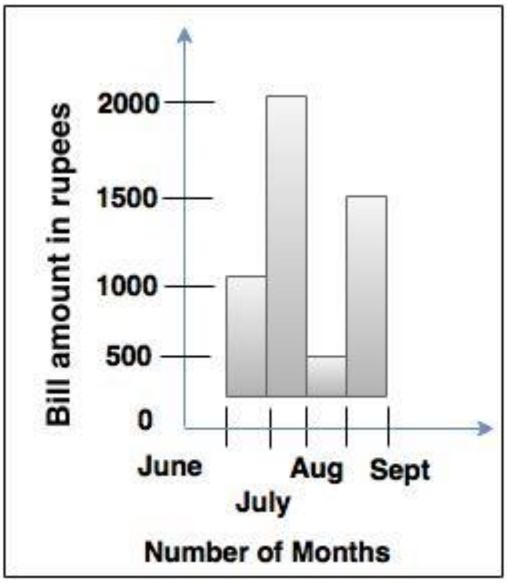
DATA MINING (COCSC16)
UNIT 3

Knowledge Representation

- Knowledge representation is the presentation of knowledge to the user for visualization in terms of trees, tables, rules graphs, charts, matrices, etc. **ForExample:** Histograms

Histograms

- Histogram provides the representation of a distribution of values of a single attribute.
- It consists of a set of rectangles, that reflects the counts or frequencies of the classes present in the given data.
  Example: Histogram of an electricity bill generated for 4 months, as shown in diagram given below.



Histogram for Electricity Bill

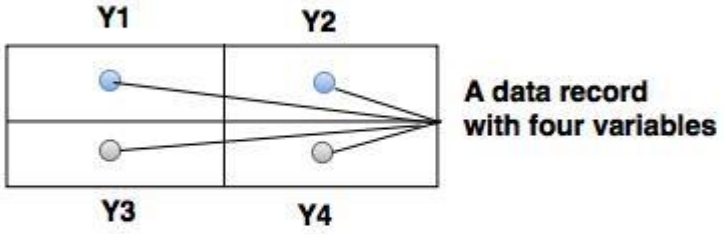
Data Visualization

- It deals with the representation of data in a graphical or pictorial format.
- Patterns in the data are marked easily by using the data visualization technique.

Some of the vital data visualization techniques are:

1. Pixel- oriented visualization technique

- In pixel-based visualization techniques, there are separate sub-windows for the value of each attribute and it is represented by one colored pixel.
- It maximizes the amount of information represented at one time without any overlap.
- Tuple with 'm' variable has different 'm' colored pixel to represent each variable and each variable has a sub window.
- The color mapping of the pixel is decided on the basis of data characteristics and visualization tasks.



Pixel visualization

2. Geometric projection visualization technique

Techniques used to find geometric transformation are:

i. Scatter-plot matrices
It consists of scatter plots of all possible pairs of variables in a dataset.

ii. Hyper slice
It is an extension to scatter-plot matrices. They represent multi-dimensional function as a matrix of orthogonal two-dimensional slices.

iii. Parallel co-ordinates

- The parallel vertical lines which are separated defines the axes.
- A point in the Cartesian coordinates corresponds to a polyline in parallel coordinates.

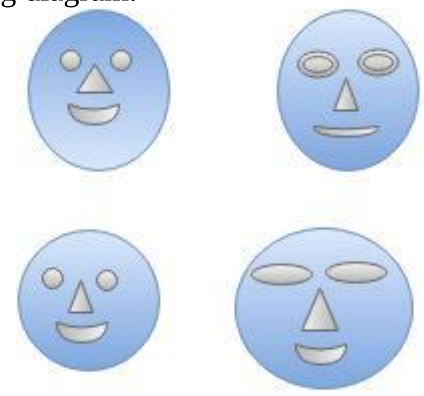
3. Icon-based visualization techniques

- Icon-based visualization techniques are also known as **iconic display techniques.**
- Each multidimensional data item is mapped to an icon.
- This technique allows visualization of large amount of data.
- **The most commonly used technique is Chernoff faces.**

Chernoff faces

- This concept was introduced by **Herman Chernoff** in 1973.

- The faces in Chernoff faces are related to facial expressions or features of human being. So, it becomes easy to identify the difference between the faces.
- It includes the mapping of different data dimensions with different facial features. **For example:** The face width, the length of the mouth and the length of nose, etc. as shown in the following diagram.



Chernoff faces

4. Hierarchical visualization techniques

- Hierarchical visualization techniques are used for partitioning of all dimensions in to subset.
- These subsets are visualized in hierarchical manner.

