

FREQUENT PATTERN MINING ON STREAM DATA

Project Report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Adarsh Pal (141350)

Ishita Dewan (141368)

Under the supervision of

Dr. Pardeep Kumar

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this report entitled “**Constrained based Frequent Pattern Mining on Stream Data**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2017 to May 2018 under the supervision of **Dr. Pardeep Kumar**, Associate Professor.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Adarsh Pal, 141350

Ishita Dewan, 141368

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pardeep Kumar

Associate Professor

Department of Computer Science

Dated:

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our guide Dr. Pardeep Kumar for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by his time to time shall carry us a long way in the journey of life on which we are about to embark.

The in-time facilities provided by the Computer Science department throughout the project development are also equally acknowledgeable.

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

Date:
(141350)

Adarsh Pal

(141368)

Ishita Dewan

TABLE OF CONTENTS

| Serial Number | Topics | Page Numbers |
|---------------|--|--------------|
| | | |
| 1 | Chapter-1. Introduction | 12 |
| 2 | 1.1 Introduction | 12 |
| 3 | 1.2 Problem Statement | 18 |
| 4 | 1.3 Objectives | 19 |
| 5 | 1.4 Organisation | 19 |
| | | |
| 6 | 2. Literature Survey | 20 |
| 7 | 2.1. Constraint Frequent Pattern Mining: A Pattern Growth Approach | 20 |
| 8 | 2.2. Improved Apriori Algorithm with Pruning Unnecessary Candidate Set for Reducing Execution Time | 21 |
| 9 | 2.3. Improving the efficiency of Apriori Algorithm in Data Mining | 22 |
| 10 | 2.4. Research and Improvement of Apriori Algorithm | 22 |
| 11 | 2.5. From Data Mining to Knowledge Discovery in Databases | 23 |
| 12 | 2.6. Study on the Application of Apriori Algorithm in Data Mining | 23 |
| 13 | 2.7. Frequent pattern mining: current status and future directions | 24 |
| 14 | 2.8. An improved Apriori Algorithm for Association Rules | 24 |
| 15 | 2.9 Application of Apriori Algorithm in Predicting Flood Areas | 25 |

| | | |
|----|---|----|
| 16 | 2.10 Implementation of Association Rules with Apriori Algorithm for increasing the Quality of Promotion | 25 |
| 17 | 2.11 Is the number of constraints limited? | 26 |
| 18 | 2.12 Web Log Mining using Matrix Apriori Algorithm | 26 |
| 19 | 2.13 Hp-Apriori Algorithm for frequent dataset mining | 28 |
| 20 | 2.14 A More Advanced Apriori Algorithm based on Recurrent Matrix | 29 |
| 21 | 2.15 Investigation and Enhancement of Apriori Algorithm for Association Rules | 30 |
| 22 | 2.16 A Frequent-Pattern Tree Approach to Candidate set generation | 30 |
| 23 | 2.17 Implementing the Apriori algorithm in parallel by Map Reduce | 31 |
| 24 | 2.18 Using Power set on Hadoop improving the Apriori algorithm | 32 |
| 25 | 2.19 Reviewing the algorithm based on Apriori | 32 |
| 26 | 2.20 Sentiment Analysis of Music using Association Rule Mining | 33 |
| 27 | 2.21 Mobile e-commerce approval system using the improved Apriori algorithm | 34 |
| 28 | 2.22 Use of Auto-Adjust Apriori algorithm | 35 |
| | | |
| 29 | Chapter-3 System Development | 36 |
| 30 | 3.1 Design | 36 |
| 31 | 3.2 Model Development | 36 |
| 32 | 3.2.1 MODULE 1: Analytical Model Development | 36 |
| 33 | 3.2.2 MODULE 2: Computational Model Development | 36 |
| 34 | 3.2.3 MODULE 3: Experimental Model Development | 36 |

| | | |
|----|---|----|
| 35 | 3.2.4 MODULE 4: Mathematics Model Development | 36 |
| 36 | 3.2.5 MODULE 5: Statistical Model Development | 36 |
| | | |
| 37 | Chapter-4 Performance Analysis | 38 |
| 38 | 4.1 Analysis of system developed | 38 |
| 39 | 4.2 Implementation | 38 |
| 40 | 4.3 Output | 41 |
| | | |
| 41 | Chapter-5 Conclusion | 50 |
| 42 | 5.1 Conclusions | 50 |
| 43 | 5.2 Future Scope | 51 |
| 44 | 5.3 Applications | 52 |
| 45 | 5.3.1 Ordering and similitude hunt of complex organized information | 52 |
| 46 | 5.3.2 Spatiotemporal and media information mining | 52 |
| 47 | 5.3.3 Mining data streams | 53 |
| 48 | 5.3.4 Web mining | 53 |
| 49 | REFERENCES | 54 |

LIST OF FIGURES

1. Candidate item set generation
2. Algorithm for Apriori
3. Transactions for market-basket analysis
4. Read Write Operation
5. Phases of Web Mining
6. Association Rules
7. Formulae for association rules

LIST OF GRAPHS

1. Graph between transactions and items.
2. Graph between items and itemFrequency.
3. Graph between sequence and support.

LIST OF TABLES

Table 1-Table 8: Numerical Illustration of Algorithm.

Table 9: Improved algorithm vs. Traditional Algorithm.

Table 10: Original vs. Proposed Algorithm.

ABSTRACT

The issue of frequent pattern mining has been generally contemplated in the writing in view of its various applications to an assortment of information mining issues, for example, clustering and characterization. Furthermore, frequent pattern mining likewise has various applications in diverse spaces, for example, spatiotemporal information, programming bug discovery, and biological information. The algorithmic parts of frequent pattern mining have been investigated generally. Frequent pattern mining is one of four noteworthy issues in the information mining area. This part gives an overview of the significant subjects in frequent pattern mining. The earliest work here was focussed on deciding the effective calculations for frequent pattern mining, and variations, for example, long pattern mining, interesting frequent mining, constraint based example mining, and compression. Lately scalability has turned into an issue in light of the gigantic measures of information that keep on being made in different applications. Furthermore, due to propels in information gathering innovation, propelled information composes, for example, worldly information, spatiotemporal information, graph information, and uncertain information have turned out to be more typical. Such information writes have various applications to other information mining issues, for example, bunching and characterization. What's more, such information writes are utilized frequently in different worldly applications, for example, the Web log investigation. Requirement based example mining frameworks are frameworks that with negligible exertion can be modified to discover diverse kinds of examples fulfilling limitations. They accomplish this generosity by giving (1) abnormal state dialects in which software engineers can without much of a stretch determine imperatives; (2) nonexclusive look calculations that discover designs for any errand communicated in the determination dialect. The improvement of nonspecific frameworks requires a comprehension of various classes of limitations. The greater part of the information mining calculations were intended to mine the incessant example from exact information. Notwithstanding, vulnerability exists in numerous genuine circumstances, for example, sensor system and security saving applications. To separate significant data from unverifiable information various successive example mining calculations have been proposed. While managing dubious information U-Apriori, UF-development, UFP-development, UH-mine, PUF-development, TPC-development calculation are cases of existing successive example mining calculations,

which use distinctive ways to deal with mine continuous example. One imperative perception is that calculations carry on totally extraordinary in the indeterminate database when contrasted with the exact database due of the incorporation of likelihood esteem. Frequent configuration mining has been connected with point in data mining research .For more than 10 years. Unlimited composition has been focused on this investigation what's increasingly, immense progress has been made, going from compelling and adaptable counts for visit itemset mining in return databases to different investigate backcountry, for instance, progressive case mining, sorted out case mining, relationship mining, subsidiary course of action, and general case based gathering, and their far reaching applications. Mining incessant examples from exchange database, time arrangement and information stream is an imperative undertaking now. A decade ago, there are for the most part two sorts of calculations on visit design mining. One is Apriori in light of creating and testing, the other is FP-development in view of partitioning and vanquishing, which has been generally utilized as a part of static information mining. Be that as it may, with the new necessities of information mining, mining incessant example isn't limited in the static datasets any more. For information stream, the regular example mining calculations must have solid capacity of refreshing and changing in accordance with additionally enhance its effectiveness.

CHAPTER-1. INTRODUCTION

1.1 Introduction:

Frequent patterns are thing sets, game plans, or structures that show up in an informational index with recurrence in excess of a client indicated limit. For instance, an arrangement of things, for example, eggs and drain, which show up as often as possible together in an exchange informational index, is a successive itemset. A subsequence, for example, purchasing initial a work area, at that point a DSLR camera, and afterward a pen drive is a consecutive example, on the off chance that it happens every now and again in a spending history record. An organisation can allude various basic structures, for example, sub diagrams, sub trees, or sub cross sections, that might be joined to item groups or sub groupings. A substructure can allude to various auxiliary structures, for example, sub diagrams, sub trees, or sub lattices, which might be joined with thing sets or sub successions. In the event that a substructure happens often in a chart database, it is known as a (visit) auxiliary example. Finding successive examples assumes a fundamental part in mining affiliations, connections, and numerous other intriguing connections among information. Additionally, it helps in information ordering, arrangement, bunching, and other information mining undertakings also. Visit design mining is a critical method and an engaged topic in information mining research. It has been a drawn in subject in data digging research for more than 10 years. Bounteous written work has been given to this investigation and monstrous progress has been made, reaching out from profitable and versatile figuring for visit itemset mining in return databases to different research unsettled areas, for instance, back to back illustration mining, sorted out case mining, association mining, agreeable gathering, and progressive case based clustering, and also their wide applications. Visit configuration mining research has through and through extended the degree of data examination and will have significant impact on data mining frameworks and applications as time goes on.

Aim of frequent pattern mining:

Discovering connections among the things in a database is the essential issue of incessant example mining. The issue can be expressed as takes after. Given a database D with exchanges T_1 to T_n .

The parameter s can be communicated either as an absolute number, or as a small amount of the total number of exchanges in the database. Every transaction T_i may be referred to as sparse paired vector, or as an arrangement of distinct qualities speaking to the identifiers of the twofold credits that are initialised to the estimation of 1. The matter got initially projected with regards to market basket analysis keeping in mind the complete aim to discover frequent gatherings of things that are purchased together. Accordingly, in this situation, each credit relates to a thing in a superstore, and the twofold esteem speaks to regardless of whether it is available in the transaction. Since the issue was initially proposed, it has been connected to various different applications with regards to information mining, Network log mining, consecutive pattern removal, and software design bug examination. The first model of successive example mining, the issue of discovering affiliation instructions has additionally been projected that is firmly identified with that of continuous examples, when all is said in done affiliation guidelines can be viewed as a "moment organize" yield, which are gotten from visit designs. Think about the arrangements of items U and V . The control $U \Rightarrow V$ is viewed as an association rule at least support s and least confidence c , when the accompanying two conditions remain constant:

1. The set $U \cup V$ is frequent pattern.
2. The ratio of the support of $V \cup U$ to that of U is minimum c .

The base confidence c is dependably a portion under 1 in light of the fact that the rule of the set $U \cup V$ is constantly not as much as that of U . Since the initial step of finding frequent examples is normally computationally all the more difficult one, a large portion of the exploration around there is focussed on the previous. Regardless, some computational and showing issues also develop in the midst of the second step, especially when the continuous example mining issue is used as a piece of the setting of other data mining issues, for instance, arrange. A few cases of critical applications are as per the following;

- **Client Transaction Analysis:** For this situation, the transaction represents to a set of things that co-happen in client purchasing conduct. For this situation, it is attractive to decide frequent pattern of purchasing conduct, since they can be utilized for settling on choice about rack supplying or proposals.
- **Data Mining Problems:** Frequent design mining can be used to empower other real information mining issues, for example, characterization, and bunching and anomaly examination. This is on account of the utilization of incessant examples is very central in the logical procedure for large group of information removal issues.

