CHAPTER 3

ASSOSIATION RULE MINING AND REGRESSION

This chapter discusses the Mining rule of the Association, its types and application of the Mining rule of the Association. Also discussing the Apriori Algorithm and Regression in this chapter

3.1 Overview of Association Rule Mining (ARM)

The ARM proved a very good technique to extract useful information from large databases. Association Rules discover all arrangements of things (itemsets) with help (number of exchanges) over least help (extensive itemsets) and after that use expansive things to make the standards you need which are more dependable than least trust. An ordinary and generally utilized case of the utilization of affiliation rules is showcase bushel examination.

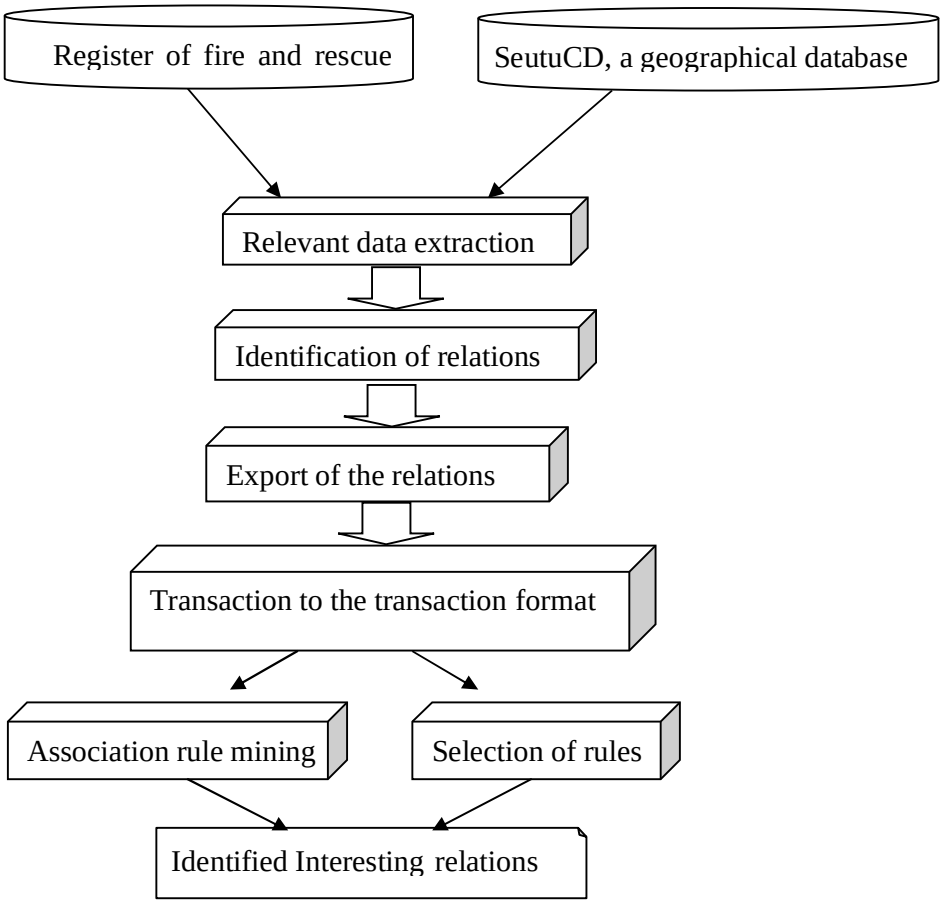

Figure 3.1 Architecture of Association Rule Mining (Source [138])

ARM architecture is explained in Figure 3.1which consists of

Find all Common Itemsets: Each one of these items sets shall, by definition, occur at least as often as the minimum default support number.

Generate Strong Association Rules from Frequent Itemsets: By definition, these guidelines must fulfil least help and least certainty. Extra intriguing measures can be connected whenever wanted. The second step is the two's most effortless. The initial step decides the general execution of the tenets of the mining affiliation.

Association means that related items are grouped from a itemset. For the purpose of finding association rules, a simple example is to analyze a large supermarket transaction database. This is called affiliation mining tenets or market crate investigation. Affiliation rule mining is utilized to discover visit examples, affiliations and connections between the arrangement of things or articles in value-based databases and social databases. A set of items is called an itemset. An itemset that contains k items is called k-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset. This is also known as the frequency or support count of the itemset. An itemset satisfies minimum support if the occurrence frequency of the itemset is greater than or equal to the product of minimum support and the total number of transactions in the entire database. The number of transactions required for the itemset to satisfy minimum support is referred as the minimum support count. If an itemset satisfies minimum support, then it is called a frequent or large itemset.

An association rule mining algorithm is divided into two parts: -

- Frequent itemsets generation i.e. all the itemsets having support greater than the user specified minimum support.

- Frequent itemsets generated in the step 1 will be used to generate association rules that satisfy user specified minimum confidence.

This can be used by retailers and entrepreneurs to advertise their businesses for improvement. The analysis based on the market finds the buying habits of the customer. We identify and analyse various customer buying habits in MBA to find associations among items that customers buy. In order to identify the frequent product combination, this analysis is done on the customer basket. MBA is a technique that helps to understand what items are likely to be purchased together under the association rules, primarily to identify

cross-selling opportunities. A supermarket can use this technique to organize and place frequently sold together products in the same area. MBA can be used by direct marketers to find out which new products their clients can offer. Using data mining tools generally facilitates the application of MBA. Marketers can identify the product in demand by making use of the MBA and the products "combined take-up rates" can be known. How often the items are purchased together is defined by the combined take rates. Association analysis helps retailers plan marketing, product placement and inventory management strategies of different types. Two basic entities of association rules are,

Support: Support is a measure of the fraction of the population that satisfies both the antecedents and the consequences of the rule

Confidence: Confidence is a measure of how often the consequences are true when the antecedents are true.

At the time of using association rules in relation database management systems, generally turn the database into a (tid, item), where tid is the transaction ID and item is the basis of several items that customers buy. There are multiple entries for a particular transaction ID, since one transaction ID indicates that one specific client is purchased and a client is allowed to purchase as many items as he wishes. This may look like an association rule:

buys (X, "computer") -> buys (X, "Windows software") [support =1%, confidence=50%]

Where,

$$Support = \frac{\text{The number of transactions that contain Computer and Windows software}}{\text{The total number of transactions}}$$

$$\text{Confidence} = \frac{\text{The number of transactions that contain Windows software}}{\text{The number of transactions that contain Computer}}$$

The rule above shall apply if its support and trust are equal to or greater than the minimum support and trust specified by the user. The main aim is to identify the group of items or items which appear together in several transactions. The vital links between the objects must, therefore, be discovered in a manner that results in the presence, in the same transaction, of certain objects.

This type of data is provided as if, then by association rules, statements. The rules are calculated from the data and, unlike the if-then logic rules, are probabilistic in nature.

The precedent (if a part of the if-then statements) and the resulting one (the next part) are disjointed item groups in the analysis of the associations (then part).

The use of customary affiliation calculations will be basic and productive. In any case, affiliation principle mining calculations ordinarily find a tremendous amount of guidelines and don't ensure that every one of the standards found are applicable. Backing and certainty variables can be utilized for acquiring fascinating principles which have values for these components greater than an edge esteem.

Despite the fact that these two parameters permit the pruning of numerous affiliations, another regular requirement is to demonstrate the qualities that must or can't be available in the predecessor or subsequent of the found principles.

Another arrangement is to assess, and post-prune the got standards so as to locate the most fascinating guidelines for a particular issue. Customarily, the utilization of target intriguing quality measures has been proposed, for example, backing and certainty, referenced beforehand, just as others gauges, for example, Laplace, chi-square measurement, relationship coefficient, entropy gain, intrigue, conviction, and so on. These measures can be utilized for positioning the acquired principles all together than the client can choose the standards with most noteworthy qualities in the measures that he/she is progressively intrigued.

Emotional measures are ending up progressively significant, at the end of the day estimates that depend on abstract components constrained by the client. The majority of the abstract methodologies include client investment so as to express, as per his or her past information, which guidelines are of premium.

3.1.1 Types of Association Rules

Multi-Level Association Rule: Information collected in a database contains an integrated hierarchy of multiple concept levels in several applications in the database. A concept hierarchy, for example, may replicate a categorization of items selling in a department store such as electronics, computers, speakers, computers, etc. Users may want to detect association rules only at the same level or association rules between items in such a database that extend across multiple levels. This process is called Multi-Level Mining Association Rule.

Utilizing ideology of hierarchies of command, which characterize a progression of mappings from a lot of low-level ideas to higher-level ideas, it is easy to mine staggered affiliation rules resourcefully. By supplanting low-level ideas inside the information with their higher-level ideas or connections from an idea chain of importance, information can be summed up.

Quantitative Association Rules: Many, if not most, databases in practice contain a large quantity of data and are not restricted to individual items only. The quantitative case is not directly interpreted by the definition of category association rules. Therefore, it is essential to provide a definition of association rules in the case of a data base containing quantitative attributes to the categorical definition. The basis of this definition is the reference by numerical value intervals of quantitative values to categorical events. Every fundamental event is therefore either a categorical item or a collection of numerical values. An example of this rule based on the definition would be:

$$sex = male\ and\ age \in [20, 30] \Rightarrow wage \in [\$5, \$10]\ (conf.\ 85\%)$$

Redundant Association Rules: The discovery of the Association rule has serious problems, particularly if the thresholds for support and trust are low, since the number of transactions is increased. With the increasing number of frequent items, the number of rules submitted to the user usually increases in proportion. Many of these rules can be redundant. To address the issue of rules of redundancy, four types of research have been conducted on mining association rules. Firstly, the rules based on user-defined templates or item restrictions were extracted. Secondly, to select only interesting rules, researchers have developed interesting measures. Third, scientists have suggested inferential rules or inferential schemes for pruning redundant rules and submitting smaller, more intelligible association rules to users. Finally, a new framework for the mining partnership rule was proposed to find rules for the association of different formats or features.

Negative Association Rules: Typical association rules only take into account items listed in transactions. Such rules are referred to as positive association rules. Negative rules of association also take negated items into account. (I.e. not involved in transactions). In Market-Basket Analysis, identifying products that conflict or complement each other, negative association rules are useful. Mining Negative Association rules are a challenging task because there are significant differences between positive and negative mining rules of association.

In negative association rule mining, the researchers address two key issues:

- Effectively searching for interesting itemsets and
- Effectively identifying negative association rules of interest.

3.1.2 Increasing the Efficiency of Association Rules Algorithm

Mining rules can reduce computational costs in four ways

- Reducing the number of database passes
- Sampling the database
- Adding additional pattern structure constraints
- Parallelization There has been a lot of progress in all these directions.

3.2 Apriori Algorithm

Apriori Algorithm is one of the great strategies used for finding or mine affiliation runs in affiliation examination. On account of the base assistance and least trust limit, it attempts to discover all the connection regulates whose estimations of the standard assistance are more noticeable than or proportional to the base guideline support edge regard and moreover the trust regards more conspicuous than or equal to the base trust edge regard. So as to discover regular things, a PC program dissects a wide scope of procurement records (exchanges). Such sets are regularly alluded to as a lot of things. The "visit" definition depends on a recurrence given by the client and is alluded to as the essential help. Endless supply of the continuous itemsets, these affiliations can be utilized by a different procedure for recommending extra things that a present client may purchase.

Apriori computation is definitely not hard to execute and direct, is used to mine beginning and end ceaseless itemsets in database. The estimation makes various endeavours in database to find visit itemsets where k-itemsets are used to make k+1-itemsets. Each k-itemset must be more significant than or comparable to least assistance point of confinement to be repeat. Else, it is called contender itemsets. In the essential, the figuring check database to find repeat of 1-itemsets that contains only a solitary thing by incorporating everything in database. The repeat of 1-itemsets is used to find the itemsets in 2itemsets which hence is used to find 3-itemsets, and so on until there are no more k-itemsets. If an itemset isn't visit, any colossal subset from it is furthermore non-visit; this condition prune from chase space in database.

Apriori estimation encounters some weakness paying little respect to being clear and essential. The principal limitation is costly wasting of time to hold endless contender sets with much progressive itemsets, low least assistance or gigantic itemsets.

