An Efficient Method to Identify the Most Frequent Item Sets Using Database Count Algorithm and Effective Aco Algorithm in a Peer to Peer Network

S.Veena[#1], DR.P.Rangarajan [*2]

[#1] *RESEARCH SCHOLAR, Sathyabama University, Chennai*
[*2]*PROFESSOR & HEAD, R.M.D Engineering College, Chennai*

***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 Database count algorithm is used to mine the local frequent item sets and then generate the global frequent item sets from a Peer to peer network. Then the global item sets are optimized by using the Effective Ant colony Optimization algorithm to generate the most frequent item sets.**

Keywords - Peer to Peer network, Database Count 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 [1]. 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 [2]. 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 [3]. 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 [4]. Association rules was first introduced by Agarwal [3]. 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 the global frequent item sets by using an enhanced apriori algorithm and Database count 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 Database Count 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, enhanced apriori algorithm and the Database Count 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 [5] 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.* [6] 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.

Mining of the frequent itemset is an important tasks in association rule mining which finds the frequent itemsets in a database which consists of transactions. It plays an important role in many data mining tasks which finds the interesting patterns from various data, such as association rules, classifier, clustering and correlation, etc [7].

In many databases which consists of transactions, there are large amounts of data, the scalability is high in a distributed systems and the centralized database is partitioned and distributed and it is very important to investigate efficient methods for mining the association rules in a peer to peer network. This paper [8] describes various interesting relationships between large itemsets that exists locally and globally. In this paper, the author had described an interesting association rule mining algorithm in a distributed system, FDM (Fast Distributed Mining of association rules), which generates a small number of candidate sets and reduces the number of messages to be passed at mining association rules. Chewang.D.W had demonstrated that FDM has a superior performance over the direct application of a typical sequential algorithm.

Association Rule mining is one of the important data mining techniques currently designed to group the data together from large databases which is used to extract the interesting correlation and relation among large amount of data. Thabet Slimani proposes current trend in association mining and compare the performance of different algorithms [9].

III Database Count Algorithm

A. Association Rule

An association rule is an implication expression of the form $P \rightarrow Q$, where P and Q are disjoint itemsets, i.e., $P \cap Q = \emptyset$.The strength of an association rule can be measured in terms of its support and confidence. Support determines how often a rule is applicable to a given data set, while confidence determines how frequently items in Q appear in transactions that contain P. The formal definitions of these metrics are

$$\text{Support, } s(P \rightarrow Q) = \frac{\sigma(P \cup Q)}{N}$$

$$\text{Confidence, } c(P \rightarrow Q) = \frac{\sigma(P \cup Q)}{\sigma(P)}$$

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.

B. Enhanced Apriori algorithm

An Enhanced Apriori algorithm is an improvement of Apriori algorithm which reduces the time consuming for candidates itemset generation. First, it scan all transactions to generate P1 which contains the items, their support count and transaction ID where the items are found. And then we use P1 later as a helper to generate P2 , P3,... Pk. When we want to generate S2, we make a self-join P1* P1 to construct 2-itemset S (x, y), where x and y are the items of S2. Before scanning all transaction records to count the support count of each candidate, use P1 to get the transaction IDs of the minimum support count between x and y, and thus scan for S2 only in these specific transactions.The same thing for S3, construct 3-itemset S (x, y, z), where x, y and z are the items of S3 and use P1 to get the transaction IDs of the minimum support count between x, y and z, then scan for S3 only in these specific transactions and repeat these steps until no new frequent itemsets are identified.

The enhanced apriori algorithm can be described as follows:

//Input the items, support, and their transaction ID

Step a. P1= find_fis (T);
Step b. For (x = 2; Px-1≠Φ; x++) {
//Generate the Sx from the Px-1
Step c. Sx= candidates from Px-1;
//get the item Ez with minimum support in Sx using P1, (1≤z≤x).
Step d. p = Get _item_min_sup(Sx, P1);

// obtain the the final transaction IDs that contain item p.
Step e.Ta = get_Transaction_ID(p);
Step f. For each transaction t in Ta Do
Step g.Increment the count of all items in Sx that are found in Ta;
Step i. Px = items in Sx≥min_support;
Step j.End;
Step k. }

C. Database Count algorithm

We consider a type of distributed system namely peer to peer network, where the data are distributed evenly among the peers in a network. The data are distributed homogenously in a peer to peer network. Database Count algorithm is a typical parallel algorithm and it is based on apriori algorithm. The database in each peer scans the data base to calculate the support of each candidate set and then set the sum of the entire candidate set as the global support .If the global support is greater than the min_support ,we consider the itemset as the global frequent item.

