International Journal of Engineering Research and Technology.
ISSN 0974-3154 Volume 11, Number 2 (2018), pp. 287-306

http://www.irphouse.com

Mining Frequent Patterns from Big Data Sets using Genetic Algorithm

Chandaka Babi
Research Scholar, Dept of IT,
Gitam Institute of Technology, Gitam University,
Visakhapatnam, India

Dr. M. Venkateswara Rao
Professor, Dept of IT,
Gitam University, Visakhapatnam, India.

Dr. Vedula Venkateswara Rao
Professor, Dept of CSE,
Sri Vasavi Engineering College, Tadepalligudem, India.

Abstract

Frequent pattern mining is crucial data mining job, with a goal of determining knowledge in the form of repetitive patterns. Many proficient pattern mining algorithms have been revealed in the last two decades, yet most do not scale to the type of data we are presented with today, the so-called "Big Data". Scalable parallel algorithms hold the key to solving the problem in this context of Big Data**.** Most of the existing algorithms toward this issue are based on exhausting search methods such as Apriori, and FP-growth. However, when they are applied in the big data applications, those methods will suffer for extreme computational cost in searching association rules. Huge amounts of digital information are stored in the World today and the amount is increasing by quintillion bytes every day. Processing of this information using traditional techniques is becoming increasingly difficult. The challenges include capture, storage, search, transfer, analysis, and visualization of big datasets. 90% of the data stored in the World today have been created in the last two years alone and continue to grow at the same pace. However, these sheer amounts of data bring significant benefits and make it possible to do many things that could not be done previously: predict future behavior of Internet users, diagnose diseases, identify crime suspects, forecast financial trends, etc. Approximate (probabilistic) data mining algorithms are very important to efficiently deal with such amounts of information and to deliver the results in an acceptable

timeframe required by various real-world applications. While approximate algorithms do not deliver precise results, the availability of theoretical estimations of their errors enables data analysts to decide if the precision offered by an approximate method satisfies the needs of the particular application. Exact data mining methods, on the contrary, tend to be slow and are best employed where the precise results are of the highest important. In this paper we focus on hybrid Apriori to generate frequent patterns and then the association rule generated by the Apriori algorithm is optimized using genetic algorithm. To generate strong association rules, Genetic Algorithm operators like selection, crossover and mutation have been applied on association rule generated by Apriori algorithm. The parallel algorithm has been proposed to mine the frequent patterns with a user specified minimum support. The work is shared among n number of processors to compute frequent item sets. So there will be communication between the processors. The time taken to finish the task is very less when compared to other algorithms.

Keywords: Big Data, Data Mining, Frequent Pattern Mining, Frequent Item Sets, Hybrid Apriori, Association Rules, Genetic Algorithm, Optimization.

I. INTRODUCTION

1.1 Overview:

Data Mining is the most valuable and widely used process for the exploration and analysis of large quantity of data to acquire valid, novel, potentially useful and intelligent patterns hidden in database. Areas like finance, banking, retail sales, production, population study, employment, monitoring of human or machines etc., have ways to record known information, but cannot tackle the uncertainties of the future due to lack of tools to use this known information. The exponential growth of data and databases every now and then has necessitated the development of new techniques and tools that could intelligently transform data into useful information and knowledge. This is achieved by cleaning the data, integrating the data, extracting information stored in the database, transforming data; discovering the valuable patterns and knowledge presentation etc. Data integration merges the data from multiple data stores and improves the accuracy and speed of the subsequent data mining process by avoiding redundancies and inconsistencies in the resulting data set. Data cleansing routines attempt to fill in missing values, smooth out noise and correct inconsistencies in the data. In data transformation, data is transformed into forms, appropriate for mining, by smoothing, aggregation, normalization and discretization. Various Knowledge representation techniques are used to present the mined knowledge to the user. In business, planning and estimating of future values are very important. The commodities industry needs prediction for forecasting of supply, sales, and demand for production planning, sales, marketing and financial decisions.

1.2 Frequent Pattern Mining

Frequent patterns are patterns that appear frequently in a data set. Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data. For example, a set of items, such as paste and brush, which appear frequently together in a transaction data set, is a frequent itemset. Consider the scenario, such as buying first a PC, then a data card, and then a pen drive, and if this pattern occurs frequently in a shopping history database, then that pattern is a frequent pattern. A substructure can refer to different structural forms, such as sub graphs, subtrees, or sub lattices, which may be combined with itemsets or subsequences. Frequent pattern-mining searches for recurring relationships in a given data set. Market basket analysis is the earliest form of frequent pattern mining for association rules.

1.3 Motivation

For mining frequent patterns, several algorithms are available, where the problem developed was, to finding an efficient algorithm and extracts the knowledge with the database information. This scenario motivated to find the suitable algorithm for mining frequent patterns and to handle the issues involved in this processing more efficiently in the real time development system. Knowledge discovery in the database is the task of understanding the problem domain and extract the useful new knowledge. Knowledge discovery requires technical knowledge and domain knowledge and should have experience based on knowledge mining responsibilities. This motivated to find the suitable algorithm to mine a frequent pattern from a massive datasets. In this study, different categories of algorithms like fast algorithm for mining frequent patterns from massive data sets, optimized association rule mining using genetic algorithm, parallel algorithm for mining frequent patterns and partition algorithm for mining frequent pattern were analysed and proposed an improved algorithm for mining frequent patterns. Massive implementation works have been carried out to compare the results obtained from proposed algorithms with earlier algorithm. From the result, partition algorithm has better performance over existing algorithms.

