International Association of Scientific Innovation and Research (IASIR)
(An Association Unifying the Sciences, Engineering, and Applied Research)

ISSN (Print): 2279-0063
ISSN (Online): 2279-0071

International Journal of Software and Web Sciences (IJSWS)
www.iasir.net

Pruning Large Data Sets for Finding Association rule in cloud: CBPA (Count Based Pruning Algorithm)

Nidhi Khurana* , Dr. R.K. Datta**
*Research Scholar , Singhania University.
*Faculty, Guru Nanak Institute of Management, New Delhi
**Director, M.E.R.I.T., New Delhi

Abstract: *Organizations are more interested in the interesting data rather than the bulk of data. So they need a systematic and scientific approach to extract meaningful data out of heaps of the data and to find out the relations among these patterns. To analyze "big data" on clouds, it is very important to research data mining strategies based on cloud computing paradigm from both theoretical and practical views. In this paper, based on the original Apriori algorithm, an improved algorithm is proposed which adopts a new count-based method to prune candidate item sets and uses generation record to reduce total data scan amount and also make it more modeling oriented. Experiments demonstrate that by performing our algorithm of on given datasets we will find the Solution of problems in association and apriori algorithm.*

Keywords: *Cloud, data mining , association rule, apriori algorithm , pattern.*

I. Introduction

All the organizations big or small have bulk of data which needs to be stored or retrieved systematically to form information. The repository of data is known as Database. With the advancement in computer science, the database has taken many shapes. According to the applications, starting from the traditional file system to hierarchical, Network, Relational, Object Oriented, Associative, now it has reached to Data Warehouses and Data Marts etc.

But every piece of data stored in these databases may not be useful for the Organization. They need to filter the useful data from the bulk of data which can be used for decision making, reporting or analysis. These useful patterns or pieces of data are known as interesting patterns.

II. Related Work

A. Association Rule Mining

Association Rule mining is the scientific technique to dig out interesting and frequent patterns from the transactional, spatial, temporal or other databases and to set associations , relations or correlations among those patterns (also known as item sets) in order to discover knowledge or to frame information.

Association rules can be applied in various fields like network management, Basket data analysis, catalog design, clustering, classification, marketing etc. Association rules establish the relationship between different variables to analyze the present situation. For e.g. to find the relationship between the various items sold at a shopping mall, the association rule can be applied on the huge amount of data recorded by the Shopping mall.

For e.g. the rule {Computer, Printer} → {UPS} found in the sales data of a mall would indicate that if a customer buys Computer and Printer together, he or she would definitely also buy UPS. This information can be used making the decision regarding keeping the stock of the products as well as to analyze the customer buying habits and promotional activities for future. Association rule works on the database of transactions where every transaction contain list of item set(patterns).

Measures of the rule are Support and Confidence. Support of rule is proportion of transaction in the data set that contains the item set to the total number of transactions. The Confidence of a rule is ratio of total number of transactions with all the items to the number of transaction with the A item set. For e.g if Dataset *T* is given the an itemset *A* has number of occurrences in it. An association rule is the relationship between two itemsets *A* and *B* . such as A B means when A occurs B also occurs .

B. Example & Explanation of Association rule

To illustrate and understand the basic terms we consider a small database of 6 transactions and 3 items. The rule is

{Computer, printer} {UPS}

This implies that if customer buy Computer and Printer, he tend to buy UPS also. Out of 6 transactions 3 transactions support this rule .In 3 records all the three items are brought together.

- The Support of rule denoted as Supp***(***A***)*** is proportion of transaction in the data set that contains the itemset to the total number of transactions. In the above example, the itemset {Computer, Printer, UPS} has a support of 3/ 6 = 0.5 since it occurs in 50% of all transactions (3 out of 6 transactions).

- The Confidence of a rule (denoted as conf(A B)=Ratio of total number of transaction with all the items to the number of transaction with the A item set . for e.g Computer and Printer are purchased 4 times and out of 4 transactions UPS is purchased three times with Computer and Printer i.e A so the conf(A,B)= ¾ =.75 i.e 75%.

So the association rule is the technique to set the relation between item sets to draw important conclusions. It set the minimum support and confidence threshold and evaluates the frequent itemset and then use the evaluated itemset to frame desired results.

C. Types of Association rule

Association rule mining can be broadly classified into following categories :

- Boolean or quantitative associations
- Single dimension or multidimensional associations
- Single level or multilevel associations

D. Modified Level of Association Rule

Multiple level association rule mining can work with two types of support- Uniform and Reduced.

