Grenze International Journal of Engineering and Technology, Jan Issue

Frequent Itemset Generation using Clustering

M.Sinthuja[1], Vijaya Kumar B.P[2], Hemanth K[3], Abhay Desalli[4], N Prajwal Pai[5] and Snehil Tewari[6]
[1-6]M. S. Ramaiah Institute of Technology/ISE, Bangalore, India
Email: sinthujamuthu@gmail.com, vijaykbp@yahoo.co.in

***Abstract*—Data mining examines pre-existing large databases to generate new information. For large databases, research is needed to improve mining performance and accuracy, therefore, much of the current focus on association rule mining lies in new mining theories, algorithms, and mining algorithms. improve old methods. Association rules mining is a function of the data mining research domain and arouses many researchers' interest to design a highly efficient algorithm to mine association rules from transaction databases. With the increasing size of the database, we have a large amount of data, but unfortunately we cannot use raw data in our daily decisions/inferences. We desperately need specific information. This information is in most cases in the gathered data, but the extraction of it is a very time and resources consuming operation. In the single machine environment, the problems of Apriori and FP-Growth algorithm in large-scale data association rules mining are high memory consumption, low computing performance, poor scalability and reliability, choosing appropriate minimum support and so on. The purpose of this project is to introduce an improved FP-growth algorithm that will help to resolve the bottleneck problems of the traditional algorithm and has more efficiency than the original one. Therefore, we put forward a new implementation method which depends on a cluster based approach for mining frequent item sets to generate association rules and is verified by using real datasets. Experimental results show that the proposed algorithm was able to reduce the size of constructed trees and the execution time was significantly less than that of the conventional FP-Growth algorithm.**

***Index Terms*— Association Rule Mining, Clustering, Data Mining, FP-growth, Frequent Itemset Mining.**

I. INTRODUCTION

Digitization of all trades and markets has been achieved because the perfect documentation is available to every individual. Digital transformation and the resulting business model innovation has fundamentally changed consumer expectations and behaviour, putting enormous pressure on traditional businesses and disrupting many markets. So AI and machine learning (ML) became mainstream terms outside of the high-tech field, developers were encoding human knowledge into computer systems as rules that get stored in a knowledge base These rules define all aspects of a task, usually in the form of an "If" statement ("if A, then do B, else if X, then do Y"). While the number of rules that need to be written depends on how many actions you want the system to handle (e.g. 20 actions means writing and coding at least 20 manually), systems based on rules usually require less effort, more profit and less risk because the rules will not change or update on their own. While a rule-based system can be thought of as having "fixed" intelligence, a machine learning system on the other hand is adaptive and tries to emulate human intelligence. There is still a basic layer of rules, but instead of a human writing a

Grenze ID: 01.GIJET.9.1.716

fixed set of rules, the machine is capable of learning new rules on its own and discarding those that no longer work. In fact, a machine can learn in many ways, but supervised training - when the machine is fed data to train - is often the first step in a machine learning program. Eventually, machines will be able to interpret, classify, and perform other tasks on their own with unlabeled data or unknown information.
Rules based machine learning algorithms are helpful when simple rules don't apply - when there is no easily definable way to solve a task using simple rules; speed of change - when situations, scenarios, and data are changing faster than the ability to continually write new rules and Natural language processing - tasks that call for an understanding of language, or natural language processing.