II. RELATED WORK

2.1 Introduction to Data Mining:

The important trend in information technology is to identify the meaningful information from the enormous amount of data stored in files, databases, and other repositories and to develop powerful means for analysis and interpretation of such data for the extraction of interesting knowledge that could help in decision-making. The two main reasons for using data mining are, having very small information with too much data and another one is to extract useful information from huge amount of data. Data mining helps for discovering meaningful new correlation, patterns and potentially useful diamonds of knowledge from large amount of data using statistical

and mathematical techniques. The concept of data mining has emerged since many years, but the term data mining have been introduced in 1990. Data mining have gone through several research and developmental phases for many years, and because of three different families, it has reached its form and those families are statistics, artificial intelligence (AI) and machine learning. Data mining is built on statistics which is the foundation of most technologies, e.g. standard deviation, standard variance, regression analysis, standard distribution, clusters analysis etc. Artificial intelligence is also the base for data mining, which tries to simulate human thought process or human intelligence in statistical problems. Another core area for data mining is machine learning, and it is a combination of statistics and AI. Data mining is basically the adaptation of machine learning techniques to business applications. Data mining is the collection of historical and recent developments in statistics, AI, and machine learning. These techniques are used to study and find hidden patterns or knowledge available in the data. Data mining task can be divided into two types; they are descriptive data mining and predictive data mining. Descriptive data mining tasks specifies the general properties of the existing data, and predictive data mining performs inference on the collected data sets and attempt to make predictions. The data mining functionalities are Characterization, Discrimination, Association Analysis, Classification, Prediction, Clustering and Outlier Analysis.

2.2 Frequent Patterns Mining Evolution

Frequent patterns are item sets, sub sequences, or substructures that appear in a data set that satisfies the user-specified minimum support count with frequency not less than a user-specified threshold. Agarwal et al, proposed an efficient algorithm that generates all the association rules between the items in the customer transaction database and this algorithm incorporates novel estimation, buffer management and pruning techniques. Agarwal et al proposed two algorithms Apriori and Apriori Tid for solving the problem of mining frequent patterns, and then combined both of these algorithms. The Apriori and AprioriTid algorithms generate the candidate item sets to be counted in a pass by using only the itemsets found large in the previous pass not considering the transactions in the database. Srikant et al presents the mining of generalized association rules from a large database of transactions. Jiawei Han et al describe mining association rules at multiple concept levels resulting in the discovery of more specific and concrete knowledge from data. Ming-syan chen et al., presents a comparative study and classification of the available data mining techniques. Han developed DBMiner a relational data mining system at Simon Fraser University, for mining multiple kinds of rules at different abstraction levels, like characteristic rules, discriminant rules, association rules, classification rules etc. Show-Jane Yen et al. presents an efficient graph based algorithm to mine the frequent data items in the database and this algorithm constructs association graph to indicate the associations between items.

2.3 Big Data

Big data is a term for data sets that are so large or complex that primitive data processing application software is inadequate to deal with them. Big data represents a new period in data study and utilization. It is a leveraging open source technology- a robust, secure, highly available, enterprise-class Big Data platform. Challenges include capture, storage, analysis, querying, and updating data safely and securely. While the term "big data" is relatively new, the doing of collecting and storing plethora of information for eventual analysis is ages old. The significance of big data is not based on how much data we have, but how we use that data. We can take data from any source and analyze it to find responses that enable us to produce results in reduced cost and time with smart decision making. Here in this paper we are trying to combine big data with genetic algorithms for generating efficient analysis of data. The reason for the interest in genetic algorithms is that these are very powerful and broadly applicable search techniques. As said earlier also, Big Data refers to large volume, complex, growing data sets with numerous, self-directed sources. Big Data are now rapidly expanding in all fields like science and engineering, including physical, biological and biomedical sciences with the fast development of networking, data storage, and the data collection capacity

2.4 Introduction of Hadoop

Hadoop is the Apache open source framework indicted in Java which sanctions distributed processing of astronomically immense datasets across collection of computers system utilizing programming framework. The Hadoop framework application works in an environment that provides storage in distributed manners and computation across clusters of computers. Hadoop is setup to such a manner that single server have a lot of machines working and each will provide local computation process and storage system. Those are some costly to create larger servers with more efficient configurations that will process large scale data, but as another way, we could add together many cluster of computers with single-CPU, as a single functional distributed system. The clustered machines could read the dataset in parallel and give a higher throughput results

2.5 Architecture of Hadoop

In the architecture of hadoop we describe the various components of hadoop like mapreduce job processing, handling the data and architecture of file system. Hadoop deploys master/slave architecture for distributed computation and distributed storage. The various component of Hadoop:

1. Processing / computation layer (MapReduce)
2. Storage layer (Hadoop Distributed File System)
3. Hadoop command
4. Hadoop YARN