- **Web Mining:** For such a situation, the Network logs might be handled so as to decide essential examples in the perusing conduct. This data can be utilized for Web webpage outline proposals, or significantly exception investigation.
- **Software Bug Analysis:** For this situation, the Web logs might be handled so as to decide essential examples in the perusing conduct. This data can be utilized for Web webpage outline proposals, or significantly exception investigation.
- **Compound and Biological Analysis:** Chemical and biological information are frequently represented on charts and arrangements. Various strategies have been proposed in the writing for utilizing the regular examples in such diagrams for a wide assortment of uses in various situations.

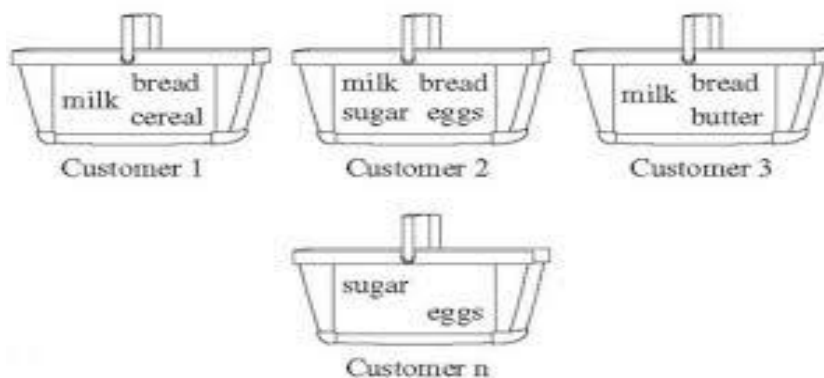


Figure 1

Frequent Pattern Mining Algorithms

- ✓ The greater part of the calculations for frequent design mining have been outlined with the conventional support confidence system, or for specific structures that produce all the more intriguing sorts of examples. These specific system may utilize distinctive sorts of intriguing quality measures, show negative standards, or utilize imperative based structures to decide more applicable examples.
- ✓ **Constrained Frequent Pattern Mining:**

Off-the-rack frequent configuration mining computations find incalculable examples which are not profitable when it is needed to choose plans in view of more refined criteria. Frequent configuration mining procedures are as often as possible particularly significant concerning obliged applications, in which rules satisfying particular criteria are found. For instance,

one may want particular things to be available in the rule. The first arrangement is to principally mine all the thing sets, and after that empower internet mining from this arrangement of base examples. Be that as it may, pushing limitations straightforwardly into the mining procedure has a few focal points. This is on the grounds that when limitations get pushed straightforwardly into the mining technique, the mining can be done at much lesser sustainable levels than that may be performed by utilizing a two-stage methodology. That may be the situation whenever substantial count of middle of the road competitors may get trimmed by the requirement grounded example mining calculation. An assortment of subjective limitations would likewise become available for examples. The real issue in these techniques becomes that the requirements would bring about some infringement of the descending conclusion character. Since many successive example mining calculations depend significantly on this property, its infringement is a difficult issue. By and by, numerous limitations have particular properties in light of which specific calculations can be created. Obligated techniques have additionally been created for the consecutive example mining issue. In genuine presentations, the yield of the vanilla regular example mining issue might be very broad, it that is done by putting limitations into the pattern mining process, which valuable application-particular examples could be found. Constraint frequent design mining techniques may be firmly identified by the issue for example grounded arrangement, on the grounds that the last issue expects us to find discriminative examples from the hidden information. One of the primary explanations behind the abnormal state of enthusiasm for visit design mining calculations is because of the computing test of the errand. Notwithstanding of a direct estimated dataset, which is exponential to the span of the exchanges in the dataset. This normally makes tests for itemset to generate, when the support stages are little. Actually, in most down to earth situations, the help levels at which one can mine the relating thing sets are restricted (limited underneath) by the memory and computational requirements. Accordingly, it is necessary to have the capacity to play out the search in a space-and time-effective way. Amid the initial couple of years of research here, the essential focal point of work was to discover FPM calculations with better computational effectiveness. A few courses of calculations are created for visit design mining, a considerable lot of what are firmly identified with each other. The candidate generation procedure of the most previous calculations utilized joins:

Join-Based Algorithms:

Join-based calculations create $(k + 1)$ - competitors from visit k -designs with the utilization of joins. These competitors are then approved against the exchange database. The Apriori strategy utilizes joins to make applicants from visit designs, and is one of the most punctual calculations for visit design mining

Apriori Method:

Apriori is a calculation for frequent item set mining and association rules learning over important databases. It proceeds by perceiving the continuous individual things in the database and extending them to greater and greater thing sets as long as those item sets show up sufficiently much of the frequency in the database. The continuous item sets managed by Apriori can be used to choose association rules which highlight general examples in the database: this has applications in spaces, for instance, grandstand case examination. The Apriori count was proposed by Agrawal and Srikant in 1994. Apriori is planned to take a shot at databases containing trades (for example, gatherings of things bought by customers, or unpretentious components of a site frequentation). Diverse estimations are expected for finding alliance regulates in data having no transactions, or having no timestamps (DNA sequencing). Each trade is seen as a course of action of thing. Specified a limit, the Apriori calculation distinguishes the thing sets which are subsets of at any rate transactions in the database. Apriori employs a "base up" approach, where frequent subgroups are expanded once at a time (a stage known as competitor age), and gatherings of hopefuls are exasperated alongside the information. The scheming ends when no further productive expansions are found. Apriori exploits breadth first search and a Hash tree structure to tally candidate item sets proficiently. It creates applicant item sets of length from thing sets of length. At that point it prunes the opponents which have an occasional design. As showed by the downward conclusion lemma, the applicant set has all regular - length thing sets. From that point forward, it checks the exchange database to decide frequent thing sets among the competitors. Apriori, while generally critical, experiences various wasteful aspects or exchange offs, which have generated different calculations. Applicant age creates substantial quantities of subsets (the calculation endeavours to stack up the competitor set with however many as could be expected under the circumstances previously each output). Base up subset investigation (a broadness first traversal of the subset grid) finds any maximal subset S simply after the greater part of its appropriate subsets.

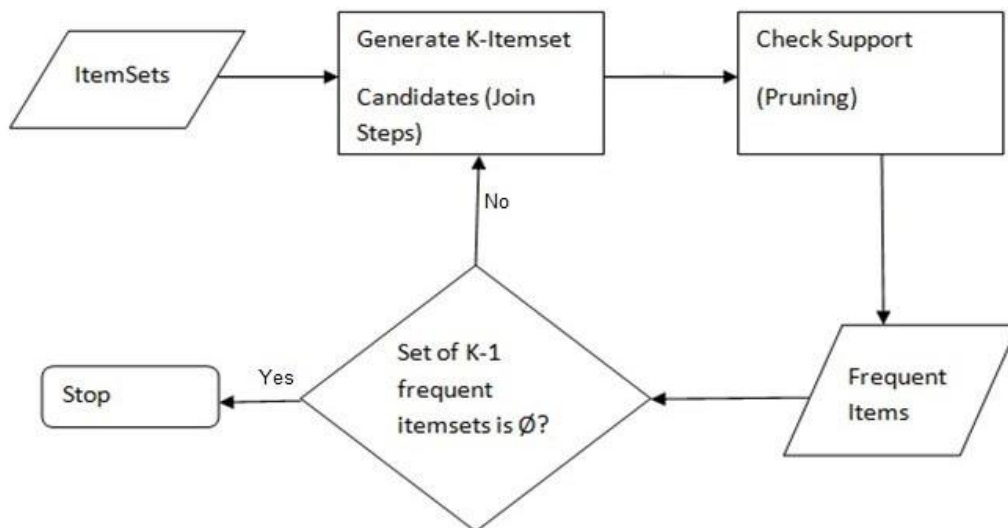


Figure 2

Association rules:

Association rule learning is a rule based machine learning technique for finding fascinating relations between factors in significant databases. It is planned to distinguish solid tenets found in databases using a few measures of intriguing quality. For instance the administer found in the business data of a supermarket would exhibit that if a customer buys onions and potatoes together, they are most likely going to in like manner buy cheeseburger meat. Such information can be used as the purpose behind decisions about exhibiting works out, for instance, e.g., constrained time assessing or thing game plans. Despite the above case from feature bushel examination connection rules are used today in various application locales including Web use mining, intrusion distinguishing proof, endless age, and bioinformatics. Strikingly with progression mining, alliance lead adjusting usually does not consider the demand of things either inside a trade or across finished transactions.

| ID | Items |
|-----|------------------------------|
| 1 | {Bread, Milk} |
| 2 | {Bread, Diapers, Beer, Eggs} |
| 3 | {Milk, Diapers, Beer, Cola} |
| 4 | {Bread, Milk, Diapers, Beer} |
| 5 | {Bread, Milk, Diapers, Cola} |
| ... | ... |

} market basket transactions

{Diapers, Beer} Example of a frequent itemset

{Diapers} → {Beer} Example of an association rule

Figure 3

Parameters for Association Rule Mining:

- ❖ **Support:** Support means that how much of the time the itemset shows up in the dataset. The help of X concerning Y is characterized as the extent of transactions t in the dataset which contains the itemset X.
- ❖ **Confidence:** Confidence means that how frequently the control has been observed to be valid. The certainty estimation of a govern, $X \rightarrow Y$, concerning an arrangement of exchanges T, is the extent of the exchanges that contains X which likewise contains Y.
- ❖ **Lift:** The lift of a rule is the ratio of the support of the items on the LHS of the rule co-occurring with items on the RHS divided by probability that the LHS and RHS co-occur if the two are independent.
- ❖ **Conviction:** can be interpreted as the ratio of the expected frequency that X occurs without Y is to if X and Y were independent divided by the observed frequency of incorrect predictions.

1.2 Problem Statement:

Constraint-based mining begins with the observation that many pattern mining problems can be seen as instances of the following generic problem statement:

Given

- a data language L
- a database $D \subseteq 2^L$ with transactions
- a pattern language L_P
- a constraint $\phi: L_P \times 2^L \rightarrow \{0, 1\}$

Find all patterns $\pi \in L_P$ for which $\phi(\pi, D) = 1$

The pattern language typically describes the composition of the patterns we wish to find. Constraints typically describe the statistical, syntactical, or parameters on which we wish to describe the data.

1.3 Objectives:

1. To incorporate global optimization strategy for enhancing the optimal solution
2. To study the performance analysis of designed "Frequent Pattern Mining Algorithms".
3. To hybridize the existing algorithm to achieve effective and efficient solution for pattern mining.

1.4 Organization:

The main body of the report is preceded by detailed contents including lists of figures, tables, and annexes followed by units used in the report. This is followed by executive summary giving briefly the scope and objectives of the study, importance of the topic, methodology, limitations, major observations / findings, and recommendations & action plan.

Chapter 1 explains the importance of the topic and scope of data collection and analysis. Chapter 2 discusses the methodology and literature survey of the project. Chapter 3 discusses the design of the system in the project. Chapter 4 discusses the numerous variations of algorithms used in the course of project and

Chapter 5 specifies the Test Plan of the project. This includes the involved Data Sets, the Metrics, the Test Setup and several other required aspects of the project. Chapter 6 explains about the results and the performance analysis of the project.

Chapter 8 gives the conclusions, recommendations & action plan.

All chapters are preceded by a brief synopsis of the chapter, and key words. References which have been used for certain inputs are listed after the key words. Wherever these references have been quoted / data or technical specifications taken in the text, these have been cross-referred by their serial number (appearing as superscripts in the report) in the list of References.

