Method For Mini Frequent Patterns From Large Data-Sets

M. Krishnamoorthy[1], Dr.R. Karthikeyan[2]

[1]Research Scholar, Department of Computer Science and Engineering
Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India.
[2]Associate Professor, Department of Computer Science and Engineering
Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India.

ABSTRACT

Frequent pattern mining is an important field of research in data mining. It has piqued the interest of many researchers since its inception. Data generation and collecting increase in size exponentially diagonally. Knowledge discovery and decision making necessitate the ability to process and extract relevant information from "Big" Data in a scalable and efficient manner. The use of refined analysis to vast volumes of data in order to discover new knowledge in the form of patterns, trends, and associations is known as data mining. Decision-making and information retrieval necessitate a scalable and effective approach for processing and extracting important information from Big Data. Data mining is the sophisticated study of huge amounts of data to discover new information in the form of patterns, trends, and relationships. With the spread of the World Wide Web, the amount of data stored and made available electronically has increased dramatically, and methods for retrieving information from such large amounts of data have grown in importance for both the business and scientific research communities. Frequent Item Set Mining is one of the most widely used methods for extracting relevant information from data. Recent advances in parallel programming have provided excellent methods for overcoming this challenge. Nonetheless, these tools have technical limitations, such as unbiased data sharing and inter-communication costs. In this paper, we investigate the use of Frequent Item Set Mining in the Map Reduce architecture. Big-Frequent-Item set Mining is a new method for extracting big datasets. This approach is designed to work with exceedingly large datasets. Our method is similar to FP-growth, but it employs a distinct data structure based on algebraic topology. We also focused on hybrid Apriori to generate frequent patterns, and the Apriori algorithm's association rule is then optimised using a genetic algorithm. To generate strong association rules, Apriori algorithm association rules were subjected to Genetic Algorithm operators such as selection, crossover, and mutation. To mine recurrent designs with a user-specified lowest provision, a parallel algorithm has been proposed. To compute frequent item sets, the work is distributed among n processors. As a result, the processors will communicate. When compared to other algorithms, the time obligatory to comprehensive the task is very short.

INTRODUCTION

Above past several years, through progress in storage space expertise, repository devices are turn into greater and further efficiently practical. By way of a consequence, commerce, big establishments, etc., are in progress putting away and extracting numerous kinds of data in arrangement of huge data sets. The principle of keeping information is bifold. Initially to fetch the data in the forthcoming, and next, for analysis and finding correlations among data items placed in data sets. The job of finding associations among the data items is called as association rule mining. The main stimulus originates from market basket analysis [Fan Jiang et al]. As soon as in a superstore a consumer approaches to purchase objects, the likelihood is the customer to purchase several other precise s. In recent times, investigators ensure discovering directions for data wherever presence of item is unsure [Kun He et al.]. Data mining has involved a prodigious covenant of interest in IT industry and in civilization as intact in current time, outstanding to extensive convenience of enormous quantities of information also impending obligation for spiraling thoseinformation into practical data and facts.

FREQUENT ITEMSET EXTRACTION

Frequent itemsets are orders found regularly in a data source. Learning recurrent itemsets is critical for removing associations and other inspiring influences among data. Reflect a example, a tilt of items, such as brush and paste, that appear to be frequently self-possessed within transaction data. Consider the following scenario: if you buy a computer first, then a data card, and finally a pen drive, and if this pattern occurs frequently in a spending past data set, that pattern is a frequent itemset. A substructure can refer to various essential methods, such as sub-graphs and sub-trees, that are combined with itemsets. Extraction of Frequent Itemsets looks for repeated connections in a data set. Market basket scrutiny is the first method of obtaining association rules by extracting common itemets.

PROBLEM STATEMENT

Every frequent itemset is typically extremely large, necessitating the use of multiple applications. The subset required by these requests frequently comprises a small number of itemsets. As a result, more time is consumed seeing all unwelcome recurrent itemsets in order to extract frequent itemsets. Furthermore, memory is wasted by storing all unimportant frequent itemets. As a result, restraints can be familiarized to eliminate these insignificant itemsets. As a result, it is necessary to endorse a capable technique for discovering recurrent itemsets using Big Dataset constraints. Using almost entirely Frequent Item-Set removal approaches, frequent 1-itemsets are generated to regulate the support value (incidences) of each item current in the entire data set. This job is the aforementioned a mind-numbing job in producing frequent itemsets once bearing in mind the immensity of contemporary data sets presented. On no account unequivocal approach has been delineated in these methods to achieve the afore mentioned job. So proficient data structure and algorithms can be proposed to determine support value of each item.

CONTRIBUTIONS

This paper proposed the improved Apriori approach and it signifies compared with the Apriori algorithm. The algorithms are applied on massive data groups and it has been proved that improved Apriori is more effectual than Apriori method. Association rules produced by Apriori method is enhanced using genetic method approach, and then it is compared with the association rule produced by Apriori method by taking automatically generated data sets and then it has been proved that genetic algorithm based procedure has generated an effective association rule when compared to the Apriori algorithm association rule.

Parallel algorithm have been proposed by having 9 processors and large data sets, implementation work was carried out and compared with the other algorithm, and the time taken by the parallel method is low when equated with former methods and the disadvantage is cost. It is very costlier when associated to other algorithms. Partition algorithm has been projected and likened with Apriori and improved Apriori algorithm. Massive datasets have been taken for comparison and study of results demonstrates that suggested method has an outstanding impact compared to preeminent current methods.

METHODOLOGY

The methodology to be followed to complete this research and achieve its objectives is demonstrated in Figure.1.1 also consists of the subsequent phases:

Data Collection: we collect largest freely public available Big Data that represents market basket analysis from document with multiple domain.

Data Preprocessing: Some preprocessing is carried out. It consists of removing stop words, tokenizing the string to words, and then applying appropriate term stemming.

Design the Parallel FP Tree Approach we build the parallel FP-Tree algorithm for large volume data based on MapReduce model.

