MINING FREQUENT PATTENTS

Association Rule Mining is to find out association rules or Frequent patterns or subsequences or correlation relationships among large set of data items that satisfy the predefined minimum support and confidence from a given database.

Frequent patterns are patterns (such as itemsets, subsequences, or substructures) that appear in a data set frequently. For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set is a *frequent itemset*.

A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (*frequent*) *sequential pattern*. A *substructure* can refer to different structural forms, such as subgraphs, subtrees, or sub lattices, which may be combined with itemsets or subsequences.

Association rule mining: Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.

Applications:

Basket data analysis

Cross-marketing

Catalog design

Loss-leader analysis

Clustering

Classification, etc.

Market Basket Analysis:

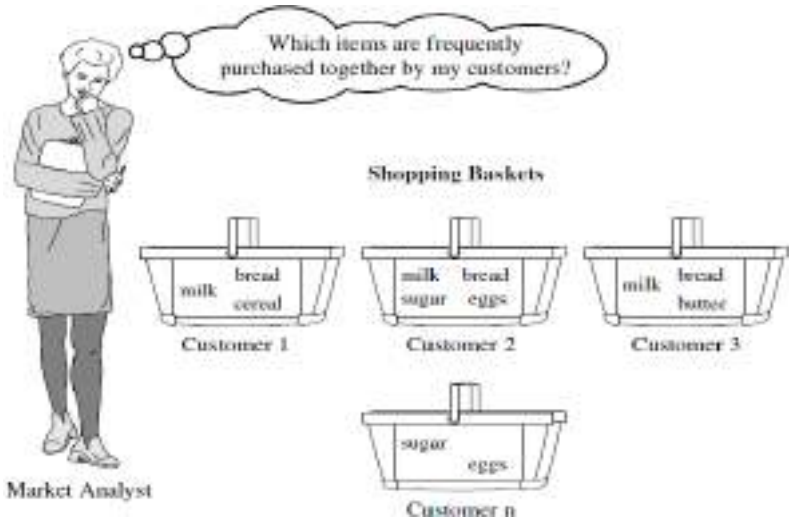
Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets. A typical example of frequent itemset mining is market basket analysis.

This process analyzes customer buying habits by finding associations between the different items that customers place in their “shopping baskets”.

The discovery of such associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers.

Frequent Itemsets, Closed Itemsets, and Association Rules

Let I ={*I*1, *I*2, ... , *Im*} be a set of items. Let *D*, the task-relevant data, be a set of database transactions where each transaction *T* is a set of items such that *T* _ I . Each transaction is



associated with an identifier, called TID.

An association rule is an implication of the form *A* →*B*, where *A* → I , *B* → I , and *A* →*B*=f. The rule *A* →*B* holds in the transaction set *D* with support *s*, where *s* is the percentage of transactions in *D* that contain *A* →*B* (i.e., the *union* of sets *A* and *B*, or say, both *A* and *B*). The rule *A* → *B* has confidence *c* in the transaction set *D*, where *c* is the percentage of transactions in *D* containing *A* that also contain *B*. This is taken to be the conditional probability, *P*(*B*|*A*). That is,

$$support(A \Rightarrow B) = P(A \cup B)$$
$$confidence(A \Rightarrow B) = P(B|A).$$

If the relative support of an itemset *I* satisfies a prespecified minimum support threshold (i.e., the absolute support of *I* satisfies the corresponding minimum support count threshold), then *I* is a frequent itemset. The set of frequent *k*-itemsets is commonly denoted by *Lk*.

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

Support is used to eliminate uninteresting rules. Low support is likely to be uninteresting from a business perspective because it may not be profitable to promote items.

Confidence measures the reliability of the inference made by a rule. For rule A⍰B, the higher confidence , the more likely it is for **B** to be present in transactions that contain **A**. It is used to estimate conditional probability of **B** given **A.**

From Association Analysis to Correlation Analysis

A correlation measure can be used to augment the support-confidence framework for association rules. This leads to *correlation rules* of the form

A=>*B* [*support*, *confidence*, *correlation*]

That is, a correlation rule is measured not only by its support and confidence but also by the

correlation between itemsets*A* and *B*. There are many different correlation measures from which to choose.

Lift is a simple correlation measure that is given as follows. The occurrence of itemset *A* is independent of the occurrence of itemset *B* if $P(A \cup B)$ = *P*(*A*)*P*(*B*); otherwise, itemsets *A* and *B* are dependent and correlated as events. This definition can easily be extended to more than two itemsets.