For example, if there are 104 from consistent 1itemsets, it need to deliver more than 107 candidates into 2-length which along these lines they will be attempted and store up. Also, to recognize customary precedent in size 100 (e.g.) v1, v2… v100, it has to create 2100 contender itemsets that yield on excessive and wasting of time of cheerful age. Finally it will check for certain sets from cheerful itemsets, in like manner it will inspect database commonly again and again for finding contender itemsets. Apriori will be low and inefficiency when memory utmost is limited with tremendous number of trades.

3.2.1 Apriori Rule Mining

Efficient method for finding rules for associations by effectively finding sets of articles which meet the minimum criterion for support:

- Form size of itemsets K which justifies the support criterion with the help of items of size K-1
- Use the principles of certain itemsets to limit the work to be done

Each itemset size built in one pass through the database

3 .2.2 Apriori Properties

- For a support threshold of S, if any items I meet the threshold level S, then any non-empty subset I also meet the threshold as well. If any items I does not meet the threshold level S, any superset (including 0 or more) won't meet the threshold level S either.
- One can use these properties to prune the search space and reduce the work required. One major shortcoming of data-mining association rules is that too many rules are often generated by the support confidence framework. While ordinary Apriori algorithm can distinguish significant itemsets and assemble affiliation rules, it experiences the drawback of creating various hopeful itemsets that should be contrasted more than once and the whole database. Various changes and augmentations have been made to improve the Apriori Algorithm, in any case

3.2.3 Apriori Approach

- Apriori Algorithm is the most important algorithm used to find frequent mining itemsets.
- Apriori's basic principle is: Every subset of a frequent set of items must be frequent.

  Using frequent itemsets, association rules are generated.

Association rules:

- Unsupervised learning
- Used to discover the patterns
- Each of the rule has the following form: A -> B, or Left -> Right

For example, "70% of customers who buy 2% milk also buy entire wheat bread." Association rules are used to explore data mining in order to look for sound rules:

- Find the big itemsets (i.e. most common item combinations)
- Apriori algorithm is the most commonly used algorithm.
- Generate rules of association for the sets of items above.

An association rule's strength is measured using the following parameters:

- Using support/confidence
- Using dependence framework

Support: Support shows the frequency of the patterns in the rule; it is the percentage of transactions that contain both A and B, i.e.

Support = Probability (A and B)

Support = (no. of transactions involving A and B) / (total number of transactions).

Confidence: The strength of the implication of a rule is confidence. This is the proportion of transactions containing B if they contain A, then

Confidence = Probability (B if A) = P(B/A)

Confidence = (no. of transactions involving A and B) / (total number of transactions that have A).

Apriori Algorithm has been taken into account all database transactions in the market basket setting. This technique organised in recognition of patterns, became known through innovation under the next rule: "Often customers buy slippers and beers together at the grocery store on Thursdays". The Apriori Algorithm's outputs are easy to understand and it is possible to identify many new patterns. The sheer number of association rules, however, may make it problematic for interpreting the results.

A second algorithm fault is that large items are highly computational because of the exponential complexity of the algorithm. Therefore, there is a great need for a novel and effective market bucket analysis technique.

3.2.4 Common Steps in Apriori Algorithm

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

Join Operation: In order to frequently set k as indicated by L_k, a candidate, denoted by C_k, consists of the attachment of L_{k-1} to itself.

Prune operation: The number of each subset of C_k is calculated to find the frequency set as all Ck members are not frequent. Therefore, all members with less than support value are deleted. The remaining members make up the frequency set. If subset of C_k of size k-1is not present in L_{k-1} then it's not a frequent candidate. Thus it is removed from C_k.

The pseudo code for generating the frequent itemset is shown below.

Apriori Algorithm for Frequent Itemset Generation (Source [35])

Input: Dataset of Transactions and Minimum Support Threshold values

Step 1: F_1=Frequent 1-itemset

Step 2: Initialise k=1;

Step 3: F_k={i|I Є I ∧ σ({i}) >=N*minsupp} //find all frequent 1-itemsets

repeat;

k=k+1;

C_k=apriori-gen(F_{k-1}) //generate candidate itemset

Step 4: for each transaction t Є T do

C_t=subset(C_k,t) // identify all candidate that belong to t

for each candidate itemset

c Є C_t do

σ(c)= σ(c)+1 //increment support count

end for

end for

Step 5: F_k={c|c Є C_k ∧ σ(c) >=N*minsupp} //extract the frequent k-itemset

Until F_k=Θ;

Step 6: result= U_k*F_k

/* Procedure Apriori-gen(F_{k-1}) */

Step 1: for each itemset f_1 Є F_{k-1} do

Step 2: for each itemset f_2 Є F_{k-1} do

if(f_1[1]=f_2[1]) ∧ (f_1[2]=f_2[2]) ∧… (f_1[k-2]=f_2[k-2]) ∧ (f_1[k-1]=f_2[k-1])) then

c= f_1 × f_2; // join step of generate candidates

for each(k-1)-subsets of c do

if (s Є F_{k-1}) then

delete c; // prune step to remove candidates

else add c to F_k;

end for

return F_k;

end if

end for

end for

C denotes the set of candidate k-itemsets and k F denotes the set of frequent k-itemsets:

- The algorithm initially passes the data to determine each item's support. Once this step is completed (step 1 and 2), the set of all frequent items one F will be celebrated.
- The frequent (k−1) itemsets array found in the previous iteration (step 5) can then iteratively generate the algorithms, using the new candidate k-itemsets. Candidate generation is obliged to use an operation called apriorigen.
- To count the candidates ' support, the algorithm must construct another leave out of the information set (steps 6–10). The set operate is utilized to see all the candidate itemsets contained in k C in each dealings t.
- All candidate itemsets will be eliminated by algorithm whose support counts are only min sup (step 12) after tallying their supports.
- Once no new frequent itemsets are generated, i.e. K F (step 13), the algorithm ends. A part of the Apriori algorithm has 2 necessary features for frequent generation of itemsets.

In principle, there are many ways of developing candidate itemsets. A list of requirements for a good candidate generation procedure may be as follows:

- Too many needless candidates should be avoided. A candidate itemset does not make sense if at least one is sporadic in each of its sub-sets. Such a candidate must be sporadic in accordance with one of the support properties of the antimonite.
- Make sure that the set of candidates is complete, i.e. that the candidate generation procedure does not overlook any frequent itemsets. The set of candidate itemsets should include the set of all the frequent itemsets in order to ensure completeness.
- The same candidate itemset should not be generated once. Generating duplicate candidates ends up in wasted calculations and should therefore be avoided for reasons of potency.

While Apriori Algorithm is generally huge, it is ineffectual as it endeavours to stack however many applicants as could be expected under the circumstances before the second sweep, making a substantial number of subsets. It brings forth a ton of Apriori in that capacity-like calculations. The minsupp is a parameter to check the combinatorial extension of the exponential calculation. Since Apriori utilizes just a single minsupp esteem, some container size can be over dimensioned while others can be under dimensioned.

The Apriori Algorithm's first step generates market basket sets. I_k is defined as a set of items with k items that are frequently purchased. First, the algorithm filters the items

more often than the miniature, creating I1. In the next phases, it generates I_k+1 applicants in every I_k, for example I_k/I_k+1. For every I_k+1 candidate, the algorithm removes baskets that are lower than the minsupp. The cycle ends when reaching Imax.

Market baskets are generated by the Apriori Algorithm in the second step and then generates association rules Left hierarchy. The support and confidence measures are calculated for each rule.

The following transaction – analytical tasks store must be performed before a new customer even enters, namely

- Analysis of frequent item sets transactions that fulfill the minimum support necessary.
- Construct frequent item sets to the candidate rules.
- Print the candidate's rules which fail to satisfy the minimum confidence required.
- The following basket analysis tasks, i.e., have to be done when a new customer enters the store and puts a product into the basket.
- Find all rules which correspond to the actual basket content of the customer.
- To propose additional items to the customer, select one or more rules to be applied.

A classical algorithm used when searching for frequently selected items (stage 1 of transaction analysis tasks), i.e. Apriori performance has decreased significantly in the last decade, such as algorithms. But the output was always small market baskets. There is a need to find larger shopping carts for thousands of items in the retail industry.

The yields of the Apriori calculation are justifiable and numerous new examples can be recognized. Be that as it may, the sheer number of guidelines of affiliation can make translating the outcomes troublesome. A second shortcoming in calculations is the tremendous time it takes to discover vast things because of the exponential unpredictability of the algorithm [94].

3.2.5 Formal Problem Description

Give me a chance to be give me a chance to be a lot of items $I=\{i_1, i_2 \ldots i_n\}$. Give D a chance to be a database or a lot of exchanges. Every exchange is a lot of things, t being a legitimate subset of I. At the point when X is an appropriate subset of t, an exchange t bolsters X, a lot of things in I. A standard of affiliation is an implification of the $X \rightarrow Y$ structure, in which X and Y are subsets of and $X \cap Y = \emptyset$, where X is known as the point of reference and the subsequent part is called Y.

Guideline X→Y bolster alludes to the proportion of the quantity of exchanges in the database containing the itemset X and Y to the absolute number of exchanges in database D. Rule trust is the proportion of the quantity of exchanges in the database containing the X and Y itemset to the quantity of exchanges containing X. A standard X→Y is solid when it achieves the base edge of help and the base limit of certainty. Affiliation rules mining calculations filter the exchange database and compute the help and certainty of the standards just with help and certainty over the base client backing and trust limit specified. Find visit things and produce affiliation rules. In most cases, the finding of the continuous set overwhelms the execution of the whole procedure.

3.2.6 Frequent Itemset Generation

Visit object sets are those arrangements of things whose events in the database exceed a predefined edge. Generally speaking, the computational prerequisites for generating visit itemsets are more expensive than those of run generation. Generating all the subsets of things, say I={i_1, i_2, ... in} for large value n, is practically a lot of trouble because of the gigantic search space in fact, a linear number of things suggest an exponential number of itemsets developing.

Additionally, the way to generate incessant itemsets can be isolated into two sub-issues: the handle generation of large candidate itemsets and the preparation of successive itemsets. Itemsets that are normal or plan to be large or incessant are referred to as candidate itemsets and are viewed as regular itemsets among those itemsets whose help surpasses the aid edge. To find continuous itemsets, a beast drive approach is to decide the help each candidate itemset means. For this reason, a counter can be used and adjusted to zero for every itemset. All transactions are scanned at that point and at whatever point one of the candidates is perceived as transaction subset, their counter is increased. Another approach is by setting crossing points to determine the help values. Everything can be associated with a tidlist, the rundown of transactions containing the thing. Specific transaction identifier (tid) is used to indicate each transaction.

Accordingly, there will be a tidlist associated with each itemset X and to obtain the tidlist of a candidate itemset U=X ∪ Y, then calculate X. tidlist ∩ Y. tidlist. Finally, the actual help is generated by deciding |U. tidlist|. Whichever approach is utilized; the computational unpredictability of the procedure is always remarkable. Many approaches can diminish this intricacy and enhance the performance of the procedure. Apriori standard is one such approach is proposed.

Continuous examples, for example, visit thing sets, substructures, arrangements term-sets, express sets, and sub diagrams, for the most part exist in true databases. Distinguishing successive thing sets is a standout amongst the most significant issues looked by the learning disclosure and information mining network. Visit thing set mining assumes a significant job in a few information mining fields as affiliation rules warehousing, relationships, grouping of high-dimensional organic information, and arrangement. Given an informational collection d that contains k things, the quantity of thing sets that could be created is 2k - 1, barring the void set.

So as to looking through the continuous thing sets, the help of everything sets must be figured by filtering every exchange in the dataset. A savage power approach for doing this will be computationally costly because of the exponential number of thing sets whose help checks must be resolved. There has been a great deal of fantastic calculations produced for extricating continuous thing sets in exceptionally enormous databases. The productivity of calculation is connected to the size of the database which is amiable to be dealt with. There are two regular procedures received by these calculations: the primary is a successful pruning system to decrease the combinational pursuit space of competitor thing sets (Apriori methods). The second methodology is to utilize a packed information portrayal to encourage for preparing of the itemsets (FP-tree systems)

The Apriori rule states that all its subsets should also be visited if an itemset is visited. On the other hand, if an itemset is occasionally, all its supersets will also have to be rare. This can help reduce the quantity of candidate itemsets being investigated as the subsets of continuous itemsets and supersets of rare itemsets do not require a separate investigation.