Design the hybrid Apriori Approach: we build the parallel Apriori algorithm for large volume data using the MapReduce model.

Designing an Optimization Approach: We create an optimization process that is used in conjunction with Hybrid Apriori to reduce association rules that do not fit the given confidence level.

The algorithm is implemented using the Java programming language and the Hadoop platform with a multicomputer cluster on the largest freely available business transaction-based big data sets.

Evaluation: Using various performance metrics, the proposed approach is evaluated for speedup and execution time**.**

Results and argument: In this section, we analyse the obtained results and justify the effectiveness of the suggested and planned approach**.**

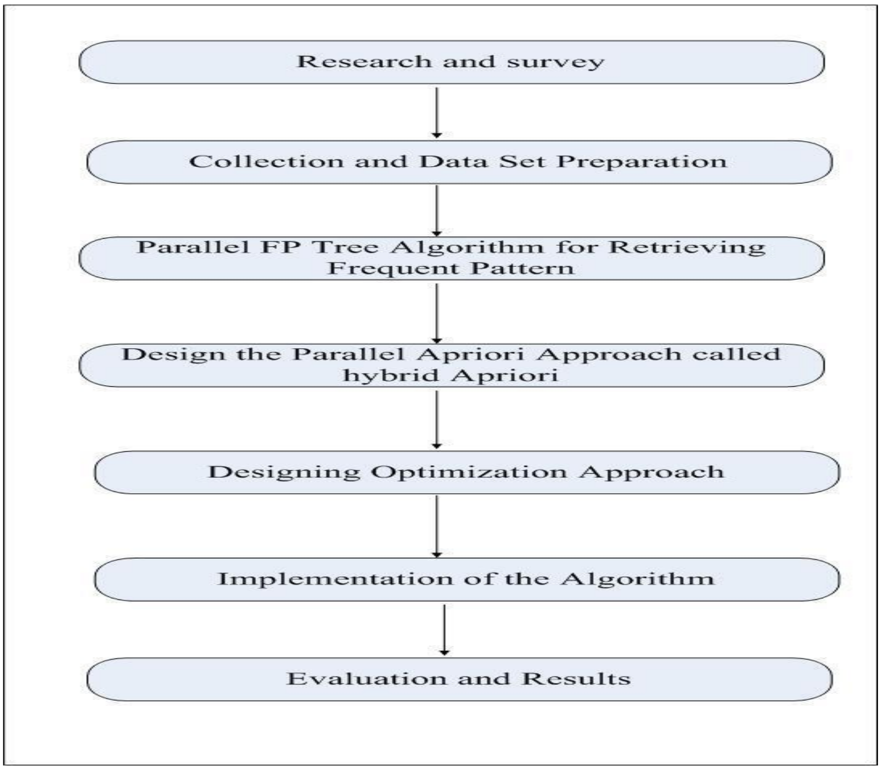

Research and survey
Collection and Data Set Preparation
Parallel FP Tree Algorithm for Retrieving Frequent Pattern
Design the Parallel Apriori Approach called hybrid Apriori
Designing Optimization Approach
Implementation of the Algorithm
Evaluation and Results

Figure.1.1 Stages of the Research Methodology

PARTITION METHOD TO MINE FREQUENT PATTERNS FROM LARGE DATA-SETS

Partitioning becomes, to begin with, carried out in Oracle database technology 8.0 and is the maximum necessary and useful practical feature of the Oracle database. Partitioning is a way of dividing information sets, tables, indices, and tables organized by way of index into smaller remains known as dividers, permitting statistics set gadgets to be performed and retrieved at a larger scale. Every partition is diagnosed by using its specific call and has its own characteristics which include storing and directory. To show the separating method, consider a Human Resource supervisor who has one large field containing employee documents. Each record specifies the worker's beginning date. Queries for personnel who joined in a given month are routinely performed. One response is to create an index based totally on the employee connection date that specifies the positions of the documents scattered in the course of the container. Another choice is to use a partitioning approach. This uses some of the decreased containers, with every container containing documents for personnel who joined in a selected month. There are several blessings to the usage of smaller packing containers.

PARTITIONING STRATEGIES

A database or a data set As elementary partitioning approaches, partitioning provides three essential data dispersal methods that regulate how Information is located in discrete walls. There are three of them: range, hash, and listing. Data units may be partitioned using single-degree partitioning or complex-level partitioning, depending on the data dispersal approaches. Single-stage partitioning employs only one information distribution manner, together with variety, , list or hash, in extra or one columns as the dividing key.

Compound dividing is an aggregate of two facts distribution approaches that define a compound divider table. A data set is to begin divided through one records dispersion method, and then every divider is similarly alienated into subpartitions by another data dispersal technique. Meta-data refers to the partitions of a composite partitioned table and does not denote specific data storage.

MINING FREQUENT ITEMSETS USING PARTITIONING ALGORITHM

A divider impression has been deliberate to growth the strolling time pace at the bottom viable value. In facts insert into the facts set, distinct partition directions are formed for each object-set, i.E., 1-itemset, 2-itemset, three-itemset, etc. To start, a list of common 1-itemsets is diagnosed via analyzing the dataset and etching numbers of incidences of each object from the partition of that unique item set using the pointer; item-sets enjoyable the minimum guide be counted are blanketed inside the common-1itemset L1. Apriori-methodL1aim to decide L2, the list of L2then worn to discover L3, and so on, till no more common ok-itemsets are located. It is not necessary to study the entire information set to reap Lk; the procedure is sufficient for pursuing the tally of each facts object-set from its divider. Originally, vast belongings called the Apriori belongings to become used to reduce the search area with a view to generating common object-set. The common itemsets may be obtained thru steps: Join and Prune. Lk is determined in a be a part of operation from a listing of candidate-okay itemsets Ck created via connectingLk- 1over itself.