II. Literature Review

Association rule mining [1] is an important data analysis method and data mining technology. Although the author proposed the Apriori algorithm, the algorithm uses an iterative process for the data subset and uses the candidate itemsets produced earlier to generate frequent itemsets later, which results in low efficiency of the algorithm and being difficult to be used in the mining of massive data.
The algorithm Pascal[2] which introduces a novel optimization of the well-known algorithm Apriori. Given a certain threshold of minsup, Pascal detects all the common patterns by performing as few counts as possible. To get support for larger models without accessing the database whenever possible, we use the knowledge of the support of some of their submodules,, the so-called key patterns. This method was implemented in the Pascal algorithm that is an optimization of the simple and efficient Apriori algorithm. Pascal's comparative test with the Apriori, Close and MaxMiner algorithms, each representative of the frequent pattern discovery strategy, shows that Pascal improves the efficiency of extracting frequent patterns from correlated and non-correlated data incur additional running time when data is weakly correlated.
This paper [3] discusses the approach of defining a new notion of cluster centroid, that represents the common properties of cluster elements. Thus, similarity within a cluster is measured using cluster representation, which also becomes a natural tool for finding an explanation of the cluster's population.
Their definition of cluster centroid is based on a data representation model which simplifies the one used in document clustering. In fact, we use compact representation of boolean vectors that states only presence and absence of items, while document clustering requires storing the frequencies of items (words). In this paper they show that using our concept of cluster centroid associated with jaccard distance we obtain results having a quality comparable with other approaches used in this task, but we have better performances in terms of execution time. In addition, cluster representatives provide an advanced explanation of cluster characteristics. A particular remark is due to the performance of our approach, which is relevant in case of real-time applications, e.g., clustering of search engine query results or web access sessions for personalization.
In these cases, response time is a fundamental requirement of clustering algorithms, and performances of traditional methods are unacceptable. An alternative to hierarchical methods is to exploit partitioning methods, such as KMeans and its variants, which are linearly scalable to the size of the dataset. However, these methods do not provide an exhaustive form for variable-sized datasets containing categorical data. Constrained frequent pattern mining algorithms on Hadoop platform based on the in-depth analysis of constrained frequent pattern mining algorithms is introduced[4]. First, three pairs of Map and Reduce functions are used to perform the entire process, including mapping transactions to frequent elements that aid in counting, building a constrained regularity tree, and so on. forced (CFPTree), mining frequent patterns, and aggregating frequent patterns results. A parallel mining algorithm of the constrained frequent pattern, called PACFP, is proposed using the MapReduce programming model. We are proposing a load balancing strategy based on frequent item support. The experiments confirm the usability of the algorithm and the load balancing strategy using the celestial spectral dataset. Frequent Pattern Mining (FPM) is emerging as a significant approach to discover fascinating knowledge concealed in the data[5]. With the advancement of big data, growth of the dataset size has increased tremendously. For this reason, traditional algorithms cannot scale to achieve better performance. To achieve better scalability, MapReduce framework is introduced. Execution time for FPM is another major shortcoming in big data mining. To solve this problem, Hadoop with Map Reduce Model is used.
This paper [6] envisages that digitization of data or documents were a revolutionary change which brought out major advancements in almost all sectors. In today's world, digital documents or data have replaced traditional book keeping and data collection methods. The Internet is one of the major areas where a lot of unstructured high dimensional digital documents/data are found. The rate of growth of digital data has made it tough for us to classify or to cluster documents. Document clustering refers to dividing data into similar groups, each of which has similar objects. The clustering is done in such a way that documents present inside a cluster will have higher

degree of similarity and for the documents that are present in different clusters will have lower degree of similarity.

Knowledge discovery in databases (KDD) which is defined as the non-trivial extraction of valid, implicit, potentially useful and ultimately understandable information in large databases [7][12]. In this paper, the problem of the efficiency of the main phase of most data mining applications is addressed: The frequent pattern extraction. This problem is mainly related to the number of operations required to count support models in the database and we propose a new method, called model counting inference, that allows them to perform as few support counts as possible. Using this method, the support of a pattern is determined without accessing the database whenever possible, using the support of some of its sub-patterns called key patterns.

FP-growth algorithm as the representative of non-pruning algorithms is widely used in mining transaction datasets [8]. But it is sensitive to the calculation and the scale of datasets. When building an FP-tree, the search operation as the major time-consuming operation has a higher complexity. And when the horizontal or vertical dimension of the data set is larger, the mining efficiency will be reduced or even failed. Mining association rule is one of the recent data mining research [9] [10][11][13][14]. Association rules are commonly used in marketing, advertising, and inventory control. The association rules detect the current usage of the elements. This problem is fueled by applications known as market basket analysis to find relationships between items purchased by customers, i.e. what types of products tend to be purchased together. This paper presents an efficient partitioning algorithm to extract frequent item set (PAFI) using clustering technique. This algorithm finds frequent itemsets by partitioning the database transactions into clusters. Clusters are formed based on measures of similarity between transactions. Then it finds the frequent itemsets with the transactions in the clusters directly using the improved Apriori algorithm which further reduces the number of scans in the database and hence improves the efficiency.

III. Existing System Models

Association rule mining finds interesting associations and relationships between large sets of data elements. This rule indicates how often a set of items occurs in a transaction. A good example is market-based analysis. Market-based analysis is one of the main techniques used by major relationships to show associations between products. It allows retailers to identify relationships between items that people often buy together. Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrence of other items in the transaction. A supermarket will show that if a customer buys onions and potatoes together, they are likely to also buy a hamburger. This information may be used as the basis for decisions regarding marketing activities, such as promotional pricing or the creation of combined offers or product placement. Also, learning association rules often doesn't take into account the order of elements in a transaction or from one transaction to another.