3.2.7 Rule Generation

The performance of an algorithm that mines data association rule depends largely on the first step that finds the database's frequent itemsets. The rule generation stage is straight forward once frequent itemsets are in hand [120]. Suppose one of the big itemsets is Lk, Lk = $\{I_1, I_2, I_3 \ldots\ldots I_k\}$ and the rules of association are generated in a simple way with this itemset.

First rule is the following:

$\{I_1, I_2, I_3 \ldots\ldots I_{k-1}\}$ default is $\{I_k\}$

By checking confidence, this rule can be considered interesting or not interesting. Then the other rules are generated by deleting and entering the last item into the precedent in order to determine the importance of the rule. This process goes on until the preceding part is empty. So it's a simple approach, but it takes time to check the confidence of each and every possible rule and the situation gets worse when most of the rules generated are rejected because they are redundant or have low confidence

3.2.8 Traversing the Search Space

Sets of visit items are those arrangements of things whose events in the database exceed a predefined limit. Generally speaking, the computational prerequisites for generating visit itemsets are more expensive than those of run generation. Generating all the sub-sets of things, say I= $\{I_1, I_2, I_3 \ldots\ldots I_n\}$ for large value n, is practically a lot of trouble due to the huge search space, in fact a linearly evolving number of things infer an exponential number of itemsets.

Additionally, the way to generate incessant itemsets can be separated into two sub-issues: the handle generation of large candidate itemsets and the preparation of continuous itemsets. Itemsets that are normal or wish to be large or successive are referred to as candidate itemsets and are viewed as continuous itemsets among those itemsets whose help surpasses the help edge. Deciding the help means each candidate itemset is a savage drive approach to finding successive itemsets. For this reason, a counter can be used and set to zero for each itemset. All transactions are scanned at that point and at whatever point one of the candidates is perceived as a subset of a transaction, their counter is increased.

Another approach is to decide the help values by set crossing points. Each thing may be associated with a tidlist, which is the rundown of transactions that contain the thing. Each transaction is meant by one of a kind transaction identifier (tid). Accordingly, there will be a tidlist associated with each itemset X and to obtain the tidlist of a candidate itemset $U=X \cup Y$, then calculate $X.tidlist \cap Y.tidlist$. Finally, the actual help is generated by deciding $|U.tidlist|$.

Whichever approach is utilized, the computational multifaceted nature of the procedure is always remarkable, whatever approach is used. There are many approaches that can reduce this multi-faceted quality and improve the procedure's performance. One such approach proposed is the Apriori Guideline. The Apriori rule states that all its subsets should also be visited if an itemset is visited. On the other hand, if an itemset is occasionally, all its supersets will also have to be rare. This can help to decrease the

quantity of candidate items as the subsets of successive itemsets and supersets of rare itemsets do not require separate investigation [104].

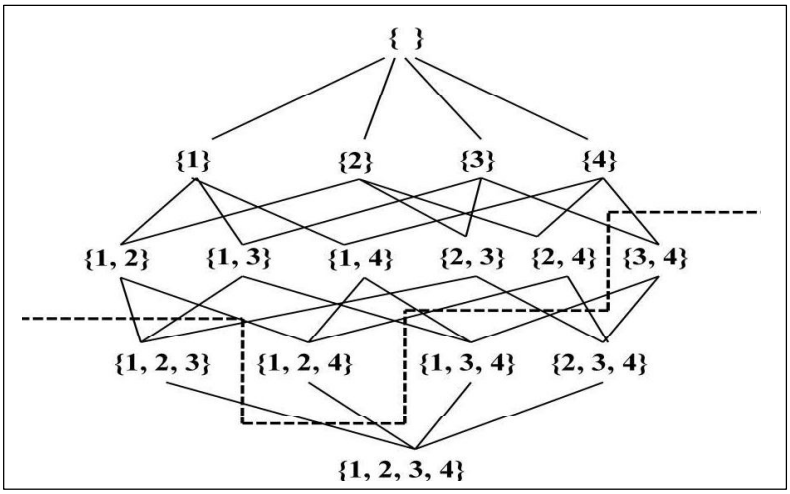

Figure 3.2 Search Space for I = {1, 2, 3, 4} (Source [139])

For the special case I = {1, 2, 3, 4} we visualize the search space that forms a lattice in Figure 3.2. The frequent itemsets are located in the upper part of the figure whereas the infrequent ones are located in the lower part. Although explicit support values for each of the itemsets are not specified, it is assumed that the bold border separates the frequent from the infrequent itemsets. The existence of such a border is independent of any particular database D and min_supp. Its existence is solely guaranteed by the downward closure property of itemset support. The basic principle of the common algorithms is to employ this border to efficiently prune the search space. As soon as the border is found, we are able to restrict ourselves on determining the support values of the itemsets above the border and to ignore the itemsets below.

3.2.9 Defining Support and Confidence

There are two important and basic measures for association rules, Support and Confidence. Support(s) of an association rule X→Y is defined as the percentage/fraction of records that contain X ∪ Y to the total number of records in the database.

$$\text{Support}(X \rightarrow Y) = \frac{|X \rightarrow Y|}{|D|} \qquad \text{---------- (1)}$$

where |X→Y| denotes the number of transactions in the database that contains the itemset X ∪Y and |D| denotes the number of the transactions in the database D. Suppose

the support (Eq.1) of an item is 0.1%, it means that 0.1 percent of the transactions contain this item. The confidence of rule is calculated by following equation:

$$\text{Confidence } (X \rightarrow Y) = \frac{|X \cup Y|}{|X|} \quad \text{---------- (2)}$$

Where |X| is number of transactions in database D that contains thing set X. certainty is a measure of the quality of the association rules. Assume certainty of the manage is 80%, it means that 80% of the transactions that contain X also contain Y together. A manage $X \rightarrow Y$ is solid if bolster $(X \rightarrow Y) \geq$ min_support and certainty $(X \rightarrow Y) \geq$ min_confidence, where min_support and min_confidence are two given least edges. Support is usually a significant measure which can occur essentially by chance with low help. from a business point of view as things that customers occasionally purchase together may not be profitable. Then again, certainty is measuring the reliability of the derivation of a show run. This ratio is called a measure of h-confidence or all-confidence (Eq.2). The numerator in the above ratio is bounded by any item appearing in the frequent itemset with the minimum support. In other words, an itemset h-confidence = $\{i_1, i_2, \ldots\ldots, i_k\}$ should not exceed the following phrase:

$$\text{Confidence } (i_1, i_2 \rightarrow \{i_3, i_{4,\ldots,} i_k\}) \leq \text{Confidence } (i_1, i_2, i_3 \rightarrow -\{i_4, i_{5\ldots} i_k\}) \quad \text{------- (3)}$$

Given a frequent itemset $\{-i_1, i_2, \ldots., i_k\}$, the rule $\{-i_j \rightarrow -i_1, i_{2\ldots\ldots\ldots} i_{j-1,} i_{j+1\ldots\ldots\ldots\ldots}, i_k\}$ has the lowest confidence (Eq.3) if $s(i_j) = \max[s(i_1), s(i_2), \ldots\ldots\ldots\ldots, s(i_k)]$ because the confidence is the ratio between the rules support and the support of the rule antecedent. Thus if the antecedent is having higher value of support the denominator in the confidence ratio will be higher than the numerator and this decreases the resulting ratio. The lowest confidence attainable from a frequent itemset is $\{-i_1, i_2, \ldots\ldots., i_k\}$

This ratio is called h-confidence or all-confidence measure. The numerator in the above ratio is restricted by the minimum support of any item that appears in the frequent itemset.

3.3 Algorithms for Association Rule Mining

As association rule mining is a two-step procedure: frequent itemsets generation and rule generation. Because second step is straight forward, most of the algorithms have proposed methods of detecting the frequent itemsets.

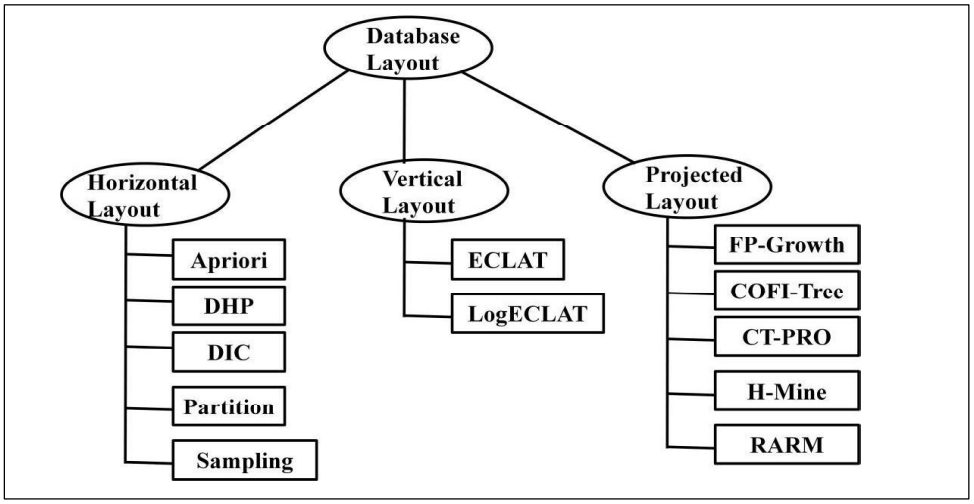

Figure 3.3 Algorithms for Frequent Itemset Mining for Different Database Layouts (Source [140])

Figure 3.3 shows different algorithms which is used to find out Frequent Itemsets. The algorithms are categories on the basis of layout of database that is used. These algorithms can be separated in three broad categories:

- Algorithms that use horizontal layout database
- Algorithms that use vertical layout database
- Algorithms that use projected layout database.

In horizontal layout, database is represented as collection of transactions with each transaction containing a list of items. Vertical layout of database consists of items with each item containing a list of transaction Ids of the transactions that contain this particular item. Projected layout based databases uses different data structures like tree to store and count support values.

3.3.1 Apriori Algorithm

The first algorithm has been proposed for frequent mining of items and Association Rules was named Artificial Immune Systems (AIS) [105]. This algorithm uses the horizontal layout database in this algorithm, Regulations are generated which have only one item as a consequence. The database is numerous times scanned to obtain common itemsets. The algorithm was shortly improved and Apriori renamed. Apriori algorithm is the most classical and important algorithm for mining frequent itemsets. It employs an iterative approach usually a breadth-first approach through the search space where k^{th} itemsets are used to explore $(k+1)^{th}$ itemsets in the beginning, 1-frequent itemsets are found

and their support is calculated to check against minimum threshold. The candidate itemsets having higher support than minimum threshold support, are selected for next pass in the second pass 1-frequent sets are used to obtain 2-frequent itemsets. The algorithm works in two stages: Join stage and Prune stage.

Prune Phase: L_k is a subset of C_k, whereas refers to this group of candidates, that is its individuals may or may not be visit, but rather all the incessant k-itemsets are incorporated into C_k. The database is scanned to figure out which individuals from C_k have bolster number higher than least edge and those individuals are incorporated into L_k. The candidate itemset C_k can be large and this computation can be unwieldy. To diminish the measure of C_k, the Apriori property is utilized to prune Ck. Utilizing this property, any (k-1) subset of C_k that is not in L_{k-1}, cannot create a regular k-itemset in next pass. So such itemsets are expelled from C_k.

The algorithm utilizes the Apriori guideline to prune the candidate sets to expel the pointless itemsets in start of the pass and lessen computation. The Apriori property indicates how the help measure decreases the quantity of candidate itemsets investigated amid the successive itemset generation. The Apriori guideline is stated as takes after:

Apriori Principle: If you visit a set of products, you should also check all of the subsets. On the contrary, the anti-monotonous property says that if a set is not visited, none of its supersets will be visited. The strategy to cut the exponential search area by measuring aid is called aid-based cutting. Therefore, utilizing these two properties the algorithm prunes the candidate itemsets which are not visit in the present pass as they cannot be utilized to create visit itemsets in consequent passes. Also if an itemset is observed to be visit, its subsets require not be checked for being successive. The computational multifaceted nature of Apriori algorithm can be dictated by several factors. A portion of the factors are:

Bolster limit: Using a low help edge brings about large number of itemsets being declared as incessant. Therefore, more candidate itemsets are generated and tallied and this has increases the computational many-sided quality of the algorithm. Number of things and number of transactions: For more things, more space is expected to store the help tallies in the event that number of incessant itemsets also increases with number of things, the computation cost will develop substantially.