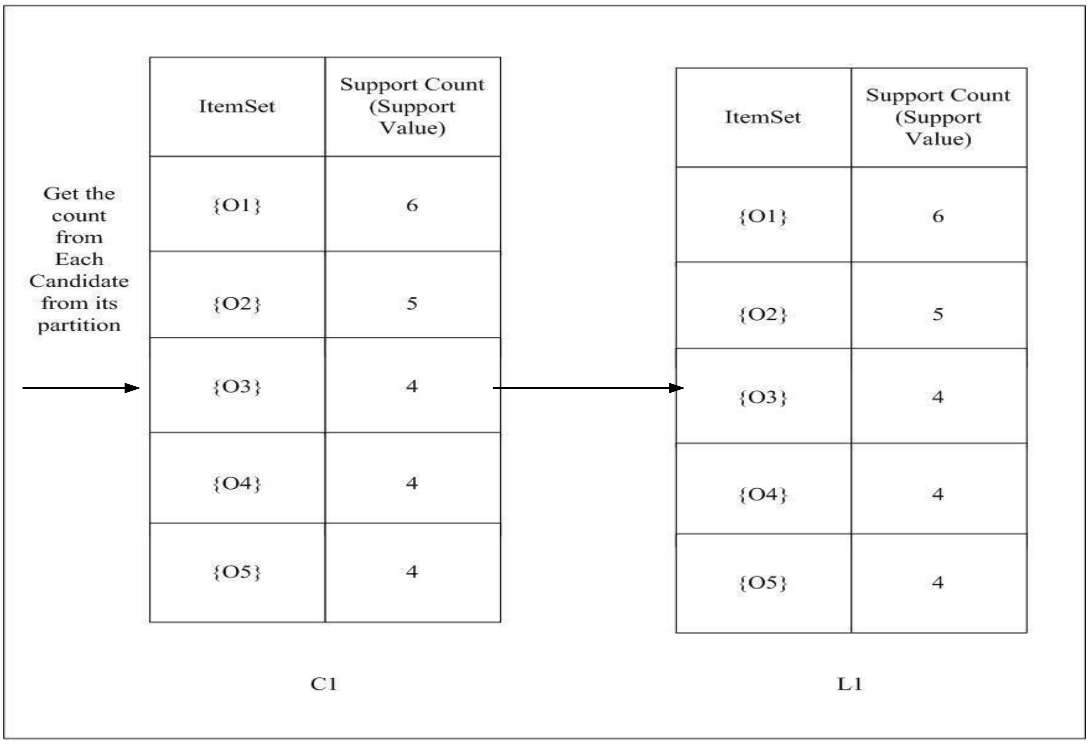
To decide the remember(quantity) of each applicant in Ck, the divider of each object-set might be verified, and the depend that is not lesser than the minimal aid count is more common and additionally pertains to Lk. The Apriori characteristic is used to reduce the dimensions of Ck. When in comparison to other existing Algorithms, the partition set of rules plays well in locating the numerous common information objects.

TID	List of Items
TR1	O1, O2, O4
TR2	O2, O4
TR3	O3, O4
TR4	O1, O4
TR5	O1, O2, O5
TR6	O1, O2, O3
TR7	O1, O2, O3, O5
TR8	O1, O3, O5

Table.5.1 Transaction Data Set for Partition Method

Consider Data Set D, which has seven transactions. Partitioning is now used to collect and get better information from the records set. There are numerous partitioning strategies to be had; in this paper, range dividing changed into used to improve enactment when removing recurrent patterns from the data set.

Table.5.1 includes eight-transaction transaction records set for the divider set of rules. In the desk underneath, transaction T1 holds I1, I2, I3, and transaction T2 holds I2, I4, and so forth. It employs applicant 1 itemset C1 to generate occurrence 1 itemset L1 by manipulative the range of incidences of the statistics items straight from the divider as opposed to studying the whole facts set once more. The minimum support count number in this case is two. Candidate-1 object-set that satisfies the minimal support be counted and is protected in L1. Figure.5.1 depicts the formation of applicant one itemsets and common one itemsets.