Following the original definition by Rakesh Agrawal, Tomasz Imieliński and Arun Swami the problem of association rule mining is defined as: Let $I = \{ i_1 , i_2 , \dots , i_n \}$ be a set of n binary attributes called items; Let $D = \{ t_1 , t_2 , \dots , t_m \}$ be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the entries in I. A rule is defined as an implication of the form : $X \Rightarrow Y$, where $X , Y \subseteq I$ and a rule is defined only between a set and a single item, $X \Rightarrow i_j$ for $i_j \in I$.

Few real-world use cases for association rules:

- Retail - Retailers can collect data about purchasing patterns, recording purchase data as item barcodes are scanned by point-of-sale systems. Machine learning models can look for co-occurrence in these data to determine which products are most likely to be purchased together. The retailer can then adjust its marketing and sales strategy to take advantage of this information.
- User experience (UX) design - Developers can collect data on how consumers use a website they create. They can then use the links in the data to optimize the user interface of the site by analyzing where users tend to click and what maximizes their chances that they engage with a call to action.
- Entertainment - Services like Netflix and Spotify can use association rules to fuel their content recommendation engines. Machine learning models analyze past user behavioral data to find frequent patterns, develop association rules, and use those rules to recommend content that users are likely to interact with or organize content in a way that is likely to put the most interesting content for a given user first. Regular pattern mining algorithms have been applied in many fields. Studying their systems model can help to better understand these.

Figure 2.1 shows the process of the system model. Users can get the necessary knowledge to get through the data mining process through the data mining platform. The data mining platform includes a data definition, a data mining designer, and a model filter. Through data definition, we can perform data pre-processing and make

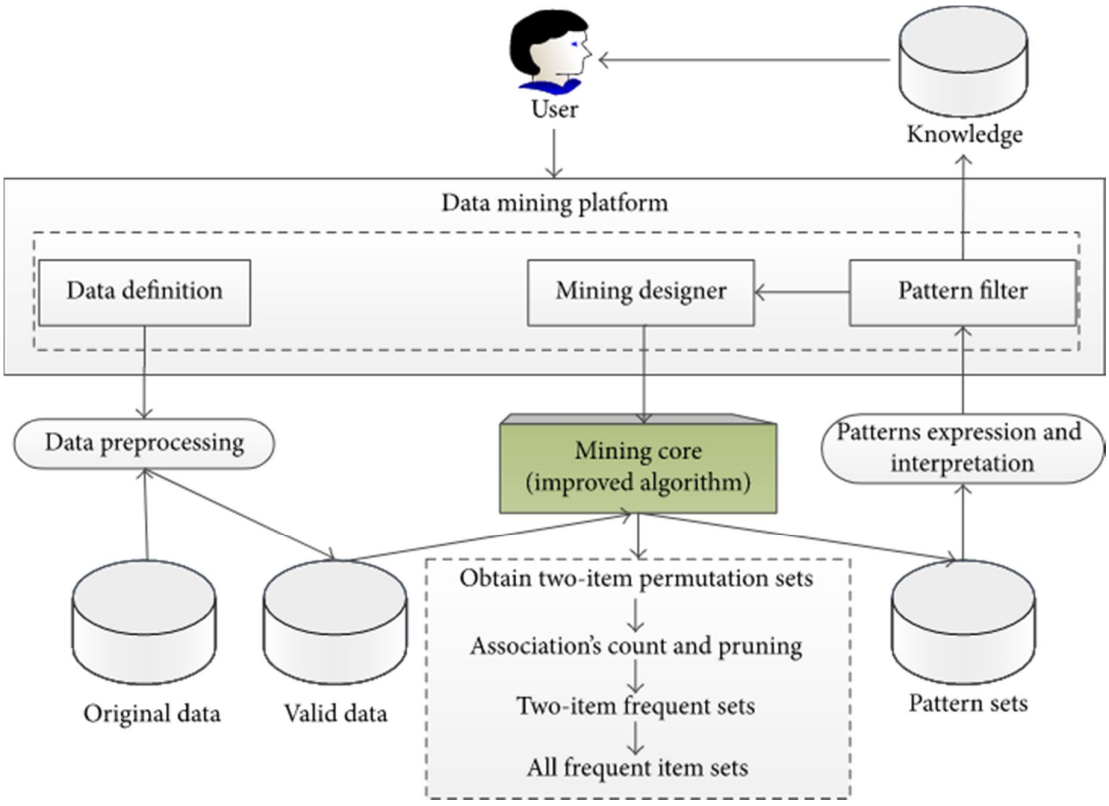

Figure 2.1. System Model

incomplete data usable; thanks to the data mining designer, we can use improved algorithms to dig into the data and get useful patterns (this is a set of frequent items); through the pattern filter we can choose interesting patterns among the obtained samples.