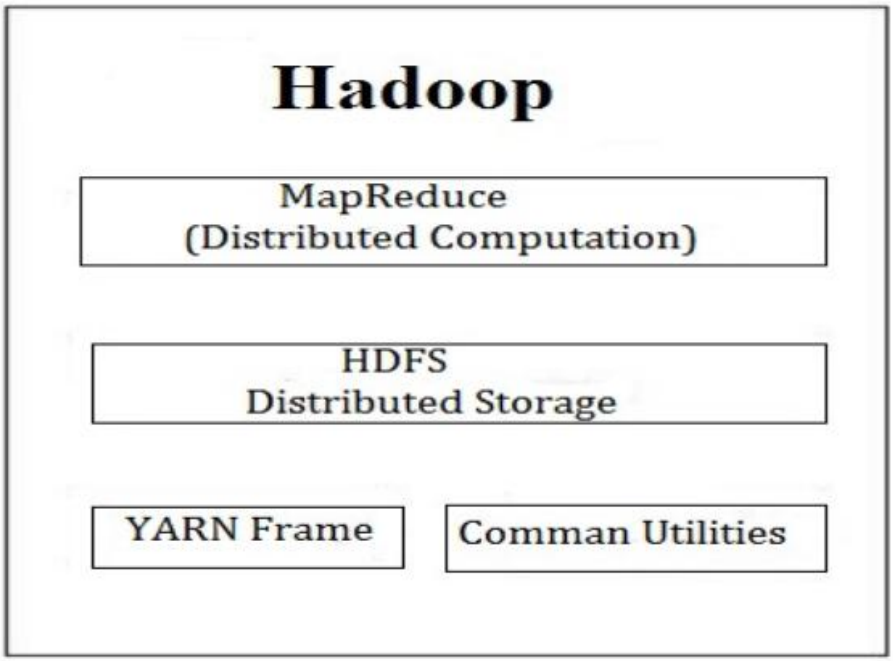

Figure 2.1 Hadoop Architecture

2.6 MapReduce

MapReduce framework are proposed by Google, used to write application to processing large dataset , collateral way , on more cluster of hardware system in efficient way. MapReduce is processing technique and programming model for distributed computing based on JAVA programming language. The MapReduce algorithms have two components namely Map and Reduce. Map takes the large volume of dataset and converts them into broken form of tuples as (key/value) pairs and Reduce phase takes the output of maps phases and combine them into smaller tuples sets. According to name sequence Reduce task will perform after the Map phase operations. MapReduce framework is designed by Google by seeing the behaviors of map reduce functional programming languages. MapReduce is enables to easily use for large distributes problems. In MapReduce framework there are two components, one is map and other is reduce. MapReduce is designed for large volume of data set. It is introduced for Big Data Analysis [16] and it is used a lots of algorithms like Breadth-First Search, Traveling Salesman problems, Finding Shortest Path problem etc. In this framework two key factor, Map and reducer. The Map which is parallely divided the data into many cluster and each cluster the data is form of key and value. The output of map phase data will goes into intermediate phase where data will be shuffling and sorting. Then using the partitioner for dividing the data parallel in different cluster according to the user. A cluster is determined by the number of reducers. The reducers will taking all iteration of data give the results in form of values.

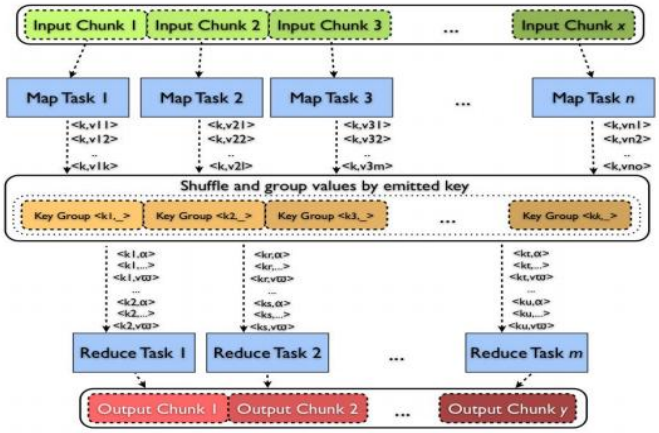

Figure 2.2 MapReduce framework.

III. HYBRID APRIORI ALGORITHM FOR FREQUENT PATTERNS FROM Big DATA

The Apriori Hybrid algorithm is shown as follows.

L1 = {Large size -1 Sequences}

C'k =Database D;

For (k=2; Lk-1 = ∅; k++)

Ck = New candidate sequences generated from Lk-1 ;

C'k = ∅;

For all entries s ∈ C'k-1

do Begin

Ct = {s∈Ck | s-C[k]) ∈ s.set-of-sequences ∧ (s-C[k]) ∈ s.set-of-sequences};

For each customer sequence c in the database

do begin

Increment the count of all candidate sequences in Ck that are contained in s;

if (Ct ≠ ∅) then C'k = C'k + < s.SID, Ct>;

End;

Lk = Candidate sequences in Ck with minimum support;

End;

Answer = ∪k Lk;

An interesting feature of the proposed algorithm is that the given customer transaction database D is not used for counting support after the first pass. Rather the set C_k is used for determining the candidate sequences before the pass begins.

Each member of the set C_k is of the form < SID, $\{S_k\}$ > where each S_k is a potentially frequent k-sequence present in the sequence with identifier SID. For k=1, C_1 corresponds to the database D, although conceptually each sequence i is replaced by the sequence {i}. For k>1, C_k is corresponding to customer sequence S is < s.SID, {s<Ck / s contained in t}>. If s customer sequence does not contain any candidate k-sequence, then C_k will not have an entry for this customer sequence. Thus, the number of sequences in the database is greater than the number of entries in C_k. The number of entries in Ck may be smaller than the number of sequences in database especially for large value of k. In addition, for large values of k, each entry may be smaller than the corresponding sequence because very few candidate sequences may be contained in the sequence. The following shows the generation of candidate and frequent itemsets.