The lift between the occurrence of *A* and *B* can bemeasured by computing

$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}.$$

- If the lift(A,B) is less than 1, then the occurrence of A is negatively correlated with the occurrence of B.
- If the resulting value is greater than 1, then A and B are positively correlated, meaning that the occurrence of one implies the occurrence of the other.
- If the resulting value is equal to 1, then A and B are independent and there is no correlation between them.

The Apriori property is a fundamental property of frequent itemsets used in the Apriori algorithm. In other words, if an itemset appears frequently enough in the dataset to be considered significant, then all of its subsets must also appear frequently enough to be significant.
In the Apriori algorithm, support refers to the frequency or occurrence of an item set in a dataset.

$$Support(A) = \frac{Number\ of\ Transactions\ in\ which\ A\ occurs}{Number\ of\ all\ Transactions}$$

In the Apriori algorithm, confidence is also a measure of the strength of the association between two items in an itemset. It is defined as the conditional probability that item B appears in a transaction, given that another item A appears in the same transaction.

$$confidence(A \Rightarrow B) = P(B/A) = \frac{sup(A \cup B)}{sup(A)}$$

Let's consider the transaction dataset of a retail store as shown in the below table.

TID	Items
T1	{milk, bread}
T2	{bread, sugar}
T3	{bread, butter}
T4	{milk, bread, sugar}
T5	{milk, bread, butter}
T6	{milk, bread, butter}
T7	{milk, sugar}
T8	{milk, sugar}
T9	{sugar, butter}
T10	{milk, sugar, butter}
T11	{milk, bread, butter}

- Let's calculate support for each item present in the dataset. As shown in the below table, support for all items is greater than 3. It means that all items are considered as frequent 1-itemsets and will be used to generate candidates for 2-itemsets.

Item	Support (Frequency)
milk	8
bread	7
sugar	5
butter	7

- Below table represents all candidates generated from frequent 1-itemsets identified from the previous step and their support value.

Candidate Item Sets	Support (Frequency)
{milk, bread}	5
{milk, sugar}	3
{milk, butter}	5
{bread, sugar}	2
{bread, butter}	3
{sugar, butter}	2

- Now remove candidate item sets that do not meet the minimum support threshold of 3. After this step, frequent 2-itemsets would be - {milk, bread}, {milk, sugar}, {milk, butter}, and {bread, butter}. In the next step, let's generate candidates for 3-itemsets and calculate their respective support values. It is shown in the below table.

Candidate Item Sets	Support (Frequency)
{milk, bread, sugar}	1
{milk, bread, butter}	3
{milk, sugar, butter}	1

- As we can see in the above table, only one candidate itemset exceeds the minimum defined support threshold - {milk, bread, butter}. As there is only one 3-itemset exceeding minimum support, we can't generate candidates for 4-itemsets. So, in the next step, we can write the association rules and their respective metrics, as shown in the below table.

Candidate Item Sets	Support (Frequency)
{milk, bread}	{butter} (Confidence - 60%)1
{bread, butter}	{milk} (Confidence - 100%)
{milk, butter}	{bread} (Confidence - 60%)

- Based on association rules mentioned in the above table, we can recommend products to the customer or optimize product placement in retail stores.

Advantages and Limitations of Apriori Algorithm

Here are some of the advantages of the Apriori algorithm in data mining -

- Apriori algorithm is simple and easy to implement, making it accessible even to those without a deep understanding of data mining or machine learning.
- Apriori algorithm can handle large datasets and run on distributed systems, making it scalable for large-scale applications.
- Apriori algorithm is one of the most widely used algorithms for association rule mining and is supported by many popular data mining tools.

Below are some of the limitations of the Apriori algorithm in data mining -

- Apriori algorithm can be computationally expensive, especially for large datasets with many itemsets. For example, if a dataset contains 104104 from frequent 1- itemsets, it

will generate more than 107107 2-length candidates, which makes this algorithm computationally expensive.

- Apriori algorithm can generate a large number of rules, making it difficult to sift through and identify the most important ones.
- The algorithm requires multiple database scans to generate frequent itemsets, which can be a limitation in systems where data access is slow or expensive.
- Apriori algorithm is sensitive to data sparsity, meaning it may not perform well on datasets with a low frequency of itemsets.