IV. Proposed System

In this research paper, an efficient Partition Algorithm for Mining Frequent Itemsets (PAFI) using clustering is proposed. This algorithm finds the frequent itemsets by partitioning the database transactions into clusters. Clusters are formed based on the similarity measures between the transactions. Then it finds the frequent item sets with the transactions in the clusters directly using an improved Apriori algorithm which further reduces the number of scans in the database and hence improves the efficiency. By further experiments the efficiency can be increased to a significant level.

Figure 4.1 shows the various preprocessing steps applied on the dataset and then the data is pruned based on the clustering method that will be explained. Thus the size of the dataset is reduced. The final step of this algorithm is applying frequent pattern growth on this newly obtained pruned dataset as shown.

Clustering is a data mining technique used to place data items into related groups without prior knowledge of the clustering definition. Clustering can be considered as the most important unsupervised learning problem; so, like any other problem of its kind, the problem is finding structure in unlabeled data sets. Thus, a cluster is a collection of objects that are "similar" to each other and "dissimilar" to objects belonging to other clusters. We used clustering as a processing technique to truncate the data. The various steps in the implementation part are :

- One hot encoding on the dataset:

One hot encoding is a process by which categorical values can be converted into binary values so that it can be applied to machine learning algorithms.

In our case the input dataset which consists of several transactions each containing different itemsets is converted into binary values. This is done for ease of implementation and for the fact that many machine learning algorithms cannot run on categorical values.

