IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-9 E-ISSN: 2347-2693

Generating Optimized Association Rule for Big Data Using GA and MLMS

Arsha Sultana [1*] and S. Madhavi [2]

[1*,2] Dept. of Computer Science and Engineering,
Prasad V Potluri Siddhartha Institute of Technology, Kanuru, India

www.ijcseonline.org

Received: Sep /09/2015 Revised: Sep/17/2015 Accepted: Sep/26/2015 Published: Sep/30/ 2015

***Abstract*—** For mining association rule different algorithms are used such as Apriori, tree based algorithm which take too much computerized time to accomplish all the frequent items. These obstacles are eliminated by using GA and MLMS and also improving the performance. In this method used a multi level minimum support of data table as 0 and 1. Genetic algorithm is indiscriminate search algorithm model based on natural selection, works in an iteration manner and is very adequate in large amount of data. Genetic algorithm is implemented in Hadoop to reduce computation cost. Hadoop supports for manipulating large data and operate them in parallel manner for better performance. The optimal frequent items are access that satisfies fitness, support and confidence.

***Keywords*—** Association Rule, Apriori algorithm, Genetic algorithm, Hadoop ,MapReduce

I. INTRODUCTION

Large amount of data is collected from different perception and gradually rising. Data storage has grown significantly from an analog to digital. Data is accumulating from gigabytes to petabytes and it cannot be processed using traditional database management system. To conquer this obstacle Bigdata is used. Dissemination is essential to manage infinite amount of data. For dissemination distributed file system is used. One of the most popular distributed file systems is Hadoop. Hadoop is a java based programming framework that supports the processing of large amount of data in a distributed environment. The hadoop ecosystem consists of hadoop kernel, MapReduce, HDFS, HBase and Zookeeper.

Hadoop Distributed File System that provides a high-throughput access to data. HDFS uses a master/slave architecture in which master consists of single Name Node and slave one or more Data Node [6]. Hadoop MapReduce is a software framework easily to write applications which process the big data in parallel on large clusters. The two tasks that are performed on MapReduce program are Map Task and Reduce Task. The key/value pair is used in map task and reduce task. In map task input data is converting into a dataset and disintegrate into key/value pair. After map task, the reduce task takes the output of map task as input and combine into a smaller tuple [8].

HBase is an open source distributed database model which is written in java. It runs on the top of HDFS providing Big table. HBase stores large amount of sparse data. Tables in HBase provide the input and output for MapReduce jobs run in hadoop and access through API. Zookeeper is a consolidated service for maintaining configuration information, naming and providing group service [9].

II. RELATED WORK

In Data Mining, association rules are useful for analyzing and anticipating customer behavior. Association rule mining is to find out association rules that satisfy the minimum support and confidence. Frequent and infrequent items are produced by using different algorithms. A minimum support inception is applied to find all frequent items in a data. A minimum confidence constraint is applied to those frequent items to produce rules. The association rule is of the form F1->F2 where F1 is antecedent and F2 is consequent.

Support: It is the possibility of item in a given transactional data. Support (F1->F2) =Support (F1UF2).

Confidence: It is the conditional possibility. Confidence (F1->F2) = Support (F1UF2)/Support (F1).

Most well known algorithm for generating association rules are Apriori algorithm. It uses breadth-first search to count minimum support and produce candidate items. By using this candidate items association rules are generated. The Apriori property is all subsets of frequent items must also be frequent [1]. The following are the steps that are used in Apriori algorithm

Step 1: Scan all the data to count number of existence of each item.

Step 2: Assign Minimum Support Count Value.
Step 3: Produce Candidate items and scan for support count (minsup>sup).
Step 4: Finally, Frequent items are generated.

Genetic algorithm is a universal search algorithm used to produce good optimal solution to obstacle that cannot be solved easily by using other approaches. They consist of population of individual solutions that are pursuing upon by a set of genetic operators in order to produce new and improved solution. The following are the steps that are used in genetic algorithm.

Step 1: An arbitrary population of solutions is produced used as initial population.
Step2: In the population, the solutions are classified to determine their fitness.
Step3: Based on their fitness, solution pairs are selected and then combined to produce offspring, which are added to the next generation of population.

The figure1 that is analyzed and used in project for the genetic algorithm [3]. First, the dataset is loaded for mining the association rule which is based on Support and confidence. After accomplished frequent and infrequent items, rules are produced based on correlation factor.