The **FP Growth algorithm** is a popular method for frequent pattern mining in data mining. It works by constructing a **frequent pattern tree (FP-tree)** from the input dataset. The **FP-tree** is a compressed representation of the dataset that captures the frequency and association information of the items in the data.

Transaction ID	Items
T1	{M, N, O, E, K, Y}
T2	{D, O, E, N, Y, K}
T3	{K, A, M, E}
T4	{M, C, U, Y, K}
T5	{C, O, K, O, E, I}

Let's scan the above database and compute the frequency of each item as shown in the below table.

Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
U	1
Y	3

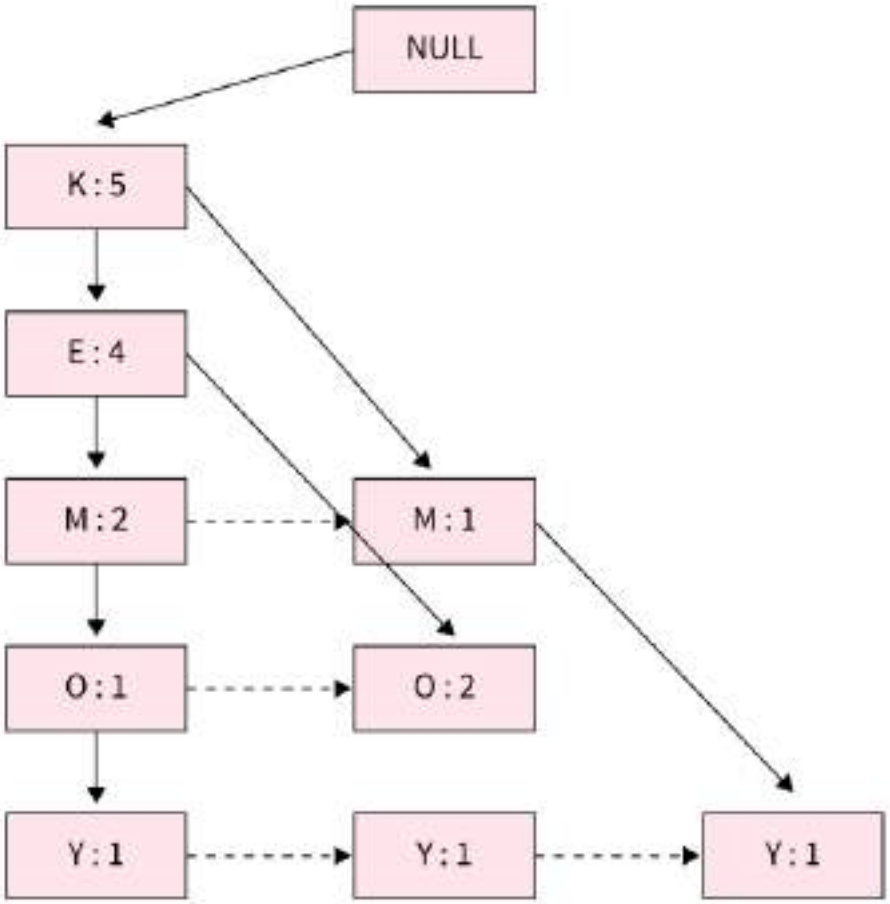
Let's consider minimum support as 3. After removing all the items below minimum support in the above table, we would remain with these items - {K: 5, E: 4, M : 3, O : 3, Y : 3}. Let's re-order the transaction database based on the items above minimum support. In this step, in each transaction, we will remove infrequent items and re-order them in the descending order of their frequency, as shown in the table below.

Transaction ID	Items	Ordered Itemset
T1	{M, N, O, E, K, Y}	{K, E, M, O, Y}
T2	{D, O, E, N, Y, K}	{K, E, O, Y}
T3	{K, A, M, E}	{K, E, M}
T4	{M, C, U, Y, K}	{K, M, Y}
T5	{C, O, K, O, E, I}	{K, E, O}

Now we will use the ordered itemset in each transaction to build the FP tree. Each transaction will be inserted individually to build the FP tree, as shown below -

- **First Transaction {K, E, M, O, Y}:**
  In this transaction, all items are simply linked, and their support count is initialized as 1.