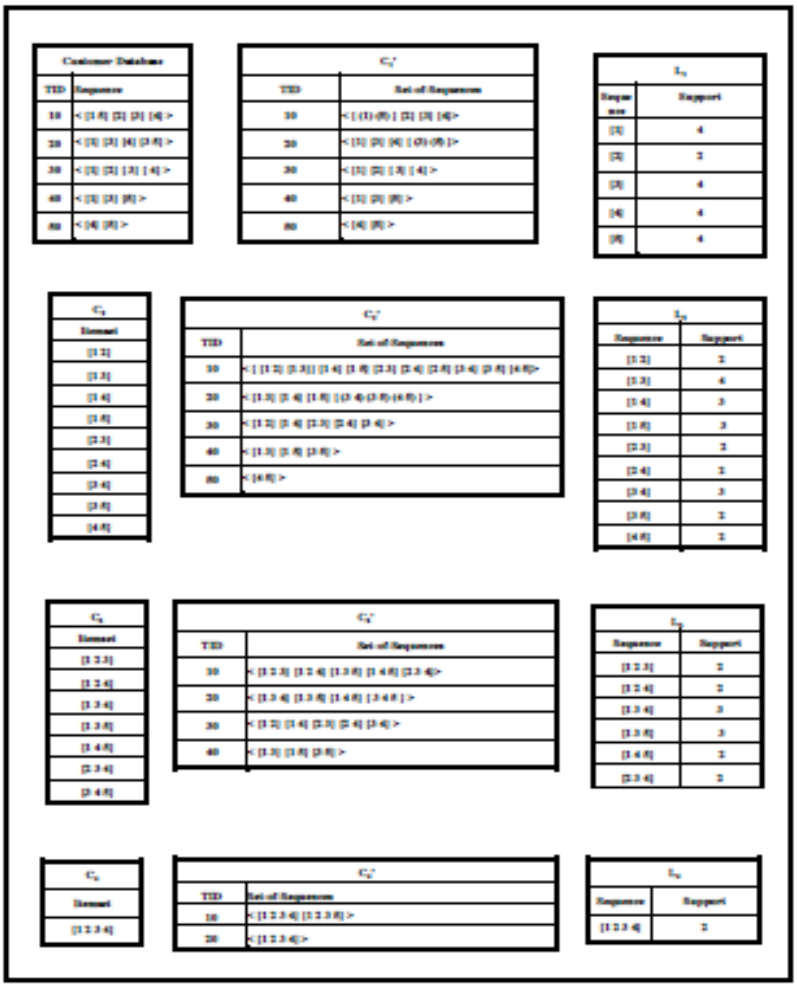

Figure 2.3 Generation of Candidate and Frequent Itemsets

IV. GENETIC ALGORITHM FOR OPTIMIZATION OF FREQUENT PATTERNS AND GENERATION OF ASSOCIATION RULES FROM BIG DATA

Genetic algorithm is one of the best ways to optimize the rules. A new fitness function has been designed for the optimization of the rule set that uses the concept of supervised learning. Then the Genetic Algorithm will be able to generate the stronger rule set. Concepts of mining associations are given as follows.

Let I = {I1 ,I2... Im } be a set of items and D = {t1,t2,…….tn} be a set of transactions, where ti is a set of items such that ti $\subset$ I . An association rule is an implication of the form X $\Rightarrow$ Y, where X,Y $\subset$ I and X $\cap$Y = $\not\in$.

The rule X $\Rightarrow$ Y holds in the set D with support and confidence, where support is the percentage of transactions in D that contain both X and Y and confidence is the percentage of transactions in D containing X that also contain Y. An association rule must satisfy user-set minimum support (min_sup) and minimum confidence (min_conf).

The rule X $\Rightarrow$ Y is called a strong association rule if support $\geq$ min_sup and confidence $\geq$ min_conf. usually association analysis is not given decision attributes so that find association and dependence between attributes to the best of our abilities. But the aimless analysis may take much time and space. Decision attributes determined can reduce the amount of candidate sets and searching space, and then improve the efficiency of algorithms to some extent. In addition, users are not interested in all association rules, but they are just concerned about the associations among condition attributes and decision attributes.

4.1 Genetic Algorithm

The Genetic Algorithm was developed by John Holland in 1970 and it is based on the genetic processes of biological organisms. Over many years, the principles of natural selection, natural population evolve and survival of the fittest was first clearly stated by Charles Darwin in the Origin of Species. Genetic Algorithm is an adaptive method which may be used to solve search and optimization problems. Genetic algorithm is a type of searching algorithm. It searches a solution space to find an optimal solution to a problem. Genetic Algorithms are powerful in their ability to model both qualitative and quantitative search spaces. The algorithm creates a "population" of possible solutions to the problem, and lets them "evolve" over multiple generations to find better and better solution.

Algorithm stars with a set of solutions called population. Solutions from one population are taken and are used to form a new population. The unsuccessful individuals would result in a value of zero for the fitness function. Successful individuals result in a maximum value for the fitness function. Some individuals result in intermediate values of the fitness function. Individuals with higher fitness have a higher probability of being selected for mating, and individuals having low fitness are killed off with a low probability of mating. The genetic processes of

crossover and mutation are applied to the individuals in the mating population, and a new generation is created.

All offsprings are made by crossover. If crossover probability is 100%, and new generation is made from exact copies of chromosomes from old population. If the crossover probability is 0% and the mutation is performed, then the part of chromosome is changed. If mutation probability is 100%, whole chromosome will be changed. If it is 0%, no change has occurred.

The algorithm operates through a simple cycle

- Creation of a population of strings.
- Evolution of each string.
- Selection of the best string.
- Genetic manipulation to create a new population of strings.

This new form of solution is called offspring, which are selected according to their fitness and this process is repeated until the condition is satisfied.

4.2 Genetic Algorithm operators

The Genetic operators determine the search capability and convergence of the algorithm. Genetic operators hold the selection, encoding of chromosomes, crossover and mutation on the population and generate the new population. In selection, Chromosomes are selected from the population. The problem is how to select these chromosomes. According to Darwin's evolution theory, the best one should survive and create new offspring. There are many methods to select the best chromosomes. For example, roulette wheel selection, Boltzmann selection, tournament selection, rank selection and, steady state selection is some of the methods for selecting the better chromosomes. Roulette wheel selection is a way of choosing members from population of chromosomes and there is no guarantee that the fittest member goes through to the next generation. To choose a chromosome, spin the ball and grab the chromosome at the point it stops. Each chromosome has one binary string and each bit in this string represents some characteristic of the solution. There are many ways of encoding, and it depends mainly on the solved problem. For example, one can encode directly integer or real numbers; sometimes it is useful to encode some permutations and so on.

The most used way of Encoding of chromosomes is a binary string and it is shown as follows.

Chromosome 1	11011100100110110
Chromosome 2	11011111000011110

Figure 4.1 Encoding of Chromosomes

Crossover selects gene from parent chromosomes and creates a new offspring. There are different ways like single point crossover and other is multipoint crossover. Single point crossover, is to choose a random crossover point and everything before this point copy from a first parent and then everything after a crossover point copy from the second parent. In multipoint cross over more than one crossover points will be chosen. The figure 4.2 shows the example of single point crossover.

Chromosome 1	11011 | 00100110110
Chromosome 2	11011 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11011 | 00100110110

Figure 4.2 Crossover of Chromosomes

Next to crossover is Mutation. The mutation operator introduces a certain amount of randomness to the search. It can help the search to find the better solutions. Mutation changes randomly the new offspring. The mutation depends on the encoding as well as the crossover. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Figure 4.3 shows the mutation operation.

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

Figure 4.3 Mutation of Chromosomes

4.3 Steps in Genetic Algorithm

1. Generate random population of n chromosomes.
2. Evaluate the fitness function f(x) of each chromosome x in the population
3. Create a new population by repeating following steps until the new population is completed

   Selection: According to the fitness function, select two parent chromosomes from a population. Chromosomes which are having better fitness will be selected.