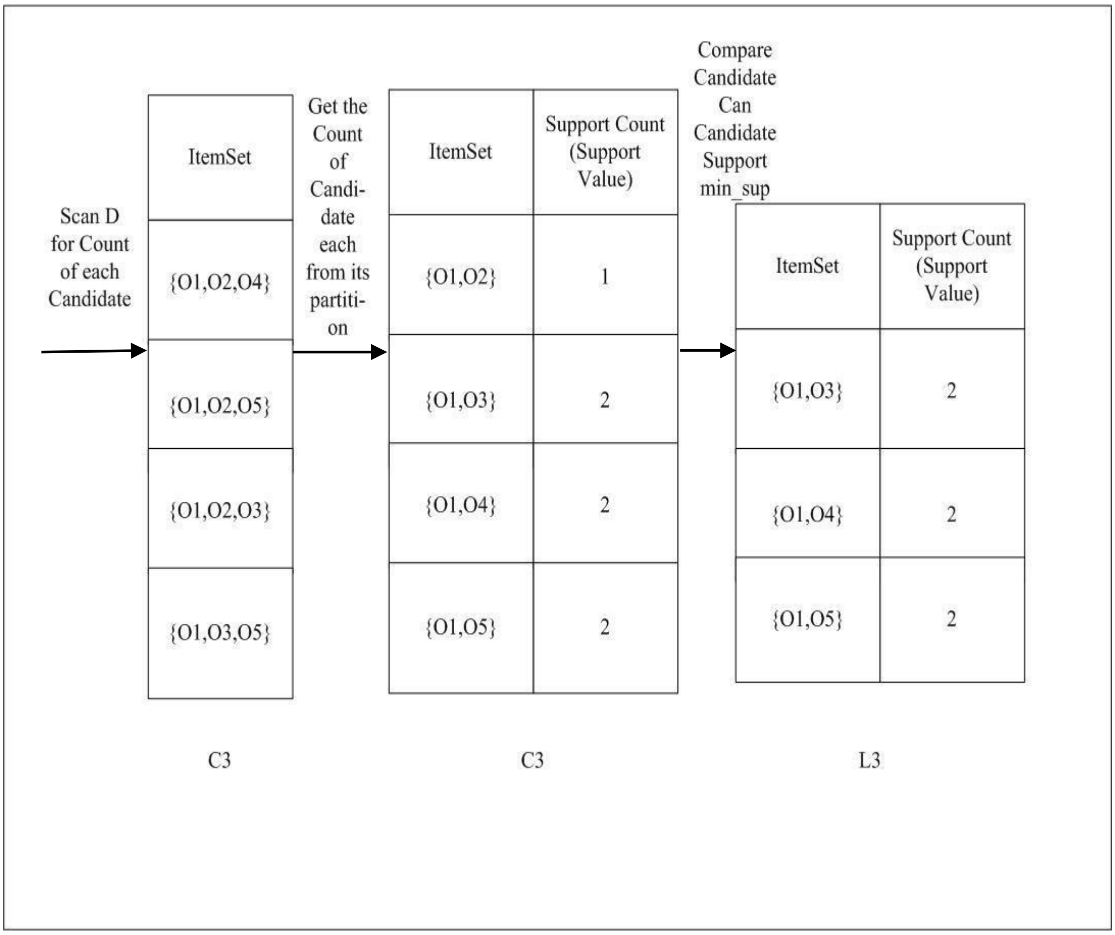


ItemSet	Support Count (Support Value)
{O1}	6
{O2}	5
{O3}	4
{O4}	4
{O5}	4

ItemSet	Support Count (Support Value)
{O1}	6
{O2}	5
{O3}	4
{O4}	4
{O5}	4

Figure.5.1 the development of candidate-1 itemsets and frequent-1 itemsets.

The union of L1 produces the candidate two itemsets, including the verification of whether the subset of the recurrent itemsets is more frequent, because in L1 the total items have been included, and there is no clipping here. Instead of interpretation the entire data set to compute the support value, this is sufficient to retrieve the count of the appropriate divider. The frequent-two itemsets will be comprised of the candidate-2 itemsets that meet the minimal support value. Figure.5.2 shows how the partition determines the creation of frequent-two itemsets. The last eight frequent two itemsets were created and used to generate the candidate two itemsets.


Scan D for Count of each Candidate
Get the Count of Candi-date each from its partiti-on
Compare Candidate Can Candidate Support min_sup
C3
C3
L3

ItemSet
{O1,O2,O4}
{O1,O2,O5}
{O1,O2,O3}
{O1,O3,O5}

ItemSet	Support Count (Support Value)
{O1,O2}	1
{O1,O3}	2
{O1,O4}	2
{O1,O5}	2

ItemSet	Support Count (Support Value)
{O1,O3}	2
{O1,O4}	2
{O1,O5}	2

Figure.5.2 Generation of Frequent 2- Itemset using Partition Algorithm

The applicant 3 itemsets are made by union of L2 by itself and determining whether the subset of common item-sets is again recurrent. Only the itemsets O1, O2, O4, O1, O2, O3, O1, O2, O5, and O1, O3, O5 have been combined for the next stage, which is known as applicant 3 itemsets since its subset is also a recurrent itemset. Because its subset is not recognised as frequent, the residual itemsets have been separated. Instead of reading the entire data set to estimate each supported value, simply notice the count from the appropriate partition.

EXPERIMENTAL EVALUATION

The partition algorithms was tested using Big Data sets and associated to the Apriori algorithms. The graph below depicts the performance of the Partitioning algorithm.

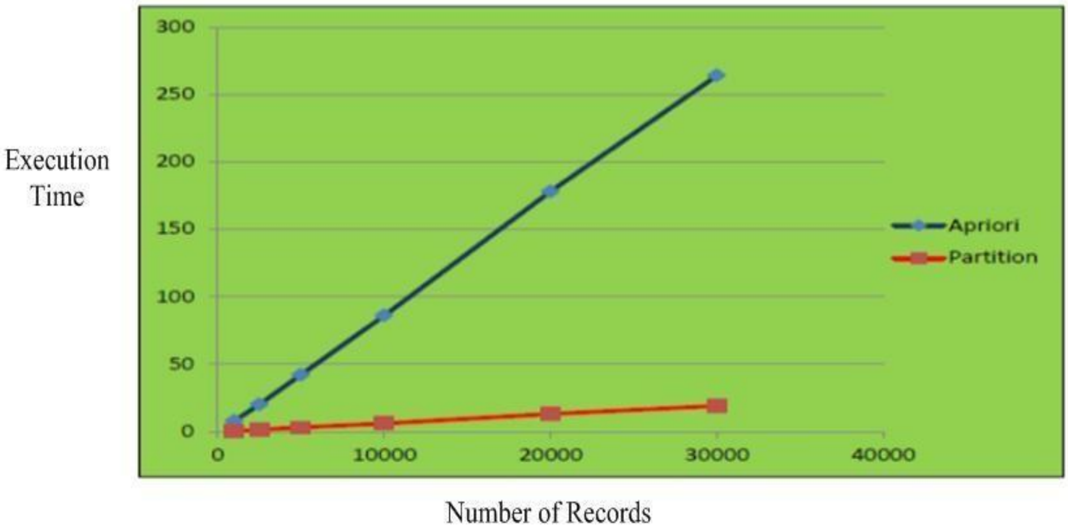


Figure.5.4 Performance evaluation of Partitioning algorithm

Figure.5.4 shows how frequent itemsets are created, as well as an association rule generation graph and table. Apriori and partition algorithms have been graphed in Figure 5.1. The x-axis in this graph represents the total numbers of record, and the y-axis represents the execution time. According to the graphical representation, the partition algorithm outperforms the Apriori algorithm.