The algorithm also makes repeated transitions to the database, with a wider number of transactions the run time increases. Average width of transactions: it affects the

multifaceted nature of the algorithm by two different means and by increasing the average width of the transaction, the maximum size of the regular product is tending to increase. Second more itemsets are contained in the transaction with large things and this will increase the quantity of scanning performed amid help numbering. The key strides in the algorithm are laid out in figure 3.3.

Apriori Algorithm generation for Itemset (Source [90])

Inputs: Transactions Database, Minimum Support Threshold value

Output: All Frequent Itemsets whose Support is higher than Minimum Support Threshold

step 1: Initialize L_1=Large 1-itemsets;

step 2: for (k=1; L_{k+1}!=NULL; k++)

C_k=Apriori-gen ($L_{k+1,}$ minsup); //generate new candidate from L_{k+1}

for all transactions T ∈ D do begin

C_r=subset(C_k,T);

for all candidates C ∈ C_t do

Count(C)= Count(C)+1;

end

L_k={C ∈ C_t | Count(C)>=minsupthreshold}

end

step 3: L_g=U_k *L_k

step 4: Return L_g

/* procedure Apriori-gen */

Step 1: for each itemset I_1 ∈ L_{k-1}

Step 2: for each itemset I_2 ∈ L_{k-1}

if(I_1[1]=I_2[1]) ∧ (I_1[2]=I_2[2]) ∧…(I_1[k-2]=I_2[k-2]) ∧ (I_1[k-1]=I_2[k-1])) then

c= I_1 × I_2;

if infreq_subset(c, L_{k-1}) then

delete c;

else add c to C_k;

step 3: return C_k

RID	Items List
R10	R_1, R_2, R_5
R20	R_2, R_4
R30	R_2, R_3
R40	R_1, R_2, R_4
R50	R_2, R_3
R60	R_2, R_3
R70	R_1, R_3
R80	R_1, R_2, R_3, R_5
R90	R_1, R_2, R_3
R100	R_1, R_2, R_5, R_6

(i) Database of transactions

R Items	Count Number
R_1	7
R_2	8
R_3	6
R_4	2
R_5	3
R_6	1

(ii)

R Items
R1
R_2
R_3
R_5

(iii)

R Items	Count Number
R_1, R_2	5
R_1, R_3	4
R_1, R_5	3
R_2, R_3	4
R_2, R_5	3
R_3, R_5	1

(iv)

R Items
R_1, R_2
R_1, R_5
R_2, R_5
R_2, R_3
R_1, R_3

(v)

R Items	Count Number
R_1, R_2, R_5	3
R_1, R_2, R_3	2

(vi)

Table 3.1 Generation of Frequent Itemsets using Apriori

Example: Let us consider the database shown in Figure 3.4. The subsequent figures show how Apriori algorithm works to find frequent itemsets.

The value of support threshold is taken as 2.

Apriori property is utilized to avoid the unnecessary activities of checking backing of the candidate itemsets. The conceivable candidate 3-itemsets are (I_1, I_2, I_5), (I_1, I_2, I_3), (I_2, I_3, I_5) and (I_1, I_3, I_5). Because the 2-itemset (I_3, I_5) is not visit as appeared in figure (d), there is no expectation that any of its superset will be visit in this way the itemsets (I_2, I_3, I_5) and (I_1, I_3, I_5) are not considered and eliminated with no further checking.

Apriori algorithm has the drawback of scanning the entire database many circumstances and generation of large number of candidate itemsets in any case, it successfully discovers all the conceivable regular itemsets. As the database dimensionality increases with the amount of things, there is a need for more search space and I/O costs. This also requires more database scans and large amount of storage space for storing the candidate itemsets. This thus increases the overall computational cost in this way many variations have been made in Apriori algorithm to limit these limitations. Generally, there are two approaches to enhance the issues viz. diminishing the quantity of scans over the database and to investigate various types of pruning strategies to decrease the candidate itemsets.

3.3.2 DHP Algorithm

DHP generates k-itemsets of L_{k-1} as well as Apriori instead of including all k's from the L_{k-1} to the L_{k-1} to the L_{k-1} to the C_k, DHP uses a hash table, which has been worked out in the past pass. The k set is only added when the k-set is hung into a hash area with a value greater than or equal to the basic transaction bolster required. The span of candidate C_k is significantly lowered in these lines. DHP also dynamically decreases the database estimate by reducing the amount of transactions in the database and reducing each transaction measure.

3.3.3 DIC Algorithm

DIC is further variation of Apriori algorithm. DIC diminish the separation of tallying and generating candidates in this approach, the database is split into start focused parts and at any point new candidate element sets, unlike in Apriori, can be added, deciding on new applicants itemsets immediately before each full database scan is completed. The method uses the lesser bound of the actual tally as far as possible. At whatever point a candidate itemset achieves least help even before it is compared against all transactions, DIC starts generating candidates based on it. This leads to less database scans than Apriori for discovering all incessant itemsets. For this a prefix tree is utilized in contrast to a hash tree, each leaf or internal hub is assigned to exactly one candidate itemset.

3.3.4 Partitioning Algorithm

Partitioning algorithm is based on the idea of isolating the database into smaller parts and finding the incessant itemsets in partitions separately [106]. It overcomes the large database storage problem that does not fit into main memory because small parts of the database fit easily into their main memory. The algorithm is divided into n parts in two passes in the main pass whole database. Each partitioned database is loaded one by one and the following local components will be found in the main memory.

At that point all locally visit components are consolidated and global candidate set is obtained in the second pass, the globally visit components from these candidate itemsets are found. A local successive itemset may or may not be visit regarding the whole database. Any object set which is likely to be visited must therefore be included in one of the following partitions. However, the expected time for calculating a recurring candidate generation in each partition is greater than the database scan and thus increases computational costs. This algorithm may diminish the database scan for continued itemset generation however.

The partition algorithm is based on apriori algorithm. It firstly partitions the data into a number of non-overlapping partitions and processes each partition separately to generate frequent itemsets local to each partition and finally it combines all the local frequent itemsets to generate global frequent itemsets. It reduces the number of complete database scans up to two and hence improves the performance of mining algorithm.

The Incremental Mining algorithm is another useful technique for speeding up the mining process when new data is added to the database. Sampling algorithm is also based on apriori algorithm. Rather than mining entire database, here draw out a random sample of data form the database and then finds out frequent itemsets in that sample instead of the entire database. Finally, the rest of the database is used to compute the actual support of the frequent itemsets that found in the sample.

Because of searching for frequent itemsets in the sample, it is possible that many may miss some global frequent itemsets. To lessen these use lower support than minimum support for the sample. In this way trade off some degree of accuracy against efficiency. There are various mechanisms so that find out all the missing frequent itemsets those are not find out in the sample. Most of these algorithms are In-memory algorithms, in which data is directly read from flat files or first extracted from database to the flat files and then

processed in main memory. Most of these algorithms build specialized data-structures and implement their own buffer management schemes.

3.3.5 Sampling Algorithm

The sampling algorithm is utilized to conquer the limitation of I/O overhead in case of large databases. This method does not consider the whole database for checking the recurrence of the arrangement of things. Rather it utilizes a random sample of the database which can be accommodated in the main memory. This random sample is utilized to discover all association decides that may probably hold in the whole database. After finding the standards from the sample, the outcomes are confirmed with whatever is left of database. Therefore, the algorithm create exact association decides that hold for the entire database. Be that as it may, the approach is probabilistic and in some rare cases it doesn't deliver exact association rules in any case, the issue can be handled easily utilizing a moment pass and lower limit values of help and certainty.

The experimental outcomes with this approach demonstrate that the association standards can be found proficiently utilizing this approach. This method of identifying association rules is better if proficiency is more important than the accuracy, because this approach gives brings about small time and also defeats the limitations of memory consumption. Testing calculation is actualized in the setting segment calculation. The calculation first parcels the database into various segments. And after that accepts one parcel as an example. It at that point discovers all the neighbourhood visit itemsets in the example for the decreased least help for that example.

At that point these nearby incessant itemsets alongside their negative fringes are tried against the whole dataset for the base help in the whole dataset. Itemsets qualifying the base help are visit itemsets in the whole database. On the off chance that negative outskirt of the neighbourhood visit itemsets contain visit itemsets in the whole database, at that point just the calculation filters the database second time to discover missing regular itemsets in the database.

The presentation of inspecting calculation depends on the nature of the example picked. On the off chance that the example picked is an awful example the quantity of applicants creates for second sweep might be extremely huge henceforth second output can be wasteful. The example can be a segment of the database. All things considered the parcel is dealt with simply like an irregular example picked.

3.3.6 Equivalence Class Transformation (ECLAT) Algorithm

ECLAT is a profundity initially based algorithm. It utilizes a vertical database layout [24] instead of expressly posting all transactions; each thing is put away together with its tidlist (transaction-id rundown) and utilizations the crossing point based approach to figure the help of an itemset. It requires less space than Apriori if the itemsets are small in number. It is suitable for small databases and requires less time for visit itemset generation as compared to Apriori. Another advantage of ECLAT is that it scans the database just twice first time to generate a 2-itemset and second time to transform it into vertical shape. Éclat algorithm works in three phases:

- Initialization phase: Construction of the global means the incessant 2-itemsets.
- Transformation phase: Vertical transformation of the database.
- Asynchronous phase: Construction of the regular k-itemsets

In ECLAT algorithm, the calculation is done by intersection purpose of the Tid sets of the contrasting k-itemsets regardless, the Tid sets might be long, which takes generous memory space just as much calculation time for joining the long sets. Additionally, the identicalness classes are comprehended from base to top in reverse lexicographic solicitation. Subsequently the data on help count is accessible and may have an essential impact in pruning, anyway ECLAT algorithm does not make full use of the data in masterminding to decrease the pointless task on itemsets which will be deleted in the assistance based pruning later.

The calculation does not exploit Apriori property to diminish the amount of competitor itemsets explored in the midst of visit itemset age along these lines the amount of hopeful itemsets produced in ECLAT calculation is generously more noteworthy than that in Apriori algorithm. The circumstance weakens for itemsets with numerous thick and long examples.

3.3.7 Log ECLAT Algorithm

Log ECLAT is another algorithm that utilizations vertical partitioning of database. This algorithm is edified by ECLAT algorithm and utilizations special candidates to discover visit patterns from a continually updating database, containing essential information about regular patterns. This algorithm can discover several k-itemsets in one

time of database scanning and along these lines, the season of establishing new database is lessened. With a specific end goal to discover visit patterns ranging from L_{k-1} to L_k, log ECLAT algorithm utilizes candidates chosen from combinations which are created by each two diverse itemsets in Lk and scans database Dk. The rightness and fulfilment of log ECLAT algorithm is demonstrates with two facts.

The main fact is that, the combinations utilized by log ECLAT algorithm to scan the database can be one of t-itemset Lt ($k+1 \leq t \leq 2k$). These two itemsets are distinctive and one itemset is having k things in this manner, in one itemset there are j ($1 \leq j \leq k$) things, which is not quite the same as the other itemset and the combination of two distinctive k-itemset is made out of i ($k+1 \leq i \leq 2k$) things. The second fact is that any combinations of things which are not created by log ECLAT algorithm cannot be visit patterns. For a combination of things which is not a continuous pattern, its subset cannot be visit patterns.

Clearly, those unqualified combinations of things are the subset of combinations of things which are not visit patterns and none of incessant patterns will be remembered fondly. Because of these two facts, it is easy to get the conclusion that Log ECLAT algorithm can discover all the continuous patterns effectively. Log ECLAT also performs well with the change in number of transactions and bolster limit values.

3.3.8 FP-tree/FP-Growth

The main drawback of Apriori algorithm was handled in FP-development algorithm. The compact data structure called Frequent pattern tree or FP Tree is used in this algorithm. Visit items are generated with two database passes and without the generation of candidates handle [107]. Accordingly, it is faster than the Apriori algorithm. It is also a two stage handle which incorporates building FP-tree and generating successive patterns. A FP-tree is a tree with following properties: It comprises of one root labelled as root, an arrangement of thing prefix sub-trees as the offspring of the root and an incessant thing header table.

