An Improvement in Execution Time of Apriori Algorithm

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Sonal
(801232028)

Under the supervision of:
Mr. Ravinder Kumar
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "***An Improvement in Execution Time of Apriori Algorithm***", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mr. Ravinder Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Sonal)

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

(Mr. Ravinder Kumar)
Assistant Professor
Computer Science and Engineering Department

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

I would like to express my gratitude to my supervisor Mr. Ravinder Kumar for the useful comments, remarks and engagement through the learning process of this master thesis. Words are inadequate to express the great care and interest taken by him in all aspects of my thesis work.

I express my gratitude to Dr. Deepak Garg, Head, Computer Science & Engineering Department for the motivation and inspiration that he built in us leading to the completion of thesis work.

I would like to thank my colleagues, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together.

Last, but not least I am thankful to all those people who have directly or indirectly helped me during my thesis work.

Sonal

801232028

ME (CSE)

Abstract

Data mining which is also known as Knowledge Discovery in the databases (KDD) is an important research area in today's time. One of the important techniques in data mining is frequent pattern discovery. Finding co-occurrence relationships between items is the focus of this technique. The active research topic for KDD is association rule mining and many algorithms have been developed on this. This algorithm is used for finding associations in the item-sets. Its application areas include medicine, World Wide Web, telecommunication and many more. Efficiency has been an issue of concern for many years in mining association rules. Till date the researchers of data mining have worked a lot on improving the quality of association rule mining and have succeeded to a great extent. There are many algorithms for mining association rules. Apriori algorithm is the mostly used algorithm which is used to determine the item-sets, which are frequent, from a large database. It extracts the association rules which in turn are used for knowledge discovery. Apriori is based on the approach of finding useful patterns from various datasets. There are lot many other algorithms that are used from association rule mining and are based on Apriori algorithm. Although it is a traditional approach, it still has many shortcomings. It suffers from the deficiency of unnecessary scans of the database while looking for frequent item-sets as there is frequent generation of candidate item-sets that are not required. Also there are sub item-sets generated which are redundant and algorithm involves repetitive searching in the database. This work has been done to reduce the redundant generation of sets. The large dataset is scanned only once. As a result the overall time of execution is reduced. Also the number of transactions to be scanned are reduced.

Table of Content

List of Figures

List of Tables

Abbreviations

KDD - Knowledge Discovery in the Database

OOO - One by One

BUC- Bottom up Construction

DGP - Dimension Group Parallel

DS - Data Set

DS^T – Transposed Data Set

Chapter 1

Introduction

This chapter consists of the introduction to the association rules, their mining, and knowledge discovery from the database and Apriori algorithm.

1.1 Data Mining

Large chain of stores like Wal-Mart in U.S has large number of stores worldwide. They have to deal with terabytes of data [31]. It deals with billions of transactions every day. The major companies like telecommunication collect bulk of data. Also the banks and government agencies collect huge amount of data. Due to such growth in electronic information, there is realization with the organizations that the data stored over the years is of utmost importance in strategy making. This information contains the intelligent data useful in business decisions and also the success of the organization may depend on it. It is not practical to analyze such large data manually. Thus there is need to design techniques that may extract valuable information from a large database. Such techniques are provided by the data mining.

It is the process of analyzing the data from various perspectives and then summarizing that data into useful information. This useful information can be used to increase revenue, cut costs, or both. It is sometimes called as Data or Knowledge Discovery (KDD).

The techniques of data mining can be used for smaller data however with the larger data there are better chances of finding interesting things. Larger data ensures that the interesting things discovered is not due to some random fluctuations but due to the fact that it is consistent and can be replicated.

Traditional databases help in solving simple queries but they do not provide any intelligence about the data. Data mining provides intelligence about the data that helps in better understanding of business. In conventional systems the responsibility of

manager is more as the process of analysing hypothesis is expensive and time consuming. But in data mining large amount of work is done by the software thus reducing the efforts of the manager.

Data: Any number, facts or text that can be processed by the computer is termed as data. Organizations accumulate large amount of data in different databases and different formats which includes operational data (cost, payroll etc), non operational data (Industry Sales) and Meta Data (Data Dictionary definitions).

Information: It is termed as relationship and association between the data.

Knowledge: Information can be converted into knowledge to determine the future trends.

1.1.1 Data Warehouse

It is a system that is used for analysis of data and generation of reports. It contains summary of information collected from different systems in order to support queries that are complex. It stores data that is both historical and current. Although updation is not needed every minute but still it is done time to time in order to keep the current information. Data stored in the data warehouse may not always be best suited for data mining analysis however in many cases data in the warehouse is the only required source of information for mining.

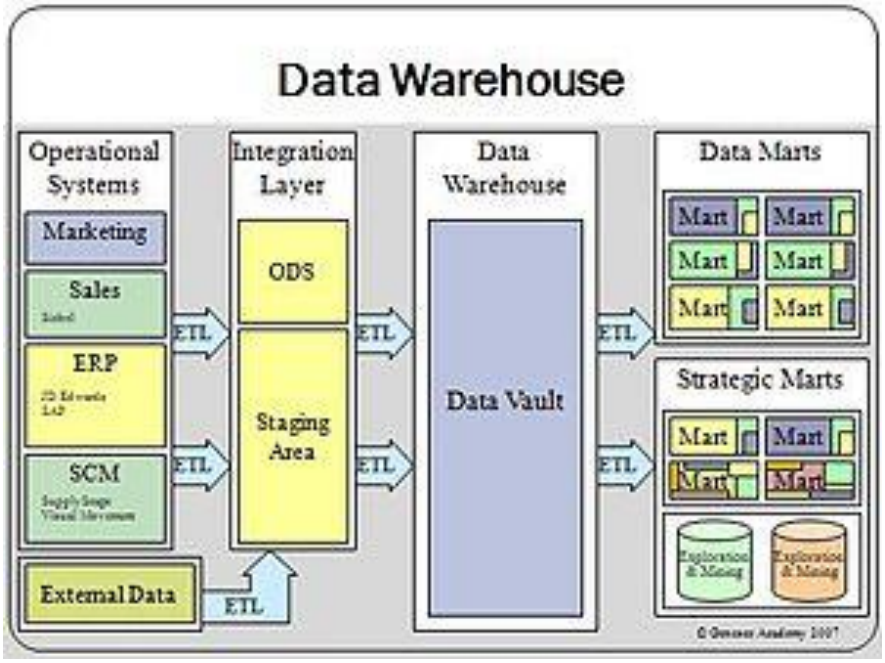

Figure 1.1: Various Parts of Data Warehouse

1.1.2 Knowledge Discovery in the Database (KDD)

The role of data mining (KDD) is very important in many of the fields such as analysis of market basket, classification, etc. If talk about data mining, the most important role presented by frequent item set which is used to find out the correlation between the various types of the field that is displayed in the database. Discovery of frequent item set is done by association rules. Retail store also use the concept of association rule for managing marketing, advertising, and errors that are presented in the telecommunication network.

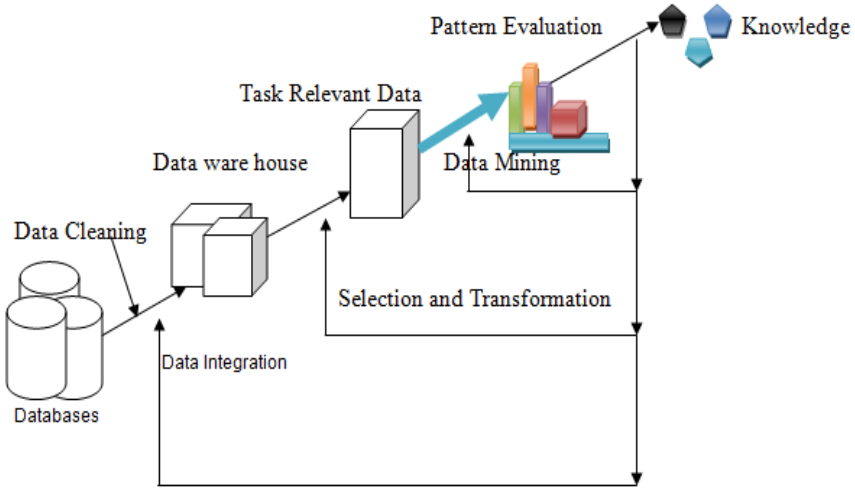

Figure 1.2: The Process of Knowledge Data Discovery

As we know information technology is growing and databases generated by the companies or organizations like telecommunications, banking, marketing, transportation, manufacturing etc are becoming huge. It is important to explore the databases and efficiently and completely as data mining helps to identify information in large amount of databases. KDD is the process designed to generate data that shows the well-defined relationship between the variables. It is the process designed to generate data that show the well-defined relationship between the variables. KDD has been very interesting topic for the researchers as it leads to automatic discovery of useful patterns from the database. This is also called as Knowledge Discovery from the large amount of database. Many techniques have been developed in data mining amongst which primarily Association rule mining is very important which results in association rules. These rules are applied on market based, banking based etc. for decision making.

The relationship among the items is done by association rule. All type of relationship between items is totally based on the co-occurrence of item.

The knowledge discovery in data can be achieved by following steps:

- *Data Cleaning*: In this step, the data that is irrelevant and if noise is present in database then both irrelevant and noisy data is removed from the database.