CHAPTER 2: LITERATURE SURVEY

2.1 Improving the Apriori Algorithm by Pruning Unnecessary Candidate Set for the purpose of Reducing Execution Time

Khare and Srivastava have come up with an algorithm for improving the efficiency of normal Apriori algorithm by improving the pruning algorithm. The new algorithm that they have introduced decreases the number of transactions whenever the database is scanned. On one side the Apriori algorithm uses the concept of candidate set generation and in every iteration the number of transactions is more as compared to the improved algorithm wherein the number of transactions decreases with every iteration. The minimum support value is premeditated at each pass which removes the formation of unnecessary sets. In the traditional Apriori algorithm database has to be scanned after every transaction whereas in the improved algorithm the transactions are filtered via database just once and then the generated filtered set is then used to further compute the results. Moreover the generation of redundant sets is also avoided. The overall time and efficiency is far better than the original Apriori algorithm. The paper also introduces a table wherein the number of transactions generated after every iteration is compared and proved the efficiency of this new algorithm. The algorithm being simple carries out the pruning process more effectively.

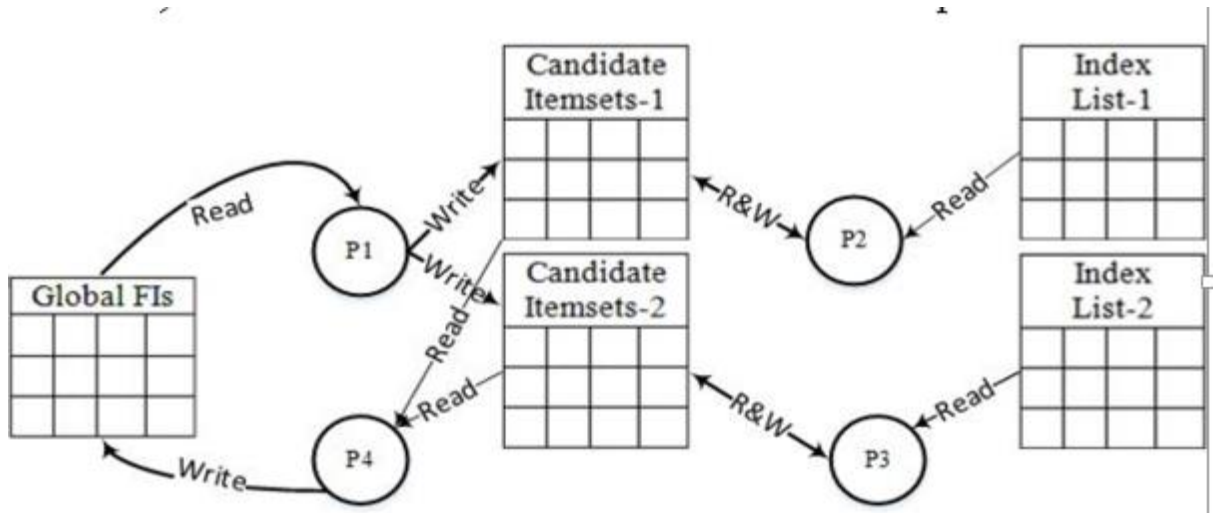


Figure 4

2.2 Improvement of the efficiency of Apriori Algorithm in Data Mining

Mangla et al., have introduced a new algorithm to overcome the shortcomings of the traditional Apriori algorithm. This algorithm uses matrix method to overcome the performance bottleneck of the traditional algorithm. This algorithm works efficiently by optimizing the way one can mine information from huge datasets. The method eliminates the less frequent element every time from the database. No repeated scans of database are required. The method proposed in this paper involves the mapping of the items and transaction from the database into a matrix Z and using this matrix for further computations. The rows are demarcated as transactions and columns as items. In every iteration the particular row or column is removed that does not satisfy the support threshold. The paper also compares both the algorithm and proves the credibility of matrix method over the slow traditional Apriori algorithm. The I/O are also reduced in every step as the transactions are cut down making the process a processor friendly and efficient at the same time. Although this algorithm is optimized and efficient but has an overhead to accomplish the new database after every matrix generation. Although this algorithm beats the traditional Apriori but it introduces an overhead of maintain the new database after every iteration making the process a little complicated.

2.3 Research and Improvement of Apriori

Du et al., proved via experiments proves the superiority of DC_Apriori over Apriori algorithm and the MC_Apriori algorithm based on matrix. The computation time of the DC_Apriori is less than the traditional Apriori and the MC_Apriori algorithm .In addition to it the MC_Apriori algorithm is able to handle massive datasets. The performance of the DC_Apriori algorithm is proved by comparing its performance with the other algorithms and a table highlighting the computation difference is shown. Many amendments have been made to the traditional priori algorithm like the use of matrix method where the scholars have made use of conversion of the entire database in a matrix form and then the “AND” operation is applied to it to further perform computations. The algorithm further minimizes the number of transactions by reducing the number of connections and the relationship between them. No repeated candidate sets are generated and hence efficiency is improved. Moreover the process of matrix making involves a lot of time consumptions for construction of a matrix every time for inserting items into the matrix and adjusting matrix structure. This improved algorithm requires only one pruning operation and connection operation of generating items is improved. Finally the performance of the three algorithms is compared on a mushroom and a retail database thereby concluding the superiority of DC_Apriori.

2.4 From Data Mining to Knowledge Discovery in Databases

Fayyad et al., has provided a glimpse of the emerging field of mining of data and knowledge discovery in databases. It explains its applications in the real world, and various other mining techniques and challenges involved along with the current and future perspective on the same. The paper also discusses various methods for data mining and their application issues. The historical account of KDD is discussed and its intersection in other spheres to get an idea as to how it influences other sectors also. The traditional method of data analysis requires manual power and becomes very cumbersome. At the basic level the KDD is concerned with deriving data that has sense or maybe adding sense to the raw data. The main issue talked by the KDD course is one of diagramming low-level data (which are naturally too capacious to interpret and outline easily) into other formulae that force be more condensed, more nonconcrete (for example, a vivid calculation or model of the procedure that created the records), or more convenient (for example, a prophetic classic for assessing the value of forthcoming cases).The actual aim is to derive the application of various data mining methods to discover patterns and their extraction.

2.5 Learning of the various applications of Data Mining

Liu has described the Apriori algorithm in association rule mining algorithm in detail and its implementation and illustrations .In this paper along with the association theory the entire Apriori algorithm is explained and illustrated .The applications are discussed in detail. Along with it the various threshold parameters such as support, confidence are discussed that form the basis of the Apriori algorithm and important for the implementation of this very algorithm. Apart from the implementation the various pros and cons of the algorithm is discussed in detail to know exactly how much the algorithm is able to achieve the motive. The paper also discusses the improvements the algorithm can undergo for better functioning and efficiency that this algorithm can achieve. The basis of the algorithm is discussed in depth and various conclusions are derived that can help to mine data efficiently so that determining a pattern from data becomes easy.

2.6 Current position and upcoming directions of frequent pattern mining

The paper discusses briefly about the current position of frequent pattern mining and get an overview of its upcoming directions in frequent pattern mining .Multiple objects or items have some or the other relationship among them. This paper studies those results to come to a desirable output. Further solicitations like Indexing ,Hypermedia statistics mining, Mining data streams, Web mining, Software virus mining are discussed briefly. Frequent pattern has a tremendous scope of success in future and this scope has been discussed in detail. A rough outline has been provided about the work that has been done on this very topic and a general idea of people from different fields has been provided in the paper. Frequent pattern mining is actually required in various fields and has achieved a huge success and has a wide variety of applications. Moreover a more depth knowledge is needed so that this particular area has a tremendous impact and a long lasting one too. Further its applications in Indexing and finding comparison of multifaceted organized data, Spatiotemporal and hypermedia data mining, mining data streams, web mining and Software virus mining and system hiding is identified. Mining correlations in storage systems in one of the all other applications of frequent pattern mining. Storage caching and prefetching are some other applications of frequent pattern mining. Recurrent XML query arrangements are also used to recover the hiding presentation of XML organisation schemes.

2.7 Improving the Apriori algorithm for related association rules

Maolegi and Arkok have come up with the method of improving the Apriori algorithm for the related association rules. The paper starts by discussing the limitation of the traditional Apriori algorithm and the overhead it has of scanning the entire database and selecting the frequent item sets every time. The paper come up with a method to remove to overhead of scanning the entire database every time. The algorithm proposed and the traditional Apriori algorithm has been compared by implementing them on various transactions and the relative performances has been compared. The proposed algorithm truly beats the traditional algorithm by almost 67.38%.The algorithm is successful in making the traditional algorithm more efficient and less time consuming. The time required to generate the candidate support count is much more less in the proposed algorithm as compared to the traditional one. As per the results whenever the gap between the improved Apriori and the original Apriori increases from view of time consumed the value of minimum support increases.

2.8 Flood Area Prediction as an application of Apriori algorithm

Harun et al., proposed an application of Apriori algorithm for predicting the flood areas. The study comprises the assortment, pre-processing of data, and data alteration .This data is then tested entirely by the Apriori algorithm. To reach the results the parameters such as support, confidence and lift are taken up and the desirable results are then fetched. The results showed that every district generated the central rules consisting of the association of the villages and the water level. The results generated can hence be used in flood management. The predicted results can also provide an early warning to the villagers as to when the situation is going to get worse. Not just can this study be limited to the small scale but can be extended to large scale applications also. The results will be highly useful for the flood potential areas and use of early measures to avoid the situations beforehand. The research has been done purely by the data provided by the newspapers and reports. The raw data has been collected from Malaysian Irrigation Department and used as an input for the study. Domain understanding is very much needed for extraction and selection of data. The relations between river flow and the flood area has been seen and the desirable results are hence generated.

2.9 A Pattern Growth Approach to Constraint based Frequent Pattern Mining

Pie and Han have explored the influence of pattern-growth methods towards frequent pattern mining with hard constraints and have signified the interesting problems that are open. The principles involving the pattern mining have been taken into consideration and sequential pattern has been discovered. The role of pattern approach has been found towards the generation of frequent pattern and their extraction from large data sets. The need for finding the patterns is discovered. The strength of the pattern growth mining over algorithms like Apriori algorithm is discussed and the importance is taken into consideration. Recent studies have highlighted how the frequent patterns help to extract data related to floods or DNA etc. and their contribution in various fields has been derived. The constraint-based mining pattern in the outline of mining recurrent item sets, connotations, associations, consecutive arrangements and other interesting patterns in large database has been studied in the paper.

2.10 The increase in the Quality of Promotion using Association Rules for Apriori algorithm

Zulfikar et al., have together studied the data of a retail company namely –XMART. The aim of the company is to increase sales along with a promotion. The total quantity of records dealt was nearly 10,000 records. The data can be further processed then using association rules over the Apriori algorithm. Using the association rules the support and the confidence is then calculated to know the importance of every rule and derive relationship between the items. Association rules have actually acted as a benchmark for the promotion of the product as it derived the necessary product. In addition to all this the association rules also help by acting as a reference to the various product and determining their layout .The location of the product really can matter to simulate the sales and hence the items within the important rules are placed together to increase their sales. And hence items under the most important rule are hence promoted and sales are hence increased which is the main aim of the store like XMART. The paper further describes that the store XMART is actually divided into a number of clusters ,numerically 8. The rule that applies to one cluster may entirely be different to the rule that is applied to other cluster. The rules hence imposed can then act as a way to promote promotion thereby increasing sales. Several other factors may

also affect such as the duration of the day. The rules may change depending on what time of the day it is. A rule that may apply to the day time may entirely be different to the one applied at night. Apriori algorithm is specifically designed for large data in the form of transaction which makes it even more useful for such type of data. What makes Apriori algorithm more useful is that it is easy to use. Main problem faced here is the generation of a large number of candidate sets which can make the entire process to be a little cumbersome and less useful. But the improved Apriori algorithm have been devised which help to overcome this problem.