   Crossover: Cross over the parents to form a new offspring with a crossover probability. Offspring is an exact copy of parents, if no crossover was performed

   Mutation: Mutate new offspring at each position in chromosome with the a mutation probability.

   Accepting: Place new offspring in a new population

   Original offspring 1 1101111000011110

   Original offspring 2 1101100100110110

   Mutated offspring 1 1100111000011110

   Mutated offspring 2 1101101100110110

4. Use the newly generated population for a further processing of algorithm.
5. If the end condition is satisfied, stop further processing of the algorithm, and return the best solution in current population.
6. Go to step 2.

A typical flowchart of a genetic algorithm is shown in Figure 4.4. One iteration of the algorithm is referred to as a generation.

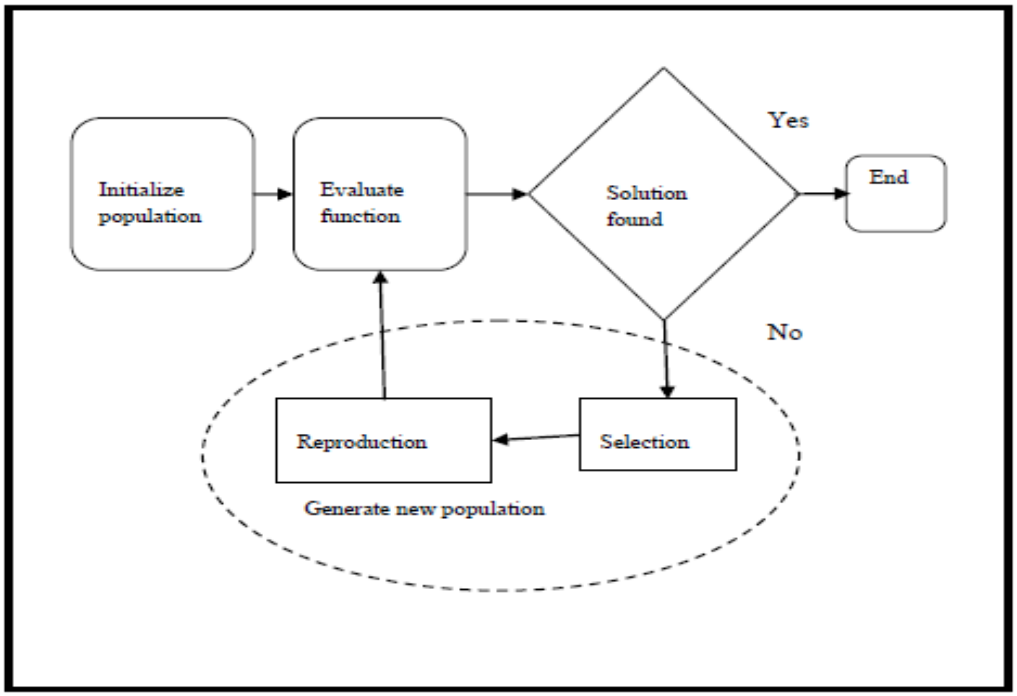

Figure 4.4 Flow chart of Genetic Algorithm.

V. EXPERIMENTAL EVALUATION

5.1 Implementation and Results of Hybrid Apriori and Genetic Algorithm

The experiment uses abandon dataset obtained from UCI machine learning repository. The data set has 4000 samples. It is composed of a discrete attribute and 8 continuous attributes. Here mining is applied to such association rules X ⇒ Y that Y was age. The initial population size taken is 1000, (ie) 1000 parent chromosomes have been taken for this experiment, and the maximum number of generations created is fixed as 500, so that, not more than 500 generations are created, and probability of cross over fixed as 0.7. According to that, crossover has happened and the mutation probability is fixed as 0.05 for this experiment.

The setting of parameters:

Initial population size—1000;

Maximum number of generations—500;

Probability of crossover—0.7;

Mutation probability—0.05;

Initial range of real-valued Strings —(0.000 067; 0.0002);

Table 5.1 Performance Evaluation between Hybrid Apriori and Genetic Algorithm

Attribute	**A**	**B**	**C**	**D**	**E**	**F**
Min Support	**2**	**4**	**6**	**8**	**9**	**7**
Min Confidence	**0.1**	**0.2**	**0.3**	**0.1**	**0.2**	**0.1**
Hybrid Apriori	**121**	**137**	**50**	**98**	**84**	**28**
Genetic Algorithm	**84**	**115**	**42**	**75**	**70**	**22**

Table 5.1 shows the result of those rule generated by Apriori and genetic algorithm. Here, 6 different values of support and confidence for 6 different attributes have been specified and number of rules generated by Apriori and Genetic algorithm were calculated and presented in the table 5.1. When the min support for attribute A is 2 and min confidence is 0.1, the Apriori Algorithm generates 121 association rules, whereas Genetic Algorithm generates 94 association rules. For the attribute C, with the minimum support 6 and the minimum confidence 0.3, Apriori algorithm generates 50 association rules, and genetic algorithm generates 42 association rules. For the attribute D, with the minimum support 8 and the minimum confidence 0.1, Apriori algorithm generates 98 association rules, and genetic algorithm generates 75 association rules. . For the attribute E, with the minimum support 9 and the minimum confidence 0.2, Apriori algorithm generates association rules and genetic algorithm generates 70 association rules. When the min support for F is 7 and min confidence is 0.5, then the association rules generated by Apriori Algorithm is 28, whereas 22 association rules have been generated by Genetic Algorithm. As a result genetic algorithm has generated less number of strong association rules when compared to Apriori algorithm. Thus the Genetic Algorithm has better performance over Apriori Algorithm in generating the association rules.

Figure 5.1 shows the Apriori/Genetic Algorithm rule generation graph. Using table 5.1, graph has been plotted for Apriori and GA.