- *Data Integration*: In this step the different types of data and multiple data sources are joined in a common source.
- *Data Selection*: In this stage, the application analyzes that what data, what type of data is retrieved from the collection of data.
- *Data Transformation*: In this stage, the selected data is changed into accurate form for the procedure of data mining.
- *Data Mining*: This is the important step in which the techniques used to extract the pattern is clever.
- *Pattern Evaluation*: In this step severely needed patterns represent acquaintances based on measures parameters.
- *Knowledge Representation*: This is the final step in which knowledge is visually represented to the user. Knowledge representation use visualization techniques to help in understanding of user and taking the output of the KDD**.**

1.1.3 Need of Data Mining

Data Mining has found applications in various fields. The reason behind is such wide range applications is the growth in data. The data is rising due to various reasons some of them are described below:

- *Due to Usage of Cards:* Due to the growing trend of usage of credit cards etc. there is growth in data. Considering the number of transactions in a day, it may amount to billion in a year.
- *Due to Web:* There is exponential growth in web data due to the development of e-commerce. Such data contains the information about the website visitors.
- *Due to Increase in Data Storage Capacity:* There is increase in storage space in last few years. This increase is in units of zettabytes. Such large storage can be used to execute data intensive applications.
- *Due to Reduction in Processing Cost:* There is reduction in the hardware cost with the increase in performance of the hardware in the last 30 years. As a result intensive applications are being carried out that were not possible in the previous times. Data mining applications are still an expensive matter.
- *Other Sources:* There are many other sources of data generated from transactions. These are:

- Banking
- Basic Utilities
- Shopping
- Telephone
- Medical

1.1.4 Data Mining System

Classification of data mining system is as follows:

- *Classification According to the Type of Data Source Mined*: In this type, data mining system is defined according to the data to be managed or handled such as spatial, Multimedia data etc.
- *Classification According to the Data Model*: In this type, data mining system defined according to the data to be managed or handled such as relational and object oriented database.
- *Classification According to the kind of Knowledge Discovered*: In this type, the data Ming system is defined according to the functionality of the knowledge established like clustering and classification.
- *Classification According to the Mining Techniques to be used*: In this type, the data mining system is defined according to the different type of approaches such as machine learning, neural network, and genetic algorithm etc.of data mining.

The steps of typical data mining system are shown in figure 1.3:

- *Requirement Analysis*: It is the most important phase in the process. There is need to clearly define the problem as the technique of data mining and data used depends on the goal.
- *Selection and Collection of Data*: In this phase database required is searched for. Most of the data is available in the organization warehouse.
- *Preparation of Data*: If data warehouse is available then preparation of data is not required as most of it is already been done. If warehouse is not available then most of the effort goes in data preparation as there exist many problems like need to identify the missing data, conflicts in data etc.

- *Validation*: Once the data is available the data is explored using a number of techniques and the results are evaluated. This is an iterative process.
- *Implementation, Evaluation and Monitoring*: After selection of model, it is implemented. It involves generation of reports, explanation and visualization of results. Best technique is chosen by checking the efficiency and accuracy of results. Tools used by managers are monitored and regularly evaluated.
- *Visualization of Results*: Results of the data mining are explained to the decision makers. Most of the data mining tools have modules for visualization. They communicate the results to the managers.

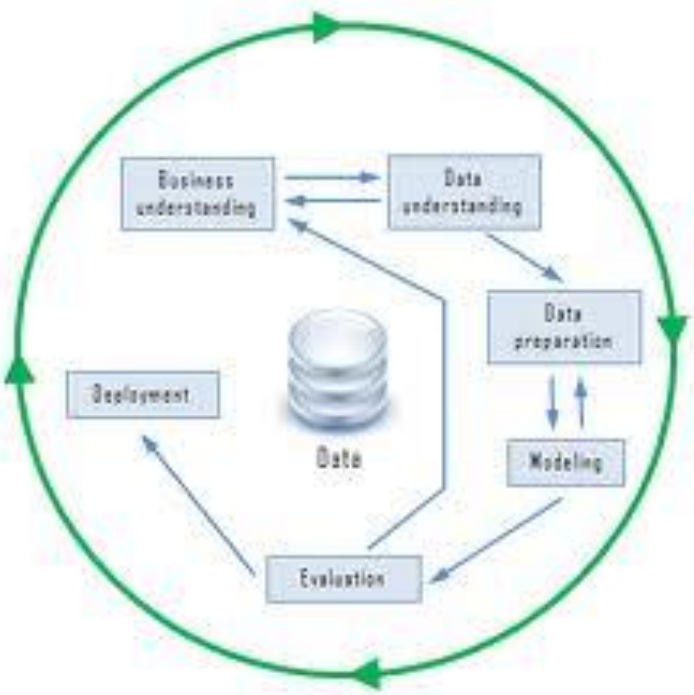

Figure 1.3: Steps of Data Mining Process [31]

1.1.5 Data Mining Applications

There is wide variety of applications. Some of them are:

- *Profiling of Customers*: The process of data mining may use the suitable information related to customers in order to identify their decisions while purchasing. Profiling helps in the identification of privileged customers.
- *Promotion of Web sites*: Data mining helps in cross-selling, i.e. a customer on web may be suggested with the items he/she may wish to buy by using the database which contains the items the customer might have ordered in previous times.

- *Fraud Detection*: This is done by identifying irregularity between the units that are similar. This done during auditing.
- *Prediction*: The techniques of data mining can be used for predicting and analysis of sales. This is done by selecting some or all attributes in the database and then using them to foretell the other variables which may be of interest.

1.2 Association Rules

It is one of the data mining techniques. The aim is to find out which items are frequently purchased together so that they are arranged accordingly on the shelves of the store. This information can also be used in cross selling. It consists of if/then statements that are used to find out the relationship between the data, stored in warehouses or other repositories, which may otherwise seem unrelated. For example if a person buys a new car he is most likely to get its insurance done. Data is analysed for finding out frequent patterns to form association rules. Then the parameters such as support and confidence are used to predict the relationships that are important. The support and confidence are defined as:

- *Support*: It is the number of times an item appears in the database.

  $Sup = \Pr(X \cup Y)$

- *Confidence*: It defines the number of times an item appears provided the other has already occurred.

  $Conf = \Pr(Y \mid X)$

Association rule consists of two parts:

- *Antecedent*: It is the 'if' part of the rule. It indicates the item that is found in the database.
- *Consequent*: It is the 'then' part of the rule. It indicates the item that is available when combined with the antecedent.

Association rules are used for the identification of rules in the database. They are introduced in [1] to find out the relationship between the large dataset of transactions.

For example the rule {Butter, Jam} →Bread, which means that if a person buys butter and jam he is also likely to buy bread. This helps in making decisions about the promotions and also about how things are placed in the stores. The order of items in the transactions or across the transactions does not matter for association rules.

From [1] the definition of association rules is given as:

Let I = $\{i_1,i_2,i_3.....i_n\}$ be the set of n items and DS = $\{T_1,T_2,.....T_k\}$ be the set of m transactions where each transaction in DS has got a new transaction id. The rule derived is of the form A => B where A, B $\subseteq$ I and A $\cap$ B = Ø.

Here A is called antecedent and B is called consequent.

1.2.1 Application of Association Rules

Association rules are used to predict the behaviour of the customers. These are used for market basket analysis. Also other areas of association rules include Web usage mining, Intrusion detection, Continuous production, and Bioinformatics. There are many algorithms for association rule mining. Apriori is traditional algorithm used for rule mining.

1.3 Apriori Algorithm

Apriori algorithm is the mostly used algorithm to extract frequent sets and find association rules in the transactional database. It begins by identifying the single frequent items and then proceeds to combine the items to form larger item-sets as long as item-sets exist in the database. Thus it is called as Bottom Up approach. The frequent sets formed are used to discover the association rules from a large database. The main aim of the data mining process is to discover from a dataset and then convert it into a form that is understandable and can be reused further.

Apriori algorithm uses level wise search where item-sets of size *k* are used to form item-sets of size *k+1*.

Finding out the frequent item-sets basically involves two steps [5]:

- *Join Operation*: In order to frequent set in pass k denoted by L_k, candidate set, denoted by C_k, is formed by joining L_{k-1} with itself.
- *Prune Operation*: The count of each subset of C_k is calculated in order to find the frequent set since all the members of C_k may not be frequent. Thus all the members with count less than support value are removed. Rest of the members form the frequent set. Also if any subset of C_k of size k-1is not present in L_{k-1} then it is not a frequent candidate. Thus it is removed from C_k.