2.11 Is the number of constraints limited?

Pie and Han have come up with the unique idea of adding more constraints to the Frequent Pattern Mining. Normally the constraints used are all on the basis of association rule mining that basically gives us support, confidence, lift etc. But as per the recent studies and research adding more number of constraints will actually help in improving the performance. This paper has proposed to add some new constraints into the mining process. Although there are some constraints that are believed to be a little “tough” and hence it gets difficult to incorporate those. This paper introduces an extension to those constraints to help the get deep into the data mining process. Other interesting classes such as convertible constraints are identified which can then be pushed deep into the data mining process. The main five classes are identified and its been concluded then that only four of these classes can be incorporated. The paper starts by discussing the convertible class and its importance which actually could not have been studied in the earlier studies. In addition to this, this particular class is then believed to be a good set of integrated constraints and can be pushed into the frequent pattern mining. Next the monotone candidates have been introduced and all the constraints are divided into 5 categories particularly Its been shown that the four categories: succinct, anti-monotone, monotone and convertible can be easily incorporated into the data mining process. Further the process of constraint pushing and the frequent pattern growth mining is integrated thereby increasing the efficiency and high performance.

2.12 Web Log Mining using Matrix Apriori Algorithm

Data mining is actually useful and has an immense importance in the Big Data. Data mining has actually become essential to extract useful information from raw set of data and also to meet other user needs. Data mining is actually successful to bring various technologies together namely: artificial intelligence, machine learning and many others. Data mining has further its sub processes: data is prepared first, then the data is mined and finally the results

are evaluated. The paper tries to throw light and find out the relationship of transaction in a web log by analysis it using association rules and then do personal recommendations. Web log mining can actually provide with a lot of information and do important personalized services and finally achieving the Web personalization. Using the association rules the relationship between various data can be analysed and the transactions can be hence evaluated. For example a company like Walmart used this process to find out the relationship between various commodities like beer diapers etc. And the aim was to bundle those products together who together make a weighted support and hence that particular rule becomes important for analysis. The driving force is the customer behaviour and how it acts in different weathers or maybe different times. Moreover the comparison between algorithm such as Apriori, AprioriTid and Apriori_LB etc. is done .Out of all the algorithms the matrix algorithm is then picked up to carry out the Web log mining. The data on which the study was carried out out was mainly the Sogou search web logs. Experimental results and analysis are carried out and the desirable results are then withdrawn. Association rule mining is actually very beneficial in deriving the relationship between the consumer behaviour and the transactions that they carry out. With the help of the web log mining the query classification becomes easy, that will be beneficial in the future.

Web Mining

Phases of Web Usage Mining

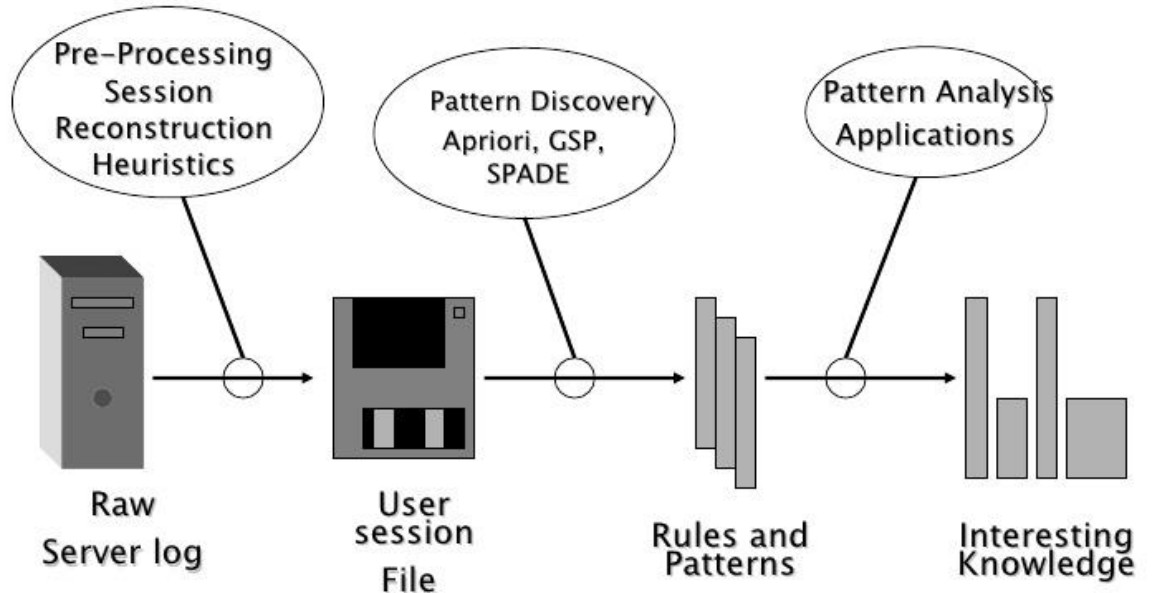


Figure 5

2.13 Hp-Apriori Algorithm for frequent dataset mining

With the increase in the volume of data from all spheres and development like networking, data storage etc. more and more data has been generated. There are multitudes of unstructured data in Big Data and more of real time analysis is actually required on them. Since large amount of data is generated they carry different characteristics that include large-volume, heterogeneity, decentralised control etc. and there is a need to explore the complex and evolving relationships among data. The ARM basically has two segments, (1) mining the Recurrent Item sets and (2) Association Rule (AR) withdrawal. For extracting data in these forms the association rule mining is considered as the most important algorithm as it is pretty much successful in deriving data in these forms. Apart from the Apriori algorithm other algorithms such as FPGrowth and Eclat are useful for deriving

similar results. As the Apriori algorithm is less efficient as compared to other algorithms as it uses the concept of Breadth-First Search and consumes more time. The Horizontal Parallel algorithm is actually derived from a base algorithm Apriori algorithm. Through this algorithm the data is mined from the Big Data in a parallel way. Also this algorithm maintains an index file so that the process can become a little more speedy and efficient. The main reason for using Apriori is that it helps to achieve parallelism and simple make the normal Apriori algorithm a bit more efficient. The proposed algorithm is compared to the traditional algorithm on approximately four datasets and the results are in favour of the proposed. So results show that that it's better to use the proposed algorithm while dealing with data sets.

2.14 A More Advanced Apriori Algorithm based on Recurrent Matrix

Niu et al., have proposed a better algorithm by using the matrix based approach. Among all the methods the Apriori algorithm has been the most famous of all and the most used as well. In this paper a more developed Apriori algorithm has been designed using the matrix based approach. The Apriori algorithm uses a bottom up approach and undergoes multiple database scans that makes the entire process a bit cumbersome and elongated at the same time. The paper introduces a new algorithm namely the Frequent Matrix Algorithm (FMA). This algorithm is successful to discretize the matrix by the minimum support as a parameter which is generated automatically. At an experimental aspect this algorithm has been proven to be far better than the traditional algorithm. Association rule mining plays a very important role in customer behaviour analysis and making that as a basis of the algorithm. It has its major application in commercial analysis. For constructing the frequent matrix firstly a single scanning of the database is done and then modelling of data takes place. Although the performance of the classical algorithm can be very inefficient when it will deal with a massive amount of data but the improved algorithm provides a relief from this very problem and provides a solution. The proposed algorithm basically has three steps-building a matrix, discretization and finally searching full-1 sub-matrix. The proposed algorithm unlike the Apriori scans the database just once and makes use of the frequent matrix and finally it searches for the valid highest sequence.

2.15 Investigation and Enhancement of Apriori Algorithm for Association Rules

Chengyu and Xiongying had proposed an algorithm for the improvement of the traditional Apriori algorithm. . With the expansion and the extensive solicitation of DBMS, large-scale database system is promoted in daily life. For renewing the association rules effectively the paper comes up with an idea of Apriori algorithm and points out its various advantages and disadvantages. The paper also analysis the classic FUP and IUA and discusses their advantages and disadvantages. It also gives a narrative for NIUP and NFUP. NFUP algorithm links robust huge item sets into unimportant measurable of candidate item sets grounded on robust large item sets concept, and approves initial clipping policy to expurgate down the periods of scanning database. The proposed algorithm basically has two steps-connecting step and pruning step. . IUA algorithm uses an exclusive candidate recurrent item set to produce procedure iua_gen that produces minor recurrent substances before perusing the DB for each time, thereby improving the effectiveness of the procedure. The proposed algorithm just similar to the Apriori algorithm uses the policy of “non-empty items of one recurrent items must be themselves also be frequent items “and hence works accordingly. The proposed algorithm is efficient as it scans the database’s IO less as compared to the original algorithm and hence saves time at the same time. The paper also describes the association rules and their applications. The proposed algorithm is simple no doubt but it creates a problem as it carries out a lot of pruning and hence has this overhead. The support degree of FUP algorithm is also taken into consideration and analysed. The study of negative association rules for updating incrementally is also carried out and analysed.

2.16 A Frequent-Pattern Tree Approach to Candidate set generation

Han and Pie have projected a novel frequent tree structure which is a prolonged prefix-tree structure for storing compacted, vital evidence about recurrent designs, and mature an efficient FP-tree based quarrying method, FP-growth, for withdrawal the whole set of frequent patterns by pattern portion progression. Efficiency is achieved using three techniques: The first technique includes the compression of a large database, FP tree avoids costly, and repeated database scans at the same time. The second is no costly generation of candidate sets. The third is a partitioning-based, divide-and conquer method to fester the excavating task into a set of smaller tasks for mining confined patterns in restricted databases, which intensely reduces the exploration space. The good thing about FP Growth pattern is that it is able to be scaled to both the large and small patterns which makes it even more efficient and reliable. The study has also shown that it is very much faster than the traditional algorithm and various other algorithms that have been designed lately for the

same purpose. The FP Growth algorithm is also extending towards some better improvisations such as mining of closed item sets .Further mining the sequential patterns and finally pushing tough constraints. Though the algorithm is quite efficient, it might cause problems such because of the repeated construction of FP trees that makes the process a lot more cumbersome. Still the other extensions of this algorithm are being worked upon that can be more efficient than the original algorithm such as mining the structured pattern by developing further this approach. Another approach could be to mine fault tolerant patterns.

2.17 Implementing the Apriori algorithm in parallel by Map Reduce

Li et al., have proposed an algorithm by which one can implement the Apriori algorithm in parallel based on Map Reduce technique. The most common type of data mining problem is to search frequent patterns in large transactional databases, it is a huge challenge to come up with an algorithm that is able to deal with certainly large datasets, that too in an efficient manner. Using a large number of computer nodes processing of huge datasets is done on some types of distributable problems. The proposed algorithm proves out to be beneficial as it is able to scale up well and process large datasets on commodity hardware. Market basket analysis by means of association rule mining is described in the paper. The algorithm tends to find the association between the item sets and other transactions. Map Reduce is an original software context announced by Google in 2004. In a massively parallel manner large datasets and their relationships can be computed. Big data has also come up with a method to pre-process, cluster and classify the various algorithms. Mining association rules from large datasets becomes very easy by using this proposed algorithm. The main step for the implementation of the Apriori algorithm is to find the frequent item sets and then proceed further. In the mth restatement, it calculates the incidences of possible candidates of size m in each of the transactions. Hence it is possible to implement the computation process of an iteration in a parallel way. Wherever the candidate sets occur, their occurrences are combined together to form a set. Then the join operation is performed on the K items and further prune operation are performed on the K+1 item sets. In addition to everything else the process of finding the frequent pattern is somewhat cumbersome and demands time and computational latency .The parallel Apriori algorithm based on Map Reduce is very much efficient in overcoming this problem and provide with the desirable results. The main changes that the traditional algorithm undergoes is size up, scaling up and finally speedup to acquire the performance of the PApriori algorithm. Surprisingly the algorithm introduced in the paper is able to offer more efficiency as the size of database

increases. Hence the algorithm introduced in the paper is able to access the data in the commodity hardware efficiently and process it better.