- **Second Transaction {K, E, O, Y}:**
  In this transaction, we will increase the support count of K and E in the tree to 2. As no direct link is available from E to O, we will insert a new path for O and Y and initialize their support count as 1.
- **Third Transaction {K, E, M}:**
  After inserting this transaction, the tree will look as shown below. We will increase the support count for K and E to 3 and for M to 2.
- **Fourth Transaction {K, M, Y}** and **Fifth Transaction {K, E, O}:**



FP Growth Algorithm Vs. Apriori Algorithm

Factor	FP Growth Algorithm	Apriori Algorithm
Working	FP Growth uses FP-tree to mine frequent itemsets.	Apriori algorithm mines frequent items in an iterative manner - 1-itemsets, 2-itemsets, 3-itemsets, etc.
Candidate Generation	Generates frequent itemsets by constructing the FP-Tree and recursively generating conditional pattern bases.	Generates candidate itemsets by joining and pruning.

Data Scanning	Scans the database only twice to construct the FP-Tree and generate conditional pattern bases.	Scans the database multiple times for frequent itemsets.
Memory Usage	Requires less memory than Apriori as it constructs the FP-Tree, which compresses the database	Requires a large amount of memory to store candidate itemsets.
Speed	Faster due to efficient data compression and generation of frequent itemsets.	Slower due to multiple database scans and candidate generation.
Scalability	Performs well on large datasets due to efficient data compression and generation of frequent itemsets.	Performs poorly on large datasets due to a large number of candidate itemsets.

Advantages of FP Growth Algorithm

The FP Growth algorithm in data mining has several advantages over other frequent itemset mining algorithms, as mentioned below:

- **Efficiency:**
  FP Growth algorithm is faster and more memory-efficient than other frequent itemset mining algorithms such as Apriori, especially on large datasets with high dimensionality. This is because it generates frequent itemsets by constructing the FP-Tree, which compresses the database and requires only two scans.
- **Scalability:**
  FP Growth algorithm scales well with increasing database size and itemset dimensionality, making it suitable for mining frequent itemsets in large datasets.
- **Resistant to noise:**
  FP Growth algorithm is more resistant to noise in the data than other frequent itemset mining algorithms, as it generates only frequent itemsets and ignores infrequent itemsets that may be caused by noise.
- **Parallelization:**
  FP Growth algorithm can be easily parallelized, making it suitable for distributed computing environments and allowing it to take advantage of multi-core processors.

Disadvantages of FP Growth Algorithm

While the FP Growth algorithm in data mining has several advantages, it also has some limitations and disadvantages, as mentioned below:

- **Memory consumption:**
  Although the FP Growth algorithm is more memory-efficient than other frequent itemset mining algorithms, storing the FP-Tree and the conditional pattern bases can still require a significant amount of memory, especially for large datasets.
- **Complex implementation:**
  The FP Growth algorithm is more complex than other frequent itemset mining algorithms, making it more difficult to understand and implement.

Mining Various Kinds of Association Rules

1. **Based on the completeness of patterns to be mined:**

- We can mine the complete set of frequent itemsets, the closed frequent itemsets, and themaximal frequent itemsets, given a minimum support threshold.
- We can also mine constrained frequent itemsets, approximate frequent itemsets,near-match frequent itemsets, top-k frequent itemsets and so on.

2. **Based on the levels of abstraction involved in the rule set:**

Some methods for associationrule mining can find rules at differing levels of abstraction.

For example, supposethat a set of association rules mined includes the following rules where X is a variablerepresenting a customer:

buys(X, ―computer‖))=>buys(X, ―HP printer‖) (1)

buys(*X*, ―*laptop computer*‖)) =>*buys*(*X*, ―*HP printer*‖) (2)

In rule (1) and (2), the items bought are referenced at different levels ofabstraction (e.g., ―*computer*‖ is a higher-level abstraction of ―*laptop computer*‖).

3. **Based on the number of data dimensions involved in the rule:**

- If the items or attributes in an association rule reference only one dimension, then it is asingle-dimensional association rule.

  buys(X, ―computer‖))=>buys(X, ―antivirus software‖)

- If a rule references two or more dimensions, such as the dimensions age, income, and buys, then it is amultidimensional association rule. The following rule is an exampleof a multidimensional rule:

  age(X, ―30,31…39‖) ^ income(X, ―42K,…48K‖))=>buys(X, ―high resolution TV‖)

4. **Based on the types of values handled in the rule:**

- If a rule involves associations between the presence or absence of items, it is a Booleanassociation rule.
- If a rule describes associations between quantitative items or attributes, then it is aquantitative association rule.