The various steps of Apriori algorithm are shown in figure 1.4

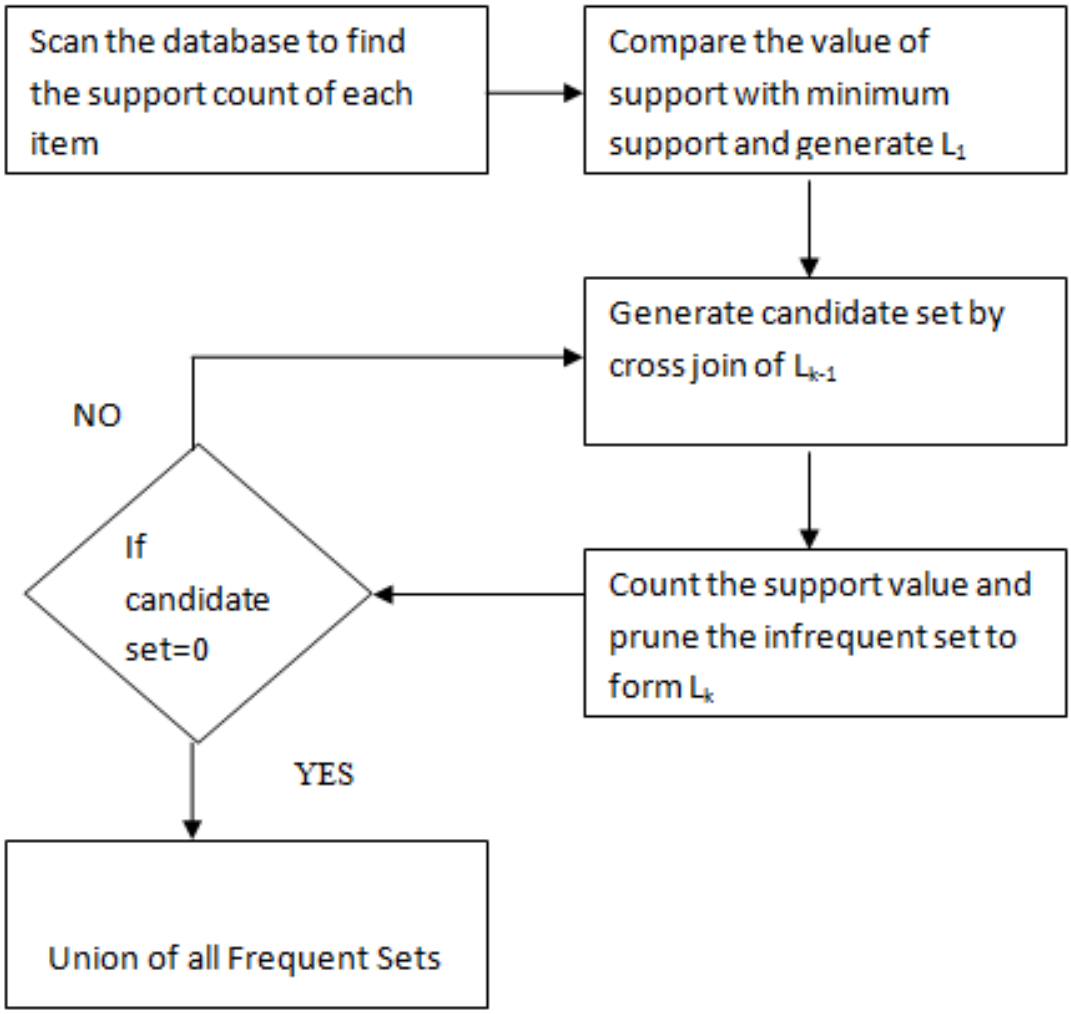

Figure 1.4: Illustration of Apriori Algorithm

1.3.1 Advantages of Apriori Algorithm

- It makes use of a large item-set.
- It is easy to use
- It is easy to implement
- It does not generate too many candidate item-sets

1.3.2 Disadvantages of Apriori Algorithm

- Requires too many database scans
- Consumes large amount of time
- Generates redundant item-sets

1.3.3 Basic Terminology

The basic terms used in the Apriori algorithm are:

- *Item*: It is a single article in the dataset
- *Transaction*: It is a set of items
- *K-ItemSet*: It is the item-set of size k i.e. the set containing k items.
- *Support*: It is also defined as the number of transactions containing item x.

$$Support = \frac{(X \cup Y).count}{n}$$

- *Frequent Set*: The set of items satisfying the minimum support value.

1.4 Thesis Outline

Thesis is organized in the following manner:

Chapter 2: This chapter contains the survey of the related work done in the field of Apriori algorithm. It discusses the various shortcomings of the algorithms and the approaches adopted to improve their efficiency.

Chapter 3: This chapter explains the problem which is the topic of research.

Chapter 4: This chapter contains the implementation work. It discusses in detail the solution given to solve the problem stated earlier.

Chapter 5: This chapter contains the comparative analysis of the proposed approach with the existing one.

Chapter 6: This chapter contains the conclusion arrived at.

Chapter 2

Literature Survey

Lots of work has been done in Apriori algorithm. There are many approaches introduced for improving the Apriori algorithm. Some of them are discussed below:

2.1 General

In [3], they proposed high dimension Apriori algorithm. Unnecessary generated sub item-sets are removed by this method. As a result higher efficiency of mining can be obtained when data dimension is high as compared to original Apriori algorithm.

In [4], a new algorithm is proposed that reduces number of times the database is scanned. This algorithm is IApriori algorithm. Also the procedure of joining frequent item-sets is optimized. This results in the reduction in the size of candidate item-set. This newly proposed algorithm performs better than the classical Apriori algorithm.

An Apriori algorithm is can also be improved with improvement of pruning operation. In [8], the same approach is adopted with the introduction of count based method in an improved algorithm named IAA. According to this if L is an item-set of dimension *k* and is a frequent set then its subsets of dimension *k-1* will also be frequent. Since each of the two subsets will generate L only once, the total time will be L_k^2. If the count of each subset of L_k is less than L_k^2 then it is considered to be infrequent. Other deficiency of Apriori algorithm is scanning the database multiple times. This is also improved in IAA. In this data is stored as <Itemset,TID> [30] which is similar to WDPA with only difference is that each candidate item-set is counted only once. Thus candidate sets are generated using count occurrence step that is based on records that were produced in prune operation. The frequent item-sets and association rules are generated synchronously the advantage of synchronized generation is that operations can be stopped before all large frequent item-sets are found in case the existing results are not according the expectation. This deviation may arise if minimum support is inappropriate or there is restriction on the number of association rules. The operation

can be performed in parallel and distributed mode and also on master and idle nodes thus saving the operation time of association rule mining. The algorithm has been implemented using C-R problem where objective is to find association rules of type $c \Rightarrow r$. Here c ∈ C and r ∈ R, C is the set of condition items and R is the set of result items.

Libing Wu. et. al presented an improved algorithm, based on Apriori Algorithm, to enhance the efficiency of mining. The concept of interest items was introduced [10]. Interest item is a group of items which are of user interest. Also frequency threshold was introduced which is defined as minimum number in association rules. It is the product of minimum support count and minimum confidence value. The transactional database is scanned and all the interested items in it are recorded. Then the algorithm traverses the interest items and finds out all the subsets of interested items. Later the association rules are formed from the subset of interested items. These are formed using minimum support and confidence values. Those items whose count is less than frequency threshold are deleted from the set of interested items and those greater than threshold value are maintained. Next if the item-sets are different and their union belongs to the set containing subset of interest items then the value of the following parameter is compared with minimum confidence as:

$$if\left(\frac{(L_i.Items \cup L_k.Items).count}{L_i.Items.count} \geq \min Conf\right) \quad (1)$$

If value in equation (1) is true then association rule is formed. This algorithm performs better when the database is large and interested items are small in number. The flowchart explaining the steps is given in figure 2.1.

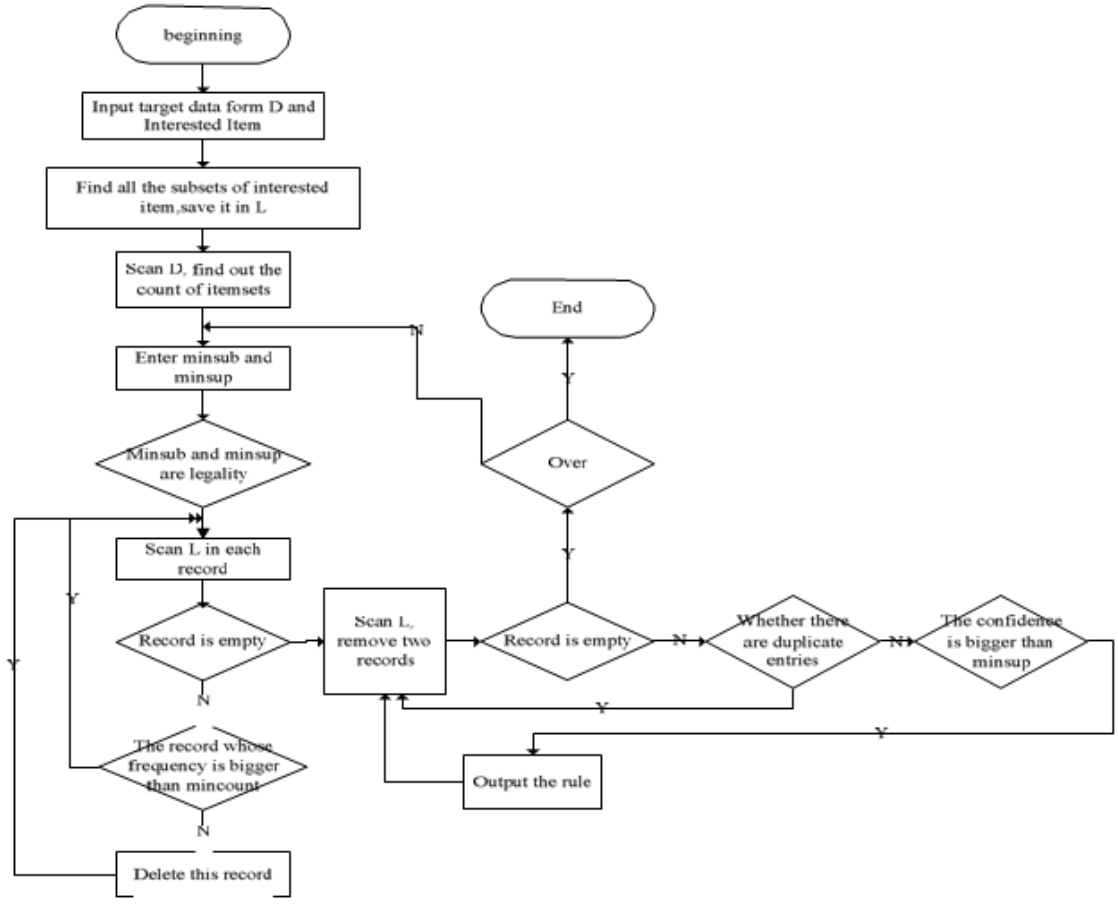

Fig 2.1: Illustration of the Improved Apriori Algorithm [10]

Parvinder S. Sandhu et. al used weight and utility attributes to determine the association rules. They discussed that classical Apriori algorithm does not consider the quantative information associated with the attributes which leads to rules that are though frequent but have less weight age. It might be the case that some of the attributes which are less frequent are more frequently occurring. These attributes might be important from the business point of view. For this purpose the weight gains (W) and utility gains (U) are added to the association rules [11]. Utility gains add profit margins to the attributes. They used Apriori algorithm to generate rules. Then UW score is calculated for each association rule using W-gain and U-gain. Those rules having UW score above the threshold set are selected.