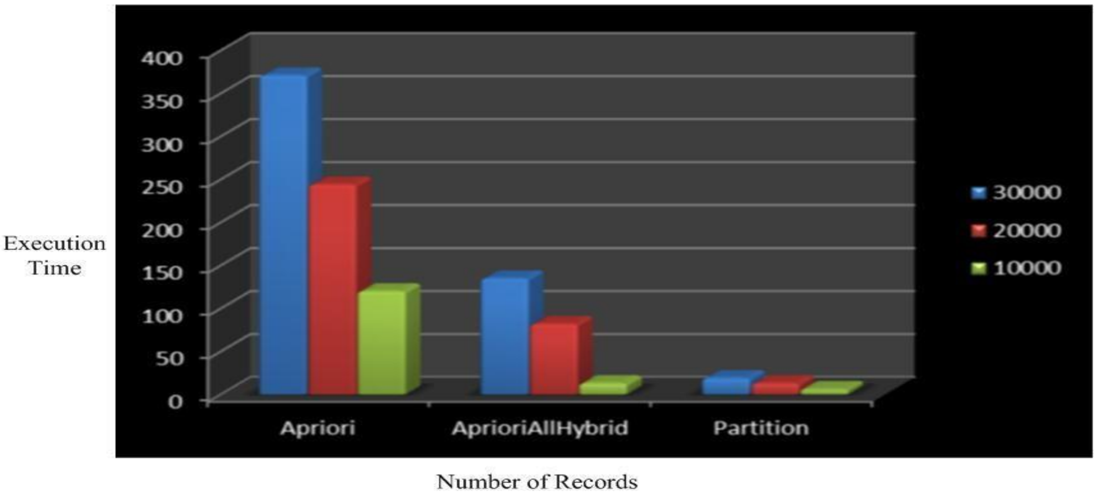


Figure.5.5 Comparison of Apriori, Apriori Hybrid and Partition algorithms

The scale of 10000 means 10 lakh, 20000 means 20 lakh, and so on as the number of customer transactions in the data set increases. When we compared the partition method to other algorithms, it took less time to identify frequent patterns and association rules in the data set by taking into account the items that belonged to transactions.

COMPARISON AND ANALYSIS OF PARALLEL ASSOCIATION RULEMINING ALGORITHMS

We developed three parallel Association Rule Mining Algorithms in our research to extract Frequent Item-Sets and Association-Rules from Big Data Sets. BigFM, Parallel

Apriori with Genetic Algorithm for Optimization, and Partition Based Apriori are the algorithms. When implemented, those algorithm outperformed corresponding traditional FP Growth, Apriori algorithm that extract Frequent Item - Set and Connotation - Rules from conventional statistics units. In this segment, we are able to compare the 3 Parallel Association Rule Mining algorithms BigFM, Parallel hybrid Apriori the use of Genetic Algorithm, and Partition based Apriori set of rules to look at how those algorithm produce connotation regulations and achieve implementation efficiency based on time difficulty and area difficulty parameters.

BIGFM Vs PARALLEL HYBRID APRIORI

This segment tested the implementation times of the BigFM and hybrid Apriori algorithms for various minimal help values. We investigated the execution time with various minimal guide values and increased the number of mapper. A tiny Hadoop-2.6.0cluster of five nodes is established, all of which are going for walks Ubuntu 14.04. One node serves as the Name Node, whilst the alternative 4 feature as Data Nodes. Name Node has four cores and four gigabytes of reminiscence, and it runs in a virtualized environment on a window host. Two Data Nodes run on special physical machines, each having four cores and two gigabytes of memory. Other Data Nodes, each with 4 cores and 4 gigabytes of memory, are strolling in a virtualized environment on the same window host. All algorithms are written in Java and hire the MapReduce 2.0libraries. Experiments had been executed on both synthetic and actual-time datasets. We measured the execution time of algorithms at the aforementioned datasets for numerous minimum support values. In all algorithms, we employed 4 reducers. Because each break-up is assigned one mapper, the wide variety of mappers is proportional to the number of input splits. The number of splits will increase because of the chunk length, i.e. The number of lines of input, decreases. We've set chunk sizes of 5K and 6.5K.

The tables below show the execution performance of Parallel BigFM and Parallel Hybrid Apriori over single and four node Hadoop clusters.

The following Table.6.1 and chart diagram compare Parallel Hybrid Apriori and Parallel BigFM Algorithms on a single-node Hadoop Cluster.

Number of Transactions	**Parallel BigFM Algorithm (Time in secs)**	**Parallel Hybrid Apriori Algorithm (Time in secs)**
20000	78.340	33.599
50000	216.492	81.694
90000	603.050	384.879
100000	661.511	626.736
150000	1311.729	1035.475
200000	3971.819	3392.179
250000	6368.553	6227.533

Table.6.1 Parallel Hybrid Apriori Vs Parallel BigFM Algorithm over single-node Hadoop.

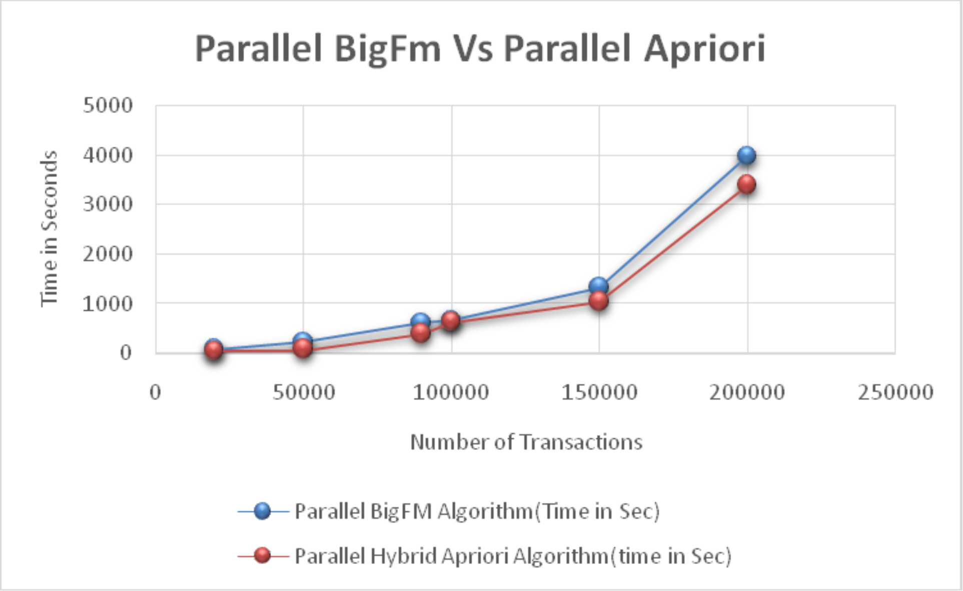


Figure.6.1 Parallel BigFm Vs Parallel Apriori Algorithm over single-node Hadoop

The following table represents Comparison of Parallel Hybrid Apriori and Parallel BigFM Algorithms over Four-node Hadoop Cluster.