5. **Based on the kinds of rules to be mined:**

- Frequent pattern analysis can generate various kinds of rules and other interesting

relationships.

- Association rule mining cangenerate a large number of rules, many of which are redundant or do not indicatea correlation relationship among itemsets.
- The discovered associations can be further analyzed to uncover statistical correlations,leading to correlation rules.

6. Based on the kinds of patterns to be mined:

- Many kinds of frequent patterns can be mined from different kinds of data sets.
- Sequential pattern mining searches for frequent subsequences in a sequence data set, where a sequence records an ordering of events.
- For example, with sequential pattern mining, we can study the order in which items are frequently purchased. For instance, customers may tend to first buy a PC, followed by a digitalcamera,and then a memory card.
- Structuredpatternminingsearches for frequent substructuresin a structured data set.
- Single items are the simplest form of structure.
- Each element of an itemsetmay contain a subsequence, a subtree, and so on.
- Therefore, structuredpattern mining can be considered as the most general formof frequent pattern mining.

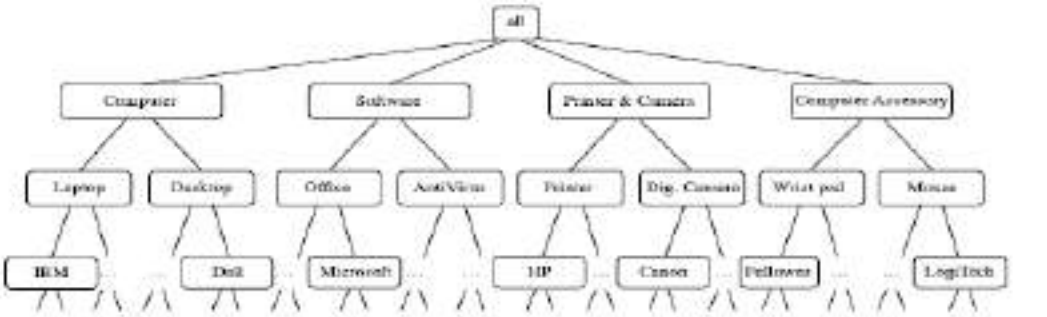
Mining Multilevel Association Rules:

- For many applications, it is difficult to find strong associations among data items at lowor primitive levels of abstraction due to the sparsity of data at those levels.
- Strong associations discovered at high levels of abstraction may represent Common sense knowledge.
- Therefore, data mining systems should provide capabilities for mining association rules at multiple levels of abstraction, with sufficient flexibility for easy traversal among different abstraction spaces.
- Association rules generated from mining data at multiple levels of abstraction are called multiple-level or multilevel association rules.
- Multilevel association rules can be mined efficiently using concept hierarchies under a support-confidence framework.
- In general, a top-down strategy is employed, where counts are accumulated for the calculation of frequent itemsets at each concept level, starting at the concept level 1 and

working downward in the hierarchy toward the more specific concept levels, until no more frequent itemsets can be found.

A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher level, more general concepts. Data can be generalized by replacing low-level concepts within the data by their higher-level concepts, or ancestors, from a concept hierarchy.

TID	Items Purchased
T100	IBM-ThinkPad-T40/2373, HP-Photosmart-7660
T200	Microsoft-Office-Professional-2003, Microsoft-Plus!-Digital-Media
T300	Logitech-MX700-Cordless-Mouse, Fellowes-Wrist-Rest
T400	Dell-Dimension-XPS, Canon-PowerShot-S400
T500	IBM-ThinkPad-R40/P4M, Symantec-Norton-Antivirus-2003
...	...



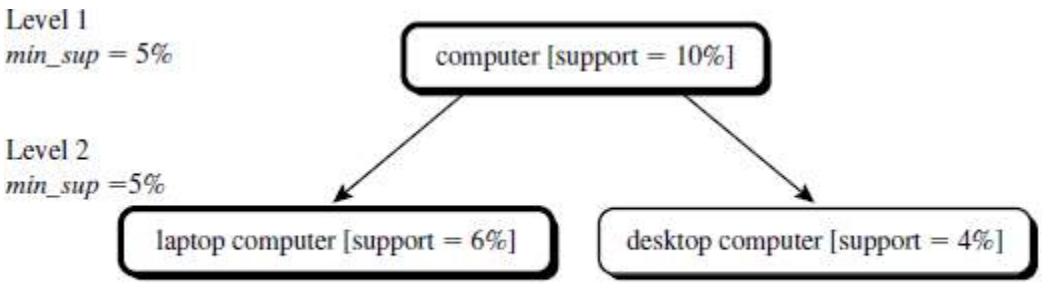
A concept hierarchy for *AllElectronics* computer items.