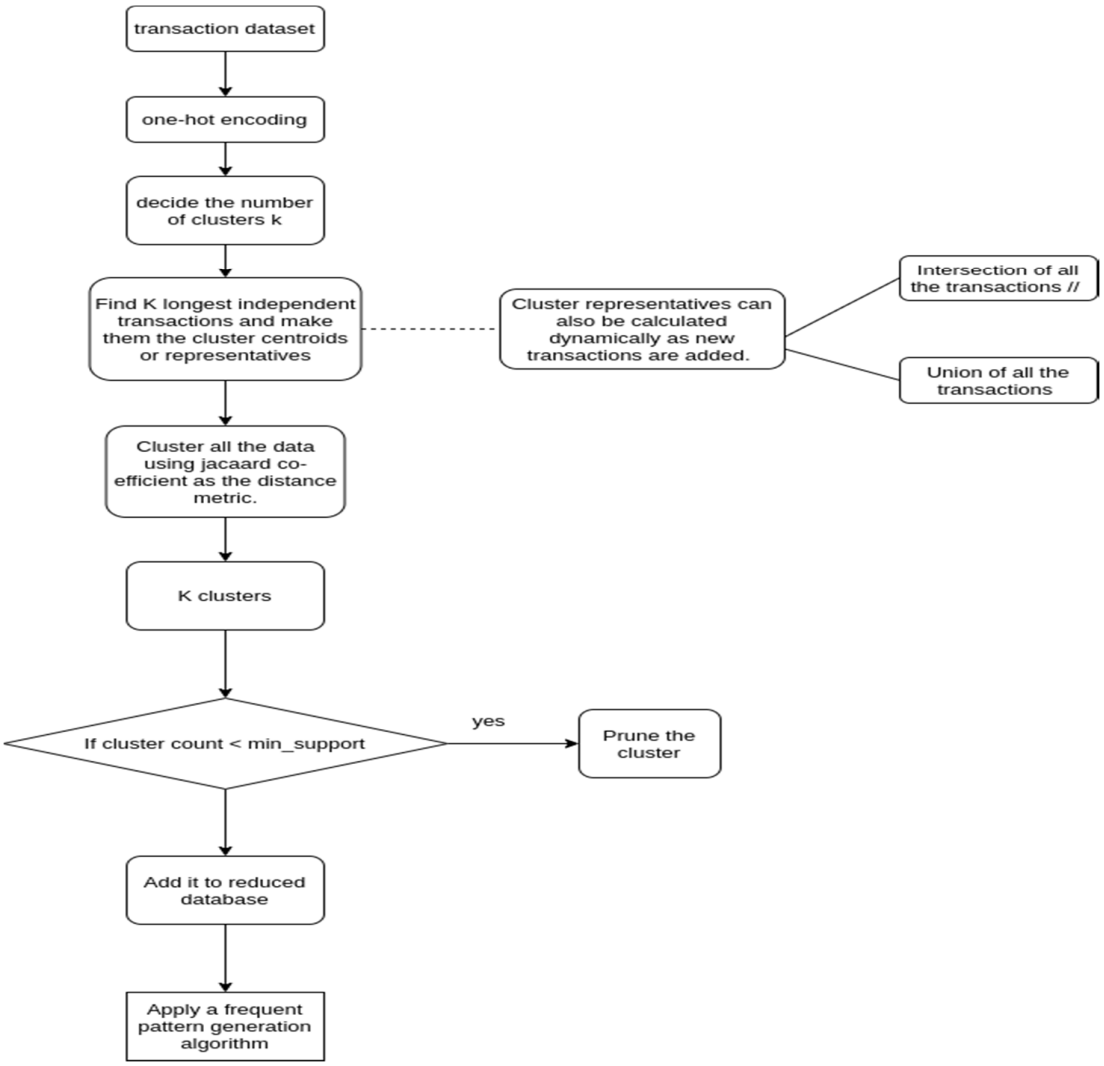

Figure 4.1. Flow Char

- Dissimilarity measure:

The standard approach used to deal with transactional data in clustering algorithms is that of representing such data by means of fixed length Boolean attributes.

Example 2. Let us suppose I = {a1,...,a10}. Itemsets I1 is represented as I1 = {a1, a2}, I2 = {a1, a2, a3, a5, a6, a7} and I3 = {a4} by means of the following vectors:

I1 [1, 1, 0, 0, 0, 0, 0, 0, 0, 0]
I2 [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
I3 [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

In this representation, position i of the boolean vector represents object ai. A value 1 corresponds to the presence of ai to the transaction, while a value 0 corresponds to its absence.

In the above example, I2 is more similar to I1 than to I3, since there is a partial match between I1 and I2 and, by the converse, I2 and I3 are disjoint. Now, the above disparity measures take into account both the presence and absence of an element in a transaction. As a consequence, sparse transactions (i.e., transactions containing a very small subset of I) are very likely to be similar, even if the items they contain are quite different.

Dissimilarity measure used by us is called Jaccard Coefficient, that computes the number of elements in common of two documents, and Cosine similarity that measures the degree of orthogonality of two document vectors. A distance measure can be straightforwardly defined from these measures. For example, we can define d(x, y) = 1

− s(x, y). In the simplified hypothesis that term-vectors do not contain frequencies, but behave simply as boolean vectors (like in the case of web user sessions), a more intuitive but equivalent way of defining the Jaccard distance function can be provided. Given two itemsets I and J, we can represent d(I,J) as the (normalized) difference between the cardinality of their union and the cardinality of their intersection:

$$dJ(I,J) = 1 - |I \cap J| / |I \cup J|$$

This measure captures our idea of similarity between objects, that is directly proportional to the number of common values, and inversely proportional to the number of different values for the same attribute.

Cluster Representative

Given a set S = {x1,...,xm}, i xi ⊆ rep(S).

We have used the intersection of all the transactions over the union to calculate the cluster representative,the intersection does not necessarily correspond to the cluster representative. Again, it can be impractical to approximate the cluster representative with the intersection. With dense population clusters, it is easy to have an empty intersection. On the other hand, the cluster representative cannot be empty.

Perform pruning

We can then perform pruning on the dataset based on the cluster representative obtained. As there are k clusters obtained we can divide the dataset into k clusters. This is done by finding if the support count of the item in a transaction is less than minimum support then it can be pruned,if not it is added to the cluster on the basis of the length of the transaction

Apply FP growth algorithm

Conventional Frequent Pattern growth algorithm is then applied on this pruned dataset and the results obtained are compared with the result obtained by applying the algorithm without the clustering part.

V. TESTING

Testing is the process of analyzing a software item to detect discrepancies between existing and required conditions and to evaluate the characteristics of the software item. The purpose of testing is verification, validation and error detection in order to find problems and the purpose of finding those problems is to get them fixed. Software Testing forms an important activity of software development and maintenance. It is the process used to help identify the correctness, security and quality of developed computer software. Testing is a technical investigative process that provides stakeholders with information about the quality of the service or product being tested. This includes the process of running a program or application for the purpose of finding errors. Software bugs will almost always exist in any moderately sized software module: not because programmers are careless or irresponsible, but because the complexity of the software is often impossible solvable. Tests can be used as measurements. It is widely used as a tool in the verification and validation process. Testers can make claims based on interpretation of the testing results, which either the product works under certain situations, or it does not work. This chapter throws light on the testing done and the results obtained and their comparison to suffice the facts of opting a particular method or approach in ultimately designing this project.

The analysis is done with the help of two datasets. The first data set is T20I6D100K which has a transaction count of 99921, it comprises 893 unique items and 19.90 is the average item count per transaction. The second data set is Market Basket Dataset which has a relatively smaller transaction count of 7500 , it comprises 120 unique items and 3.6 is the average item count per transaction.