A probabilistic approach to Apriori algorithm is applied. It involves probability and statistics. Recursive median is used for the calculation of dispersion in the list of containing transaction corresponding to each item-set. In the first phase of the

algorithm which is called as classification phase, the item-sets from the previous pass are divided into HIGH and MEDIUM groups considering the bimodality as the basis of classification. In the next phase k-candidate sets are generated from L_{k-1} using lexicographic association. Then the next phase involves two steps: firstly it identifies all the candidates in the MEDIUM group that satisfy the minimum support. Secondly: these candidate sets are removed from the C_k^{New}. This continues until no new large item-sets are formed from the previous pass. The rules created are same as those of Apriori but time taken in this new approach [14] is less.

Gurudutta Verma et. al proposed an algorithm that uses the dataset containing transactions in a transposed from. The transposition of the dataset is carried out with the help of Parallel transposition algorithm [22]. In the transposition of dataset diagonal elements remain unaffected. Those above the diagonal occupy the positions, below the diagonal, that are symmetrical and vice versa. The candidate sets are then generated using L_{k-1} and later pruning is done to find out the large item-sets. The experimental results prove that the proposed algorithm is very fast and efficient in finding out the association rules. There is reduction in the I/O overhead and the support value is retrieved faster than Apriori algorithm.

The efficiency of strong association rules is limited due to problems with Apriori algorithm. Since there are only two threshold values, support and confidence, some rules obtained were of no value. Taking this into consideration a third threshold value named interest measure is introduced [23] which helps in optimizing an Apriori algorithm. The interest measure was defined on the basis of support and confidence. Minimum value of the interest measure called as the threshold value is also set. According to the proposed approach if the interest measure is greater than or equal to the minimum interest value then only the association rule is interesting otherwise it is not meaningful. Experimental results showed that with the addition of interest measure unnecessary rules or errors were removed. Thus the algorithm is optimized.

In [27], they proposed a model which involves division of the task into two phases. The first phase the consistency of the database is improved by the resolutions of the problems like duplication of the data or bad data. Second phase involves identification of the data patterns that are frequent. This approach has an advantage that in order to find the frequent data patterns the whole database is not scanned instead the patterns

are found from the data set that is filtered. The cleansing algorithm is used to remove the impurities if they are higher in number. Pruning is generally not done if it increases the impurities. Groups are defined with the help of partitioning. Only few of the partitioned values are defined on the basis of rightness of the attributes. It is then followed by mining and rules are generated. This approach is more time efficient as compared to older Apriori algorithm. The new algorithm removes the item-sets that are not frequent from each transaction. Later they are added to the main memory thus reducing the number of items in each transaction. The memory constraint problem is resolved using this approach. The dataset is partitioned into horizontal segments from which infrequent items are removed and rest are inserted into the main memory. After insertion all the memory entries are written to the temp file which generates the support count and then support is sent to all the sites and finally frequent item-sets are calculated on each site.

In [17], an enhanced algorithm is designed and discussed which puts forward only those items that the user is interested in. These items are called seed items. The database is then scanned and all the items which are in the same transaction with the seed item are added as a part of item-set. The count structure is used to keep a record of the state of items so that item is not repeatedly visited every time database is scanned. If count structure is updated after every scan of the database then keep scanning otherwise scanning is stopped and item-set of user interest are recorded. The support value of each item is calculated taking importance of item-sets into consideration. The importance is measured by using weight as an indicator which is calculated using price of the item as a parameter. The algorithm is efficient as the data is compressed which increases the speed of the operation and the computational efficiency. The generation of frequent item-sets is faster and memory is also saved.

In [18], an Apriori algorithm is discussed that consist of three areas of improvement. Firstly the reduction in number of judgements, secondly: reduction in the number of candidate frequent item-sets and lastly the database optimization. It is assumed that the item-sets are ordered. Thus if two item-sets cannot be connected then all the item-sets after these two items cannot satisfy the condition to form a connection. Thus judgements are reduced. Secondly all the item-sets with frequency less than k-1are found and saved in I, which is the set of items, and then those frequent item-set that

contain subset of I are removed. Lastly database optimization is achieved by maintain a delete tag for all those transactions which do not contain $C_{k.}$ These items are not considered further as the algorithm proceeds.

Sometimes it may occur that a transaction in the *k+1* pass does not contain frequent *k* item-set. Identifications of such transactions are necessary so that these transactions are no read and processor time is saved. Proposed approach in [7] identifies the transactions that contain the frequent set and checks whether that transaction should be scanned further or not. This is done in first scan. In order to achieve the efficiency the data is distributed among parallel processors. This distribution is equal. Each processor is utilized to the maximum in order to maximize the efficiency.

2.2 Improved Apriori with Optimized Database

In [23], an improved algorithm which involves the scanning of the entire database only twice is discussed. In this approach there are two steps involved: firstly the database is divided into different parts. The database is scanned once and the frequent sets of each division are recorded. In the second scan the support count of each set is known and finally global frequent item-sets are found. Secondly: the candidate item-sets are added when the division of the dataset is being done. The candidate sets are added at the starting point so that candidate sets are decided before the database scan. This is an efficient approach as it reduces the count of candidate item-sets and also the storage space is saved.

In [25], they propose a new improved Apriori algorithm which uses the compressed database. The database is compressed by deleting those transactions which does not contain the items that are of no interest to the user. This reduces the database size which as a result speeds up the process due to the reduction in the number of scans and thus reducing the time. This approach is useful where there is need of classification and selection of features.

The optimization of Apriori algorithm is done by reducing the transaction records in the database. The performance is optimized with the improved algorithm discussed in [5]. Optimization is achieved with the reduction of the candidate item-set and further by reducing I/O spending. Candidate item-sets are reduced by following the property

that if T is an item-set of dimension *k* and L_{k-1} is a frequent item-set of dimension *k-1*. If all the subsets of dimension *k-1* contains T and their number is less than *k* then T is not a frequent item-set of dimension *k*. Thus the algorithm only compares the count of each element of frequent set of dimension *k-1* with that of each element of candidate set C_k of dimension *k*. If the count of element of C_k is less than *k* then it is deleted otherwise it is maintained. Secondly the I/O spending is reduced by removing the unnecessary transactions. It is obvious that those items are not part of L_{k-1} will not appear in L_k. Hence they are deleted from the database. Also the transactions that have the items less than *k* should be deleted. Hence this reduces the size of the database and speeds up the implementation.

2.3 Hash Based Approach

In [2], an algorithm is discussed that improves the shortcomings of an Apriori algorithm. In the newly proposed algorithm candidate item-sets are stored in a hash tree. The experimental results prove that the new improved algorithm is superior to the traditional Apriori algorithm. The algorithm is efficient and has wider applications in data mining.

R Chang et. al proposed a new strategy named Apriori-Improve algorithm which is faster and efficient than Apriori algorithm. This technique [19] directly generates L_2 in one scan. C_1, L_1 and C_2 are not generated in between. This is done with the help of perfect hash function. The newly proposed algorithm scans the database only once in order to generate and L_1 and L_2 which improves the efficiency of the program. In this they have used hash table in place of hash tree as it reduces the cost of searching. The data is represented in the horizontal form and storage strategy is also effectively used as it saves space and time. This method is better over Apriori as it is simple and easy in implementation.

Huiying Wang et. al discuss an improved algorithm for the mining of association rules. Candidate item-set is scanned only once for the generation of C_k. It is checked whether the item contained in L_{k-1} is a subset of item contained in C_k. If it is there then the number of occurrence is counted [20]. In case the number is less than the value of k then that item is deleted from C_k. C_k is stored using hash tree due to which candidate item-sets are quickly counted. Every node of the hash tree contains the hash table

which points to another node. The leaf nodes contain the item-sets. The depth of the tree in layer *d* is *d+1*. This algorithm performs better than Apriori algorithm as the pruning step takes $p*|L_k\text{-}1|$ in the improved algorithm whereas it takes $\sum^{|Ck|}_{i=1}\sum^{m}_{j=1}(|L_{k-1}|)$ in the Apriori algorithm.

2.4 Hybrid Approach

In [1], a new algorithm named AprioriHybrid based on the two new proposed algorithms is proposed. They combined the best features of two algorithms to design a new hybrid algorithm. These proposed algorithms perform better than the existing algorithms. However when the best features of the two proposed algorithms are combined, a new hybrid algorithm is formed which has better properties than the algorithms when taken up separately. It shows a decrease in the execution time with the increase in number of items. Also the execution time shows linear increase with increase in the number of transactions.

2.5 Application of Improved Apriori Algorithm in Various Fields

In [24], the application of improved Apriori algorithm in case of treatment of influenza by old Chinese medicine is discussed. Apriori algorithm was able to extract association rules between symptoms and syndromes only when the sample was small. For large samples an improved algorithm was proposed that mines the case of Chinese medicine to derive association rules.