- Each hub in the prefix sub-tree has three domains: name, check and hub connection, where name indicates the represented thing in the hub, tally means the number of transactions shown by the bit of the path to the hub and the port connects to the sub-tree hub.
- Each section in the incessant thing header table comprises of two fields: thing name and head of no

RID	Items List
R10	R_1, R_2, R_5
R20	R_2, R_4
R30	R_2, R_3
R40	R_1, R_2, R_4
R50	R_2, R_3
R60	R_2, R_3
R70	R_1, R_3
R80	R_1, R_2, R_3, R_5
R90	R_1, R_2, R_3
R100	R_1, R_2, R_5, R_6

Table 3.2 Database of Transactions

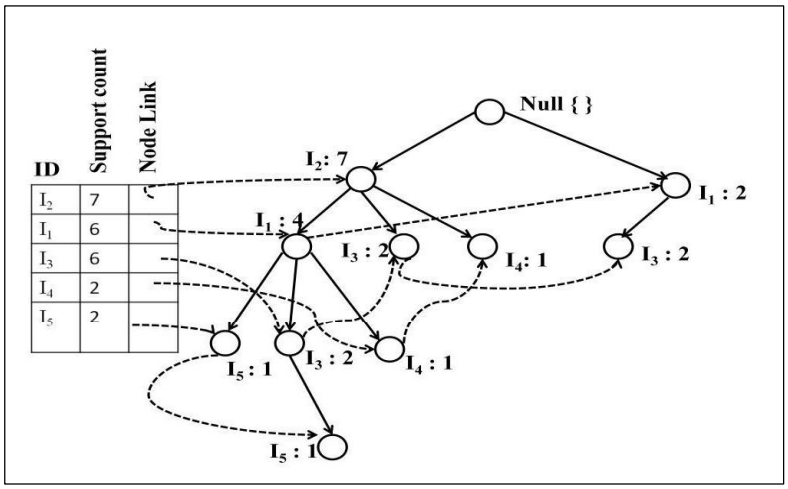

Figure 3.4 FP-Tree of the Database of Transactions (Source [141])

FP-Tree mining is uncomplicated. The process begins with every frequent pattern-1 and its conditional pattern base is built, which is a subset of prefix paths within the FP-Tree co-occurring with the suffix pattern. Afterwards his conditional FP-tree was built and recurrent mining was done on the tree. The process begins with item I_5 as the final item in descending order, when mining is done on the aforementioned tree. In Figure 3.4, I_5 takes place in two FP branches. Those branch paths are I_2, I_1, I_5, I_2, I_2, I_1, I_3, I_5. The branch paths are I_2. Thus when I_5 is suffixed, it is {I_2, I_1:1} and {I_2, I_1, I_3:1} as the conditional model base. The conditional FP tree is constructed using this conditional pattern basis as a data base for a transaction.

Item	Conditional Pattern Base	Conditional FP-Tree	Frequent Patterns Generated
R_4	{(R_2, R_1:1), (R_2:1)}	{R_1:2, R_2:2}	{R_2,R_5:2, R_1,R_5:2, R_1, R_2,R_5:2}
R_2	{(R_2, R_1:1), (R_2:1)}	{R_2:2}	{R_2, R_4:2}
R_3	{(R_2, R_1:2), (R_2:2), (R_1:2)}	{R_2:4, R_1:2}, {R_1:2}	{R_2, R_3:4, R_1, R_3:4, R_2, R_1, R_3:2}
R_1	{(R_2:4)}	{(R_2:4)}	{R_2, R_1:4}

Table 3.3 Frequent pattern generated from FP-tree

The tree contains only one {I_2:2, I_1:2} path. I_3 is left because its count of support is 1 and its threshold is lower. This is the single path:-{I1, I_5;2}, {I1, I_5:2} and {I_2, I_1, I_5:2}and the following combinations of frequent patterns. Table 3.3 above sums up the mining of the tree. If the FP-tree's size is small enough to fit into the primary memory, it can directly extract frequent articles in the best case scenario, where every transaction has the same array of elements, FP-tree only contains one branch of nodes. However, the requirement for FP–tree physical storage is higher because it takes extra space between nodes and counters for each item.

3.3.9 Co-Occurrence Frequent Item-Tree (COFI-Tree) Algorithm

COFI tree is an algorithm which uses four new ideas to achieve its productivity. To begin with, it Can use a compact memory-based data structure. Second, a relatively small autonomous tree is produced for each incessant assignment, summarizing co-events. Third, smart pruning reduces the search space dramatically. Finally, a basic and non-recursive mining process reduces memory needs as the generation of candidacy is less expected and tallying should generate all pertinent patterns. In the main phase, the algorithm works in two phases, two full transactional database I / O scans are required to assemble the structure of the FP-Tree.

This development takes place in two stages, where each progression requires a complete database I / O scan. The successive 1-itemsets are recognized by a first database scan. The goal is to generate a requested rundown of continuous things to be used when making the tree in the second step in the second step, using the rundown generated in the initial step, visit pattern tree structure is created. The second phase begins with the construction of small frequent trees of co-occurrence for each continuous thing, to begin

with, these trees are pruned to eliminate any non-visiting things as for incessant things based on the COFI-tree. The mining procedure is performed after this.

The COFI-tree algorithm's main advantage over FP-Growth is that it needs a much smaller memory impression and can mine larger transactional databases along these lines with smaller main memory available. The fundamental contrast is that COFI is trying to find a deal between a completely pattern development approach followed by the FP-Growth algorithm and an approach to the generation of total candidacy followed by the Apriori algorithm. COFI develops targeted patterns while playing in the midst of mining a lowered and centred generation of candidates. In large databases [67], this avoids the problem of recursion and stack flood [108].

3.3.10 CT-PRO Algorithm

This algorithm uses CFP-tree as an alternative tree structure. It applies non-recursive implementation and in a base-up strategy it crosses the tree. The data structure of the CFP-tree is used to represent transactions in the memory compactly. The algorithm works in the initial step in three stages, the second step identifies things whose recurrence is higher than the least help edge, the global CFP-tree is built with the regular things found in the initial step. Finally, mining is carried out on the global CFP-tree in the third step by creating local CFP-Tree. Compared to FP-Growth, one major advantage of CT-PRO is that CT-PRO avoids the cost of conditional FP-Trees. At each progression of its recursive mining, FP-Growth needs to create a conditional FP-tree. This overhead adversely affects its performance, since the quantity of conditional FP-Tree is in any case compared to the quantity of continuous itemsets, only one local CFP – Tree is created and traversed in CT – PRO for each incessant thing, not – recursively to remove any continuous item kits beginning from this regular item [109].

3.3.11 H-Mine Algorithm

H-mine is another algorithm which is the changeover FP-Tree algorithm in this method, anticipated database is created utilizing as a part of memory pointers. H-mine makes utilization of a novel hyperlinked data structure called H-struct and mines visit itemsets by adjusting joins dynamically. Before mining is performed, H-struct is created from the database. For this reason, database is scanned and a rundown of regular things is revealed. Another rundown called F-list, is created by arranging these continuous things in alphabetical request. At that point visit thing projections of all the transactions in the database are arranged in the F-list arrange. After that H-struct data structure is created in

this, a header table is created which contains the one passage for each continuous thing in the F-list. Each passage in the table also has a connection part that associates all the transactions containing that particular thing. At the point when the continuous thing projections are loaded into memory, those with the same first things in F-list arrange, are connected together by the hyperlinks as a line and the passages in the header table acts as the heads of the line.

When H-struct is created, the mining is performed on H-struct just without scanning the original database. One special feature of this algorithm is that it has exceptionally limited and totally predictable overhead space and runs quickly in memory-based conditions. The database partition can also be scaled to large databaseand FP-Tree can be built dynamically as part of the mining process when the data sets finish noticeably thick. H-mine performs superior to anything FP-Tree and Tree Projection when it is utilized for mining sparse data set because it has polynomial space unpredictability and is along these lines more space productive.

3.3.12 Rapid Association Rule Mining (RARM) Algorithm

RARM is another association which decides to use the tree to represent the original database, using an algorithm that avoids the handling of the candidate generation. This is a much faster algorithm than the FP-Tree algorithm. It uses a Support Trie Itemset (SOTrieIT) tree structure. Without second scanning of the database and without a candidate, RARM quickly generates large 1-itemsets and 2-itemsets. Every SOTrieIT hub contains one thing and its comparative support, similar to FP-Tree. The handle for large articles in two phases works. The primary phase is pre-processing phase, where database is scanned to build the TrieIT. Like FP-tree, if the thing under consideration is already in the tree, the help tally is increased by one and if not, another hub is created with help check set to one.

The major distinction between FP-tree and SOTrieIT is that in SOTrieIT the help number of just leaf hub is increased yet with FP-tree the help include of all the hubs the path is increased in the second phase, for generating large itemsets, the SOTrieIT tree is scanned top to bottom initially arrange. The scanning starts from furthest left initially level and move to second level hub whose help does not satisfy the base edge. After that it backtracks and moves to another first level hub. Since generating large 2-itemsets is the most expensive process amid the mining procedure, enhancements in this phase increases overall effectiveness of the algorithm.

3.4 Increasing the Efficiency of ARM

Several algorithms have been made from its origin in order to identify valuable association rules from the data available. Many times, the algorithms often produce thousands or even millions of unmanaged principles. There is also a broad relationship between several things between association rules. For customers, understanding such a large number of complex directives is virtually incomprehensible and hence limits the usefulness of data mining [110].

One major concern for researchers engaged in data mining research is the ability of association lead mining algorithms. Effectiveness is measured in terms of runtime required to generate rules, storage needs, number of database scans performed, number of candidate itemsets generated, number of standards generated and so on. The reduction of mining computation costs is therefore of the utmost importance [111].

Reducing the number of passes over the database: Many algorithms have been designed that focused to reduce the number of passes or scans over the database. The repeated scans of database are the major factor that contributes to high time consumption and reduced efficiency. FP-Tree algorithm was the first algorithm that reduced the number of passes to only two and eliminated the candidate generation step. Tree Projection is another algorithm proposed over time that efficiently shrink large itemsets generation time by scanning the database once or twice.

Sampling the database: Instead of performing the scanning and mining activities over the actual large database, sampling could be used to reduce the size of search space in addition to in turn reduce the overall time. In ARM algorithm the first sample the database and find all the associations in the sample. The results are then validated against the entire database. This improves the results and efficiency of the mining process.

Adding extra constraints on the structures of the pattern: Adding extra constraints regarding the pattern to be discovered also helps in increasing the efficiency of the mining process. These constraints can be categorized as post-processing constraints, pattern filtering and data filtering constraints. They filter out the patterns that don't satisfy the user-specified pattern constraints during actual mining process or after the discovery process. RARME is a tree structure process that only users have the rules that are important to users, instead of all rules, to represent their database and to mine them. RARM.

Using Parallelization: The higher speed and greater storage capacity of parallel frameworks attracted the researchers to utilize the parallelism approach in association administer mining as well. FDM algorithm is a parallelization of the Apriori and is an effective algorithm that partitions the database over autonomous machines and performs scans on local partitions. After that a disseminated pruning system is utilized to accumulate the outcomes. One extreme issue in the association is to make the provision of association directives unmanageable, especially when the aid and certainty are small, as the quantity of transactions grows.

Many approaches that attention on increasing the effectiveness of the mining procedure have to make utilization of lower values of help and certainty limit and along these lines unintentionally wind up with enormous amount of itemsets. The amount of items presented to the customer typically increases proportionately as the quantity of incessant products increases. Many of them may be redundant.

3.5 Recent Advances in Association Rule Discovery

With passage of time and advancement in technologies and tools, the mining approaches have also been revolutionized. Lots of advancements have been noted in rule discovery and pattern mining which includes redundant rules, negative association rules and alternate measures of interestingness.

3.5.1 Redundant Association Rules

To tackle the drawback of rule redundancy, lot of research has been performed that tries to remove the redundant rules either on the basis of user-specified constraints or with some inference systems that prunes redundant rules, which have found rules on association with different formats or properties, have proposed a new framework for mining association rules. It is proposed to identify, firstly and then eliminate the redundant rules, the rules that have the same significance. However, the methods never lose high trust or interesting rules.

3.5.2 Negative Association Rules

Positive association rules only consider the association among the items present in the transaction but negative association rules in addition to this also consider the items absent from the transactions. Such negative correlations are helpful for analysing patterns or products that conflict. The term negative association rule was proposed for the first time. They use statistical tests to verify the type of correlation among the data.