For example - { burgers, meatballs, eggs } is a transaction wherein burgers, meatballs and eggs are the items.

Figure 5.1 and 5.2 show the snapshot of both the data sets.

VI. RESULTS AND DISCUSSION

Aim of research is to provide a feasible user-friendly solution to solve the problems occurring due to limitations of the traditional FP Growth algorithm. The findings of the test are shown in the Table 6.1 and Table 6.2 for the T20I6D100K dataset and the Market Basket dataset respectively.

Comparisons were made for both the algorithms and then the graphs were plotted in accordance to the results obtained. Figure 6.1 and Figure 6.2 are Time vs Support Count comparisons for Dataset 1 and 2 respectively. Figure 6.3 and Figure 6.4 are Dataset Size vs Support Count comparisons for Dataset 1 and 2 respectively.

752	826	834					
720	752	834					
685	751	801	811	852			
685	751	801	811	852			
661	695	716	721	727	747	938	
641	721	948	980				
641	721	948	980				
624	654	691	702				
615	630						
601	856	900	938				

Figure 5.1 Datasets

burgers	meatballs	eggs			
chutney					
turkey	avocado				
mineral water	milk	energy bar	whole wheat rice	green tea	
low fat yogurt					
whole wheat pasta	french fries				
soup	light cream	shallot			
frozen vegetables	spaghetti	green tea			
french fries					
eggs	pet food				

Figure 5.2 Datasets

TABLE 6.1 T20I6D100K DATASET

Support Count	Dataset Size	Pruned Dataset Size	Time taken for FP Growth without Clustering (in mins)	Time taken for FP Growth with Clustering (in mins)
1000	99921	96397	15.11	14
2000	99921	70191	13.5	6.3
3000	99921	37534	10.5	1.17
4000	99921	17343	7.0	0.11
5000	99921	5658	3.47	0.0151

TABLE 6.2 MARKET BASKET DATASET

Support Count	Dataset Size	Pruned Dataset Size	Time taken for FP Growth without Clustering (in mins)	Time taken for FP Growth with Clustering (in mins)
150	7500	7483	0.25	0.35
350	7500	7273	0.15	0.15
700	7500	6131	0.0371	0.0218
1050	7500	4979	0.0259	0.01426
1400	7500	4630	0.0123	0.006857

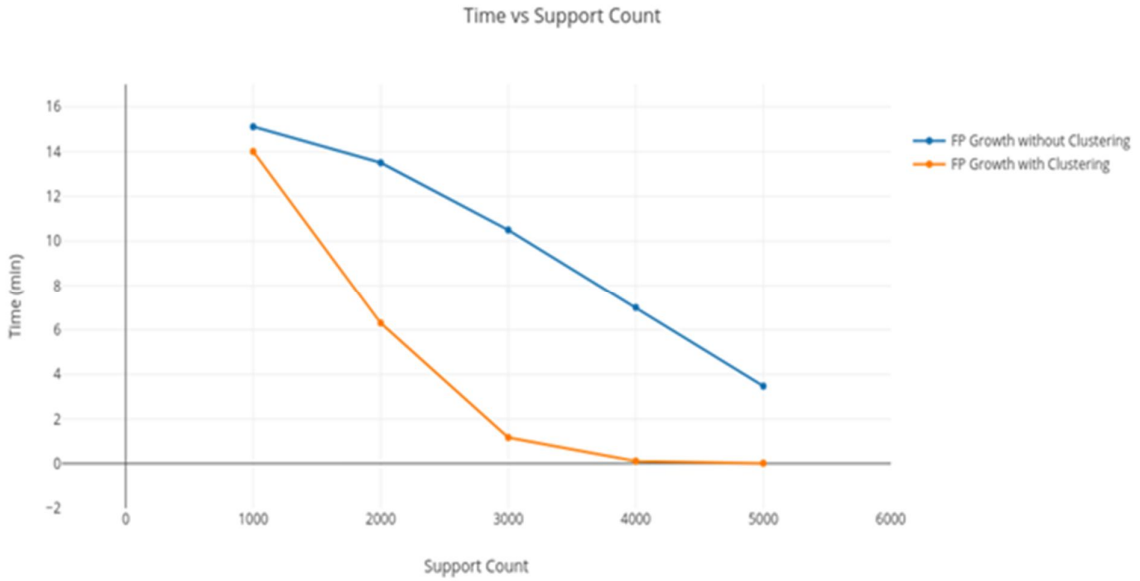

Figure 6.1 Comparison of proposed FP Growth with clustering and the existing FP growth without clustering algorithms