2.18 Using Power set on Hadoop improving the Apriori algorithm

Imran and Ranjan have come up with a way to improve the traditional Apriori algorithm using the Power set on Hadoop. Since it's an age of Big Data, hence a large amount of data is produced. Since a large amount of data is to be dealt with it becomes difficult to store and analyse the data. A better method to implement Map Reduce Apriori algorithm has been proposed using a vertical layout of database with power set. The concept of Set theory of Intersection has also been proposed. This very concept is also used to carry out the process of determining the support value. The results that were derived showed significant improvement than the traditional Map Reduce method. Since data mining has to be with huge accuracy when dealing with a large data this algorithm proves out to accomplish this very task. This algorithm is a big "yes" because it uses two concepts namely-map reduce and the other is vertical layout of data. The implementation of this algorithm requires two main functions to perform-map () and reduce ().The main advantage of using the vertical data format was that it reduces the significant amount of time that was required in order to calculate the support value. By using the power set one was able to generate the set L1 in one single iteration. The best part about using the map () and reduce () function was that one is able to implement this algorithm on a dataset of any size. Another thing that makes this algorithm worth the use is that the time required for calculation of the candidate item sets is much less as compared to the other algorithm. This hereby increases the efficiency of the algorithm and the computation time for the algorithm is greatly reduced. The process of intersection helps to filter out some of the transactions in each iteration. The algorithm in this paper was implemented on around 88,000 records to achieve desirable results. There is a huge reduction in the time required to process the transactions and find out the significant relation among them. The next set L2 is generated from the previously generated item set L1 efficiently in just a single step thereby reducing the computation cost and time.

2.19 Reviewing the algorithm based on Apriori

Singh et al., presented an algorithm to modify the Apriori Algorithm on a map reduce framework. There are lot many algorithms that have been implemented to enhance the Apriori Algorithm for distributed and parallel processing. All of the proposed algorithms so far differ from one another in some or the other aspect such as load balancing, memory

systems etc. The problem with most of the algorithm is the need for a high level language for programming purpose. If we go with other kinds of computing like the grid computing then node failures is a problem that has a very high probability. This might also force for executing the tasks multiple times and makes the process hectic and unmanageable at times. To overcome all these problems the paper has come with a Map Reduce framework that was originally given by Google. Map Reduce is actually an efficient and a simplified model for distributed computing on fairly large distributed commodity softwares. It even has a big role to play in cloud computing. Map Reduce Framework works with the help of two main functions Map () and Reduce (). Also the Apriori algorithm is further implemented on the basis of the functions used eg-1-phase vs. k-phase, I/O Mapper etc. Apriori algorithm has various implementations and this paper discusses their impact on one another. Map Reduce framework has its own advantages and disadvantages and all of these are discussed in this paper too. It gives an idea as to when, where and which Apriori algorithm is suitable to run on a Map Reduce framework. The main aim of the algorithm introduced in the paper is to come up to a stage wherein parallelisation can be achieved. Also various amendments to the traditional Apriori algorithm were analysed and a conclusion was arrived. Apart from this approach there are many approaches to the implementation of Apriori algorithm on a Map Reduce framework. Apart from all that there are some bottlenecks to the proposed algorithm like Scheduling invocations and overhead of time obviously. Such a problem is looked up by algorithms like FPC and DPC. Being such a great platform Map Reduce can also create certain computational problems at the same time which needs to be taken care of.

2.20 Sentiment Analysis of Music using Association Rule Mining

This paper has been inscribed by Gomez and Caceres in which Apriori application is professed in Sentiment Analysis. Listening to music is very much influenced by a person's behaviour and can heavily affect the mood of a person. Feelings like sad, angry, cry, happy etc. are all the emotions that a person can have and can affect a person greatly and his/her choice in music as well. But to infer emotions via music is a very complex task and to use association rule mining for the same can very much help to achieve the task. For achieving a relationship between the emotions and music needs to be deduced. This is where association Rule Mining has a role to play. For that very purpose a dataset that has rhythmic features needs to be selected and the relationships can then be deduced. Various data mining algorithms such as Random k-Label sets, Multi label sets are taken into consideration. The thing is characterized by the various emotions a person undergoes while hearing to different

types of music. Mainly Six different emotions are taken into consideration and hence classified. The classification problem of music needs to be resolved and this algorithm aims to do the same. The dataset that was being worked upon here was firstly analysed then classified and finally it was interpreted. The detection of emotions as an approximation for doing analysis of sentiments is a must. Data mining can also be used to extract useful information. So by analysing the customer behaviour it can be analysed what kind of music the customer is usually interested in and then the results can be achieved through the same. The main aim is to identify the influence of different music types on human behaviour and its major influence.

2.21 Mobile e-commerce approval system using the improved Apriori algorithm

Guo et al., proposed the application of improved Apriori algorithm in a mobile e-commerce recommendation system. The main aim of this paper is to come with an idea that can make the mobile e-commerce system more convenient and user friendly at the same time. In addition to it the information overload needs to be maintained and handled at the same time. The characteristics of the mobile e-commerce system were further combined with the improved Apriori algorithm to come up with a way for the application of the recommendation system. Data mining efficiency is improved by recommending the products to the user. The paper has also used a Taobao online dress shop for making the Apriori algorithm system a bit more effective. The problem of visual interface in a mobile terminal is looked after. In addition to it the mass data continuously generated is handled. Moreover the prediction accuracy that the algorithm gives is far better than any other algorithm. Other limitations and advantages of this algorithm have also been taken into consideration. The major role here plays is predicting the interests of the user and accordingly the products can be considered. High-profit products are taken into consideration and are then promoted. The mobile e-commerce system adopts improved Apriori algorithm. The mobile recommendation system may use different recommendation systems for the purpose and offer good performance. The paper has the capability of extending to various other directions in the near future as well. One can integrate a lot of data into the mobile. Consuming such a rich dataset, upcoming investigation could straight subordinate produce references with customers' glancing and acquisition actions for more precise inquiries. Future research can also explore the effectiveness of the algorithm for more accurate investigations. Positioning functions can be used for increasing efficiency.

2.22 Use of Auto-Adjust Apriori algorithm

Miniar et al., proposed a Predictive Performance Analytics Scheme using Auto-Adjust Apriori Algorithm. There are many academic organizations and all of them have to analyse their student performance. Analysing helps to find out the strengths and weaknesses in every student and perform the algorithm accordingly. And since it is an academic organization it has a large amount of data mainly the result data. And hence it is necessary to process this information so that desirable patterns can be looked upon. Also in an academic context some or the other courses are always related to each other. The scheme has been implemented under .Net and is very much optimised and efficient. Also according to the students pattern the teachers and students are able to actually predict that which of the subjects should the student not pursue as his or her performance has not been that good then. The student then can accordingly select his current subjects so that he does not flunk in those subjects and is able to analyse his performance. These topic sets could be used to classify connected topics as each consequences dependencies. Students can easily make out which current semester subject can affect them most in the next semester. . If a student dislikes a subject X, he can find out the other subject Y in which almost equally backlogs occurred in past. The student can then make out that if he is not able to perform well in a subject say X the he won't be able to perform good in the subject Y in the next semester as well. The paper has been worked upon on a set of BCA students. . At the same time, the system was tried to form using intuition and prediction of faculties. Each faculty was assigned task to find list of frequent subject sets based on their expertise without considering any of the past results. The faculties could identify 5 most frequent subject sets on average per faculty. The system could identify 7 most frequent subject sets on average per batch. The reason behind improved performance is faculties think from technical point of view only. This system works from technical point of view as well as other parameters which affected overall performance.

CHAPTER 3-SYSTEM DEVELOPMENT

3.1 Development:

The System development of “Constrained based Frequent Pattern Mining” is as follows:

3.2 System Design:

The algorithm on which we have worked are :

Apriori Algorithm:

```
 $C_k$ : Candidate item set of size k
 $L_k$ : Frequent item set of size k
 $L_1 = \{\text{frequent items}\};$ 
For ( $k=1; L_k \neq \Phi; k++$ ) do begin
 $C_{k+1}$  = candidates generated from  $L_k$ ;
For each transaction  $t$  in database do
    Increment the count of all candidates in  $C_{k+1}$ 
    Those are contained in  $t$ 
 $L_{k+1}$  = candidates in  $C_{k+1}$  with min_support
End
Return  $\cup_k L_k$ 
```

Association Rules:

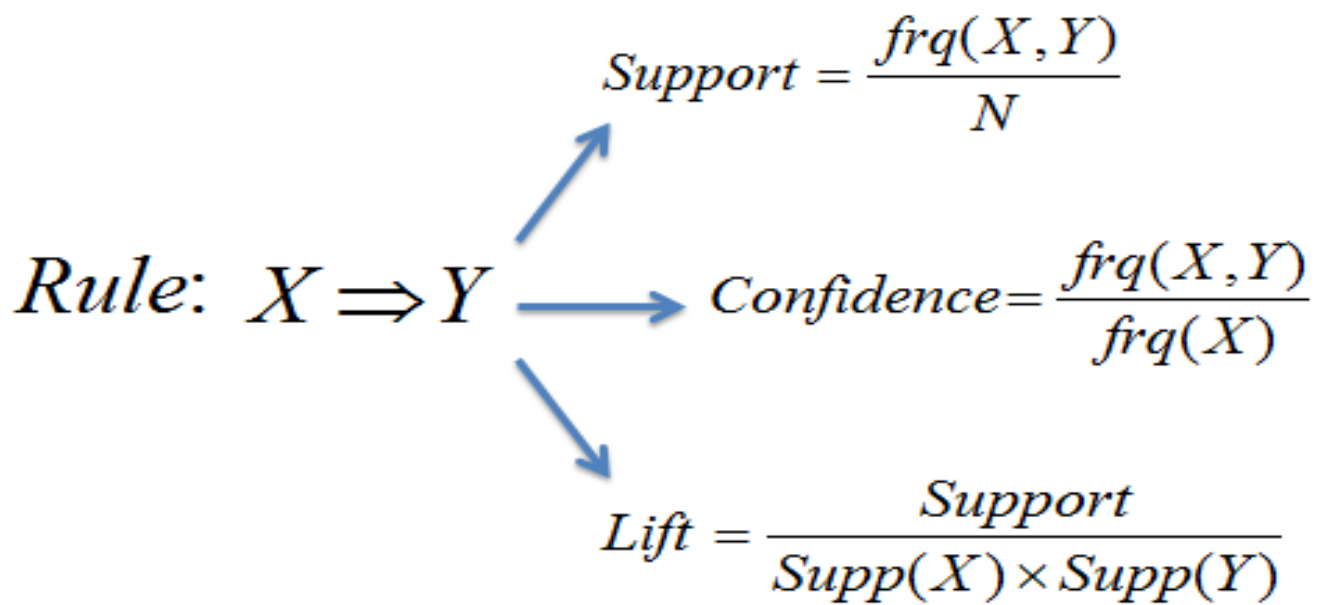


Figure 6

$$\text{Support} = \frac{\text{Number of transactions with both A and B}}{\text{Total number of transactions}} = P(A \cap B)$$

$$\text{Confidence} = \frac{\text{Number of transactions with both A and B}}{\text{Total number of transactions with A}} = \frac{P(A \cap B)}{P(A)}$$

$$\text{ExpectedConfidence} = \frac{\text{Number of transactions with B}}{\text{Total number of transactions}} = P(B)$$

$$\text{Lift} = \frac{\text{Confidence}}{\text{Expected Confidence}} = \frac{P(A \cap B)}{P(A) \cdot P(B)}$$