The concept hierarchy has five levels, respectively referred to as levels 0to 4, starting with level 0 at the root node for all.

- Here, Level 1 includes computer, software, printer&camera, and computer accessory. • Level 2 includes laptop computer, desktop computer, office software, antivirus software • Level 3 includes IBM desktop computer, . . . , Microsoft office software, and so on.
- Level 4 is the most specific abstraction level of this hierarchy.

Approaches For Mining Multilevel Association Rules:

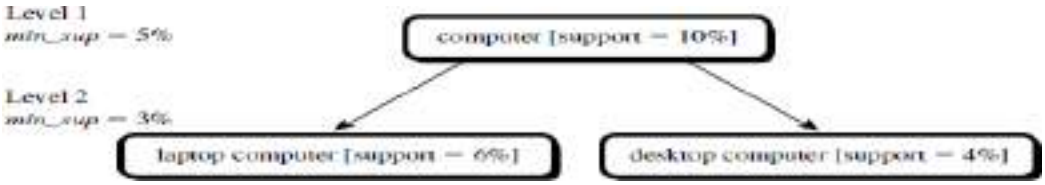
1. **Uniform Minimum Support:**
   - The same minimum support threshold is used when mining at each level of abstraction. • When a uniform minimum support threshold is used, the search procedure is simplified.
   - The method is also simple in that users are required to specify only one minimum support threshold.
   - The uniform support approach, however, has some difficulties. It is unlikely that items at lower levels of abstraction will occur as frequently as those at higher levels of abstraction.
   - If the minimum support threshold is set too high, it could miss some meaningful associations occurring at low abstraction levels. If the threshold is set too low, it may generate many uninteresting associations occurring at high abstraction levels.



Multilevel mining with uniform support.

2. **Reduced Minimum Support:**

- Each level of abstraction has its own minimum support threshold.
- The deeper the level of abstraction, the smaller the corresponding threshold is.
- For example, the minimum support thresholds for levels 1 and 2 are 5% and 3%,respectively. In this way, ―computer,‖ ―laptop computer,‖ and ―desktop computer‖ areall considered frequent.



3. Group-Based Minimum Support:

- Because users or experts often have insight as to which groups are more important than others, it is sometimes more desirable to set up user-specific, item, or group based minimal support thresholds when mining multilevel rules.
- For example, a user could set up the minimum support thresholds based on product price, or on items of interest, such as by setting particularly low support thresholds for laptop computers and flash drives in order to pay particular attention to the association patterns containing items in these categories.

Mining Multidimensional Association Rules from Relational Databases and Data Warehouses:

- Single dimensional or intra dimensional association rule contains a single distinct predicate (e.g., buys)with multiple occurrences i.e., the predicate occurs more than once within the rule.

  buys(*X*, ―*digital camera*‖)=>*buys*(*X*, ―*HP printer*‖)

- Association rules that involve two or more dimensions or predicates can be referred to as multidimensional association rules.

  age(X, "20…29")^occupation(X, "student")=>buys(X, "laptop")

- Above Rule contains three predicates (age, occupation,and buys), each of

which occurs only once in the rule. Hence, we say that it has norepeated predicates.

- Multidimensional association rules with no repeated predicates arecalled interdimensional association rules.
- We can also mine multidimensional associationrules with repeated predicates, which contain multiple occurrences of some predicates.These rules are called hybrid- dimensional association rules. An example of sucha rule is the following, where the predicate buys is repeated:

age(X, ―20…29‖)^buys(X, ―laptop‖)=>buys(X, ―HP printer‖)

Mining Quantitative Association Rules:

- Quantitative association rules are multidimensional association rules in which the numeric attributes are *dynamically* discretized during the mining process so as to satisfy some mining criteria, such as maximizing the confidence or compactness of the rules mined.
- In this section, we focus specifically on how to mine quantitative association rules having two quantitative attributes on the left-hand side of the rule and one categorical attribute on the right- hand side of the rule. That is

*Aquan*1 ^*Aquan*2 =>*Acat*

Where *Aquan*1 and *Aquan*2 are tests on quantitative attribute interval

Acat tests a categorical attribute from the task-relevant data.