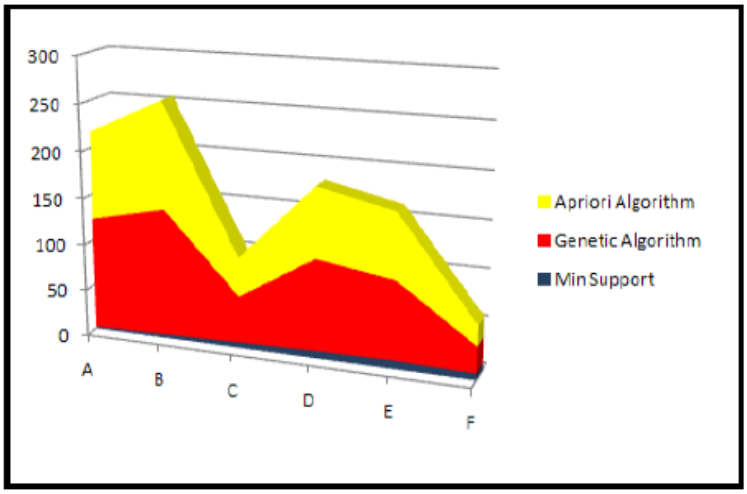

Figure 5.1 Rule Generation Graph of hybrid Apriori/Genetic Algorithm

In this graph, the attributes were taken in the x-axis and number of rules generation was taken in the y-axis. The graph shows the association rules have been optimized using Genetic Algorithm. The performance graph shows that Genetic Algorithm approach is efficient and better when compared to Apriori algorithm.

The following diagram explains comparison between base Apriori and proposed combination of Apriori and Genetic Algorithm in terms of Execution time and number of association rules. This shows the improvement of proposed algorithm.

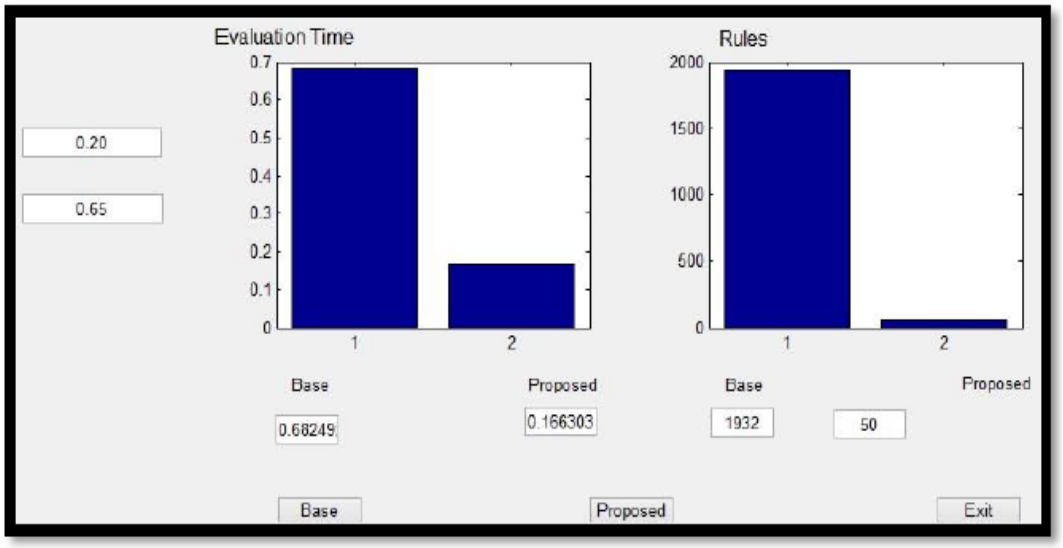

Figure 5.2 Comparison of Base Apriori and Apriori with Genetic Algorithm (Hybrid Apriori)

Table 5.2 and Figure 5.3 show the execution time of the proposed work over base work. The Execution time estimated on different support and confidence.

Table 5.2 Execution time

S.No	Support (%)	Conf (%)	Base Apriori (Time in Seconds)	Proposed Apriori_GA (Time in Seconds)
1	15	60	0.7967	0.2022
2	20	65	0.6824	0.1663
3	25	70	0.7498	0.1877
4	30	75	0.2293	0.1658
5	35	80	0.2289	0.1762
6	40	85	0.2262	0.1799
7	45	90	0.2287	0.1559
8	50	95	0.2280	0.1677