Figure 7

CHAPTER 4 : PERFORMANCE ANALYSIS

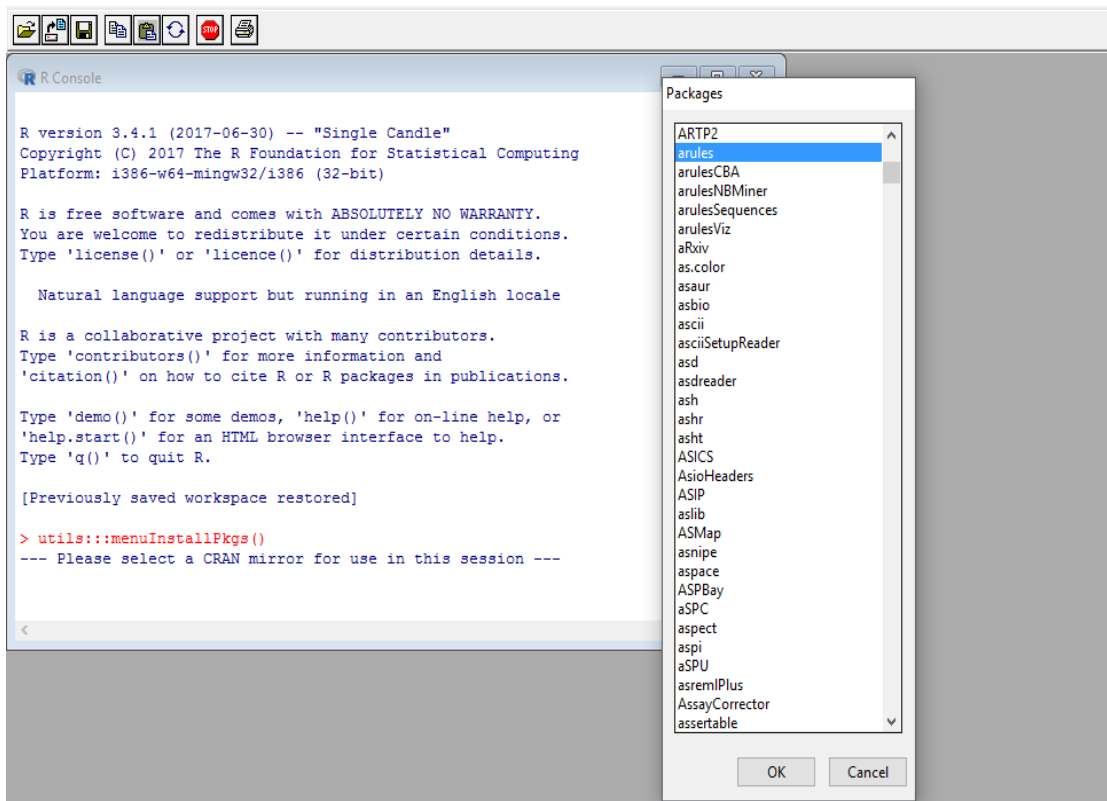
4.1 Analysis:

The information related to the topic was gathered and analysed. Past work being done was analysed and improvements were gathered.

4.2 Implementation:

We implemented the Apriori Algorithm to find the Frequent Pattern and perform Data Mining using RStudio:

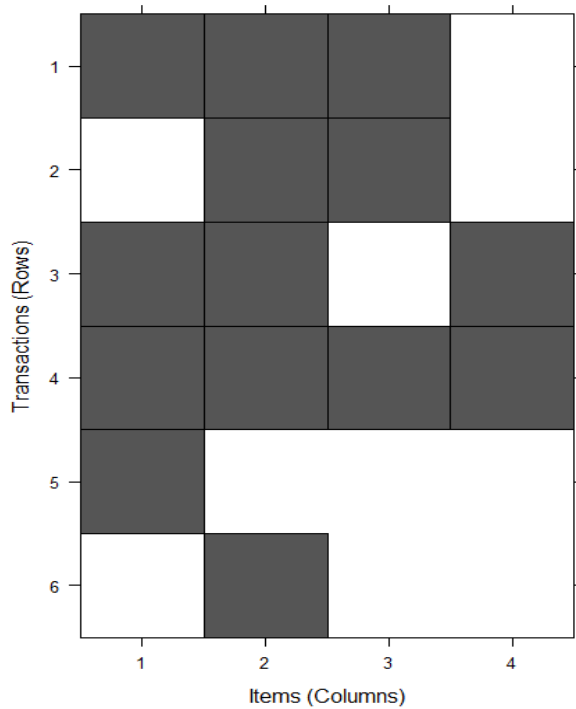
Step1: Our implementation started with installing arules in R.



Step2: We worked on a small database that we created ourselves.

```
ish.txt - Notepad
File Edit Format View Help
A,B,C
B,C
A,B,D
A,B,C,D
A
B
```

Step3: We implemented the commands in RStudio as:



Graph 1

Step4: The output is as follows:

```

Algorithmic control:
  filter tree heap memopt load sort verbose
    0.1 TRUE TRUE FALSE TRUE 2 TRUE

Absolute minimum support count: 3

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[4 item(s), 6 transaction(s)] done [0.00s].
sorting and recoding items ... [3 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [7 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> inspect(rules)
  lhs rhs support confidence lift count
[1] {} => {C} 0.5000000 0.5000000 1.0 3
[2] {} => {A} 0.6666667 0.6666667 1.0 4
[3] {} => {B} 0.8333333 0.8333333 1.0 5
[4] {C} => {B} 0.5000000 1.0000000 1.2 3
[5] {B} => {C} 0.5000000 0.6000000 1.2 3
[6] {A} => {B} 0.5000000 0.7500000 0.9 3
[7] {B} => {A} 0.5000000 0.6000000 0.9 3
> |

```

We implemented the **Apriori Algorithm** on a pre installed dataset “groceries” within the **arules** package:

The commands are as follows:

```

RGui (64-bit)
File Edit View Misc Packages Windows Help

Error in inspect(Groceries[1:3]) : could not find function "inspect"
> inspect(Groceries)
Error in inspect(Groceries) : could not find function "inspect"
> itemFrequencyPlot(Adult,topN=20,type="absolute")
Error in itemFrequencyPlot(Adult, topN = 20, type = "absolute") :
could not find function "itemFrequencyPlot"
> inspect(Groceries)
Error in inspect(Groceries) : could not find function "inspect"
> library(arulesViz)
Loading required package: arules
Loading required package: Matrix

Attaching package: 'arules'

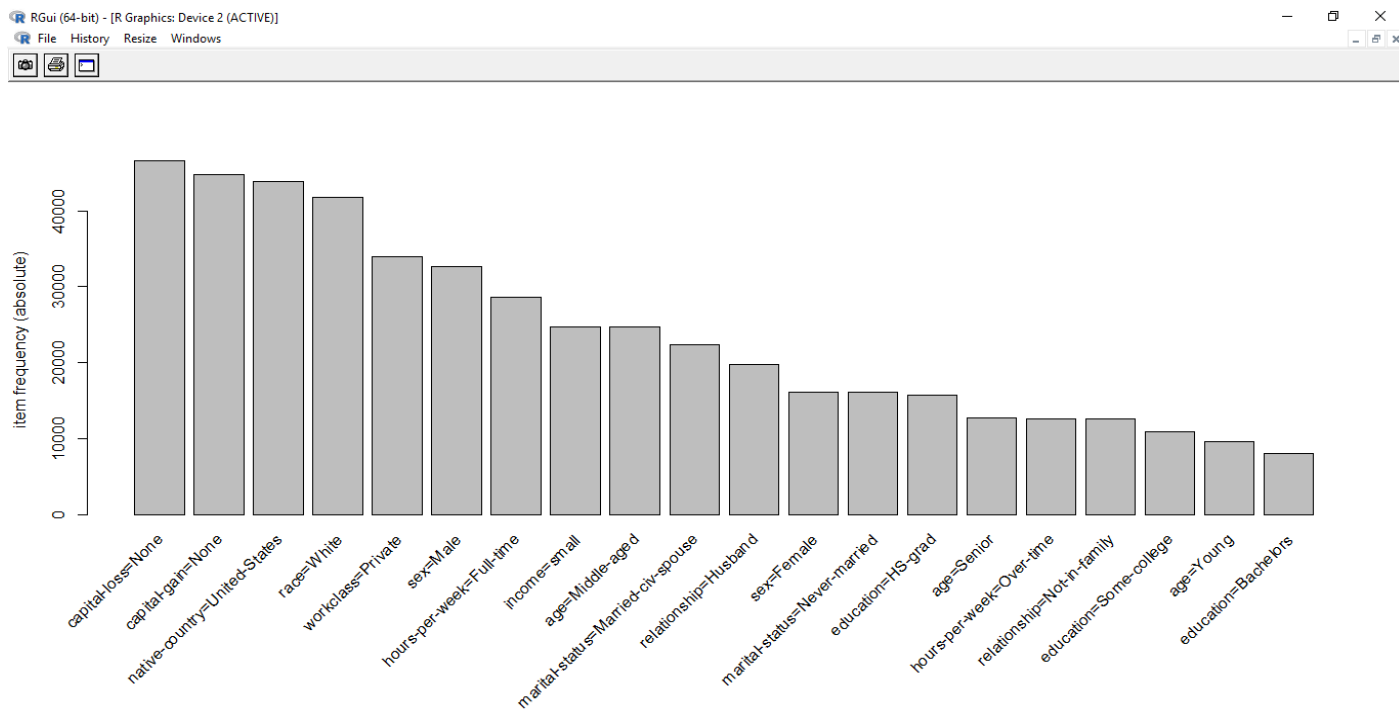
The following objects are masked from 'package:base':

  abbreviate, write

Loading required package: grid
Warning message:
package 'arules' was built under R version 3.4.2
> library(datasets)
> data(Groceries)
> itemFrequencyPlot(Adult,topN=20,type="absolute")
> |

```


4.3 Output:



Graph 2

We used the result and methodology of Apriori algorithm to come up with a method to find the most frequent sequence in a given dataset.

The dataset that we worked on was called 'sequence.txt'. The items were given an event id and a sequence id.

The steps and screenshots of the algorithm are as under:

Step1:

We attached packages that were necessary to implement this algorithm. Namely the

- 1.arules package
- 2.Matrix
3. arulesSequences

```

R Console

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> library(Matrix)
> library(arules)

Attaching package: 'arules'

The following objects are masked from 'package:base':

  abbreviate, write

Warning message:
package 'arules' was built under R version 3.4.4
> library(arulesSequences)

```

Step2: We imported the file that had various transactions and from which we had to find the sequence.

```

> x <- read_baskets("D:/Sequence.txt", info = c("sequenceID", "eventID", "SIZE"))
> as(x, "data.frame")
  items sequenceID eventID SIZE
1  {C, D}          1     10    2
2  {A, B, C}        1     15    3
3  {A, B, F}        1     20    3
4 {A, C, D, F}      1     25    4
5  {A, B, F}        2     15    3
6   {E}             2     20    1
7  {A, B, F}        3     10    3
8  {D, G, H}        4     10    3
9   {B, F}          4     20    2
10 {A, G, H}        4     25    3
> |