In [26], practical application of Apriori algorithm in enterprise decision making is discussed. The correlation between different properties is found out by analyzing the attribute values. Five candidate factors were chosen that could affect the strategies in decision making in different areas. The first step in the process involved sorting of the data on the basis of practice survey. Then the association between candidate factors is found out and finally conclusion is drawn to show the direct influence of enterprise innovation on corporate reputation.

In [9], an application of data mining technology into enterprise stock management is discussed. They have described the Apriori algorithm along with the theory of data mining in detail. The implementation part is done in c# using .net framework. The

advantages and disadvantages of the algorithm are discussed in detail and also the methods of improvement are pointed out.

Lin Lu et. al, designed and implemented the OOO algorithm after discussing the shortcomings of Apriori algorithm. OOO scans the database only once resulting in the reduction of cost of accessing the database. Its advantage lies in the fact that it does not produce the candidate items having support degree equal to 0. The complexity of OOO algorithm is O(k) whereas that of Apriori is $O(k^3)$ where k is the size of item-set. The application of OOO in supermarket is also discussed in [12] as it results in the reduction of time and space.

Jugendra Dongre et. al discusses the application of Apriori algorithm in e-commerce. The item-sets that are frequently purchased are used to analyze the tendency of the customer. It is successfully utilised in large number of industry applications. The profitable customers are targeted on the basis of reward points [29]. The retail industry gains and thus is more profitable in the market competition. The example here discussed is that of consumer database. The database contains the item-sets that are purchased from the bakery shop by the customer with its number. It is shown experimentally that with different values of confidence, the association rules formed are different. Also higher the value of minimum confidence, more are the rules that are accurately filtered.

Comparison of Apriori and Bottom up Construction (BUC) algorithm is discussed in [15]. Although BUC algorithm is based on Apriori algorithm but still its performance is lower than Apriori when both are used to find out the frequent item-sets. The reason for such bad performance lies in the fact that BUC uses recursion due to which space and time complexity increases. Hence a new algorithm Dimension Group Parallel (DGP) algorithm which does not use recursion. As the name suggests this algorithm avoids the repeated division in BUC by dividing the dimension. This algorithm is used for multidimensional database. Using DGP the number of item-sets is reduced in dimension due to which database scanning is reduced. Also only the specific attribute needs to be scanned avoiding the unnecessary scan of the whole database. The improved algorithm is more flexible when it comes to the choice of an attribute. Its application areas include telecom and also it is used to analyse the quality of ADSL line.

In [13], the application of Apriori and its improved algorithm in database cutting is discussed. Cutting data mining is a method that increases the efficiency and helps in cutting decisions. It also discovers the knowledge that is hidden in the cutting database. Although a large support value is required due to lack of experience but still the support value of cutting data is not very large. Both the minimum and maximum value of both the support and confidence are set such that important rules are not missed out and those that are not required can be missed. The basic steps of improved Apriori algorithm are same as those of Apriori algorithm except the last in which the rule with confidence value greater than maximum confidence is chosen to be the final rule. The conclusions derived from the results are that with the increase in hardness of the work piece, there is decrease in the cutting speed and with the increase in diameter of the cutter, the maximum feed rate of alloy steel increases.

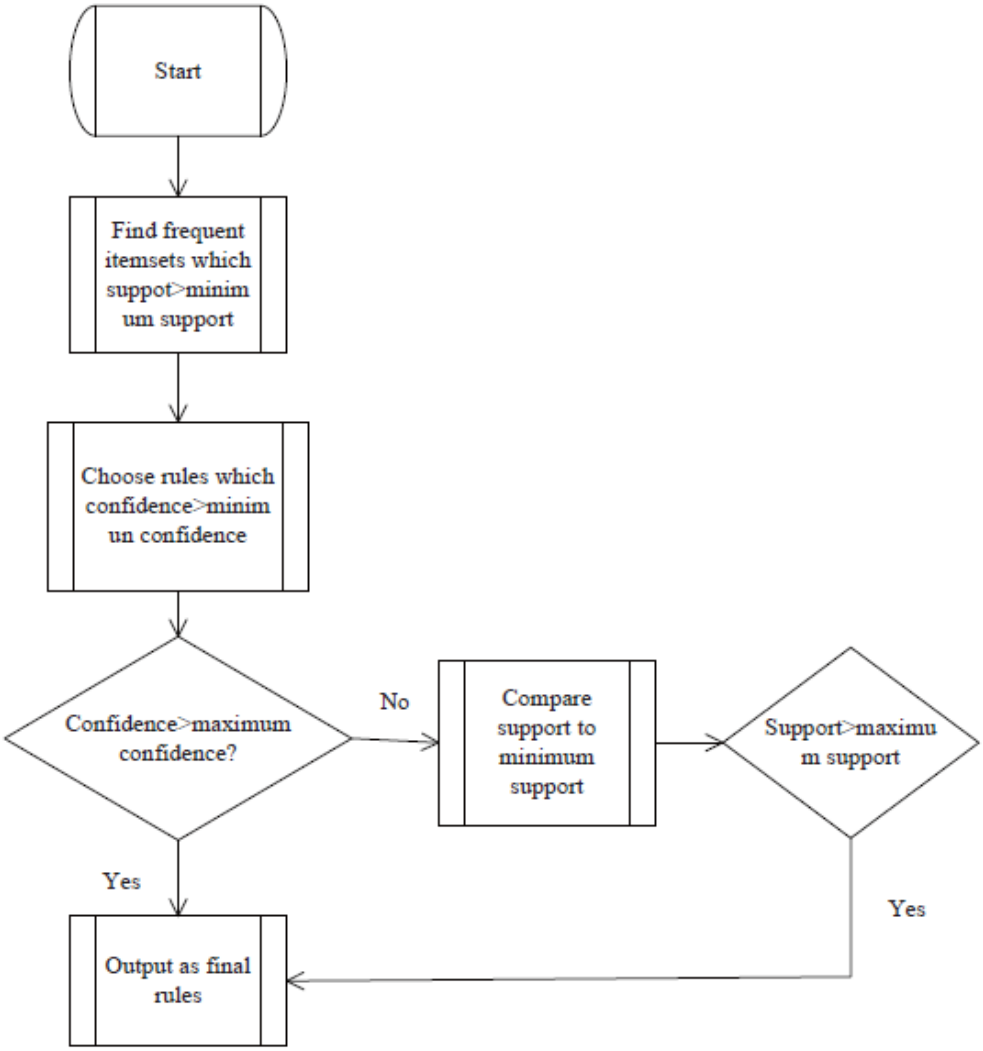

Fig 2.2: Illustration of the Improved Apriori Algorithm [13]

In supermarkets the sales vary according to the season. With the sales vary profits, trends in consumption and many other factors [16]. Yu Shaoqian proposed an algorithm with weighted support and confidence. In the algorithm commodities are divided into big categories. The weights are set for each category of commodity and those that do not satisfy the weight support threshold are pruned. Pruning ensures those rules which do not have any meaning are filtered out. Weights are set according to categories as it represents relation between the goods. Association rules are then found out that meet the requirements and thus marketing strategy is made effective.

In book recommender system when the user browses the book with its ID, he should get only those recommendations which are associated with that book. This increases the efficiency of the system. Thus during scanning only those transactions that are related to Id of book should be scanned and all other transactions that are not related can be ignored. This is done in [28]. An algorithm based on older Apriori algorithm is discussed which during the association rule mining scans only those transactions that have the similar features. The system should recommend the books on the basis of reader type. This is achieved with the addition of transaction feature. The type of reader is used as feature of transaction. Frequent item-sets are generated and then association rules are formed which have the deciding factor as that given as input. The experiment was performed using 2000 borrowing records of 600 readers with 10 books involved. It was observed that with the increase in support value time consumption of improved algorithm decreases. Time taken by Apriori algorithm is more due to the complexity involved in computation. However all strong association rules are not produced by improved Apriori algorithm due to incomplete data.

In [6], an improved Apriori algorithm is discussed that gives an early warning in case of failure of equipment. The improved algorithm aims at reducing number of database scans and candidate item-sets which have the chances of becoming frequent sets. The items that do not satisfy the minimum support do not form the frequent set and those containing these infrequent items are also removed from the original dataset. By attaining these they have proved that it is better than Apriori algorithm.

Chapter 3

Problem Statement

This chapter contains the problem which is discussed and improved in the research work.

Apriori algorithm is the traditional algorithm for finding frequent patterns. It is widely used in many areas to analyse various types of data e.g. data on diabetic patients, crimes concerning women, market data, and health care data. Although widely used, it suffers from drawback of large database scans. To calculate the support value of candidate subset, it scans the original dataset. This involves scanning large number of transactions and items, which takes large amount of time. Due to this, efficiency is degraded. This problem has been solved by reducing the number of transactions to be scanned. Also the threshold parameter is changed. This has resulted in the reduction of time of execution. As a result efficiency is improved.

Chapter 4

Details of the Research Work

In this chapter the solution to the problem discussed in the previous chapter is discussed in detail.

4.1 Introduction

The proposed methodology is based on the Apriori algorithm. The main focus is to improve the Apriori algorithm which is achieved by adopting two strategies:

- There is no need of scanning the original dataset every time to calculate the support value of each item. The scanning is carried out only once. The support value is calculated by counting the Tid set which is obtained by intersecting the transactions of each item. This saves time to a greater extent and efficiency of the Apriori algorithm is improved.
- At every step threshold value is changed. It is calculated by adding support values of all the items in a candidate item set and dividing the same by the number of unique transactions of all the items in that item set.