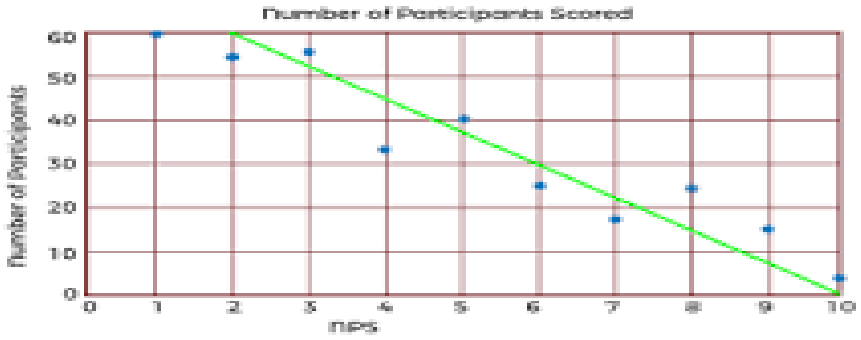
- Such rules have been referred to as two-dimensional quantitative association rules, because they contain two quantitative dimensions.
- For instance, suppose you are curious about the association relationship between pairs of quantitative attributes, like customer age and income, and the type of television (such as *high- definition TV*, i.e., *HDTV*) that customers like to buy.

An example of such a 2-D quantitative association rule is

age(*X*, ―30…39‖)^*income*(*X*, ―42*K*…48*K*‖)=>*buys*(*X*, ―*HDTV*‖)

Correlation Analysis:

Correlation analysis is a statistical method used to measure the strength of the linear relationship between two variables and compute their association. Correlation analysis calculates the level of change in one variable due to the change in the other.



Correlation analysis is a statistical method used to measure the strength of the linear relationship between two variables and compute their association. Correlation analysis calculates the level of change in one variable due to the change in the other. A high correlation points to a strong relationship between the two variables, while a low correlation means that the variables are weakly related.

Correlation is a bivariate analysis that measures the strength of association between two variables and the direction of the relationship. In terms of the strength of the relationship, the correlation coefficient's value varies between +1 and -1. A value of ± 1 indicates a perfect degree of association between the two variables.

As the correlation coefficient value goes towards 0, the relationship between the two variables will be weaker. The coefficient sign indicates the direction of the relationship; a + sign indicates a positive relationship, and a
- sign indicates a negative relationship.

Constraint based Association Mining

Data mining process may uncover thousands of rules from a given set of data, most of which end up being unrelated or uninteresting to the users. Often, users have a good sense of which direction‖ of mining may lead to interesting patterns and the ―form‖ of the patterns or rules they would like to find. Thus, a good heuristic is to have the users specify such intuition or expectations as *constraints* to confine the search space. This strategy is known as constraint-based mining.

The constraints can include the following:

- **Knowledge type constraints:** These specify the type of knowledge to be mined, such as association or correlation.
- **Data constraints:** These specify the set of task-relevant data.
- **Dimension/level constraints:** These specify the desired dimensions (or attributes) of the data, or levels of the concept hierarchies, to be used in mining.
- **Interestingness constraints:** These specify thresholds on statistical measures of rule interestingness, such as support, confidence, and correlation.
- **Rule constraints:** These specify the form of rules to be mined. Such constraints may be expressed as metarules (rule templates), as the maximum or minimum number of predicates that can occur in the rule antecedent or consequent, or as relationships among attributes, attribute values, and/or aggregates.

Graph Pattern Mining:

Graph mining is a process in which the mining techniques are used in finding a pattern or relationship in the given real-world collection of graphs. By mining the graph, frequent substructures and relationships can be identified which helps in clustering the graph sets, finding a relationship between graph sets, or discriminating or characterizing graphs. Predicting these patterning trends can help in building models for the

Enhancement of any application that is used in real-time. To implement the process of graph mining, one must learn to mine frequent subgraphs.

Sequential Pattern Mining (SPM):

Sequential pattern mining is the mining of frequently appearing series events or subsequences as patterns. An instance of a sequential pattern is users who purchase a Canon digital camera are to purchase an HP color printer within a month.

For retail information, sequential patterns are beneficial for shelf placement and promotions. This industry, and telecommunications and different businesses, can also use sequential patterns for targeted marketing, user retention, and several tasks.

There are several areas in which sequential patterns can be used such as Web access pattern analysis, weather prediction, production processes, and web intrusion detection.