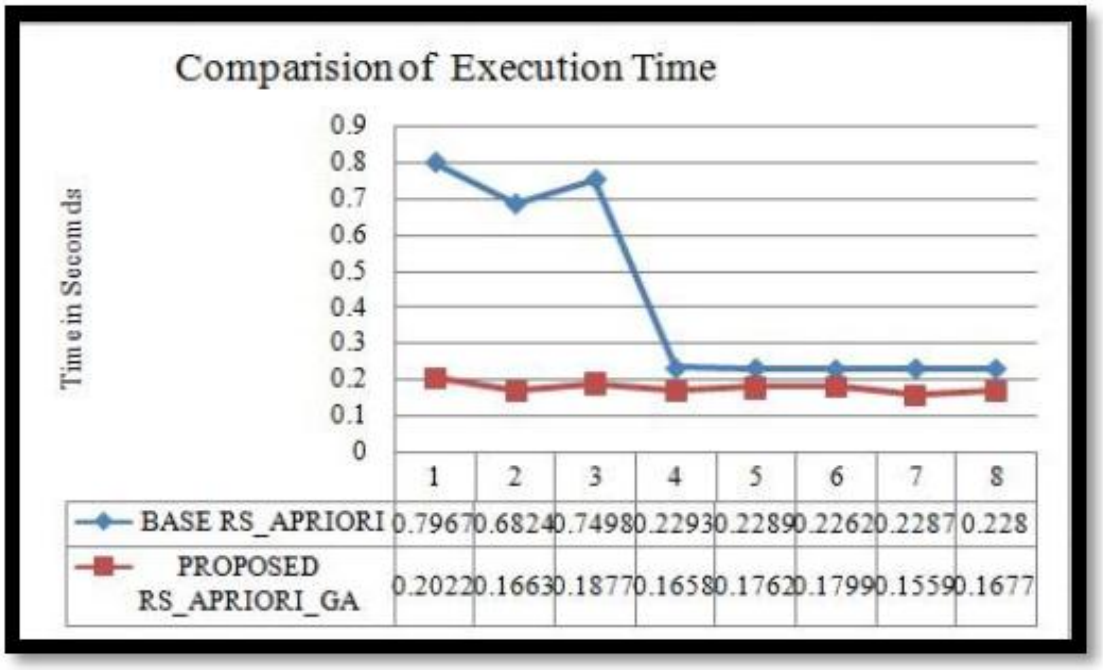

Figure 5.3 Execution Time

VI. CONCLUSION

This paper addresses the necessity of finding frequent patterns in the data mining. Problems related to frequent pattern mining have been analyzed and found a better solution for it. Basically Apriori algorithm is used to find the frequent patterns available in the database. Initially the thesis work starts with the proposed Hybrid Apriori algorithm for mining a frequent pattern which performs better when compared to Apriori algorithm and it have been proved by taking various size of database. The association rule generated by Apriori algorithm is optimized using genetic algorithm and then a parallel algorithm has been proposed which is efficient and proved with some sample data, but the disadvantage is cost. Finally partition algorithm have been proposed, in this approach control will move to the particular partition instead of scanning the entire database. Massive experimentation work was performed for evaluating and comparing Apriori Hybrid and Genetic Algorithms. This paper concludes that the proposed partition algorithm approach consistently performs well than rest of the algorithm and will support many future researches in many ways.

REFERENCES

1. Comscore: the 2010 europe digital year in review. 2011. Retrieved may 5, 2013,http://www.comscore.com/insights/presentations_and_whitepapers/2011 /2010 _europe_digital_year_in_review
2. dzemyda, g. (2001). Visualization of a set of parameters characterized by their

correlation matrix, computational statistics and data analysis, 36(1), p. 15– 30.

3. Dzemyda, g., kurasova, o., medvedev, v. (2007). Dimension reduction and data visualization using neural networks, in: frontiers in artificial intelligence and applications, vol. 160, 25–49.

4. Cheikh Tidiane Dieng, Tao-Yuan Jen, Dominique Laurent, and Nicolas Spyratos.Mining frequent conjunctive queries using functional and inclusion dependencies. VLDB J., 22(2):125 150, 2013.

5. Chengyu., and Xiongying. "Research and Improvement of Apriori Algorithm for Association Rules", IEEE International Workshop on Intelligent Systems and Applications,2010, pp.1-4.

6. Cheng-Yue Chang., Ming-Syan Chen., and Chang-Hung Lee. "Mining General Temporal Association Rules for Items with Different Exhibition Periods", Proceedings of IEEE International Conference on Data Mining, 2002, pp.59-66.

7. Chia-Chu Chiang. "Programming Parallel Apriori Algorithms for Mining Association Rules", Proceedings of IEEE International Conference on System Science and Engineering, 2010, pp.593-598.

13. Chin Chuan Ho., Hua Fu Li., and Fang-Fei Kuo."Incremental Mining of Sequential Patterns over a Stream Sliding Window", Proceedings of IEEE International Conference on Data Mining, 2006, pp.677-681.

14. Chong Wang, Jian Liu, "Mining E-shopper's Purchase Rules Based on K-trees Sequential Pattern", Proceedings of IEEE International Conference on Fuzzy Systems and Knowledge Discovery, 2010, pp.1450-1454.

15. Chuang-Kai Chiou., and Judy Tseng, C.R. "A Scalable Association Rules Mining Algorithm Based On Sorting, Indexing And Triming", Proceedings of IEEE International Conference on Machine Learning and Cybernetics, Vol.4, 2007, pp.2257-2262.

16. Chung-Ching Yu., and Yen-Liang Chen. "Mining Sequential Patterns from Multidimensional Sequence Data", IEEE Transactions on

17. Knowledge and Data Engineering, Vol.17, No.1, 2005, pp.136-140.

18. Bart Goethals, Dominique Laurent, Wim Le Page, and Cheikh Tidiane Dieng. Mining frequent conjunctive queries in relational databases through dependency discovery. Knowl. Inf. Syst., 33(3):655 684, 2012.

19. Cheikh Tidiane Dieng, Tao-Yuan Jen, and Dominique Laurent. An effcientcomputation of frequent queries in a star schema. In Database and Expert Systems Applications, 21th International Conference, DEXA 2010, Bilbao, Spain, August 30 - September 3, 2010, Proceedings, Part II, pages 225 239, 2010.

20. Kun He, Yiwei Sun, David Bindel, John E. Hopcroft, and Yixuan Li.

Detecting overlapping communities from local spectral subspaces. In *2015 IEEE International Conference on Data Mining (ICDM 2015), Atlantic City, NJ, USA*, pages 769-774, 2015.

21. Ayad Ibrahim, Hai Jin, Ali A. Yassin, and Deqing Zou. Towards privacy preserving mining over distributed cloud databases. In *Proceedings of the 2nd International Conference on Cloud and Green Computing (CGC 2012), Xiangtan, Hunan, China*, pages 130{136. IEEE Computer Society, 2012.

22. Leila Ismail and Liren Zhang. Modeling and performance analysis to predict the behavior of a divisible load application in a cloud computing environment. *Algorithms*, 5(2):289{303, 2012.

23. Fan Jiang and Carson Kai-Sang Leung. Stream mining of frequent patterns from delayed batches of uncertain data. In *Proceedings of the 15th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2013), Prague, Czech Republic*, pages 209{221. Springer-Verlag New York, Inc., 2013.

24. Alfredo Cuzzocrea, Ladjel Bellatreche, and Il-Yeol Song. Data warehousing and olap over big data: Current challenges and future research directions. In *Proceedings of the 16th International Workshop on Data Ware-housing and OLAP (DOLAP 2013), San Francisco, California, USA*, pages 67{70. ACM, 2013.

25. Malu Castellanos, Chetan Gupta, Song Wang, and Umeshwar Dayal. Leveraging web streams for contractual situational awareness in operational BI. In *Proceedings of the 2010 International Conference on Extending Database Technology/International Conference on Database Theory (EDBT/ICDT 2010) Workshops, Lausanne, Switzerland*, pages7:18. ACM, 2010.

26. Alfredo Cuzzocrea, Carson Kai-Sang Leung, and Richard Kyle MacKinnon. Mining constrained frequent itemsets from distributed uncertain data. *Future Generation Computer Systems*, 37:117{126, 2014.