Now, initialize the parameters such as selection, crossover and mutation and also it achieves number of iterations. From various solutions, initial population is chosen randomly and size depends on the problem. Select an indiscriminate point on the two parents and then split at crossover point. Childs are produced by exchanging the route. Mutation is done randomly changing 0's and 1's and it is range of 0.6 to 0.9

Parent 1 01010010 | 1101
Parent 2 10101001 | 1011

After Crossover the child's are generated

Child 1 01010010 | 1011
Child 2 10101001 | 1101

New parent is generated

Parent 101010011011

Finally, Mutation will be done to get an optimal solution

Child 10010010010

The selection is done based on fitness function. Fitness value is taken between 0 and 1. The value should be greater than are equal to fitness will be classified.

Otherwise, crossover and mutation will be done and then again the item is selected to see whether it satisfies the condition or not. The iteration will be done until the best solution is achieved.

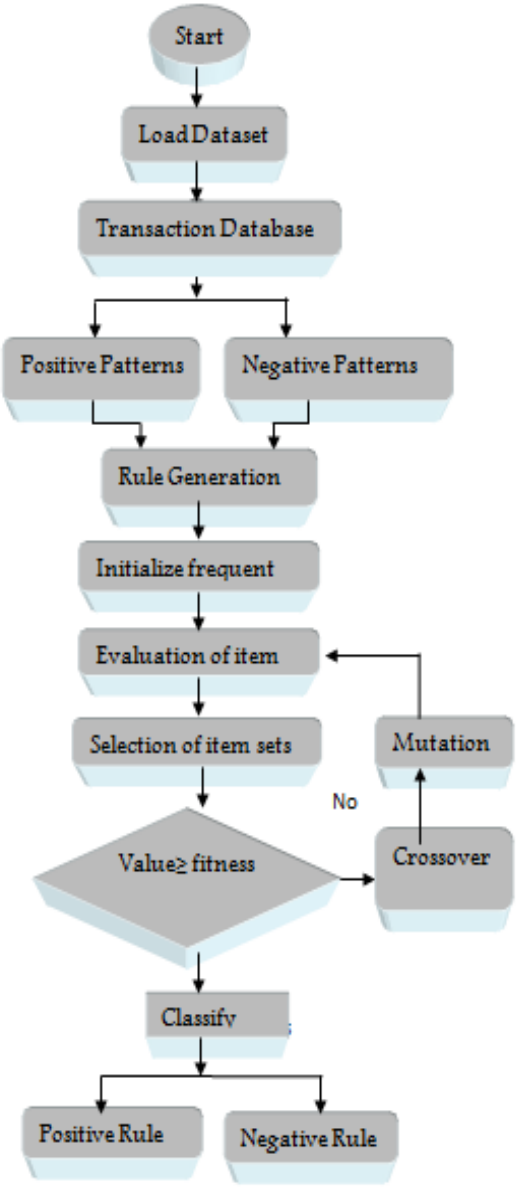

Fig.1. GA Block Diagram

III. IMPLEMENTATION

MapReduce is a functional programming used in Artificial Intelligence and it received highlight. Since Google revive to solve the obstacles to analyze Bigdata, determine more than petabytes of data in distributed environment. There are two tasks in mapreduce program those are Map task and Reduce task.

There are three important stages in mapreduce programming; they are mapper, combiner and reducer. In

mapper each item is split and counted. In combiner same items are combined and each count is given. Finally, in reducer each item counts are calculated and item with final count is displayed.

Apriori algorithm using MapReduce

Apriori MapReduce algorithm runs on parallel mapreduce framework such as Hadoop. The prune function is used to abolish the non-frequent items from a given data because it cannot be a subset of frequent items. The algorithm starts with calculate frequent items for each map node and then collect frequent items. After collecting, abolish the items that does not meet minimum support count in reduce nodes. In reduce node frequent items are calculated by joining, sorting and eliminating the equivalent nodes in map node.

