent sets of products describe how often items are purchased together.

Formally let I be the set of items. A transaction over I is a couple T = (tid, I) where tid is the transaction identifier and I is the set of items from I. A database D over I is a set of transactions over I such that each transaction has a unique identifier. We omit I whenever it is clear from the context.

A transaction T = (tid, I) is said to support a set X, if X C I. The cover of a set X in D consists of the set of transaction identifiers of transactions in D that support X. The support of a set X in D is the number of transactions in the cover of X in D. The frequency of a set X in D is the probability that X occurs in a transaction, or in other words, the support of X divided by the total number of transactions in the database. We omit D whenever it is clear from the context.

A set is called frequent if its support is no less than a given absolute minimal support threshold min_sup with 0>min_supabs>|D|. 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 * |D|]. In this paper we will mostly use the absolute minimal support threshold and omit the subscript abs.

3.1 Rapid Apriori algorithm

For learning association rules, apriori is a typical algorithm. Apriori algorithm is planned to function on databases having transactions. The apriori Algorithm is a dominant algorithm for mining regular item sets for Boolean association rules. Agrawal and Srikant improved it on 1994. To find association rules on great scale is an innovative way, permitting implication outcomes that contain more than one item. In association rule mining the input given is the database. Two main steps are there in association rule mining. First, using the minimum support value assigned, the frequent item sets are produced. Second, the association rules are produced by means of the frequency item sets produced and the minimum hope value allocated.

Pseudo code for Association rule mining

Ck: Candidate item set of size k
Lk : frequent item set of size k
Lk = {frequent items};
for (k = 1; Lk !=Æ; k++) do begin

Ck+1 = candidates generated from Lk;
for each transaction t in database do
increment the count of all candidates in Ck+1
that are contained in t
Lk+1 = candidates in Ck+1 with min_support
end
return Ck Lk;

We can increase speed by removing unnecessary transaction records from database. The items that not appear in Lk-1 will no longer need for generation of Lk. So we can delete these items from the transaction database. At the same time, after Lk-1 is generated, delete the transactions where the number of items is less than k from database then the candidate set Ck can be generated by latest D. 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: f: frequent itemsets in D
Method:
1) f1=find_frequent_1-itemsets(T);
2) For(k=2;fk-1; k++){
3) Ck=apriori_gen(fk-1, ms);
4) for each transaction t ∈D{
5) Ct=subset(Ck,t);
6) for each candidate c ∈Ct
7) c.count++;
8) }
9) fk={ c ∈ Ck |c.count_ms };
10) if(k>=2){
11) delete_value(D, fk, fk-1);
12) delete_trans (D, fk); }
13) }
14) return f=Ukfk ;

Procedure delete_value (T:Database; fk: frequent(k) - itemsets;
fk-1: frequent(k-1) - itemsets)
1) for each itemset I ∈ fk-1 and I ∈ |∉fk{
2) for each transaction t in T{
3) for each datavalue ∈t{
4) if (datavalue=i)
5) delete datavalue;
6) }}
Procedure delete_trans (D: Database; fk: frequent(k) - itemsets)
1) for each transaction t ∈ T{
5) if(datarecord.count<k){
6) delete datarecord;}
7)}

TransactionID	items
T1	S1,S3,S4
T2	S2,S3,S5
T3	S1,S2,S3,S5
T4	S2,S5
T5	S5

Item set	Support
S2 S3 S5	2

IV. ANT COLONY OPTIMIZATION ALGORITHM

The Ant Colony Optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. The algorithm which using a grouping of real ants in the real environment. They study and observe the behavior of those real ants and suggest that the real ants were able to select the shortest path between their nest and food resource, in the existence of alternate paths between the two. This ant behavior was first formulated and arranged as Ant System (AS) by Dorigo et al.. Based on the AS algorithm, the Ant Colony Optimization (ACO) algorithm was proposed. In ACO algorithm, the optimization problem can be expressed as a formulated graph G = (C; L), where C is the set of components of the problem, and L is the set of possible connections or transitions among the elements of C.

4.1 Effective ACO Algorithm
In this algorithm, each ant searches for a minimum cost feasible partial solution. An ant t has a memory M_t that it can use to store information on the path it followed so far. The stored information can be used to build feasible solutions, evaluate solutions and retrace the path backward. An ant t can be assigned a start state sts and more than one termination conditions et. Ants start from a start state and move to feasible neighbor states, building the solution in an incremental way. The procedure stops, when at least one termination condition et for ant t is satisfied. An ant t located in node i can move to node j chosen in a feasible neighborhood Nti through probabilistic decision rules. This can be formulated as follows:
An ant t in state sr =< sr-1; i > can move to any node j in its feasible neighborhood Nti , defined as Nti = {j | (j Є Ni) Λ (< sr, j >Є S)} sr Є S, with S is a set of all states.
A probabilistic rule is a function of the following.
a) The values stored in a node local data structure Ai = [aij] called ant routing table obtained from pheromone trails and heuristic values,
b) The ant's own memory from previous iteration, and
c) The problem constraints.
When moving from node i to neighbour node j, the ant can update the pheromone trails τij on the edge (i, j). Once it has built a solution, an ant can retrace the same path backward, update the pheromone trails and

CONCLUSION

The classical Apriori algorithm has performance bottleneck in the massive data processing so that we

need to optimize the algorithm with variety of methods. The Rapid Apriori algorithm optimizes by reducing the size of database in the form of deletion of values and transactions. The performance of Rapid 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 using effective Ant colony optimization algorithm.

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] Rakesh Agrawal, T. Imieliński, A. Swami, "Mining association rules between sets of items in large databases". In: Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93, 1993 pp. 207-216.

[4] Sanjeev Rao, Prianka Gupta, "Implementing Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm", In: proceeding of IJCST, ISSN 0876-8491, VOL.3, Issue 1, Jan-March 2012.

[5] B. Babcock and C. Olston, "Distributed Top-k Monitoring," Proc. ACM SIGMOD '03, pp. 28-39, 2003.

[6] Kamalika Das, Kanishka Bhaduri, Kun Liu, and Hillol Kargupta, "Distributed Identification of Top-l Inner Product Elements and Its Application in a Peer-to-Peer Network," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 19, NO. 4, 2008.

[7] Shweta and Kanwal Garg, "Mining Efficient Association Rules Through Apriori Algorithm Using Attributes and Comparative Analysis of Various Association Rule Algorithms", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, June 2013.

[8] Jaishree Singh*, Hari Ram**, Dr. J.S. Sodhi , "Improving Effeciency of Apriori algorithm using Transaction Reduction, International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.

[9] Sanat Jain, Swati Kabra, "Mining & Optimization of Association Rules Using Effective Algorithm", International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 4, April 2012.

[10] M. Dorigo, Gianni Di Caro, and Luca M. Gambardella. Ant Algorithms for Discrete Optimization. Technical Report Tech. Rep. IRIDIA/98-10, IRIDIA, Universite Libre de Bruxelles, Brussels, Belgium, 1998.

[11] M. Dorigo and M. Maniezzo and A. Colorni. The Ant Systems: An Autocatalytic Optimizing Process. Revised 91-016, Dept. of Electronica, Milan Polytechnic, 1991.

[12] M. Dorigo and G. Di Caro. New Ideas in Optimisation. McGraw Hill, London, UK, 1999.

★★★