27. Alfredo Cuzzocrea, Domenico Saccia, and Jefrerey D. Ullman. Big data: A research agenda. In *Proceedings of the 17th International Database Engineering & Applications Symposium (IDEAS 2013), Barcelona, Spain*,pages 198{203. ACM, 2013.

28. Alfredo Cuzzocrea. CAMS: OLAPing multidimensional data streams efficiently. In *Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2009), Linz, Austria*,pages 48-62. Springer verlag, 2009.

29. Yifan Chen, Xiang Zhao, Xuemin Lin, and Yang Wang. Towards frequent subgraph mining on single large uncertain graphs. In *2015 IEEE International Conference on Data Mining (ICDM 2015), Atlantic City,NJ, USA*, pages 41{50, 2015.

30. Jeferey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107{113, 2008.

31. [dson Dela Cruz, Carson Kai-Sang Leung, and Fan Jiang. Mining `following' patterns from big sparse social networks. In *Proceedings of the International Symposium on Foundations and Applications of Big Data Analytics (FAB 2016), San Francisco, CA, USA*, pages 923{930. ACM,2016.

32. Mohammad El-Hajj and Osmar R. Zaiane. Parallel bifold: Largescale parallel pattern mining with constraints. *Distributed and Parallel Databases*, 20(3):225{243, 2006.

33. Mohammad El-Hajj and Osmar R. Parallel leap: Large-scale maximal pattern mining in a distributed environment. In *Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS 2006), Minneapolis, USA*, pages 135{142. IEEE, 2006.

34. Fan Jiang, Carson Kai-Sang Leung, Dacheng Liu, and Aaron M. Peddle. Discovery of really popular friends from social networks. In *Proceed- ings of the 4th IEEE International Conference on Big Data and Cloud Computing (BDCloud 2014), Sydney, Australia*, pages 342{349, 2014.

35. Dongme Sun, Shaohua Teng, Wei Zhang, Haibin Zhu, "An Algorithm to Improve the Effectiveness of Apriori", In Proc. Int"l Conf. on 6th IEEE Int"l Conf. on Cognitive Informatics (ICCI'07), 2007.

36. Mannila, H. and Toivonen, H.,"Discovering generalized episodes using minimal occurrences", In Proc. of ACM Conference on Knowledge Discovery and Data Mining (SIGKDD), Pages 146–151, 1996.

37. Anandhavalli M, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K. (2009) Optimized Association Rule Mining using Genetic Algorithm, Advances in Information Mining, ISSN:0975-3265, Volume 1, Issue 2, 2009, pp-01-04.

38. Markus Hegland. The Apriori Algorithm – a Tutorial, CMA, Australian National University, WSPC/Lecture Notes Series, 22-27, March 30, 2005.

39. Bart Goethals, Dominique Laurent, Wim Le Page, and Cheikh Tidiane Dieng. Mining frequent conjunctive queries in relational databases through dependency discovery. Knowl. Inf. Syst., 33(3):655 684, 2012.

40. Agrawal, A.; Choudhary, A. 2011. Identifying HotSpots in Lung Cancer Data Using Association Rule Mining. 11th IEEE International Conference on Data Mining Workshops. Pp. 995 – 1002

41. L. Zhou, Z. Zhong, J. Chang, J. Li, J. Huang, and S. Feng. Balanced parallel FP-Growth with MapReduce. In Proc. YC-ICT, pages 243–246,2010.

42. M. Boley and H. Grosskreutz. Approximating the number of frequent sets in dense data. Knowl. Inf. Syst., pages 65–89, 2009.

43. L. Zeng, L. Li, L. Duan, K. Lu, Z. Shi, M. Wang, W. Wu, and P. Luo. Distributed data mining: a survey. Information Technology and Management, pages 403–409, 2012.

44. Apache hadoop. http://hadoop.apache.org/, 2013.

45. Apache mahout. http://mahout.apache.org/, 2013.

46. T. De Bie. An information theoretic framework for data mining. In Proc. ACM SIGKDD, pages 564–572, 2011.

47. [10] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In Proc. OSDI. USENIX Association, 2004.

48. J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox. Twister: A runtime for iterative MapReduce. In Proc. HPDC, pages 810–818. ACM, 2010.

49. K. Finley. 5 ways to tell which programming languages are most popular ReadWrite. http://readwrite.com/2012/06/05/5-ways-to-tell-whichprogramming- lanugages-are-most-popular, 2012.

50. F. Geerts, B. Goethals, and J. V. D. Bussche. Tight upper bounds on the number of candidate patterns. ACM Trans. Database Syst., pages 333–363, 2005.

51. A. Ghoting, P. Kambadur, E. Pednault, and R. Kannan. NIMBLE:a toolkit for the implementation of parallel data mining and machine learning algorithms on mapreduce. In Proc. ACM SIGKDD, pages 334–342. ACM, 2011.

52 B. Goethals. Survey on frequent pattern mining. Univ. of Helsinki, 2003.

53. S. Hammoud. MapReduce Network Enabled Algorithms for Classification Based on Association Rules. Thesis, 2011.

54 J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. SIGMOD Rec., pages 1–12, 2000.

55 H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang. Pfp: Parallel fp-growth for query recommendation. In Proc. RecSys, pages 107–114,2008.

56. J. Li, Y. Liu, W.-k. Liao, and A. Choudhary. Parallel data mining algorithms for association rules and clustering. In Intl. Conf. on Management of Data, 2008.

57 N. Li, L. Zeng, Q. He, and Z. Shi. Parallel implementation of Apriori algorithm based on MapReduce. In Proc. SNPD, pages 236–241, 2012.

58. M.-Y. Lin, P.-Y. Lee, and S.-C. Hsueh. Apriori-based frequent Itemset mining algorithms on MapReduce. In Proc. ICUIMC, pages 26–30. ACM, 2012.

59. M. Malek and H. Kadima. Searching frequent itemsets by clustering data: Towards a parallel approach using mapreduce. In Proc. WISE 2011 and 2012 Workshops, pages 251–258. Springer Berlin Heidelberg,2013