4.2 Data Set Used

The dataset used for implementing the proposed methodology is Extended Bakery Dataset [32]. It contains the receipt id, id of the item sold in that transaction and the quantity of the item sold out. For our purpose only the receipt id and the item id has been taken. Receipt id can be called as Transaction id or Tid. The first column represents the transaction ids (Tids) and second column represents the ids of items (Iids) that are sold in that transaction.

4.3 Technology Used

- Python 2.7.6

- Apache server 2.2

4.4 Work Explanation

The basic steps followed to improve the Apriori algorithm are as followed:

4.4.1 Dataset

The dataset as discussed above contains the transaction id and item ids corresponding to each transaction. For our illustration purpose we have taken a small dataset of transactions containing only receipt id, which is also called as transaction id, and item id. The sample of the dataset is given in the table 4.1.

Table 4.1: Sample of the Data Set (DS)

Tid	**Iids**
T_1	I_1,I_3,I_7
T_2	I_2,I_3,I_7
T_3	I_1,I_2,I_3
T_4	I_2,I_3
T_5	I_2,I_3,I_4,I_6
T_6	I_2,I_3
T_7	I_1,I_2,I_3,I_4,I_6
T_8	I_2,I_3,I_4,I_6
T_9	I_1
T_{10}	I_1,I_3

4.4.2 Transposition of the Data Set

The required dataset for the algorithm is (Iid, Support, Tid) because the candidate sets are formed by combining items. The structure of the original dataset is (Tid, Iid). In order to obtain the required dataset, transposition of the original dataset is carried out. The transposed dataset is shown in the table 4.2

Table 4.2: Transposed Data Set $(DS)^T$

Iid	S_i	Tids
I_1	5	T_1,T_3,T_7,T_9,T_{10}
I_2	7	$T_2,T_3,T_4,T_5,T_6,T_7,T_8$
I_3	9	$T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,T_{10}$
I_4	3	T_5,T_7,T_8
I_5	1	T_5
I_6	2	T_7,T_8
I_7	2	T_1,T_2

4.4.3 Large Item-Set

First of all large item-set of size 1 is formed. This is also called as frequent set and is denoted by L_1. For obtaining L_1 transposed dataset is scanned. Then the number of transactions corresponding to each item is counted. This number is the support count of each item and is denoted by S_i. This support value is compared with the minimum support value which is explained later in this section. This threshold value is updated in every pass. Those items that have the count value greater than the minimum support value forms the large item-set of size 1. Also the Tids corresponding to each Iid are written. L_1 formed is further used to form candidate item sets in subsequent pass which eliminates the need to scan the original dataset.

4.4.4 Candidate Set

The candidate set contains all the sub item sets which are capable of forming a frequent set. It is denoted by C_k. It is formed by joining the large set formed in the previous pass with itself. Candidate set of size 2 is formed by self join of L_1. Here all possible combination of items of size 2 is formed and the set is denoted by C_2.

In the subsequent passes the candidate sets are formed from the frequent sets formed in the previous pass by using the following Apriori property:

If $(I_1[1]=I_2[1]) \wedge (I_1[2]=I_2[2]) \wedge \ldots \wedge (I_1[k-2]=I_2[k-2]) \wedge (I_1[k-1] < I_2[k-1])$ then $c=I_1 \infty I_2$

which means that while forming combination of items only those subsets of frequent item-set are joined that have all but last element same and the last element of first subset is not equal to that of second subset.

During the candidate set generation the transaction set of the items that are combined is intersected. The intersection is done to calculate the support value of every item set. This eliminates the need to scan the original dataset in order to find the frequency of each combination of items. As a result the time is reduced to a large extent.

4.4.5 Frequent Set

The frequent set contains the items that are frequent which means those items that have their support greater than threshold value. The large item-set of size 1 is formed from the transposed dataset by considering those items having value no less than minimum support value. Then in every pass first candidate is formed form which frequent set is formed denoted by $L_{k.}$ The items that satisfy the minimum support value are copied into the frequent set along with the support value and the intersection of transactions.

4.4.6 Threshold Value

In traditional Apriori algorithm minimum support is taken as input from the user. If the user specified support value is small, then the time to form the frequent sets is more. This is because there will be many elements of candidate item sets that will satisfy the minimum support and form a frequent set. This frequent set is then used to form candidate sets in the next pass. Due to large number of entries in the frequent set, there will be many entries in the candidate set which are unnecessarily formed and may not satisfy the threshold value later. Rather than taking the minimum support value also called as threshold value as an input it is updated at every pass. It is the mean of all support counts at every pass. It is calculated as:

$$\text{Support} = \frac{\text{Sum of support count of all items}}{\text{Number of unique transactions}}$$

Mean value is the optimized value. It will increase or decrease depending upon the support count and number of transactions that appear after intersection. This removes

the unnecessary items formed in the candidate set very early. As a result time is greatly improved.

4.4.7 Complete Process

The complete illustration of the proposed approach is discussed here. The large database is scanned only once. The number of transactions to be scanned is reduced at every step which as a result speeds up the process. Support value is calculated from the transactions and not from the original dataset. This has reduced the great load of I/O and overall time.

The improved algorithm is applied on transposed dataset.

Pass 1: For *k = 1*

Calculate the minimum support value

$$\text{Support } (S_{min}) = \frac{5+7+9+3+1+2+2}{10} = 2.9$$

Based on S_{min}, the infrequent items are pruned from the $(DS)^T$. The items from the $(DS)^T$ which have $S_i > S_{min}$ form a frequent set shown in table 4.3.

Table 4.3: Large 1 Item-Set

Iid	**S_i**	**Tids**
I_1	5	T_1,T_3,T_7,T_9,T_{10}
I_2	7	$T_2,T_3,T_4,T_5,T_6,T_7,T_8$
I_3	9	$T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,T_{10}$
I_4	3	T_5,T_7,T_8

Pass 2: For *k =2*

In the next step candidate set of size 2 is formed represented by C_2. The possible combinations are:

$C_2=\{(I_1,I_2),(I_1,I_3),(I_1,I_4),(I_2,I_3),(I_2,I_4),(I_3,I_4)\}$=6 sets.

Also the intersection of Tids corresponding to Ids that are combined is done in order to calculate the support value. This is illustrated in the following tables. They contain the all the transactions obtained intersection and also the number of transactions obtained which is represented by S.

Table 4.4: Intersection of Transactions of I_1 and I_2

Iids	**Tids**
I_1	T_1,T_3,T_7,T_9,T_{10}
I_2	$T_2,T_3,T_4,T_5,T_6,T_7,T_8$
$I_1 \cap I_2$	T_3,T_7
$S(I_1 \cap I_2) = 2$	

Table 4.5: Intersection of Transactions of I_1 and I_3

Iids	**Tids**
I_1	T_1,T_3,T_7,T_9,T_{10}
I_3	$T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,T_{10}$
$I_1 \cap I_3$	T_1,T_3,T_7,T_{10}
$S(I_1 \cap I_3) = 4$	

Table 4.6: Intersection of Transactions of I_1 and I_4

Iids	**Tids**
I_1	T_1,T_3,T_7,T_9,T_{10}
I_4	T_5,T_7,T_8
$I_1 \cap I_4$	T_7
$S(I_1 \cap I_4) = 1$	

Table 4.7: Intersection of Transactions of I_2 and I_3

Iids	**Tids**
I_2	$T_2,T_3,T_4,T_5,T_6,T_7,T_8$
I_3	$T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,T_{10}$
$I_2 \cap I_3$	$T_2,T_3,T_4,T_5,T_6,T_7,T_8$
$S(I_2 \cap I_3) = 7$	

Table 4.8: Intersection of Transactions of I_2 and I_4

Iids	**Tids**
I_2	$T_2,T_3,T_4,T_5,T_6,T_7,T_8$
I_4	T_5,T_7,T_8
$I_2 \cap I_4$	T_5,T_7,T_8
$S(I_2 \cap I_4) = 3$	

Table 4.9: Intersection of Transactions of I_3 and I_4

Iids	**Tids**
I_3	$T_1,T_2,T_3,T_4,T_5,T_6,T_7,T_8,T_{10}$
I_4	T_5,T_7,T_8
$I_3 \cap I_4$	T_5,T_7,T_8
$S(I_3 \cap I_4) = 3$	

Now after finding out intersection of transactions, the candidate set is generated as shown in table 4.10. It contains the items combined, with intersected transactions and the support value. This set has all the possible items that can be a part of the frequent set.

Table 4.10: Candidate Set of Size 2

Iids	**S_i**	**Tids**
I_1,I_2	2	T_3,T_7
I_1,I_3	4	T_1,T_3,T_7,T_{10}
I_1,I_4	1	T_7
I_2,I_3	7	$T_2,T_3,T_4,T_5,T_6,T_7,T_8$
I_2,I_4	3	T_5,T_7,T_8
I_3,I_4	3	T_5,T_7,T_8

After generation of C_2, the minimum support value is calculated as:

$$\text{Support } (S_{min}) = \frac{2+4+1+7+3+3}{9} = 2.2$$

Based on S_{min}, infrequent item sets are pruned from the C_2 to form L_2. After pruning the frequent set formed is shown in table 4.11.