Number of Transactions	**Parallel BigFM Algorithm (Time in secs)**	**Parallel Hybrid Apriori Algorithm (Time in secs)**
20000	38.340	18.820
50000	92.054	45.84
90000	154.607	54.87
100000	250.673	101.943
150000	377.843	164.004
200000	600.154	261.13
250000	916.679	366.267

Table.6.2 Parallel Hybrid Apriori Vs Parallel BigFM Algorithms over Four-node Hadoop Cluster.

The subsequent Figure.6.2 explains the Table.6.2, assessment of Parallel Hybrid Apriori and Parallel BigFM Algorithm over Four-node Hadoop Cluster.

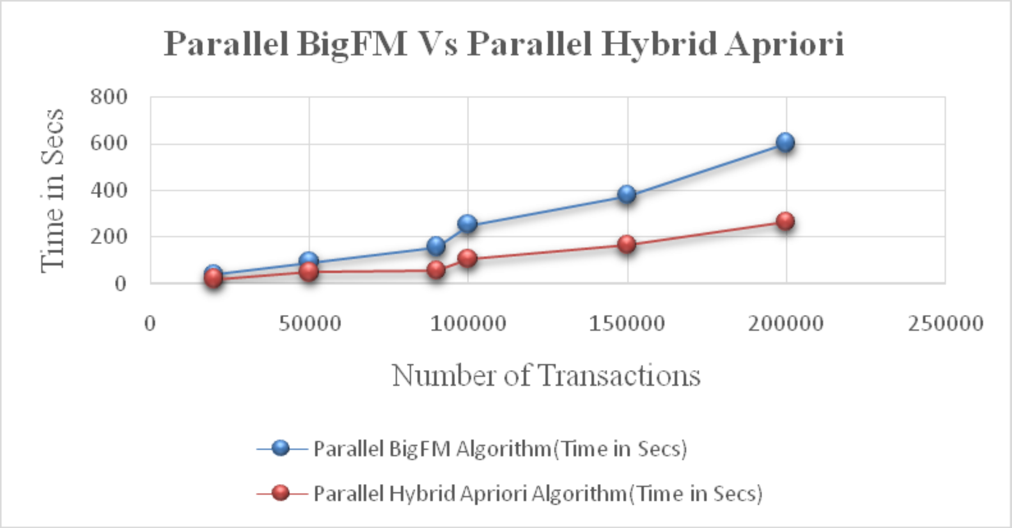


Figure.6.2 Parallel Hybrid Apriori Vs Parallel BigFM Algorithm over Four-node Hadoop Cluster.

The subsequent diagram explains performance of BigFM and Parallel hybrid Apriori using Genetic Algorithm

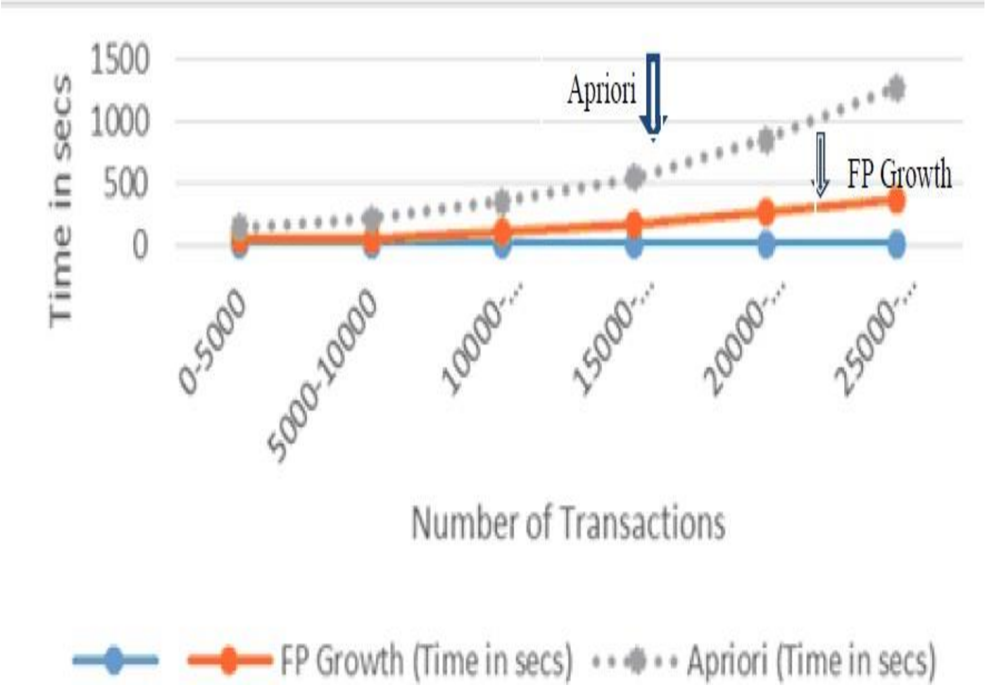


Figure.6.3 BigFM Vs Parallel hybrid Apriori using Genetic Algorithm.

The BigFM performs better than conventional FP growth algorithm. And Parallel Hybrid

Apriori performs better than conventional Apriori algorithm. Whenwe compare these two the results show the Parallel Apriori using genetic algorithm for optimization, performs better than Hybrid Apriori in terms of time complexity.