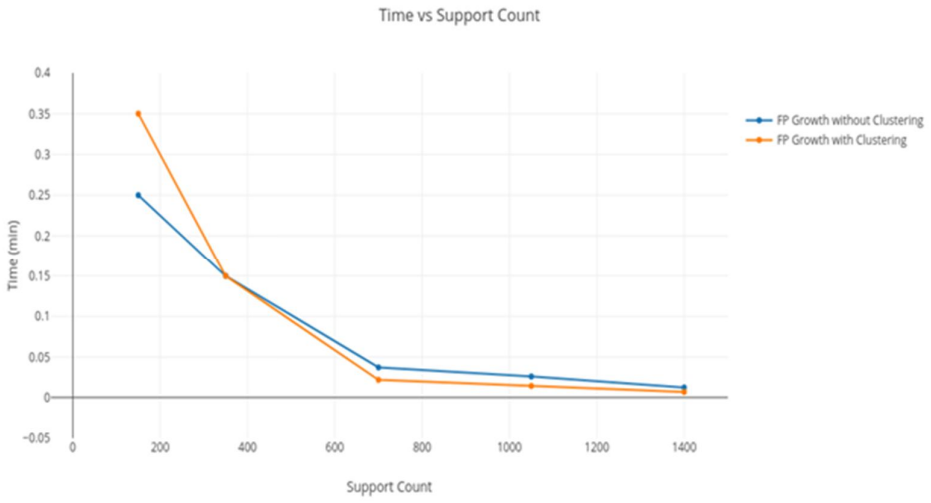

Figure 6.2 Comparison of proposed FP Growth with clustering and the existing FP growth without clustering algorithms

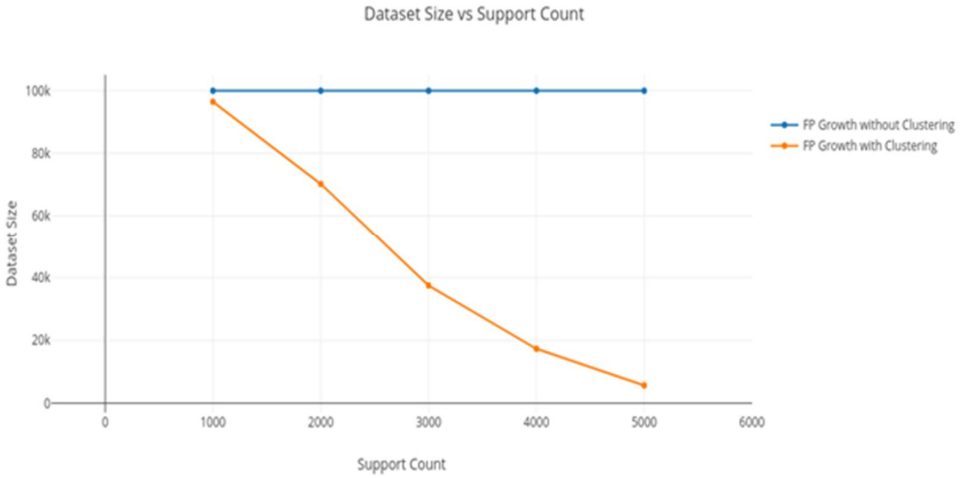

Figure 6.3 Comparison of proposed FP Growth with clustering and the existing FP growth without clustering algorithms

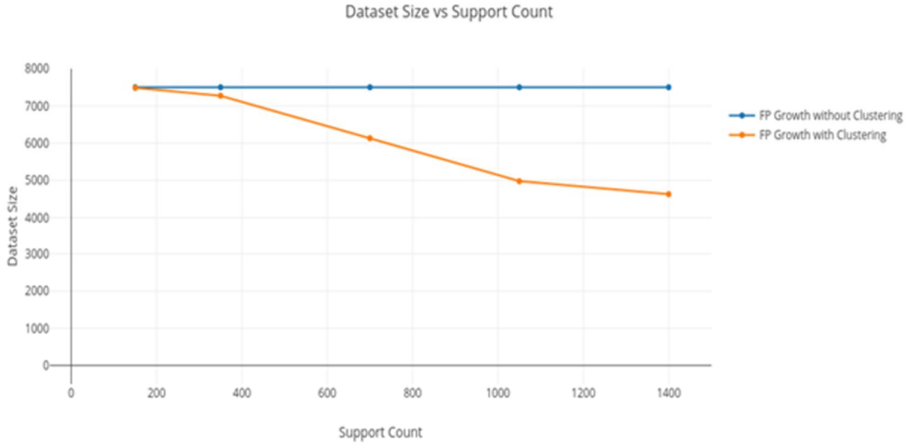

Figure 6.4 Comparison of proposed FP Growth with clustering and the existing FP growth without clustering algorithms

The above mentioned test results and graphs give a clear idea of the overall system. They help to identify the possible outcomes. The tables provide a clear picture on which functionalities are successful and their comparison with the original algorithm. Overall, the tables summarize the benefits rendered by the proposed algorithm and the graphs showcase the various comparisons.