Table 4.11: Frequent Set of Size 2

Iids	**S_i**	**Tids**
I_1,I_3	4	T_1,T_3,T_7,T_{10}
I_2,I_3	7	$T_2,T_3,T_4,T_5,T_6,T_7,T_8$
I_2,I_4	3	T_5,T_7,T_8
I_3,I_4	3	T_5,T_7,T_8

Pass 3: For *k = 3*

In all the passes for k > 2, the candidate set is generated using the Apriori property as already discussed. Following the property only (I_2,I_4) and (I_2,I_4) can be combined. The set generated is shown as:

C_3 : {(I_2,I_3,I_4)}= 1 set

Table 4.12: Intersection of transactions of (I_2,I_3) and (I_2,I_4)

Iids	**Tids**
I_2,I_3	T_2,T_3,T_4,T_5,T_6,T_7,T_8
I_2,I_4	T_5,T_7,T_8
(I_2,I_3)∩(I_2,I_4)	T_5,T_7,T_8
S((I_2,I_3)∩(I_2,I_4)) = 3	

Table 4.13: Candidate set of size 3

Iids	**S_i**	**Tids**
I_2,I_3,I_4	3	T_5,T_7,T_8

Table 4.14: Frequent set of size 3

Iids	**S_i**	**Tids**
I_2,I_3,I_4	3	T_5,T_7,T_8

Pass 4: For *k = 4*

Candidate set cannot be formed because there are no more than one item set in the frequent set in the previous pass that can be combined further. Hence no more candidate set is generated and C_4 is equal to null i.e. C_4 = Ø. Also no more frequent sets can be formed. Thus iteration is stopped here. Final result is the union of all the frequent sets formed in all the passes. It is represented by U_kL_k. The results are shown in the table 4.15.

Table 4.15: Union of all Frequent Sets

Iids	I_1,I_3	I_2,I_3	I_2,I_4	I_3,I_4	I_2,I_3,I_4

4.5 Flowchart of Improved Algorithm

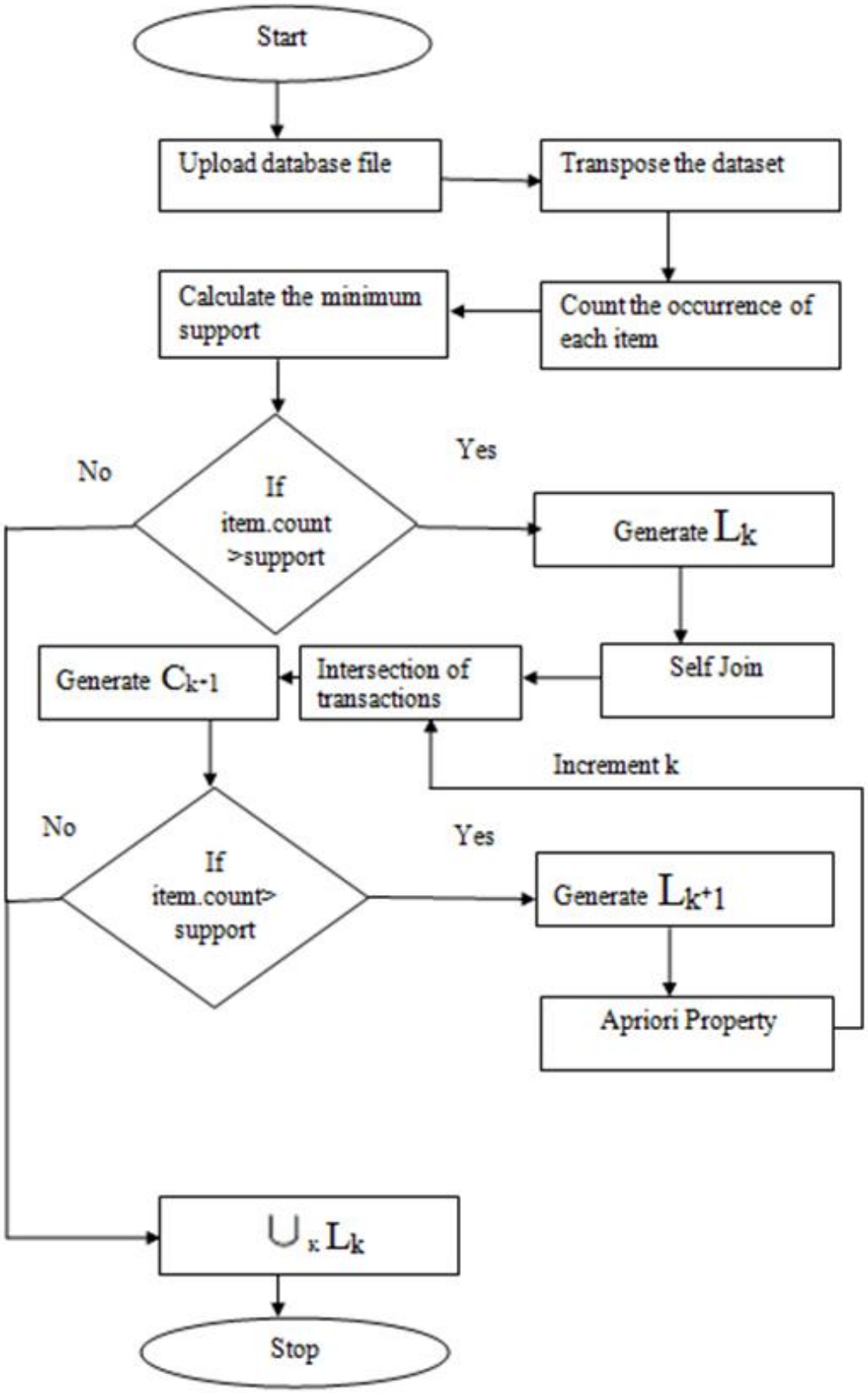

Figure 4.1: Illustration of the Complete Algorithm

Chapter 5

Results and Analysis

This chapter contains the results of the proposed approach followed by comparison of the improved algorithm with the existing one.

In the proposed methodology, the main focus has been to remove the drawbacks of classical Apriori algorithm. In order to achieve this, the two parameters have been worked upon:

- Reduction in the Number of Transactions to be Scanned
- Reduction in the Time of Execution

5.1 Reduction in the Number of Transactions

The number of transactions to be scanned is reduced by taking intersection of the transactions as already discussed. This intersection gives the support value. The comparison of the improved algorithm is done with the Apriori algorithm in terms for number of transactions. In Apriori in every pass the original database is scanned therefore the total transactions to be scanned will be equal to the product of the total number of transactions and number of passes whereas in improved Apriori transactions are reduced at every pass. Only in the first pass, it is equal to the transactions in the original dataset. In the subsequent passes it reduces. Both the algorithms are tested on datasets of different sizes and results are depicted in the table 5.1 and graphically represented in figure 5.1.

Table 5.1: Comparison of Number of Transactions

	Apriori	Improved	Reduction (%)
5k	25000	13983	44
10k	50000	28551	43
20k	120000	58448	51
75k	450000	235350	47

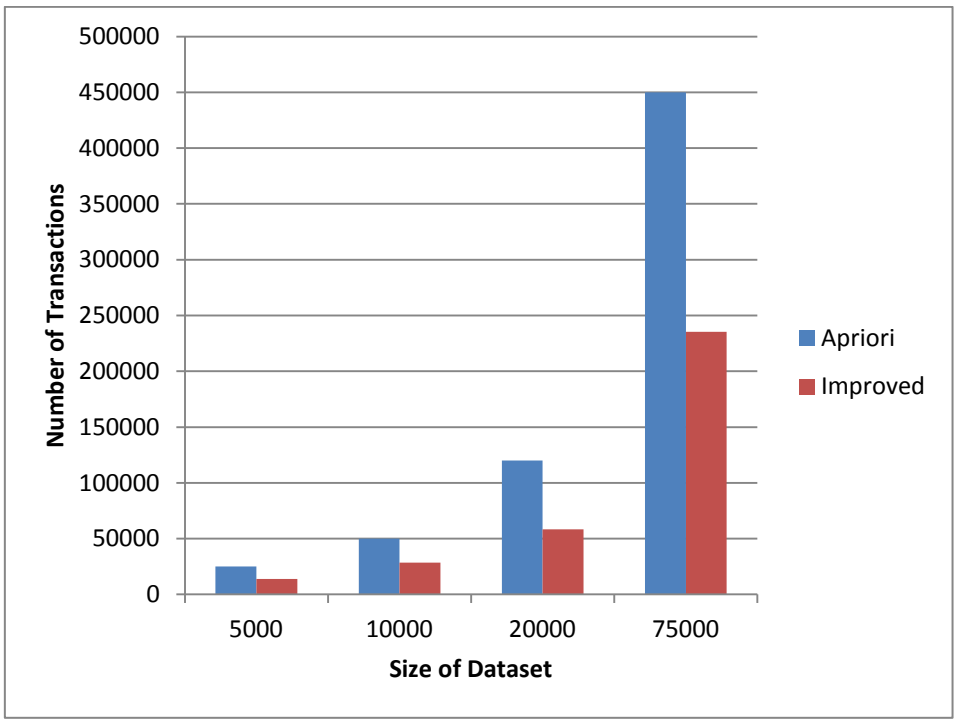

Figure 5.1: Comparison of Number of Transactions

5.2 Reduction in the Time of Execution

In order to reduce the time of execution two things were done: firstly the support value is updated at each pass and secondly as discussed above, the transactions are reduced. The results of reduced time of execution and efficiency obtained are recorded in table 5.2. The time recorded is in seconds. These results are obtained after testing both the Apriori and improved algorithm on datasets of different sizes.

Table 5.2: Comparison of Execution Time in Seconds

	Apriori	**Improved**	**Efficiency (%)**
5k	362	29	91.98
10k	515	67	86.99
20k	918	101	88.98
75k	14329	2436	82.99

Figure 5.2 depicts the graphical representation of the improved execution time over Apriori algorithm. From the figure it is clear that the improved algorithm outperforms Apriori in terms of execution time.