```

Step3: We set the support value to find the sequences having the minimum threshold value.

```

> s1 <- cspade(x, parameter = list(support = 0.4), control = list(verbose = TRUE))

parameter specification:
support : 0.4
maxsize : 10
maxlen  : 10

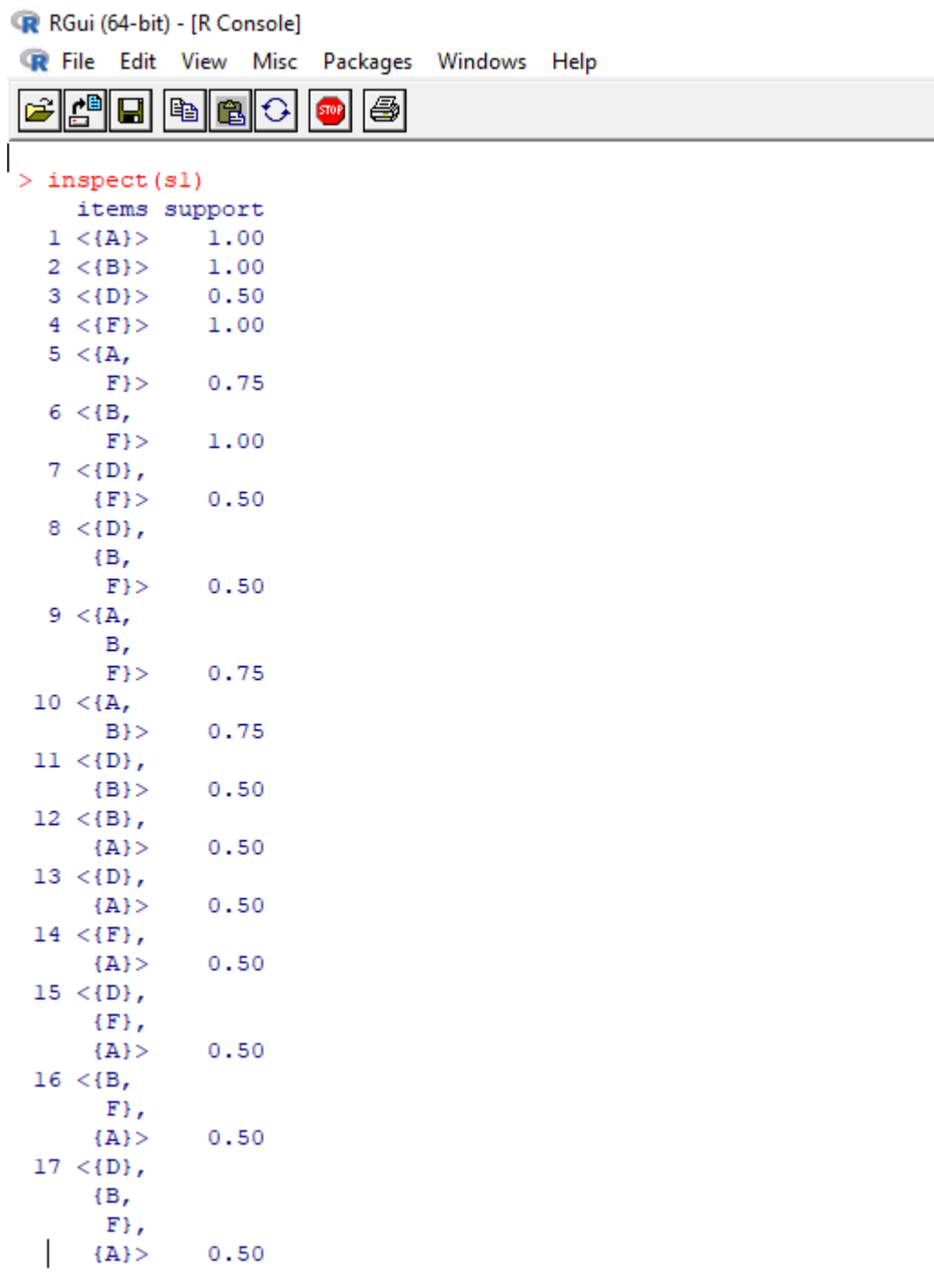
algorithmic control:
bfstype : FALSE
verbose  : TRUE
summary  : FALSE
tidLists : FALSE

preprocessing ... 1 partition(s), 0 MB [0.22s]
mining transactions ... 0 MB [0.11s]
reading sequences ... [0.03s]

total elapsed time: 0.36s
> |

```

Step4: We inspected and summarised the rules.



```
> inspect(s1)
  items support
1 <{A}>  1.00
2 <{B}>  1.00
3 <{D}>  0.50
4 <{F}>  1.00
5 <{A,
   F}>  0.75
6 <{B,
   F}>  1.00
7 <{D,
   {F}>  0.50
8 <{D,
   {B,
   F}>  0.50
9 <{A,
   B,
   F}>  0.75
10 <{A,
    B}>  0.75
11 <{D,
    {B}>  0.50
12 <{B,
    {A}>  0.50
13 <{D,
    {A}>  0.50
14 <{F,
    {A}>  0.50
15 <{D,
    {F},
    {A}>  0.50
16 <{B,
    F},
    {A}>  0.50
17 <{D,
    {B,
    F},
    | {A}>  0.50
```

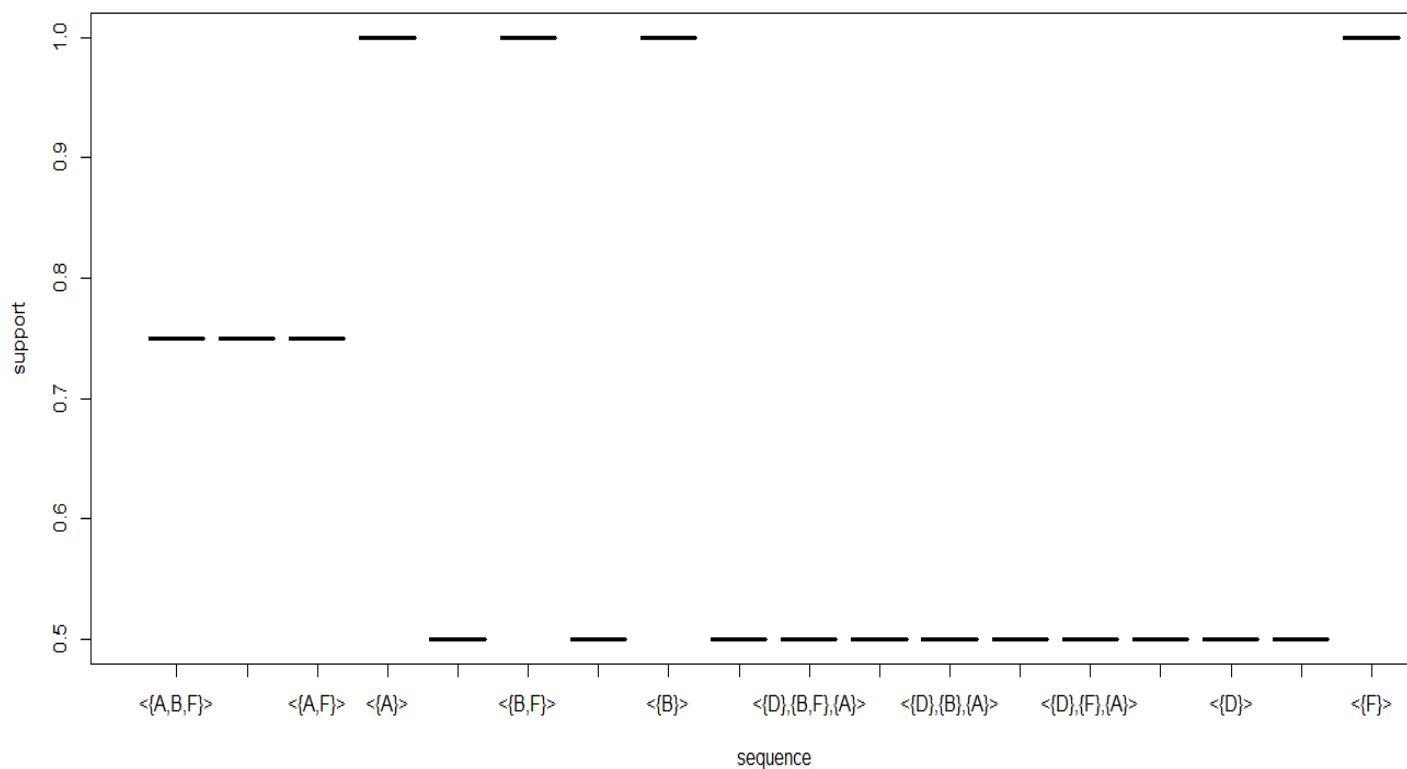
Step5: Finally we inspected the support values.

```

> y
      sequence support
1      <{A}>      1.00
2      <{B}>      1.00
3      <{D}>      0.50
4      <{F}>      1.00
5      <{A,F}>     0.75
6      <{B,F}>     1.00
7      <{D},{F}>   0.50
8      <{D},{B,F}> 0.50
9      <{A,B,F}>   0.75
10     <{A,B}>     0.75
11     <{D},{B}>   0.50
12     <{B},{A}>   0.50
13     <{D},{A}>   0.50
14     <{F},{A}>   0.50
15     <{D},{F},{A}> 0.50
16     <{B,F},{A}> 0.50
17     <{D},{B,F},{A}> 0.50
18     <{D},{B},{A}> 0.50

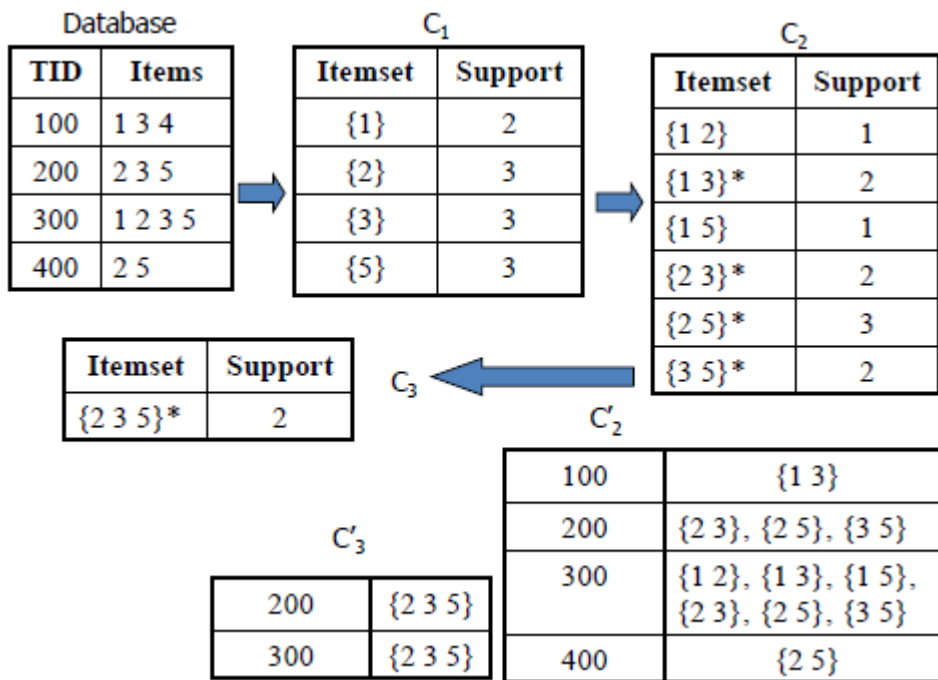
```

Step6:and then we plotted the output finding the most frequent sequence in our data.



Graph 3

Numerically:



The algorithm proposed in this paper to find the frequent sequence can be illustrated as:

This table shows the vertical format for our data:

| Seq. Id | Ev. Id | Item set |
|---------|--------|----------|
| 1 | 1 | ab |
| 2 | 2 | abc |
| 2 | 3 | cd |
| 2 | 1 | abf |
| 3 | 4 | aef |
| 3 | 2 | bd |
| 4 | 3 | abh |

Table 1

The above results can be compared to even better methods that are the improvement in this algorithm:

For a,

| Seq.Id | Ev.Id |
|--------|-------|
| 1 | 1 |
| 2 | 2 |
| 2 | 1 |
| 3 | 4 |
| 4 | 3 |

Table 2

For b,

| Seq.Id | Ev.Id |
|--------|-------|
| 1 | 1 |
| 2 | 2 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |

Table 3

For c,

| Seq.Id | Ev.Id |
|--------|-------|
| 2 | 2 |
| 2 | 3 |

Table 4

For d,

| Seq.Id | Ev.Id |
|--------|-------|
| 2 | 3 |
| 3 | 2 |

Table 5

For e,

| Seq.Id | Ev.Id |
|--------|-------|
| 3 | 4 |

Table 6

For f,

| Seq.Id | Ev.Id |
|--------|-------|
| 2 | 1 |
| 3 | 4 |

Table 7

For h,

| Seq.Id | Ev.Id |
|--------|-------|
| 4 | 3 |

Table 8

(a,b)=(1,1),(2,2),(2,1),(4,3)

(e,f)=(3,4)

(c,d)=(2,3)

(b,f)=(4,3)

are the candidates.