PARALLEL APRIORI VS PARTITION BASED APRIORI

In this section, we compare Parallel hybrid Apriori to Partition-based Apriori. The observational tests were provided to investigate the outcome of The number of information references, the number of datasets touching on numerous record, and the dimensions of the dataset in phrases of numerous items. Complete tests were carried out on 4 virtual machines that had been going for walks "Intel (R) Core (TM) i5-2450M CPU @ 2.50 GHz, 2501 MHz, 2 Core(s), 3 Logical Processor(s) Pentium(R) with 6 GB of foremost reminiscence jogging in the direction of Windows 7 Home Premium Edition." The four virtual machines have been given access to the datasets used inside the experiment. For one or more of the experiments, the following outcomes have been obtained: (i) reaction time; and (ii) communique overhead. The investigations centered on various the minimal support threshold from 0% to one hundred% of total dealings, contingent on the dataset used Sequence returns from tests revealed that the response time of Partition-based Apriori outperformed Parallel Hybrid Apriori.

The diagram below depicts the response time performance of hybrid Apriori and Partition-based Apriori.

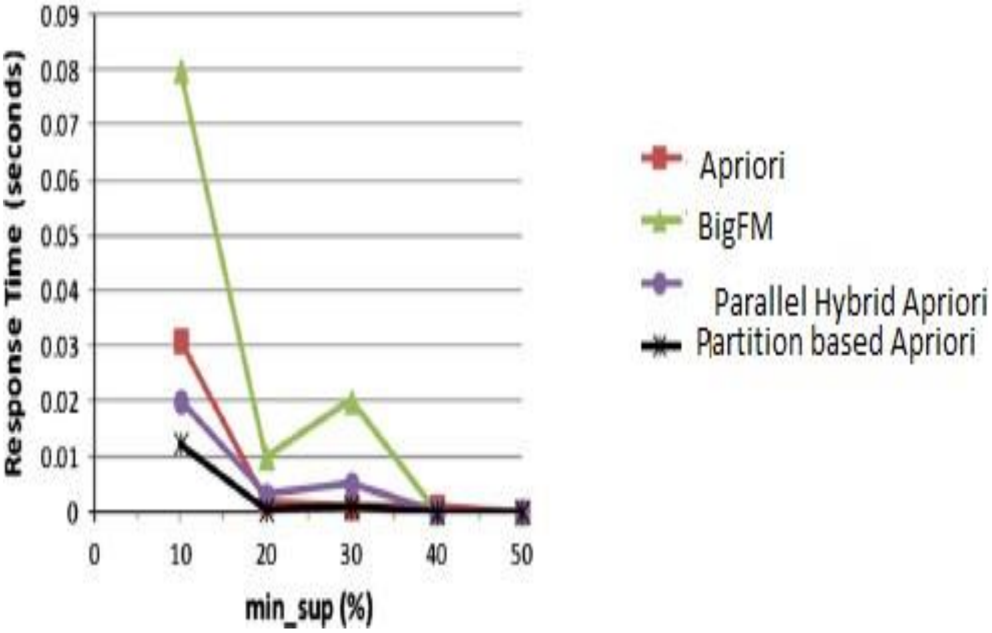


Figure.6.4 Hybrid Apriori and Partition based on Apriori algorithms byvarying min_sup, including 80 percent min-conf.

The following presents the performance Response Time and Min-Support (% (percent)).

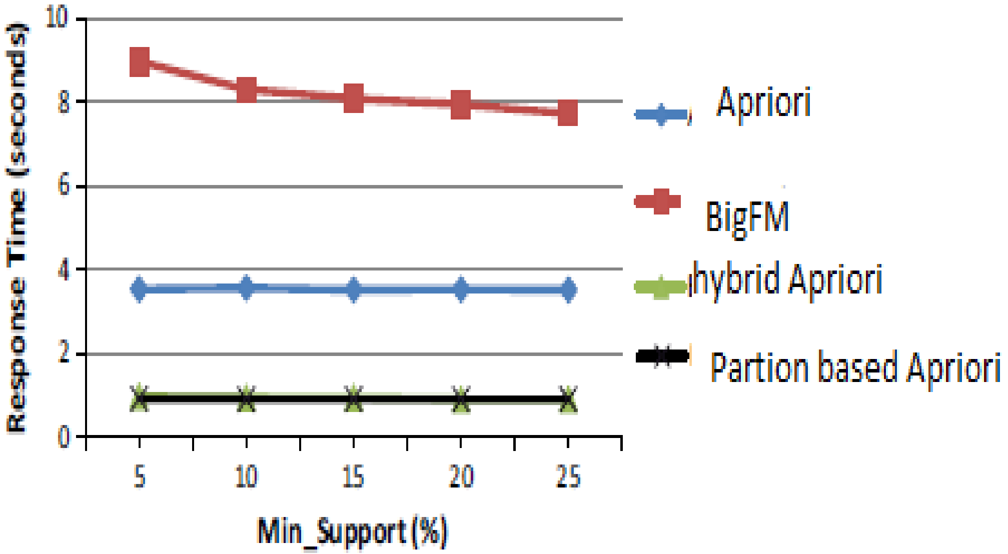


Figure.6.5 Performance Response Time and Min-Support (%).

The following shows the performance Response Time and No of Data Sets.