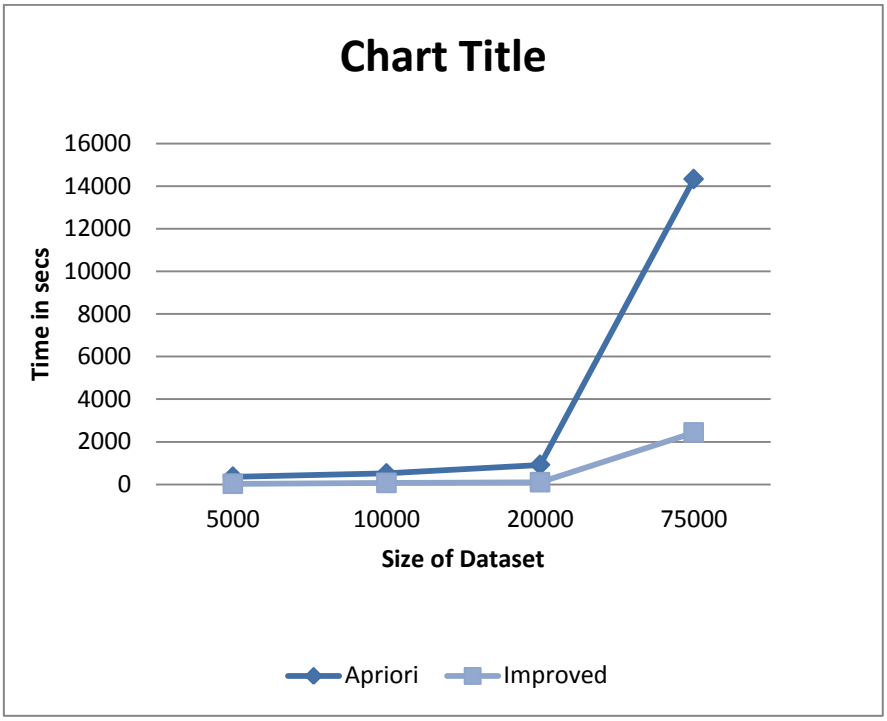

Figure 5.2: Comparison of Time of Execution (in secs)

Chapter 6

Conclusion and Future Scope

This chapter contains the conclusion that we arrived at after implementation and future scope.

6.1 Conclusion

After implementing the proposed approach we come to the conclusion that the improved Apriori algorithm proposed is an effective algorithm to reduce the number of transactions. The work is carried out on transactions rather than items which have improved its efficiency and to achieve the same the dataset is taken in a transposed manner. Instead of repeated scan of the original database, it is scanned only once to form large 1 item-set from which further computations are carried out. This reduces the time involved in scanning the dataset which in turn reduces the overall time to a greater extent. The minimum support value is also calculated at each pass which removes the unnecessary formed sets. Although the algorithm is simple, it carries out more effective pruning.

6.2 Future Scope

The results of the improved Apriori algorithm are satisfactory on single processor. The future work may include the implementation of the improved Apriori algorithm on the data distributed on parallel processors. The results are expected to be different in that case. We also wish to see whether the database scans reduces on each processor using this approach.

References

[1] R. Agrawal, R. Srikant, “Fast Algorithms for Mining Association Rules”, pp. 487-499.

[2] X. Liu, P. He, “The Research of Improved Association Rules Mining Apriori ALgorithm”, Proceedings of the Third International Conference on Machine Learning and Cybermetics, Shanghai, 26-29 August 2004, pp. 1577-1579.

[3] J. Lei, B. Zhang, J. Li, “A new improvement on Apriori Algorithm”, International Conference on Computational Intelligence and Security, Vol. 1, IEEE, 2006, pp. 840-844.

[4] Y. Xie, Y. Li, C. Wang, M. Lu, “The Optimization and Improvement of the Apriori Algorithm”, Education Technology and Training, International Workshop on Geoscience and Remote Sensing, ETT and GRS, Vol. 2, IEEE, 2008, pp. 663-665.

[5] Z. Changsheng, L. Zhongyue, Z. Dongsong, “An Improved Algorithm for Apriori”, First International Workshop on Education Technology and Computer Science, 2009, pp. 995-998.

[6] L. Jing et. al, “An Improved Apriori Algorithm for Early Warning of Equipment Failure”, 2009, pp. 450-452.

[7] K. Shah, S. Mahajan, “Maximizing the Efficiency of Parallel Apriori Algorithm”, International Conference on Advances in Recent Technologies in Communication and Computing, 2009, pp. 107-109.

[8] H. Wu, Z. Lu, L. Pan, R. Xu, W. Jiang, “An Improved Apriori-based Algorithm for Association Rules Mining”, Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009, pp. 51-55.

[9] Y. Liu, “Study on Application of Apriori Algorithm in Data Mining”, Second International Conference on Computer Modelling and Simulation, 2010, pp. 111-114.

[10] L. Wu, K. Gong, Y. He, X. Ge, J. Cui, "A Study of Improving Apriori Algorithm", 2010, pp. 1-4.

[11] P. Sandhu, D. Dhaliwal, S. Panda, A. Bisht, "An Improvement in Apriori algorithm Using Profit And Quantity", Second International Conference on Computer and Network Technology, 2010, pp. 3-7.

[12] L. Lu, P. Lu, "Study On An Improved Apriori Algorithm And Its Application In Supermarket", the research on Uncertain Reasoning Mechanism of Fuzzy Concept Map, pp. 441-443.

[13] G. Wang, X. Yu, D. Peng, Y. Cui, Q. Li, "Research of Data Mining Based on Apriori algorithm in Cutting Database", 2010, pp. 3765-3768.

[14] V. Sharma, M. Beg, "A Probabilistic Approach to Apriori Algorithm", International Conference on Granular Computing, IEEE, 2010, pp. 225-243.

[15] Y. Shi, Y. Zhou, "An Improved Apriori Algorithm", International Conference on Granular Computing, IEEE, 2010, pp. 759-762.

[16] Y. Shaoqian, "A kind of improved algorithm for weighted Apriori and application to Data Mining", The 5th International Conference on Computer Science & Education Hefei, China, August 24–27, 2010, pp. 507-510.

[17] D. Ping, G. Yongping, "A New Improvement of Apriori Algorithm for Mining Association Rules", International Conference on Computer Application and System Modelling (ICCASM), 2010, pp. V2-529.

[18] Y. Zhou, W. Wan, J. Liu, L. Cai, "Mining Association Rules Based on an Improved Apriori Algorithm", 2010, pp. 414-418.

[19] R. Chang, Z. Liu, "An Improved Apriori Algorithm", International Conference on Electronics and Optoelectronics (ICEOE), 2011, pp. 222-225..

[20] H. Wang, X. Liu, "The Research of Improved Association Rules Mining Apriori Algorithm", Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2011, pp. 961-964.

[21] V. Gurudatta, V. Nanda, "An Effectual Algorithm For Frequent Itemset Generation In Generalized Data Set Using Parallel Mesh Transposition", Advances in Engineering, Science and Management (ICAESM), International Conference, IEEE, 2012, pp. 719-724.

[22] Y. Jia, G. Xia, H. Fan, Q. Zhang, X. Li, "An Improved Apriori Algorithm Based on Association Analysis", Networking and Distributed Computing (ICNDC), Third International Conference, IEEE, 2012, pp. 208-211.

[23] J. Hu, "The Analysis on Apriori Algorithm Based on Interest Measure" ,proceedings of the International Conference on Control Engineering and Communication Technology, IEEE Computer Society, 2012, pp. 1010-1012.

[24] Y. Liu et. al, "Application and improvement discussion about Apriori algorithm of association rules mining in cases mining of influenza treated by contemporary famous old Chinese medicine", Bioinformatics and Biomedicine Workshops (BIBMW), IEEE International Conference, 2012, pp. 316-322.

[25] V. Vaithiyanathan, K. Rajeswari, R. Phalnikar3 , S. Tonge, "Improved Apriori algorithm based on Selection Criterion", 2012, pp. 1-4.

[26] Y. Peng, T. Zhou, "Research On the Apriori Algorithm in Extracting The Key Factor", 2012, pp. 90-93.

[27] C.Yadav et. al, "An Approach to Improve Apriori Algorithm Based On Association rule Mining", 4th ICCCNT ,2013, pp. 1-9.

[28] J.Yang, Z. Li, W. Xiang, L. Xiao, "An Improved Apriori Algorithm Based on Features", Ninth International Conference on Computational Intelligence and Security, 2013, pp. 125-128.

[29] J. Dongre, G. Prajapati, and S. Tokekar, "The role of Apriori algorithm for finding the association rules in Data mining", Issues and Challenges in Intelligent Computing Techniques (ICICT), International Conference, IEEE, 2014, pp. 657-660.

[30] K. Yu, J. Zhou, “A Weighted Load-Balancing Parallel Apriori Algorithm for Association Rule Mining”, Granular Computing, IEEE International Conference on 26-28 Aug. 2008, pp.756-761.

[31] G.K. Gupta, “Introduction to Data Mining with Case Studies”, Prentice Hall of India, 2006.

[32] “Extended Bakery Dataset”,https://wiki.csc.calpoly.edu/datasets/wiki/apriori.

List of Publications

[1] Sonal, Ravinder Kumar, "An Improvement in the execution time of Apriori Algorithm", accepted for publication in first International Conference for Networks and Soft Computing, ICNSC-2014, Hyderabad, August, 2014(Sponsored by IEEE).

[2] Sonal, Ravinder Kumar, "An efficient Technique to Reduce Number of Transactions in Apriori Algorithm", under communication in Seventh International Conference on Contemporary Computing , IC3-2014, Noida, August, 2014(Sponsored by IEEE).