3.5.3 Alternate measures of interestingness

Support and certainty are two popularly utilized measures of intriguing quality. Be that as it may, several different measures can be utilized to check, regardless of whether a control is fascinating or not. Coverage, lift, leverage, all-certainty, bond and conviction are a few measures that many researchers utilized for measuring intriguing quality. Some of these measures like bond apply just to subsets of databases that are created on the basis of certain time constraints. Conviction measure resembles chi-square test for correlation which is utilized as a part of place of certainty measure after the principles are generated. It is a measure of implication and not simply co-event. Liu et al. presented the idea of utilizing distinctive least backings indicated by the client for various things.

3.6 Graph Based Approach for Frequent Itemsets to Discover Association Rules

3.6.1 Overview

Finding regular itemsets is a major phase of mining administration by the association. Many of the algorithms created to locate regular object sets scan the database repeatedly and are based on the idea of the lowest bolter value. The approach proposed is based on a chart and finds items for visits without repeatedly scanning the database. In the first phase, the proposed approach performs a single scan in the database and draws a graph in which the edge is labelled with the individual transaction ids in the second stage, a table is built with all specific labelling and comparative itemsets.

The algorithm uses this method to find the largest set of items in sequence regardless of the level of assistance. From this table, the largest continuous itemset is selected based on the given selection basis. Two major drawbacks of past approaches are beaten by the proposed approach. These disadvantages are repeated scans of the database and need to repeat the whole procedure if the limit value of the base help changes. The method adopted is basic and straightforward to recognize the largest successive itemsets. To represent the entire database, a graph is used.

The graph can only be created once with a database scan. Changing the base assistance (limit value) does not affect the built-in graph or table, so no additional effort is necessary. It is not possible to detect the number of visit items unless the candidate items are generated, thus reducing the number of repeat data bases that reduce the runtime of the visit and memory consumption. The procedure is also very good where the database is constantly updated as it is just necessary to add a few hubs and borders to graphs or tables.

3.6.2 Proposed Approach

As mentioned previously, the most tedious and important part of preparing control disclosure is the divulging of large, uninterrupted items. The database shall be organized and the plate continues as a set of transaction records. Multiple circumstances are scanned in order to discover the following itemsets, calculate the aid of various arrangements and compare the help limit value. Another graph-based approach is proposed in order to distinguish between time sets to decrease the data base scan and directly generate visit items. The approach operates in three phases: Graph generated by items.

Transaction Id	Items
R1	P, R, S
R2	Q, R, T
R3	P, Q, R, T
R4	Q, T

Table 3.4 Sample Database

Corresponding to the graph and finding the frequent Itemsets. The phases of the proposed approach are outlined in the following sections.

3.6.3 Graph Construction Phase

A graph of the things in the database is constructed as an initial step. Things are represented by nodes. An edge between them associates the hubs having a place with the same transactions. The related transaction Id marks this edge. On the off chance that an edge has already associated the hubs compared to the same transactions, new transaction Id is added to the current label at that point. The edge label therefore contains all the transaction ids in which the two things (end edge hubs) are part of the related transactions. May there be five things P, Q, R, S, T and four R1, R2, R3 and R4 (or 1,2,3 and 4) transactions in the database.

- Draw as many nodes in the graph as the items in the database and represent each node with the name of the item.
- For each transaction, repeat the following steps
- By Finding all the unordered possible node pairs corresponding to the items of the same transactions.
- By drawing an edge between all these node pairs.

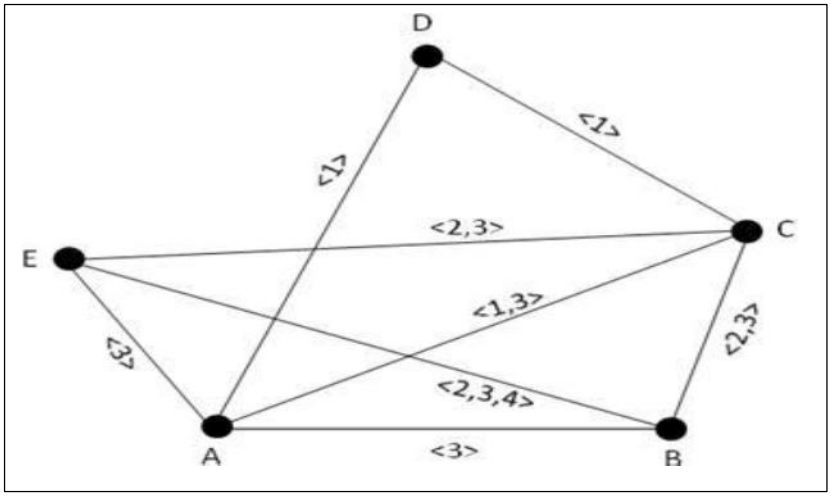

Figure 3.5 Graph of Sample Database (Source [141])

In Figure 3.4 there are three things P, R and S to start with transaction R1. There exist three conceivable unordered pairs of things (hubs) – (P, R), (P, S) and (R, S) compared to this transaction. Label < 1 > related to transaction Id R1 is assigned to the edges along with the end hubs (P, R), (R, S) and (R, S). The second R2 transaction includes three things: Q, R and T.

The unordered hubs pairs are (Q, R), (Q, T) and (R, T) related to this transaction. Label < 2 > is assigned to the three edges in the graph with end hubs (Q, R), (Q, T) and (R, T). he final hubs (P, Q), (P, R), (P, T), (Q, R) and (R, T) of the R3 transaction are also drawn to a six edges of the same chart.

The edges (P, Q) and (P, T) were drawn for the first time so that the edges were labelled < 3>. Due to the handling of the R1 and R2 transactions the edges (P, R), (Q, R), (Q, T) and (R, T) have already been drawn to update and add to current labels the edges of the labels for these edges < 1, 3 >, < 2, 3 >, >, < 2, 3 > and < 2, 3 > respectively shown in Figure 3.5. The comparative transaction Id <3 > are added. The comparing edge (Q, T) has already been drawn for transaction R4, thus adding the transaction Id 4 to the current label < 2, 3 > and updating it to < 2, 3,4>

3.6.4 Generation of Item sets

After the above-mentioned process builds the graph, a table is created to find the most common itemsets. Table of Headings-Label, Length (L), Frequency (F), L×F, Borders and set of items.: The table contains six columns:

Label	Length (L)	Frequency (F)	L×F	Edges	Itemset
(2,3)	3	1	3	(Q, T)	{Q, T}
(2,3)	2	2	4	(Q, R) (R, T)	{Q, R, T}
(1, 3)	2	1	2	(P, R)	{P, R}
(3)	1	2	2	(P, Q) (P, T)	{P, Q, T}
(1)	1	2	2	(P, S) (R, S)	{P, R, S}

Table 3.5: Calculation of Frequent Itemsets

For the above Table 3.5, the attributes are defined as below:

- Label: It represents the label of the edges generated by the Apriori algorithm.
- Length (L): It denotes the length of the Label. It is number of the transaction Ids which are part of the Label.
- Frequency (F): It is number of occurrences of the Label in the graph.
- L×F: It is product of L and F and it represents the selection criterion for finding the frequently occurring item sets. Higher the value of L×F, the corresponding itemset is likely to be more frequent.
- Edges: This column represents the edges that have been labelled with the given Label.
- Itemset: It represents the corresponding itemset. It contains distinct items which are part of the corresponding Edges.

3.6.5 Finding the Frequent Itemsets

Strategy for finding ceaseless itemsets is straight forward. When the table is created, visit itemsets can be found by breaking down the proposed assurance principle of L×F. The line having most astonishing estimation of L×F gives the biggest and the most standard itemsets in our model, the most surprising estimation of L×F is allowed continuously line of the Table 3.3, thusly the most progressive biggest itemset is – {Q, R, T}.

This standard itemset is enrolled paying little respect to some random utmost. The equivalent itemset is handled when Apriori calculation, with assistance limit estimation of 2, is connected to the example information as given in Table 3.2 if one is busy with finding the most progressive itemset of length (gauge) 2, it very well may be effectively browsed the Table 3.3. The most progressive itemset of length 2 is {Q, T} spoken to by first line of the table. It identifies with the most raised estimation of L×F for the itemsets having length 2.

3.6.6 Efficiency of the Proposed Approach and Comparison with Existing Approaches

From the database scans perspective, the proposed approach is productive. The vast majority of past methods perform repeated database scans that therefore waste CPU time and result in overall performance degradation. Another standard for proficiency measurement is the impact of aid edge value change.

Several past methods, including Apriori, require a repeat of the overall strategy if the edge value determined by the client is changed in any case once a graph and table are developed to successfully find incessant itemsets, the most common objects of a particular length can be found (without any type of adaptation) (for changing the edge value). Applied to the sample databases discussed in the proposed approach revealed the same incessant itemsets in a single database scan as created in multi-database scans by their individual approaches [112].

Using the proposed approach, it is possible to discover even the following most regular itemsets that are not examined in previous approaches. The following most incessant itemset compares for any given length to the next most noteworthy value of L/F. The proposed approach can be specifically applied to databases of small and medium size. Vertical partitioning can be used as a pre-processing venture to sift through the rare things for large databases. This step of pre-processing will make manageable the quantities of hubs and edges.

3.7 Applications of Association Rule Mining

3.7.1 Market Basket Analysis

A market basket is generally defined as a set of products in a single transaction purchased by a customer. The quantity of items purchased in a transaction of a given type is less important. The interest lies in different types of purchased items. It is necessary to analyze the accumulated transaction collection of several customers recorded since long time. he market basket is defined as a set of items purchased on a single visit to a store by a customer together. MBA is a powerful tool for cross-selling strategies to be implemented. Discovering large baskets is particularly important in retailing, as it deals with thousands of items. Market-basket analysis is a process used to analyze the habits of buyers to identify the relationship between different products in their market basket. Finding out these

relationships can help the dealer develop a sales strategy by taking into account the products often purchased together by customers [113].

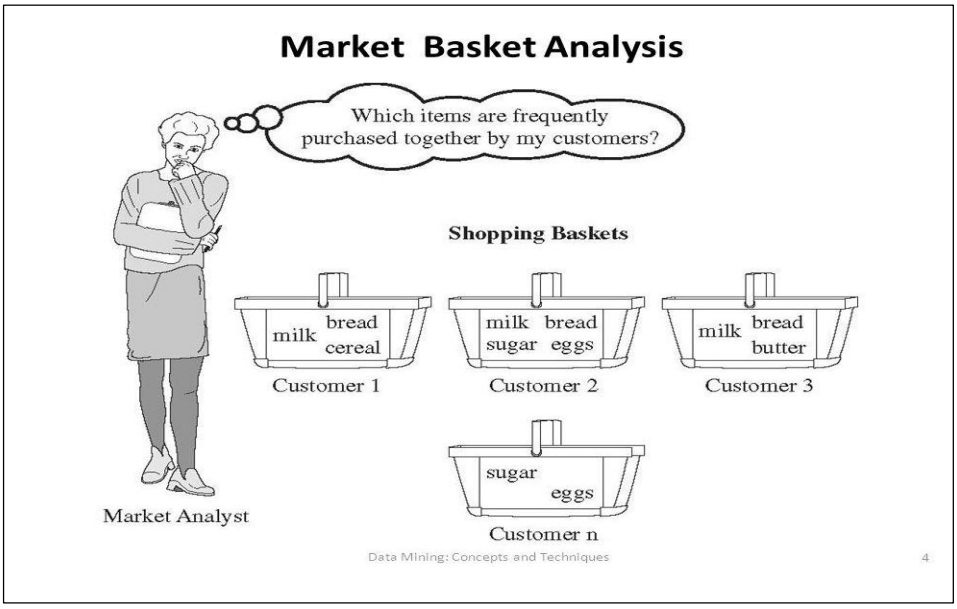

Figure 3.6: Market Basket Analysis (Source [142])

The data mining method is used to analyze the buying habits of customers using a MBA method. The buying patterns may be used in order to improve the positioning or layout of the catalog pages for those items in the supermarket. This has led to the development of techniques that search automatically for associations among the items that are stored in databases.

In unattended data mining systems, association rules are widely applied for discovery of local patterns. MBA is a modelling technique based on a theory that a customer can more or less buy another group of items when buying a specific group of items. MBA is the modelling technique. For instance, when a client is in a pub in England and buys a beer pint and doesn't buy a bar food, at the same time the client is much more likely to buy chips than anyone who did not buy beer. The number of items purchased by customers is known as a number of items and MBA attempts to identify connections between purchases.