We consider the 4 peers and the database at four peers namely E1,E2,E3and E4. The database is divided into D1, D2,D3 and D4. The minimum support is assumed as s, 20%. The local frequent itemset is generated by using a distributed algorithm from D1,D2 ,D3and D4.The global frequent item set are generated by collecting the local frequent item set from all the peers in a network based on four cases. The four cases used to generate the global frequent item set are :

Case 1: If a local frequent itemset occurs in all databases then that local frequent itemset will be a global frequent itemset.
Case 2. : If an itemset (not local) whose support value satisfies the minimum support value then it will be a global frequent itemset.
Case 3: If an item set has 2 local supports value and if the sum less than the minimum support value it may be a global frequent itemset if it satisfies the condition such that global min_support-current_minsupport< local_min_support(totcountopeers-cntoflocalsupport)
Case 4 : If an item set has 2 local supports value and if the sum less than the minimum support value it may not be a global frequent itemset if it does not satisfies the condition such that globalmin_support-current_minsupport < local_min_support (totcountofpeers-cnttoflocalsupport)

An ant a in state sr =< sr-1; x > can move to any node y in its feasible neighborhood Nax , defined as Nax = {y | (y Є Nx) Λ (< sr, y >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.

The Database Count algorithm can be described as follows :

Input:Transactional database,Support and Confidence

Step 1:Divide the Database into D1,D2,D3 and D4 for the four peers .
Step 2: Find the local frequent itemset using an enhanced apriori algorithm
Step 3 : Generate the global frequent itemset by collecting the local frequent item sets from all the peers.

Output: Global frequent Itemset

IV Ant Colony Optimization

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 behaviour of those real ants and the real ants were suggested to select the shortest path between their nest and food resource, in the existence of alternate paths between the two. This behaviour of the ants were first formulated and given as Ant System by Dorigo et al. [10][11]. Based on the Ant System algorithm, the Ant Colony Optimization algorithm was proposed [12]. In Ant Colony Optimization algorithm, the optimization problem can be expressed as a formulated graph G = (C; L), where C is the set of problem components and L is the set of possible connections or transitions among the elements of C.

A. Effective ACO Algorithm

In this algorithm**, e**ach ant searches and finds the minimum cost which is feasible partial solution. An ant a has a memory Ma that store the information on the path it has followed in the past to the present. The information which is stored can be used to build the solutions which is feasible, evaluate the solutions and retrace the path in a backward direction An ant a can be assigned a start state sas and more than one termination conditions ea. Ants start from a start state and move to the neighbor states which is feasible, building the solution in an incremental way. The procedure stops, when at least one termination condition ea for ant a is satisfied. An ant a located in node x can move to node y chosen in a feasible neighborhood Nai through probabilistic decision rules. This can be formulated as follows:

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.

V 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 Database count algorithm uses an enhanced apriori algorithm to reduce the time spent in scanning the transactions for candidate item sets by reducing the number of transactions to be scanned.The performance of Database count algorithm is improved, so that we can mine association information from massive data faster and better. Based on these knowledge or global item set extracted, the top *l* items or the most frequent item sets in the P2P network is optimized by using effective Ant colony optimization algorithm.

ACKNOWLEDMENTS

I would like to specially thank God , who guided me through the way and made all things possible. I would like to acknowledge and extend my heartfelt gratitude to my supervisor **Prof. P.RANGARAJAN** whose help, stimulating suggestions, knowledge, experience and encouragement helped me in all the times of study and analysis of the research. I am deeply indebted to **Mr.D.John Aravindhar**, Associate Professor, HITS for his vital encouragement, support and constant reminders in carrying out my research work. I would like to express my gratitude to Director, **Thiru P.Venkatesh Raja**, S.A.Engineering College, Chennai for his inspiration and support towards my research work. I would also like to thank my family members who were a great source of support and encouragement.

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] S. Rao, R. Gupta, "Implementing Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm", *International Journal of Computer Science And Technology*, pp. 489-493, Mar. 2012.

[8] Cheung.D.W,"A fast distributed algorithm for mining association rules",4th International Conference on Parallel and Distributed Information Systems,1996,P31-42.

[9] Thabet Slimani ," Efficient Analysis of Pattern and Association Rule Mining Approaches " IJITCS Vol. 6, No. 3, February 2014

[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.