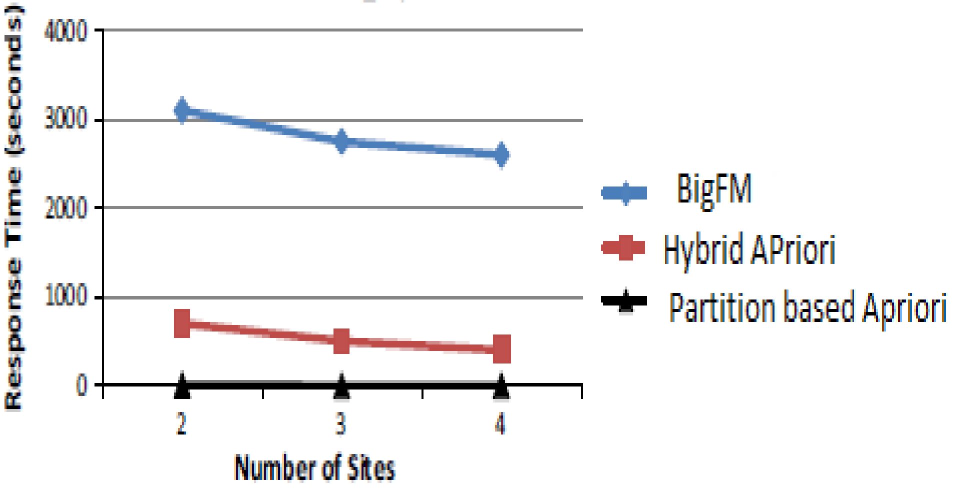


Figure.6.6 Performance Response Time and No of Data Sets.

CONCLUSIONS

An overall comparison of all proposed methods is presented. Extensive experimentation task was presented toward evaluating and examining Apriori Hybrid and Genetic Algorithms. By the above comparative analysis, the suggested and planned partition algorithm approach time and again achieved well compared to the rest of the algorithm (conventional techniques) and will assist several future types of research in many ways.

REFERENCES

1. Abdulla, Z., Herawan, T., Norazia, A. and Deris, M. M. (2014) 'A Scalable Algorithm for

2. Constructing Frequent Pattern Tree', International Journal of Intelligent Information Technologies, 10 (1), pp. 42-56.
3. Agrawal, R. and Srikant, R. (1994) 'Fast Algorithms for Mining Association Rules', In Proceeding of the 20th International Conference of Very Large Databases (VLDB), Santigo, Chile, pp. 487-499.
4. Agrawal, R. C., Agrawal, C. C. and Prasad, V. V. V. (2000) 'Depth First Generation of Long Patterns', In Proceeding of the 6th International Conference on Knowledge Discovery and Data Mining, pp. 108-118.
5. Agrawal, R., Imielinski, T. and Swami, A. (1993) 'Mining Association Rules Between Sets of Items in Large Databases', In Proceedings of the ACM SIGMOD InternationalConference on Management of Data (ACM SIGMOD '93), Washington, USA, pp. 207-216. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo, A. I. (1996) 'Fast Discovery of Association Rules', Published in Advances in Knowledge Discovery and DataMining, AAAI Press, pp. 307-328.
6. AlZoubi, W. (2015) 'An Improved Graph Based Method for Extracting Association Rules',

7. International Journal of Software Engineering & Applications (IJSEA), 6(3), pp. 1-10. Ansari. E, Dastghaibifard. G. H and keshatkaran. M, (2008), “Distributed Trie Frequent Itemset Mining,” In Proceedings of International Multi Conference of Engineers and Computer Scientists, 1(), pp. 978-988.
8. Appice, A., Berardi, M., Ceci, M., and Malerba D. (2005) 'Mining and Filtering Multi-level Spatial Association Rules with ARES', Proceedings in 15th International Symposium, ISMIS 2005, Saratoga Springs, NY, USA, pp. 342-353.
9. Ayres, J., Gehrke, J., Yiu, T. and Flannick, J. (2002) 'Sequential Pattern Mining using A Bitmap Representation', In Proceedings of ACM SIGKDD International Conference onKnowledge Discovery and Data Mining (KDD 2002), pp. 429-435.
10. Balaji Raja, N. and Balakrishnan, G. (2011) 'An Improved Algorithm of Graph and Clustering Based Association Rule Mining (GCBARM) in Discovering of Frequent Itemsets', The Journal of Management, Computer Science & Journalism, 6(2), pp. 6-10.
11. Barbara, D., Couto, J., Jajodia, S., Popyack, L., and Wu, N. (2001) 'ADAM: Detecting Intrusions by Data Mining', In Proceedings of the IEEE Workshop on Information Assurance and Security Symposium (NDSS’00), pp. 157-170.
12. Bastide, Y., Pasquier, N., Taouil, R., Stumme, G. and Lakhal, L. (2000) 'Mining minimal non-redundant association rules using frequent closed itemsets', In First International Conference on Computational Logic, pp. 972-986.
13. Bayardo, R. and Agrawal, R. (1999) 'Mining the Most Interesting Rules', In Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD '99), SanDiego, California, USA, pp. 145-154.
14. Beniwal, S. and Arora, J. (2012) 'Classification and Feature Selection Techniques in

Data Mining', International Journal of Engineering Research and Technology (IJERT), 1(6), pp. 1-6.

15. Bhalodiya, D., Patel, K. M. and Patel, C. (2013) 'An Efficient Way to Find Frequent Pattern with Dynamic Programming Approach ', In Nirma University International Conference on Engineering (NUiCONE), pp. 1-5.

16. Bhujade, V. and Janwe, N. J. (2011) 'Knowledge Discovery in Text Mining Technique Using Association Rules', Published in Computational Intelligence and Communication Networks (CICN), 2011 International Conference on, Publisher: IEEE, pp. 498-502.

17. Bica, M. (2006) 'Apriori Error Estimation in Terms of the Third Derivative for the Methodof Successive Approximations Applied to ODE'S', Journal of Applied Mathematics and Computing, 22, pp. 199-212.

18. Bodon, F. and Ronyal, L. (2003) 'Trie: An Alternative Data Structure for Data Mining Algorithms', Proceeding in Mathematical and Computer Modeling, Elsevier Ltd., 38, pp. 739-751.

19. Briandais, R. D. L., (1959) 'File searching using variable-length keys', In Western Joint Computer Conference, pp. 295-298.

20. Brin, S., Motwani, R., Ullman, J. D. and Tsur, S. (1997) 'Dynamic Itemset Counting and Implication Rules for Market Basket Data', In Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, 26(2), pp. 255-264.

21. Cao H., Jiang Z. and Sun Z. (2007) 'Fast Mining Algorithm for Multilevel Association RulesBased on FP-Tree', Computer Engineering, 19(25).