Map transaction T in data origin to all map nodes
// in each map node 'a'
C_{a1} = frequent items at the node 'a'
// measure C_1 and L_1 with C_{a1} in reduce
C_1 = frequent items
// min support = assign value
$L_1 = C_1 \cap$ min support
For (j=1; Lj! =Ø; j++)
// in each map node
$L_{aj}=L_j$ mapped to each node 'a'
// sort to abolish equivalent items
$C_{a\,(j+1)} = L_j$ join_sort L_{aj}
// Usage of Apriori property in reduce
Compute C_{j+1} with $C_{a\,(j+1)}$
If (j>=5) prune C_{j+1}
For each transaction T in data origin with C_{j+1} do
// in each map node 'a' increase the count of all candidates in $L_{a\,(j+1)}$
End
// in reduce find L_{j+1} with $L_{a\,(j+1)}$ and min supp
$L_{j+1} = C_{j+1} \cap$ min supp
End
Return L_k

In mapreduce programming, first the number of items are counted and kept in HBase table which is created by using command 'create <table name>, <column family>'. HBase table and input data are compared to create candidate sets. From these candidate sets association rules are generated.

Genetic algorithm using MapReduce

Genetic algorithms are progressively applied to large scale problems. Genetic programming is an explicit application of genetic algorithms used to derive computer programs. The basic program consists of selecting the fittest members of a population, crossing and mutating them. Crossover needs to be carefully refined because the solutions that are expand infinitely longer in strings rather than chromosome.

Genetic operators

New population members can be derived by implementing a number of genetic operators in order to associate actual chromosomes [5]. There are three genetic operators and they are

Selection: The selection process prefer candidate individuals based on their fitness from population.

Crossover: The crossover process takes place between two successive individuals with possibility specified by crossover rate. These two individuals exchange segments that are separated by crossover point.

Mutation: The mutation operator is enforced to each bit of an individual with a possibility of mutation rate. When applied, a bit whose value is 0 is changed into1.

Simple Genetic algorithm pseudo-code using MapReduce:

1. Load a sample of file..
2. Apply Apriori algorithm to find frequent items with min supp.
3. Represent each frequent item as a binary string.
4. Choose initial population of individuals.
5. Assign the fitness of each population in individual.
6. Repetition of generation is done until termination (time limit, fitness satisfactory etc)
   a. Select best-fit individual for replication
   b. Breed new individuals by using crossover and mutation to create offspring
   c. Assign individual fitness of new individuals
   d. Replace least fit population with new individuals

These Genetic operators are used in programming for producing optimal solution to an obstacle. The time complexity is also reduced by using this genetic algorithm. The association rules and candidate sets that are generated in Apriori is taken as input in genetic mapreduce program and by using fitness condition the optimal solution obtained.

IV. RESULTS

Frequent items are mined from large data by Apriori. Appropriate input data is seized and apply Apriori program on that data.

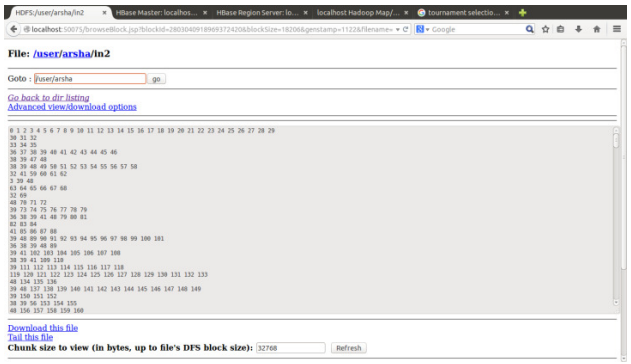

Fig.2. Input data

Now candidate sets are produced and then frequent items are also produced by using min supp condition

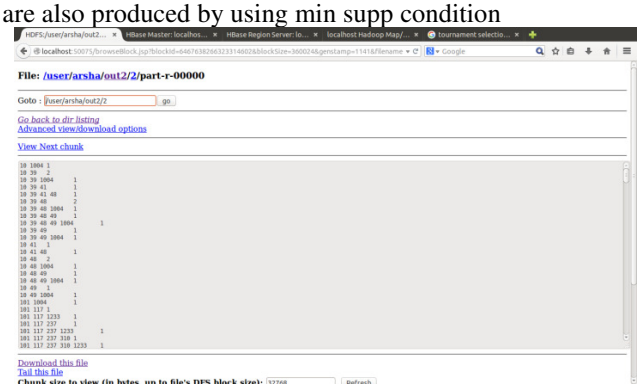

Fig.3. Candidate sets

By using these candidate sets association rules are produced. Mining frequent items are the famous in association rule