1. Uniform Support: In this approach same minimum support threshold is used at every level of Hierarchy. There is no need to evaluate itemsets containing items whose ancestors do not have minimum support. The minimum support threshold has to be appropriate. If minimum support threshold is too high the we can lose lower level associations and if too low then we can end up in generating too many uninteresting high level association rules. For e.g

At Level 1	Computer, Printer
	Minimum supp 5% [support – 10%]
At Level 2	Wipro Computer , Cannon printer Minimum support 3% [support 7%]
	Wipro Computer , HP Printer [support 3%]

2. Reduced Support : In this approach reduced minimum support is used at lower levels

There are following search strategies:

- Independently Level by Level : This technique is basically based on full breadth search. It is not required to know in advance frequent itemset for pruning. Each node is evaluated at each level , regardless of whether or not its ancestor node is found to be frequent.

- Filtering across the levels by single item set : In this technique descendants are only checked only if ancestor is found to be frequent. So item at ith level will be only checked if and only if item at i-1th level is frequent. For eg wipro computer is not examined if computer is not frequent.

- Filtering across the levels by k-itemset : In this approach a k item set at ith level is only examined if and only if its ancestor k item set at i-1 th level is frequent . For eg Wipro Computer , cannon printer will be examined only if Computer and printer are frequent.

E. Checking for redundancy :

There can be redundancy in some of the rules due to its ancestors associations between items For eg.

Rule 1 : Computer, Printer → Ups [support =10% , Confidence =70%]

Rule 2: Wipro Computer , Cannon printer → Mikrotek UPS [support =3% , confidence =70%]

In this eg first rule is an ancestor of the second rule . A rule is redundant if its support is close to the expected value, based on the rule's ancestor.

III.Apriori Algorithm

Apriori algorithms having a two-step process.

The join step: To find L_k , a set of candidate k item sets is generated by joining L_{k-1} with itself. This set of candidate is denoted C_k.
The prune step: C_k is the superset of L_k, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in C_k. A scan of the databases to determine the count of each candidate in C_k would result in the determination of L_k. (i.e. all candidates having a count no less than the minimum support count are frequent by definition, and therefore belongs to L_k)

procedure *AprioriAlg()*

begin

L_1 *:= {frequent 1-itemsets};*

for *(* $k := 2; L_{k-1} 0; k++$ *)* ***do*** *{*

C_k*= apriori-gen(*L_{k-1}*) ; // new candidates*

for *all transactions t in the dataset* ***do*** *{*

for *all candidates c* C_k *contained in t* ***do***

c:count++

}

L_k *= { c* C_k *| c:count >= min-support}*

}

Answer := $_k L_k$

end

It makes multiple passes over the database. In the first pass, the algorithm simply counts item occurrences to determine the frequent 1-itemsets (itemsets with 1 item). A subsequent pass, say pass k, consists of two phases. First, the frequent itemsets L_{k-1} (the set of all frequent (k-1)-itemsets) found in the (k-1)th pass are used to generate the candidate itemsets C_k, using the apriori-gen() function. This function first joins L_{k-1} with L_{k-1}, the joining condition being that the lexicographically ordered first k-2 items are the same. Next, it deletes all those itemsets from the join result that have some (k-1)-subset that is not in L_{k-1} yielding C_k.

The algorithm now scans the database. For each transaction, it determines which of the candidates in $C_k$are contained in the transaction using a hash-tree data structure and increments the count of those candidates. At the end of the pass, C_k is examined to determine which of the candidates are frequent, yielding L_k . The algorithm terminates when L_k becomes empty.

A. *Problem in Apiori Algorithm:*

Main Problem with Apriori algorithm is that it is less effective with time variant as well on the memory front of view because it fetch data from every time in a single apriori algorithm run on a particular data and it has to communicate with main database every time so it take more memory and time as well.To overcome this problem, there are a solution called FP-growth pattern but this solution is complex.

IV.Proposed Solution: A Case Study

Now, we are going to produce a concept of modified data association rule .In this rule, We make a copy of table of transaction and then we make another table which consist the no. of items , name of the items and list no of the items.

List no.	Name of Item	No of Item
100	A	4
101	B	3
102	C	4
103	D	2
104	E	1

Table:1-Table of transaction

List of	40-60%	60-75%	75-95%	95-100%	Item-sets
A	D	B	E	C	(A,C)
B	A	D	C	E	(B,E)
C	B	E	A	D	(C,D)
D	C	A	E	B	(D,B)
E	B	C	D	A	(E,A)

Table:2- Items record in transaction

Now, we take another table of containing Items associating Items record in transaction

For calculating` the percentage of items associate items and make item sets of every time k to n where k=2 to n and n is no. of association we want to make. For example {A,B,D} then n=3.

After making the association rule, we discard the table and take it record to the data warehouse for further use. Now, To find Percentage of items associate items , we use the formula of probability and summation of series.

$$\Pi = \sum \alpha_{i} * \beta_{i.}$$

Where α_i= total no of items with same transaction row / total no of transaction

And β_i=100.