VII. Conclusion and Future Scope

This project proposes a modified version of the FP Growth algorithm, a method wherein the trivial transactions are pruned on a cluster based method. The transactions comprising the database are divided into various clusters and then pruned according to the chosen support count. The algorithm is applied over a dataset with multiple support counts in order to get the optimal result and the same is shown in the results. The FP Growth algorithm is applied on the database to get an optimized and more efficient result. The results are presented in tabular form and graphs are plotted with respect to multiple variables for both the original and the proposed algorithm. In this project we have applied this algorithm on two datasets of varying size and detailed observations from both the datasets are mentioned in this report.

The research can be improved in the future by increasing the time and speed by using parallelization and by using map reduce.

REFERENCES

[1] Chang H Y, Lin J C, Cheng M L, et al. A Novel Incremental Data Mining Algorithm Based on FP-growth for Big Data. International Conference on NETWORKING and Network Applications. IEEE, 2016:375-378

[2] Sinthuja, M. Puviarasan, N. and Aruna, P. (2018), Mining frequent Itemsets Using Top Down Approach Based on Linear Prefix tree, *Springer, Lecture Notes on Data Engineering and Communications Technologies,* Vol.(15), September,23-32.

[3] Fosca Giannotti, Cristian Gozzi, and Giuseppe Manco. Clustering Transactional Data. ISI-CNR, Italy.

[4] Sinthuja, M. Puviarasan, N. and Aruna, P.(2019), An Efficient Maximal Frequent Itemset Mining Algorithm based on Linear Prefix Tree, *Book chapter from CRC Press, Taylor and Francis Group*, December,92-97.

[5] J. Ragaventhiran and M.K. Kavithadevi. Map-optimize-reduce: CAN tree assisted FP-growth algorithm for clusters based FP mining on Hadoop, Future Generation Computer Systems (2019)

[6] N. Ramakrishnan, M. Nair J., D. Jayaprakash, H. Ananthakrishnan and S. Rani S., Hypergraph based clustering for document similarity using FP growth algorithm, 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 332-336, doi: 10.1109/ICCS45141.2019.9065630.

[7] M. Sinthuja, N. Puviarasan, and P. Aruna." Proposed Improved FP-Growth Algorithm with Multiple Minimum Support Threshold Value (MISIFP-Growth) For Mining Frequent Itemset", *International Journal of Research in Advent Technology, (IJRAT),* Vol. 6, No.5,pp. 471-476, May 2018.

[8] Bastide, Yves & Pasquier, Nicolas & Stumme, Gerd. (2000). Levelwise Search of Frequent Patterns with Counting Inference.

[9] Sinthuja, M. Puviarasan, N. and Aruna, P., Frequent Itemset Mining using LP-Growth algorithm based on Multiple Minimum Support Threshold Value (MIS-LP-Growth), *Journal of Computational and Theoretical Nanoscience,* Volume 16, No(4), pp. 1365- 1372(8) April 2019.

[10] L. Deng and Y. Lou, Improvement and Research of FP-Growth Algorithm Based on Distributed Spark, *2015 International Conference on Cloud Computing and Big Data (CCBD)*, 2015, pp. 105-108, doi: 10.1109/CCBD.2015.15.

[11] M.Sinthuja, Dr.N.Puviarasan, Dr.P.Aruna."Geo Map Visualization for Frequent Purchaser in Online Shopping Database Using an Algorithm LP-Growth for Mining Closed Frequent Itemsets" in Elsevier, procedia computer science, Volume 132, Pages 1512-1522, June 2018.

[12] D Kerana Hanirex and Dr. M.A. Dorai Rangaswamy, Efficient Algorithm for Mining Frequent Itemsets using Clustering Techniques, International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397, Vol. 3, 2011

[13] M. Sinthuja, Dr.N.Puviarasan, Dr.P.Aruna. "Improved FP-Growth Algorithm Based on Top Down Approach in Mining Frequent Itemsets" *International Journal for Research in Engineering Application & Management (IJREAM) ISSN : 2454-9150* Vol-04, Issue- 02,pp.618-623, May 2018.

[14] M.Sinthuja, Dr.N.Puviarasan, Dr.P.Aruna. "Research of Improved FP-Growth (IFP) Algorithm in Association Rules Mining" in *International Journal of Engineering Science Invention, (IJESI)* ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726, pp.24-31,2018.