ISSN: 2321-7782 (Online)
Volume 2, Issue 2, February 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

A Survey on Mining Distributed Frequent Itemset

Poonam Modgi[1]
PG student
Parul Institute of Technology
GTU
Gujarat - India

Dinesh Vaghela[2]
Prof
Parul Institute of Technology
GTU
Gujarat - India

Abstract: ***In the current scenario there has been growing attention in the area of distributed environment especially in data mining. Frequent pattern mining is active area of research in today's scenario. In this paper we will present a survey on frequent itemset mining with distributed environment .The evaluation of algorithm with frequent itemsets and association rule mining has been growing rapidly. We will present characteristics of algorithms through comparison matrix and reach at conclusion of the best strategy followed for finding frequent items.***

Keywords: frequent itemset, ARM, trie, distributed mining, gossip protocol.

I. INTRODUCTION

Data mining is the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies as well as research focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions.

The association rule mining (ARM) is very important task within the area of data mining Given a set of transactions, where each transaction is a set of literals (called items), an association rule is an expression of the form X ->Y, where X and Y are sets of items. The intuitive meaning of such a rule is that transactions of the database which contain X tend to contain Y.

Apriori is one of the most popular data mining approaches for finding frequent itemsets from transactional datasets. The Apriori algorithm is the main basis of many other well-known algorithms and implementations. The main challenge face by many frequent itemset mining algorithm is its execution time. The complexity can vary base on the message passing overhead.

II. GUIDELINES

In the following section we will discuss different algorithms proposed by different author in increasing order of efficiency. We will see the most basic distributed version of Apriori and gradually we will discuss the comparison matrix and in future work we will propose one algorithm that will give less communication overhead and more efficiency.

1. Count Distribution (CD): [1]

In this algorithm each site finds local support count of Ck Candidate itemsets in its own local database. Each Site exchanges its local support with other sites to obtain entire support for all candidate itemsets. Each site obtains the entire support for all candidate itemsets its local support with other sites to obtain the entire support for all candidate itemsets. Each site obtains LK, the entire frequent itemset, the candidate itemset with length of k+1 obtained from each site by execution of Apriori gen()function on Lk.

The CD algorithm's main advantage is that it doesn't exchange data tuples between processors-it only exchanges the counts. The algorithm's communication overhead is O(|C|*n) at each phase, where |C| and n are the size of candidate itemsets and number of datasets, respectively.[2]

2. Fast distribution mining (FDM): [2]

In each site play different roles, in the beginning a site consider as "home site" for produced set of candidate set sand sub sequently it changes to a polling sites to get response time from other sites, it changes to a polling site to get response time from other sites, it become remote site. The different stages for FDM algorithm with consideration of different roles for each site.

FDM's main advantage over CD is that it reduces the communication overhead to O(|Cp|*n), where |CP| and n are potentially large candidate itemsets and number of sites, respectively. FDM generates fewer candidate itemsets compared to CD, when the number of disjoint candidate itemsets among various sites is large. [2]

3. Optimized distributed association mining (ODAM): [2]

ODAM calculates 1-itemset from each site and broadcast those itemsets and discovers global frequent 1-itemsets.Each site generates candidate 2 itemsets and computes its support count and same time eliminate infrequent itemsets ODAM generates globally frequent 2 itemsets and iterates through main memory transaction and generates the support count of that respective length and the final frequent itemset is generated, The efficiency is more than CD and FDM algorithm.

The ODAM's message exchange size increases linearly as we increases the number of sites. It exchanges fewer messages than FDM.ODAM requires minimal number of comparison and update operation to generate support count.

4. Distributed trie frequent itemset mining (DTFIM): [3]

In this algorithm each site scans its local database and determines local count (1-itemsets) a vector is kept for making support count of every item. At end each site synchronize their data structure, using L1 the trie copies are alike. In second pass the candidate 2-itemsets are calculated,2-d array is used for this purpose, at end count are synchronized and global support count for candidate2-itemset are calculated and trie copies are updated. From here on, in each pass k (k>=3) a candidate itemsets are calculated and the process is repeated, the pruning is performed simultaneously at each stage. The final output is frequent itemset.

The complexity of DTFIM is $O(n^2)$ which is less than CD, FDM, ODAM. As the algorithm uses trie structure it is useful for pruning at local site.

5. Mining distributed frequent itemsets using a gossip

Based protocol: [4]

In this algorithm, gossip based communication mechanism is used that is purely based on random communication between sites.The trie based Apriori structure is used to improve the performance. First the local frequency is computed and support count is checked then after gossip Based global aggregation is done. In this algorithm improvement is achieved through employing trie data structure and grouping of nodes is done.

The complexity of this algorithm is O(nlogn) which outperforms all above algorithm. Gossip based communication can reduce the computation overhead of finding frequent itemset. The scalability is high as it requires minimum communication and comparison costs. As the complexity is low compared to DTFIM it is faster.

III. COMPARISON MATRIX

Algorithm	Complexity	Message passing overhead
CD	High(O(|c_k|n) where $c_k$is candidate itemset is number of site)	less
FDM	High, less than CD(O(|c_p|n),where Cp is union of all local candidate itemsets	less
ODAM	Low compare to CD,FDM(O (| C R + P (F D |* n), where C R is the intersection of all local frequent itemsets, P (F D) is the total number of disjoint local frequent itemsets that have higher probability, and n is the totalnumber of sites)	high
DTFIM	Low compare to CD,FDM,ODAM (O(n^2))	high
GOSSIP BASED DISTRIBUTED FREQUENT ITEMSET MINING	Low compared to all above (O(nlogn))	high

IV. CONCLUSION AND FUTURE WORK

In this paper, after reviewing various algorithms the conclusion can be drawn such as when we use the most basic algorithm the complexity is high but the message passing overhead is low as we move forward to higher algorithm the complexity decreases but the overhead is more. As CD is distributed version of Apriori it is simple. The FDM is higher version with fast distribution while ODAM combines the CD and FDM and outperforms than both in terms of complexity as well as communication overhead. The DTFIM is using trie structure and it enables local pruning easy. The gossip based uses trie structure as well as gossip protocol for communication and gives highest performance.

In future, we can propose such algorithm which will use trie structure and grouping methodology but it will locally prune dataset at base level as well as at intermediate level by grouping node into cluster. At the end final aggregation of frequent itemset is done at global level. This algorithm will have low complexity as well as message passing overhead.

References

1. Distributed Count Association Rule Mining Algorithm, International Journal of Computer Trends and Technology- July to Aug Issue 2011
2. Ashrafi,Taniyar and K.smith 2004,Optimized DistributedAssociation Rule Mining, IEEE distributed system online,5:3
3. E ansari, M.keshatkaran march 2008,Distributed Trie Frequent Itemset Mining,IMECS Vol II
4. Mining distributed frequent itemset using gossip basedprotocol,IEEE,2012 9th international conference
5. Ferenc bodon,A trie based apriori implementation for mining frequent itemset.ACM,NEW YORK,USA, 2005, 56-65
6. A decentralized approach for mining event correlations indistributed system monitoring, J. Parallel Distrib. Comput. 73 (2013) 330–340
7. A fast distribution algorithm for mining association rules
8. Mohammed j zaki Parallel and distributed association mining :A survey