A. The Proposed Algorithm

CBPA algorithm is used on the customized version as data miner required as per there requirement. The proposed algorithm is memory effective and best used for data mining because its record save in cloud for future calculation.

New CBPA(Count Based Pruning Algo):

Steps:

Step 1. Access sphere of cloud where data is reside which is to be patterned

Step 2. Make copy of that database in RAM or in Cache Memory of client Sphere.

Step 3. Perform proposed Algorithm After making three table according to the requirement which is mention earlier.

Step 4. Do

For k=2 to n

Find π

And update counter T as well as table .

Step5.

Select t.items, t.items. . . . . From table T.

Step 6 . perform again

Find π and update counter T as well as table till k=n

Step7. If k=n

Then stop and make pattern decision using last counter T update values.

V.Cloud and CBPA

Cloud computing provides cost-efficient solutions of storing and analyzing mass data.By cloud we can say that it is an infrastructure that consists of services delivered through shared Data Centers and appearing as a single point of

access for consumers' computing needs and also provides demanded resources and/or service over the internet. Sector storage cloud is a distributed storage system that can be deployed over a wide area network and allows users to consume and download large dataset from any location with a high-speed network connection to the system. Sector automatically replicates files for the better reliability, access and availability. Sphere compute cloud is a computation service which is built on the top of the sector storage cloud. It allows developers to write certain distributed data intensive parallel applications with several simple APIs. Data locality is the key factor for the performance in the Sphere. The exact location of data in the cloud is often unknown. Data may be located in systems in other countries, which may be in conflict with regulations prohibiting data to leave a country or union. It is the responsibility of cloud providers to keep data in specific jurisdictions and whether the providers will make contractual commitments to obey local privacy requirements on behalf of their customers.

VI. Conclusion

To analyse "big data" on clouds, it is very important to research data mining strategies based on cloud computing paradigm from both theoretical and practical views. For this purpose, we study a strategy of data mining on cloud using association rule mining as an example. The paper proposes a fast mining algorithm of association rules based on cloud computing namely, CBPA (Count Based Pruning Algorithm). The experimental results suggest that CBPA is fast more time effective and frequent data analysis on basis of associations and Rule of association should be imply more on other area than market basket analysis

References

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases", Proceedings of the ACM SIGMOD Conference on Management of data, pp.207-216, May 1993.

[2] R. Agrawal, and R. Srikant, "Fast Algorithms for Mining Association Rules", In Proc. VLDB 1994, pp.487-499.

[3] Sujni Paul, and V. Saravanan, "Hash Partitioned Apriori in Parallel and Distributed Data Mining Environment with Dynamic Data Allocation Approach", Computer Science and Information Technology, 2008. ICCSIT'08. International Conference on Aug. 29 2008-Sept. 2 2008, pp.481-485

[4] Lei Ji, Baowen Zhang, and Jianhua Li, "A New Improvement on Apriori Algorithm", Computational Intelligence and Security, 2006 International Conference on Volume 1, Nov. 2006, pp.840-844

[5] Kun-Ming Yu, Jia-Ling Zhou, "A Weighted Load-Balancing Parallel Apriori Algorithm for Association Rule Mining", Granular Computing, 2008. GrC 2008. IEEE International Conference on 26-28 Aug. 2008, pp.756-761

[6] R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. In Journal of Parallel and Distributed Computing (Special Issue on High Performance Data Mining), 2000.

[7] Farah Hanna AL-Zawaidah and Yosef Hasan Jbara, "An Improved Algorithm for Mining Association Rules in Large Databases " in World of Computer Science and Information Technology Journal (WCSIT)ISSN: 2221-0741 Vol. 1, No. 7, 311-316, 2011

[8] P.Velvadivu1and Dr.K.Duraisamy,"An Optimized Weighted Association Rule Mining On Dynamic Content" in IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 5, March 2010.

[9] Jinxin Zhang, Mangui Liang, A new architecture for converged Internet of Things, Internet Technology and Applications, 9 (2010), 1-4

[10] R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98, 85-93, Seattle, Washington.

[11] Abhishek Kajal et al. / Indian Journal of Computer Science and Engineering (IJCSE) ISSN : 0976-5166 Vol. 3 No.3 Jun-Jul 2012 521

[12 The Hadoop architecture, http://hadoop.apache.org/

[13] Savasere A, Omiecinski E, Navathe SM. An efficient algorithm for mining frequent itemsets. In: Proceedings of the 21th International Conference on VLDB, Zurich, 1995, pp.432-444.

[14] Han J W, Pei J, and Yin Y. Mining frequent patterns without Candidate Generation[C]. Proceedings of the 2000 ACM SIGMOD international conference on Management of data. ACM Press, 2000, pp.1-12.