A typical MBA is shown in the following figure. This illustrates perfectly how the mining of associations is shown. It is a fact that every manager in a shop or department store wishes to know about the purchasing behaviour of each customer. This system of MBA helps managers to understand what sets of items customers are likely to buy. All customer transaction details in retail stores can be analyzed for this. These findings guide

you to plan your respective marketing or advertising approaches. For example, MBA will also help managers propose new arrangements for shop layouts. This analysis allows items regularly bought from each other to be placed close to one another to further promote the sale of such items. If consumers who buy computers are also likely to buy virus software simultaneously, it helps to increase both items ' sales by placing the hardware display close to the software display. The aim of the MBA is to find sets of "associated" items and the fact of their partnership is often referenced as a rule of association. Related objects often appear together intuitively. Three more accurate association measures have been used,

- **Support:** In many baskets, the items must appear.
- **Confidence:** Because the others are in the basket, the probability of one item must be high.
- **Interest**: If the items were purchased at random, this probability must be suggestively higher than or lower than the probability expected.

Most shopping is bought at retail impulses. MBA gives insights and ideas on how to buy a customer if the idea is given. It is therefore possible to use MBA to decide where products are located and promoted in a supermarket. If a customer buys Barbie dolls, the customer will be more likely to buy candy and then high-margin candy can be placed near the Barbie doll display. Different basket analysis can find interesting associations and relations and can also eradicate the problem of a possibly large volume of trivial outcomes. This is the first level of analysis however, Customer-to-customer results are compared between different weekdays and different time-seasons in different demographic groups in differential analysis. If the rule has something interesting about that store, but does not hold something else (or does not hold it in one storeroom, but holding it in all others).

It could be due to its customers, or a new and more advantageous way to display it. Investigation of these differences can provide useful insights to improve company's sales. In order for doctors to treat the patients, applying association rules in the diagnosis process. The common problem with inducing sound diagnostic rules is complicated, because the exactness of induced hypotheses is hypothetically not possible alone.

3.7.2 Medical Diagnosis

Medical diagnostics would be helpful for physicians when applying the association rules. The common problem of inducing sound diagnostic rules is complex because no induction process can be hypothetically accurate by itself.

3.7.3 Census Data

The census provides both researchers and the general public with a wide range of common statistical information on society. In addition to public business, public information (for startup new factories, shopping malls, or banks and also for marketing certain products) can be predicted in the planning of public services (education, health, transport, funds), as well as for public companies. The use of data mining techniques in census data and, in general, administrative data has enormous potential to support good public policy and the effective operation of a democratic society [114].

3.7.4 Agriculture

In agricultural precision, resource discovery, community planning and other areas, the extraction of interesting patterns and rules from farm data sets may be of significant importance. Alternatively, it is not undemanding and still in the early stages it requires demanding methodological research. information mining procedures can adequately be connected to discover the connections to clients ' obtaining conduct and to build deals among the things in an extensive database. Besides, these affiliation leads on mining can likewise be utilized to pick the correct harvest for the correct zone, in view of topographical conditions and compost levels, for expanding crop creation [115].

3.7.5 Other Application Areas

Although MBA are mainly applicable to shopping carts and supermarket shoppers, the fact that they can be used in many other areas is important to realize. This includes:

- Analysis of purchases by credit card
- Analysis of telephone calling patterns
- Identification of fraudulent claims for medical insurance
- Analysis of purchase by telecom service

There is no need to buy all the items simultaneously. The algorithms can monitor the time sequence-buying (or events) spreads. A Predictive MBA can be used to find sets of items that usually occur during a series of purchases, which are of interest to direct marketers, criminologists and many others.

3.8 Regression

Regression is a data mining function that predicts a number. It is all possible to predict the use of regression, age, weight, distance, temperature, income or sales. For example, in view of their age, weight and other factors, a regression model could be used to predict the height of the kids. A regression task begins with a set of data that knows the destination values. For example, over the course of a period, using observed data for a number of children a regression model which can predict the height of the child could be developed.

The statistics could track age, height, weight, development milestones, history of the family, etc. The goal would be height other attributes would be the predictors and data for each child would be the case. Regression analysis is a statistical process for estimating the relationships between variables. It contains several techniques for modelling and analyzing multiple variables when concentrating on the relationship between a dependent variable and one or more independent variables (or "predictors"). The variation of the variable that depends on the regression function is also relevant for the regression analysis and can be described by probability distributions.

Many techniques were developed for regression analysis. family methods like linear regression and regular minimum square regression are parametric in that the regression function is defined by a limited number of unknown data parameters. Non-parametric regression refers to techniques that enable the regression function to be in an infinite-dimensional set of functions.

In practice, the performance of regression analysis methods depends on the way in which the data are generated and how the regression approach is used. Since the true form of the data-generator process is not commonly known, the analysis of regression often depends to some extent on making assumptions about this process. These assumptions can sometimes be tested if sufficient information is available. However, regression can lead to misleading results in many applications, particularly with small effects or causal problems based on observational data [116]. Prompt regression models often are useful even if moderately breached.

3.8.1 Regression Models

Regression models involve the following variables:

- The **unknown parameters**, denoted as **B**, which represent a scalar or vector.

- The **independent variables**, denoted as **X**.
- The **dependent variable**, denoted as **Y**.

Different terminologies are used in different fields of application instead of dependent and independent variables.

A model of regression relates Y to X & B function.

$$\mathbf{Y=f\ (X,\ B)} \qquad \text{---------- (4)}$$

The approximation is usually officially done as E (Y / X) =f (X, B). To perform regression analysis, the form of function f must be specified. The form of this function is sometimes based on knowledge which does not rely on information about the Y&X connection. If such knowledge does not exist, a flexible or convenient f-form is chosen. Suppose now that the unknown parameter B vector is k in length. To perform a regression analysis, the user must provide information about the dependent variable Y(Eq.4):

- If N data points of the form (Y, X) are observed, where N < k is not possible to perform most classical regression analysis approaches: since the regression model defining equation system is undetermined, there is not enough data to recover B.

- If N = k data points are accurately observed and the f function is linear, the Y = f (X, B) equations can be resolved exactly rather than approximately. This reduces the resolution of a set of unknown N equations (elements of B), which has a unique solution as long as the X is linearly independent. If f is not linear, there may not be a solution or there may be many solutions.

- N > k data points are observed at most common situation. In this case, the data contain sufficient information for estimating a unique value for B which in some sense best fits the data and when applied to the data, the regression model can be viewed as an over determined B system.

In the final case, the regression analysis provides the tools for:

- To find a way to minimize the distance between measured and predicted Y variable values (also known as the least square method) for unknown parameters B.

- In order to provide statistical information about unknown parameters B and predicted values of dependent variable Y, the regression analyzes uses information surplus from certain statistical assumptions.

3.8.2 Regression Workflow

The mathematics used to create quality data recovery models for regression analysis do not have to be understood. Be that as it may, it is useful to comprehend certain basic ideas. The motivation behind relapse investigation is to distinguish a capacity's parameter esteem that best fits a lot of information perceptions you give. The relapse investigation These connections are communicated in the accompanying images. Relapse is appeared to be a methodology for assessing the incentive for the consistent goal (y) (Eq.5) as the capacity (F) of at least one indicators (x1; x2; xn), parameter set (e), blunder estimation (e), parameter set (e).

$$y = F(x, \theta) + e \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

The regression model training process involves finding the best parameter values for the function to minimize an error measure, such as the sum of squared errors

3.8.3 Types of Regression

There are dissimilar regression function families and different ways to measure the error [153].

- **Linear Regression**

Linear regression with a single predictor is the simplest form of regression to view. If, according to the Figure below, the relationship between x and y is approximated to a line, a linear regression technique can be used [103].

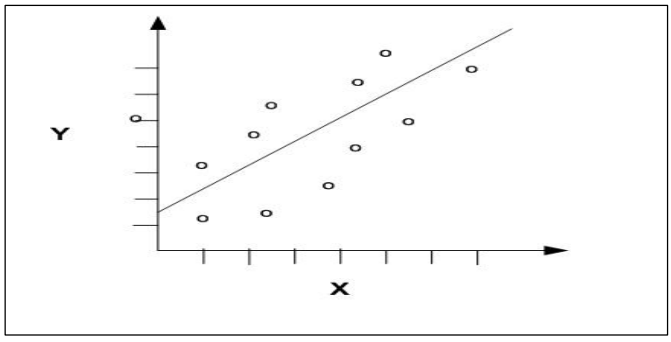

Figure 3.7: Linear Regression (Source [143])

Linear regression, generally is one common technique for analysis of statistical data. It is useful for determining the extent to which a dependent variable has a linear relationship

with one or more independent variables. There exist three linear regression types. These are:

- Simple linear regression
- Multiple linear regression.
- Multivariate linear regression [117]

In linear regression, a single variable is used to predict the value of a dependent variable.

or could say that the simple regression is related to one dependent variable (y) and one independent variable (x). Two or more separate variables are used to predict the value of a dependent variable in multiple linear regression. The difference between these two variables is the number of separate variables. Only one dependent variable exists in both cases or can say that, **Multiple regression (multivariable regression)** pertains to **one** dependent variable and **multiple** independent variables:

$$\boxed{y=f(x)} \quad \text{---------- (6)}$$

Multivariate regression is a technique that estimates more than one outcome variable for a single regression model. If a multivariate regression model contains more than one predictor variable, the model is a multivariate multiple regression.

In Multivariate regression, multiple dependent variables and multiple independent variables are involved: y1, y2, ..., ym = f (x1, x2, ..., xn). You may encounter problems where both dependent and independent variables are arranged as variables matrices (e.g. y11, y12, ... y11, y12, ... and x11, x12, ... x11, x12, ...), so that the expression can be written as Y = f(X) (Eq.6), where capital letters indicate matrices.

The parameters of regression (also called coefficients) are used in the linear scenario with a single predictor ($y = \theta_2 x + \theta_1$):

The **slope** of the line (θ_2)-the angle between a data point and the regression line and The ***y* intercept** (θ_1)-the point where x crosses the y axis ($x = 0$)

There are many practical uses of linear regression. Most applications fall into one of the following two broad categories:

- If the goal is prediction, or prediction, a linear regression can be used to match an observed y and X data set predictive model. When an additional value of X is given

after developing such a model, the adapted model can be used to predict the value of y without its supporting value of y.

- Because of the variable y and a number of variables X1, ..., the linear regression analysis can be carried out in relation to y, to quantify the strength of the y-Xj relationship, to estimate which Xj is not at all associated with y and to identify what sub-sets of Xj include redundant y information. For example, there is an independent variable in simple linear regression for modeling n data points: xi and two parameters, B0 and B1:

$$\text{Straight line: } y_i = B0 + B1x_i + e_i,\ i=1, 2, \ldots\ldots\ldots n. \qquad \text{---------- (7)}$$

There are several independent variables or functions of separate variables in multiple linear regression.

Adding a term in xi^2 to the preceding regression gives:

$$\text{Parabola: } y_i = B0 + B1x_i + B2x_i^2 + e_i,\ i = 1, 2, \ldots. n \qquad \text{---------- (8)}$$

While the expression on the right side is quadratic in the independent variable xi, the expression on the B0 (7) and B1 (Eq.8) parameters is linear in B2 parameters. In either case, ei is a mistake and the subscript I indexes a specific observation.

- **Non- Linear Regression**

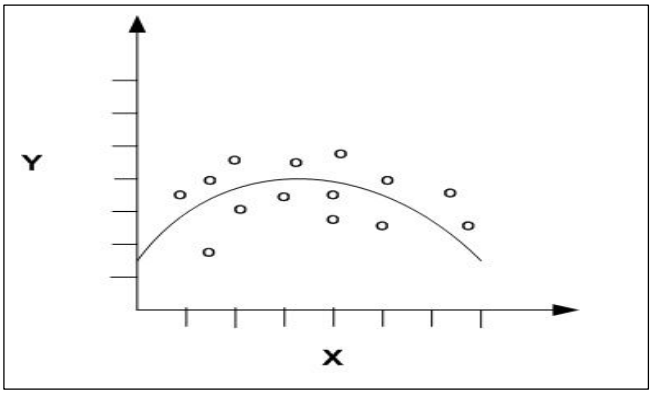

Figure 3.8: Non-Linear Regression (Source [143])

Often the relationship between *x* and *y* cannot be approximated with a straight line. In this case, a nonlinear regression technique may be used. Alternatively, the data could be pre-processed to make the relationship linear. Nonlinear relapse is a type of relapse

examination where observational information is demonstrated by a non-straight mix of model parameters, contingent upon at least one factors. A strategy is utilized to fit the information with progressive approximations. Nonlinear relapse is a method used to portray nonlinear connections in exploratory information. Nonlinear relapse models are commonly thought to be parametric where the model is portrayed as nonlinear. For nonlinear non-parametrical relapse, AI techniques are typically utilized.