1. Transaction Reduction:

The traditional computation is inefficient because of such a large amount of productions of record. Moreover, if the record is extensive, it sets aside an excess of opportunity to inspect the database. In this paper we will fabricate a strategy to get the successive thing set by utilizing an alternate way to deal with the traditional Apriori computation and applying the idea of transaction decrease and another outline technique, along these lines wipe out the contestant having a subgroup that isn't frequent.

Basic algorithm :

1. Firstly create a matrix say B
2. We will set $p=3$

```

3. While (p<=t)
If (columnsum(colj)<min_support)
If (rowsum(row i)==p)
Delete row i

Merge (col j, col j+1)

p=p+1;
4.end while
5. Display B

```

| Data Size | Apriori | Proposed algorithm | Reduction(%) |
|-----------|---------|--------------------|--------------|
| 40 | 34 | 20 | 41 |
| 20 | 15 | 10 | 34 |

Table 9

Method 2:

```

1. find_frequent_1_itemsets
2. For (p = 2; Lp-1 ≠ ∅; p++)
{
//Generate the Cp from the Lp-1

(3) Cp = candidates generated from Lk-1; //get the item Iw with minimum support in Cp using L1,
(1≤w≤p).

(4) x = Get _ item_min_sup(Cp, L1); // get the target transaction IDs that contain item x.

(5) Tgt = get _ Transaction_ID(x);

(6) For each transaction t in Tgt Do

(7) Increment the count of all items in Cp that are found in Tgt;

(8) Lp= items in Cp ≥ min_support;

(9) End;

```


(10) }

| | Original algorithm | Improved |
|-----------|--------------------|----------|
| 1-itemset | 26 | 26 |
| 2-itemset | 19 | 13 |
| 3-itemset | 12 | 7 |
| Sum | 57 | 46 |

Table 10

The time consuming in improved Apriori in each group of transactions is less than it in the original Apriori, and the difference increases more and more as the number of transactions increases.

CHAPTER 5: CONCLUSIONS

5.1 Conclusions:

Frequent design mining is out of four noteworthy issues in the information mining space. The part gives a diagram of the real themes in frequent design mining. The most punctual work around there was focussed on deciding the effective calculations for visit design mining, and variations, for example, long design mining, interesting example mining, imperative based example mining, and compression. Lately scalability has turned into an issue due to the gigantic measures of information that keep on being made in different applications. What's more, as a result of advances in information gathering innovation, propelled information writes, for example, fleeting information, spatiotemporal information, diagram information, and indeterminate information have turned out to be more typical. Such information composes have various applications to other information mining issues, for example, grouping and order. The pattern in imperative based mining has been to assemble progressively nonexclusive frameworks. While at first requirement based mining frameworks gave uncommon reason dialects that lone upheld somewhat a larger number of limitations than particular regular itemset mining calculations did, as of late the scope of imperatives has extended, and the genericity of the dialects associating the imperative based mining, coming full circle in the incorporation with nonspecific requirement fulfilment frameworks and dialects. A few open difficulties remain. These incorporate a nearer reconciliation of imperative based mining with design set mining, improving comprehension of how to coordinate measurable prerequisites in requirement based mining frameworks, and mining organized databases, for example, diagram or arrangement databases utilizing adequately bland dialects.

5.2 Future Scope:

Plenteous writing is distributed in explore into visit design mining. However, in light of our view; there are as yet a few basic research issues that should be unravelled before visit design mining can turn into a foundation approach in information mining applications. To begin with, the most engaged and broadly contemplated point in visit design mining is maybe versatile mining strategies. When we are working with information streams still it is an examination test to infer a conservative however fantastic arrangement of examples that are most helpful in applications. The arrangement of successive examples inferred by the majority of the present example mining techniques including our own give inexact examples as stream is streaming persistently and a few information is lost during the time spent examining the stream. There are proposition on lessening of such a tremendous informational index, including closed patterns, maximal patterns, surmised designs, consolidated example bases, agent designs, grouped examples, and discriminative regular examples, yet at the same time it is look into issue to mine example sets in both minimization and delegate quality for a specific application. To make visit design mining a basic assignment in information mining, much research is expected to additionally create design based mining techniques. For instance, arrangement is a fundamental errand in information mining. Development of better characterization models utilizing successive examples than most other grouping techniques is again an exploration issue. Another significant research zone in visit mining is understanding of examples i.e. semantic explanation for visit designs, and logical investigation of successive examples. The semantic of a regular example incorporates further data: what is the significance of the example; what are the equivalent word designs; and what are the ordinary exchanges that this example dwells? On one side, it is vital to go deeply part of example mining calculations, and dissect the hypothetical properties of various arrangements. Much work is expected to investigate new utilizations of incessant example mining. For instance, bioinformatics has raised a considerable measure of testing issues, and we accept visit design mining may contribute a decent arrangement to it with additionally investigate endeavours.

5.3 Applications:

Frequent pattern mining has utilizations of two sorts. The main sort of utilization is to other real information mining issues, for example, grouping, anomaly identification, and characterization. Frequent designs are frequently used to decide significant groups from the basic information. What's more, control based classifiers are regularly developed with the utilization of continuous example mining strategies. Visit design mining is likewise utilized as a part of nonspecific applications, for example, Web log investigation, programming bug examination, substance, and organic information.

5.3.1 Ordering and similitude hunt of complex organized information

Complex objects, for example, transaction arrangement, occasion logs, proteins and pictures are broadly utilized as a part of numerous fields. Proficient pursuit of these items turns into a basic issue for some applications. Because of the expansive volume of information, it is wasteful to play out a successive sweep in general database and inspect protests one by one. Superior ordering instruments in this way are in substantial request in sifting objects that clearly abuse the query prerequisite.

5.3.2 Spatiotemporal and media information mining

A spatial database stores a lot of room related information, for example, maps, preprocessed remote detecting or medicinal imaging information, and VLSI chip design information. A spatiotemporal database stores time-related spatial information, for example, climate flow, moving items, or provincial improvements. Spatial information mining alludes to the extraction of learning, spatial connections, or other intriguing examples from spatial information. So also, spatiotemporal information mining is to discover spatiotemporal learning and examples. Visit design investigation in mixed media information assumes a comparative critical part in sight and sound information mining. To mine incessant examples in media information, each picture protest can be dealt with as an exchange and much of the time happening designs among various pictures can be found.

5.3.3 Mining data streams

Colossal and conceivably interminable volumes of information streams are regularly created by ongoing observation frameworks, correspondence systems, Internet activity, online exchanges in the budgetary market or retail industry, electric power matrices, industry generation forms, logical and designing trials, remote sensors, and other dynamic situations. Dissimilar to customary informational indexes, stream information stream all through a PC framework constantly and with fluctuating refresh rates. It might be difficult to store a whole information stream or to look over it numerous circumstances because of its colossal volume. To find learning or examples from information streams, it is important to create single-filter and on-line mining strategies.

5.3.4 Web mining

Web mining is the utilization of information mining systems to find examples and learning from the Web. There are three distinct sorts of web mining: web content mining, web structure mining, and web use mining. Web content mining is a learning disclosure errand of discovering data inside website pages, while web structure mining expects to find information covered up in the structures connecting site pages. Web utilization mining is centred on the examination of clients' exercises when they peruse and explore through the Web. Traditional cases of web use mining incorporate, however not restricted to, client gathering (clients that regularly visit a similar arrangement of pages), page affiliation (pages that are gone by together), and consecutive click through investigation (a similar peruse and route arranges that are trailed by numerous clients).

REFERENCES:

1. Agrawal R, Srikant R. *Fast Algorithms for Mining Association Rules in Large Databases* (International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc, 1994), pp.487-499.
2. Karimi-Majdu A M, Mahootchi M. *A new data mining methodology for generating new service ideas* (Information Systems and e-Business Management, 2015, 13(3)), pp.421-443.
3. Wang J, Li H, Huang J, et al. *Association rules mining based analysis of consequential alarm sequences in chemical processes* (Journal of Loss Prevention in the Process Industries, 2016(41)), pp.178-185.
4. Borgelt C. *Frequent item set mining* (Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2012, 2(6)), pp.437-456.
5. Han J, Pei J, Yin Y. *Mining frequent patterns without candidate generation* (Acm Sigmod Record, 2000, 29(2)), pp.1-12.
6. Bhaskar R, Laxman S, Smith A, et al. *Discovering frequent patterns in sensitive data* (ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 2010), pp.503-512.
7. Vo B, Chi M, Minh H C. *Fast Algorithm for Mining Generalized Association Rules* (International Journal of Database Theory & Application, 1994, 2(12)), pp.161-180.
8. Rao S., Gupta R., *Implementing Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm* (International Journal of Computer Science and Technology, 2012), pp. 489-493.

9. Park J.S., ChenyM.S., Yu P.S. *Using a Hash-Based Methodwith TransactionyTrimming and DatabaseuScan Reduction for MiningyAssociation Rules* (IEEE Transactions on Knowledge & Data Engineering, 1997,9(5)), pp.813-825.
10. Zaki M.J., *Parallel uand distributed associatiomyining: A survey* (IEEE Concurrency, Special Issueton Parallel Mechanisms fortData Mining, 1999, 7(4)), pp.14-25.
11. Toivonen H., *Sampling LargeyDatabases for Association Rules* (Proc Vldb, 2000), pp.134-145.
12. Dong J., Han M., *BitTableFI: Antefficient mining frequent itemsetsralgorithm* (Knowledge-Based Systems, 2007, 20(4)), pp.329-335.
13. Bhandari A., Gupta A., Das D.,*yImprovised Apriori Algorithm using frequent pattern tree for real time applications in dataymining* (Procedia Computer Science,2015(46)), pp.644–651.
14. Zhao BG., Liu Y., *An Efficient Bittable BasedrFrequent Itemsets MiningyAlgorithm* (Journal of Shandong University, 2015(5)), pp.23-29.
15. Lazcorreta E.,tBotella F., Fernández-Caballero A. *Towardstpersonalized recommendation by two-stepumodified Apriori data mining algorithm* (Expert Systems with Applications, 2008, 35(3)), pp.1422-1429.
16. Jiao Y. *Research of anyImproved Apriori Algorithm in Data Mining Association Rules* (International Journal of Computer & CommunicationtEngineering, 2013, 2(1)), pp.25-27.
17. Fu S., Zhou HJ., *The Research and Improvement of Apriori Algorithm for Mining Association Rules*
- 18.Agrawal R, Srikant R. *Fast Algorithms foryMining AssociationRules in LargerDatabases* (International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc, 1994), pp.487-499.
19. Grahne O. and Zhu J. EfficientlyyUsing Prefix-trees inrMining Frequent Itemsets, In Proc. of theeIEEE ICDM Workshopeon FrequentuItemset Mining, 2004.

20. Christian Borgelt. An Implementation of the FP-growth Algorithm. Workshop Open Source Data Mining Software.
21. R Development Core Team. R Language Definition. Version 2.12.0 (2010-10-15) DRAFT.
22. Luís Torgo. Introdução à Programação em R. Faculdade de Economia, Universidade do Porto, Outubro de 2006.
23. Santhosh Kumar and K.V.Rukmani. Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms. Int. J. of Advanced Networking and Applications, Volume: 01, Issue: 06, Pages: 400-404 (2010).
24. Christian Borgelt. Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination. Workshop Open Source Data Mining Software (OSDM'05, Chicago, IL), 66-70. ACM Press, New York, NY, USA 2005.