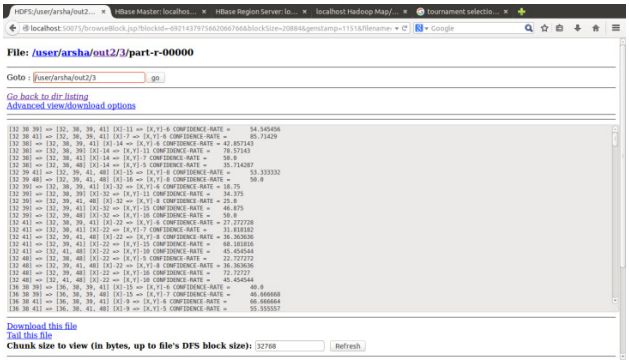

Fig.4. Association rule

Finally, Genetic program is applied to these rules. In genetic programming the input data is converted into bits 0 and 1. For crossover, mutation it is needed to be in 0 and 1. The output is displayed in 0 and 1.

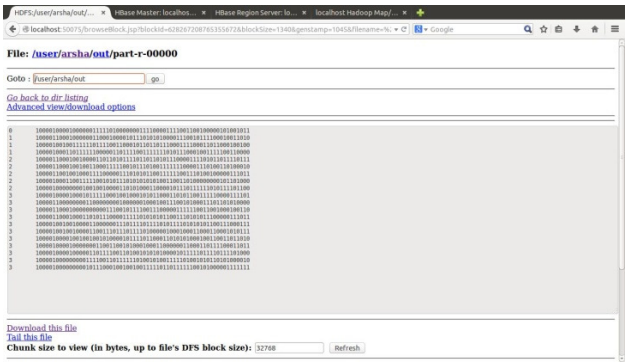

Fig.5. Output of GA

In Apriori the execution time for producing candidate set is 2min57sec and association rules are generated in 1min. Now the optimal result of Genetic algorithm is 52sec.

V. CONCLUSION AND FUTURE WORK

In Apriori mapreduce programming the candidate sets and association rules are generated in 2min57sec and 1min. The candidate sets generated are more and also computation time is high. To eliminate these obstacles genetic mapreduce program is used and results of apriori are taken as input and convert into 0's and 1's for genetic operators. The result obtained in 52sec better compared to apriori.

In Genetic optimal solutions are produced. Any input to Genetic algorithm is converted in 0 and 1 because crossover, mutation is done in bits of string. In future these bits are transmitted into normal data.

REFERENCES

[1] Mohammed Al-Maolegi, Bassam Arkok, "An Improved Apriori Algorithm for Association Rules", Int. Jounal on Natural Language Computing, Volume-03, No.1, Page No (21-29), February 2014.

[2] Soumadip Ghosh, Sushanta Biswas, Debasree Sarkar, Partha Pratim Sarkar, "Mining Frequent Itemsets Using Genetic Algorithm", Int. Journal of Artificial Intelligence and Applications, Volume-01, No.4, Page No (133-143), October 2010.

[3] Nikky Suryawanshi Rai, Susheel Jain, Anurag Jain, "Mining Interseting Positive And Negative Association Rule Based On Improved Genetic Algorithm", Int. Journal of Advanced Computer Science and Applications, Volume-05, No.1, Page No (160-165), 2014.

[4] D. Kerana Hanirex and K.P. Kaliyamurthie, "Mining Frequent Itemsets Using Genetic Algorithm", Middle-East Journal of Scientific Research, 19 (6), Page No (807-810), 2014.

[5] Pratibha Bajpai, Dr. Manoj Kumar, "Genetic Algorithm- an Approach to Solve Global Optimization Problems", Int.

Jounal of Computer Science and Engineering, Volume-01, No-03, Page No (199-206).

[6] Apache Hadoop, http://hadoop.apache.org/ , Monday, April 6, 2015.

[7] Stephen Kaisler, Frank Armour, J. Alberto Espinosa, William Money, "Big Data: Issues and Challenges Moving Forward", 46th Hawaii International Conference on System Sciences, Page No (995-1004), 2013.

[8] Srinath Parera, Thilina Gunarathane, "Hadoop MapReduce Cook Book", [PACKT] publishing, ISBN: 9781849517287, Page No (5-115), Jan 2013.

[9] Apache HBase, http://hbase.apache.org/ , Friday, July 10, 2015.

AUTHORS PROFILE

MD. Arsha Sultana is presently M.Tech Student, Dept. of Computer Science & Engineering, Prasad V Potluri Siddhartha Institute of Technology (Autonomous), kanuru, India.

Dr. S. Madhavi is presently Professor, Dept. of Computer Science & Engineering, Prasad V Potluri Siddhartha Institute of Technology (Autonomous), kanuru, India.