The needy variable (likewise called reaction) is a blend of the nonlinear parameters and at least one freely (called indicators) parametric nonlinear Regression models. The model may either be univariate (single variable answer) or multivariate. Exponential, trigonometric, control or other nonlinear capacity can be parameters. To decide the nonlinear parameter appraises, a run of the mill iterative calculation is used.

$$\mathbf{Y=f\ (X,\ B)\ +\ e} \qquad \text{---------- (9)}$$

Where, B (Eq.9) is the calculation of nonlinear parameter estimates and e is the error terms. All the models were talked about so far were linear in the parameters (i.e., linear in the beta's). For example, by using higher-ordered predictor values, polynomial regression was used to model curvature in our data. The final regression model, however, was only a linear combination of higher-ordered predictors.

The popular algorithms for fitting a nonlinear regression include:

- **Newton's method:** A classic approach based on a gradient approach that can be highly dependent on good starting values and computationally challenging.

- **The Gauss-Newton algorithm**: A modification of Newton's method which provides a good approximation of the solution which Newton's method should have achieved but which is not guaranteed to converge.
- **The Levenberg-Marquardt method**: Which can handle computer problems with other methods but may require a tedious search for the optimum value of the tuning parameter.

3.8.4 Difference Between Linear and Non-Linear Regression

- **Linear Regression**

Response = constant + parameter * predictor + ... + parameter * predictor

$$Y = b_o + b_1X_1 + b_2X_2 + ... + b_kX_k \quad \text{---------- (10)}$$

A model is linear when a parameter and a predictor variable are either a constant or the product of each term. By adding the results for each term, a linear equation is constructed. This limits the equation to one basic form only. In statistics, if it is linear in the parameters, a regression equation (or function) is linear (Eq.10). While in the parameters the equation must be linear, the predictor variables can be transformed in ways that produce curvature. For example, to produce a U-shaped curve, you can include a squared variable

$$Y = b_o + b_1X_1 + b_2X_1^2 \quad \text{---------- (11)}$$

Although the predictor variable is squared, the parameters of this model are still linear. In parameters that are linear, you can also use log and reverse functional forms to produce different types of curves (Eq.11).

- **Non-Linear Regression**

A linear equation has one fundamental form, there are many different forms that nonlinear equations can take. A focus on the term "nonlinear" itself is the easiest way to determine whether an equation is nonlinear. It's not linear, literally. If the equation for a linear equation does not meet the above criteria, it is nonlinear. This covers many forms, so nonlinear regression provides the most flexible curve–fitting function. These are the parameters and X represents the predictor in nonlinear functions. In contrast to linear regression the functions can have more than 1 parameter per variable predictor [118].

3.8.5 Some other types of Regression

Logistic Regression: It models a relationship between predictor variables and a categorical response variable. For example, we could use logistic regression to model the relationship between various measurements of a manufactured specimen (such as dimensions and chemical composition) to predict if a crack greater than 10 mils will occur.

Logistic regression helps us estimate a probability of falling into a certain level of the categorical response given a set of predictors can choose from three types of logistic regression, depending on the nature of the categorical response variable [119].

3.8.6 Common Applications of Regression

Regression modelling has numerous applications in trend analysis, corporate planning, and commercial planning, financial forecasting, time series prediction, biomedical and drug reaction modelling and environmental modelling. Regression analysis is commonly used to predict and predict when its use significantly overlaps with machine education. Regression analysis is also used to understand and explore forms of the relationships related to the dependent variable between the independent variables. In limited circumstances, regression analysis can be used to deduce causal relations between the independent and dependent variables.

3.9 Proposed Methodology

The proposed methodology, methodology of research and problems in existing techniques are discussed in this chapter. Apriori's algorithm and regression are explained in the research methodology. The technical overview, the flow chart and the algorithm used in this research are briefly discussed here. The suggested algorithm for this chapter is Regression Apriori for improved analysis of market basket. Another technique for MBA dependent on the Apriori calculation. This shows an adjusted adaptation of Apriori's outstanding Data Mining calculation, which guides clients in choosing the most ideal blend of products in a littler grocery store. This methodology essentially analyzes the way toward discovering affiliation runs in less time in expansive archives of this sort. Most association rule mining calculations experience the ill effects of unnecessary execution time issues and create such a large number of affiliation rules.

While customary Apriori calculations can recognize important itemsets and construct affiliation rules, they endure the disservice of producing various applicant itemsets that should be more than once appeared differently in relation to the whole database. A lot of memory is likewise used to process the regular calculation. This methodology is accordingly critical for successful market bin investigation and enables clients to purchase their things all the more easily, which thus expands the market deals rate.

3.9.1 Problem Specification

One of the most investigated techniques of database mining is association rule mining. It aims for interesting relations between transaction databases and other data repositories, frequently used patterns, associative or informal structures. The mining rule

Association is widely used in different fields such as market, medical diagnosis, agriculture, etc. The Apriori algorithm is one of the fundamental and standard assembly rules mining algorithms. Apriori is a proficient mining decide affiliation calculation that investigates mining at a dimension level. For Boolean affiliation rules, A powerful calculation for incessant mining thing sets is the Apriori calculation. All thing blends happen with a base recurrence in various exchanges. Regularly these blends are called itemsets. Apriori ascertains the likelihood that a thing is available in various things since different things are available.

Apriori discovers designs with recurrence over the base opposition limit. In this manner, to discover relationship with uncommon occasions, the algorithm needs to keep running with extremely low least help esteems. Be that as it may, this could prompt the blast of the quantity of things recorded, particularly for extensive quantities of things. This could fundamentally build the runtime. The reason for the Apriori calculation is to discover the connections between various informational indexes. The supposed advertise container examination each dataset has various things and is known as an exchange. The yield of Apriori is a lot of principles that tell how regularly things in informational collections are incorporated. Two techniques are usually used by Apriori Algorithm. One of these is to reduce the number of passes in the entire database or replace only a fraction of it, based on the existing frequency sets, while the other technique is to use various categories of size methods to make a much smaller number of items.

The following are Apriori Algorithm's major demerits.

- Apriori follows breadth-first search strategy to calculate itemset support and uses a function to generate candidates that follows the support's downward closure property.
- It generates lots of candidate sets and uses most of the time, space and memory for its complex process.

3.9.2 Proposed Methodology

This section addresses the technique of regression. For MBA, the proposed approach uses effective association rule mining techniques

- Apriori Algorithm
- Regression

Regression algorithms predict one or more continuous numerical variables based on other attributes in the dataset, such as profit or loss. Regression models are tested to measure the difference between forecast and expected values by calculating the different statistics. The proposed methodology uses apriori-algorithm regression techniques to remove the unwanted pattern from the supermarket and find frequent patterns in less time.

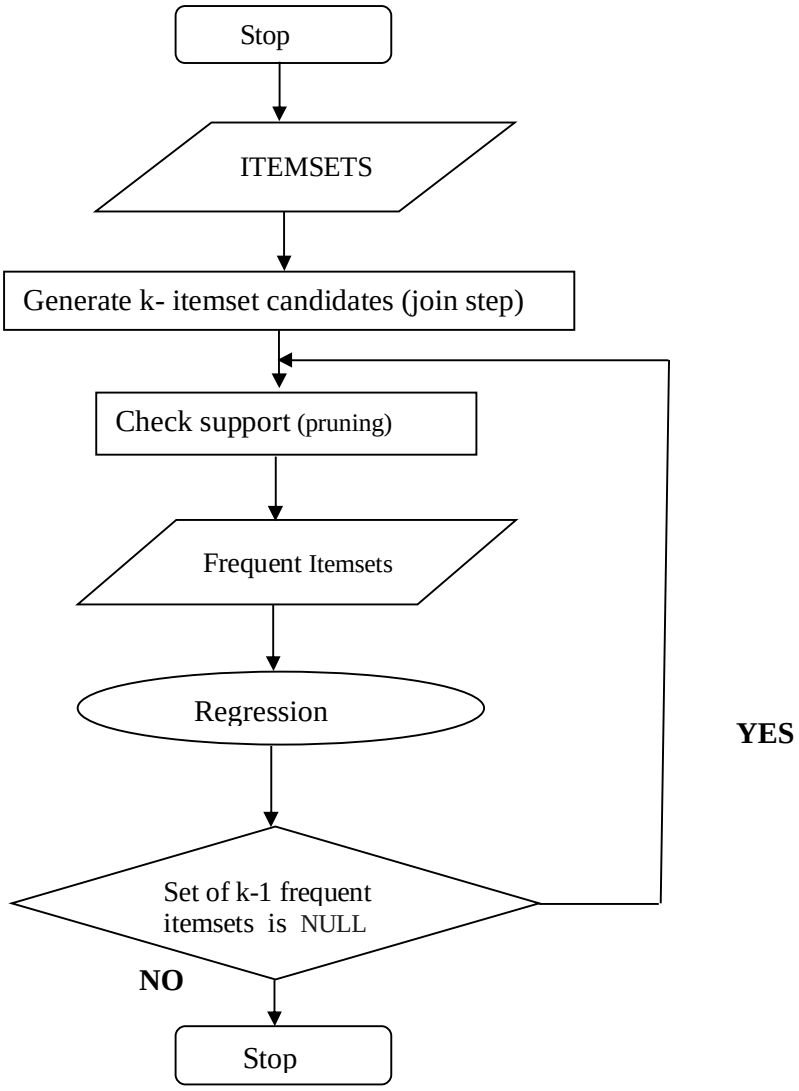

Figure 3.9 Flowchart of Apriori Algorithm with Regression (Source [143])

In this methodology, the following definitions are needed:

Assume $T=\{T_1, T_2, \dots, T_m\}$,(m=1) is situated for transactions, $Ti= \{I_1, I_2, \dots, I_n\}$,(n=1) may be those situated about items, Also k-itemset = $\{i_1, i_2, \dots, i_k\}$,(k=1) is likewise the situated for k items, and k-itemset?.

Suppose (itemset), may be the support count of itemset or the recurrence of event

from claiming an itemset in transactions.

Assume C_k is the candidate itemset about size k, and L_k is the frequent itemset for span k.

By performing these changes(adding regression) in Apriori algorithm, we achieve the new algorithm which is described in above figure.

C_k denotes the set of candidate k-itemsets and F_k denotes the set of frequent k-itemsets

Apriori Algorithm with Regression (Source [90])

Step 1: Scan the database D, generating candidate 1- setC_1;

Step 2: According to the min_sup, frequent item 1- set L_k is generated from the candidate 1- setC_k;

Step-3: According to the min_sup, frequent item (k+1) – set L_{k+1} is generated from the candidate (k+1)-set C_{k+1};

Step-4: Get frequent item sets L_k;

Step-5: Generate frequent itemset L_k from candidate items C_k.;

Step-6: Prune the results to find the frequent item sets using Linear Regression L_k from candidate (k+1) – set C_{k+1} are generated.

Step-7: Goto Step-4 till outliers are removed.

Step-8: Generate strong association rules from frequent item sets. According to the min_sup, frequent item (k+1) - set L_{k+1} is generated from the candidate (k+1) - set C_{k+1};

Step-9: A Rule which satisfy the min. support and min. confidence threshold.

3.10 Summary

In the present chapter a detailed description of the technique of association rule mining is given. Various approaches for discovery of frequent sets and association rules proposed in literature are discussed with pros and cons of each. Also, some techniques for increasing the efficiency of association rule mining technique and some recent

advancement in the field are elaborated. At last an approach based on graph is proposed that can discover frequent itemsets in efficient manner in the next chapter, the popular meta-heuristic ACO is introduced. The core theme of the current research work is to find suitability of the ant colony Meta -heuristic in discovering frequent itemsets and in mining association rules. Thus, a broad discussion of various algorithms and approaches under ant colony meta- heuristic is presented in the next chapter.