Some of the visualization techniques are:

i. Dimensional stacking

- In dimension stacking, n-dimensional attribute space is partitioned in 2-dimensional subspaces.
- Attribute values are partitioned into various classes.
- Each element is two-dimensional space in the form of xy plot.
- Helps to mark the important attributes and are used on the outer level.

ii. Mosaic plot

- Mosaic plot gives the graphical representation of successive decompositions.
- Rectangles are used to represent the count of categorical data and at every stage, rectangles are split parallel.

iii. Worlds within worlds

- Worlds within worlds are useful to generate an interactive hierarchy of display.
- Innermost word must have a function and two most important parameters.

- Remaining parameters are fixed with the constant value.
- Through this, N-vision of data are possible like data glove and stereo displays, including rotation, scaling (inner) and translation (inner/outer).
- Using queries, static interaction is possible.

iv. Tree maps

- Tree maps visualization techniques are well suited for displaying large amount of hierarchical structured data.
- The visualization space is divided into the multiple rectangles that are ordered, according to a quantitative variable.
- The levels in the hierarchy are seen as rectangles containing the other rectangle.
- Each set of rectangles on the same level in the hierarchy represents a category, a column or an expression in a data set.

v. Visualization complex data and relations

- This technique is used to visualize non-numeric data. **For example:** text, pictures, blog entries and product reviews.
- A tag cloud is a visualization method which helps to understand the information of user generated tags.
- It is also possible to arrange the tags alphabetically or according to the user preferences with different font sizes and colors.

Task Relevant Data

These primitives allow us to communicate in an interactive manner with the data mining system. Here is the list of Data Mining Task Primitives −

- Set of task relevant data to be mined.
- Kind of knowledge to be mined.
- Background knowledge to be used in discovery process.
- Interestingness measures and thresholds for pattern evaluation.
- Representation for visualizing the discovered patterns.

Set of task relevant data to be mined

This is the portion of database in which the user is interested. This portion includes the following −

- Database Attributes
- Data Warehouse dimensions of interest

Kind of knowledge to be mined

It refers to the kind of functions to be performed. These functions are −

- Characterization
- Discrimination
- Association and Correlation Analysis
- Classification
- Prediction
- Clustering
- Outlier Analysis
- Evolution Analysis

Background knowledge

The background knowledge allows data to be mined at multiple levels of abstraction. For example, the Concept hierarchies are one of the background knowledge that allows data to be mined at multiple levels of abstraction.

Interestingness measures and thresholds for pattern evaluation

This is used to evaluate the patterns that are discovered by the process of knowledge discovery. There are different interesting measures for different kind of knowledge.

Representation for visualizing the discovered patterns

This refers to the form in which discovered patterns are to be displayed. These representations may include the following. −

- Rules
- Tables
- Charts
- Graphs
- Decision Trees
- Cubes

Association Rule Mining

- Association rule mining is a technique used to identify patterns in large data sets. It involves finding relationships between variables in the data and using those relationships to make predictions or decisions.

- The goal of association rule mining is to uncover rules that describe the relationships between different items in the data set.

- For example, consider a dataset of transactions at a grocery store. Association rule mining could be used to identify relationships between items that are frequently purchased together. For example, the rule "If a customer buys bread, they are also likely to buy milk" is an association rule that could be mined from this data set. We can use such rules to inform decisions about store layout, product placement, and marketing efforts.

- Association rule mining typically involves using algorithms to analyze the data and identify the relationships. These algorithms can be based on statistical methods or machine learning techniques.
- The resulting rules are often expressed in the form of "if-then" statements, where the "if" part represents the antecedent (the condition being tested) and the "then" part represents the consequent (the outcome that occurs if the condition is met).
- Association rule mining is an important technique in data analysis because it allows users to discover patterns or relationships within data that may not be immediately apparent.
- By identifying associations between variables, association rule mining can help users understand the relationships between different variables and how those variables may be related to one another.
- This can be useful for various purposes, such as identifying market trends, detecting fraudulent activity, or understanding customer behavior.
- Association rule mining can also be used as a stepping stone for other types of data analysis, such as predicting outcomes or identifying key drivers of certain phenomena. Overall, association rule mining is a valuable tool for extracting insights and understanding the underlying structure of data.

Use Cases of Association Rule Mining:

Association rule mining is commonly used in a variety of applications, some common ones are:

1. Market Basket Analysis

One of the most well-known applications of association rule mining is in market basket analysis. This involves analyzing the items customers purchase together to understand their purchasing habits and preferences.

For example, a retailer might use association rule mining to discover that customers who purchase diapers are also likely to purchase baby formula. We can use this information to optimize product placements and promotions to increase sales.

2. Customer Segmentation

Association rule mining can also be used to segment customers based on their purchasing habits.

For example, a company might use association rule mining to discover that customers who purchase certain types of products are more likely to be younger. Similarly, they could learn that customers who purchase certain combinations of products are more likely to be located in specific geographic regions.

3. Fraud Detection

You can also use association rule mining to detect fraudulent activity. For example, a credit card company might use association rule mining to identify patterns of fraudulent transactions, such as multiple purchases from the same merchant within a short period of time.

4. Social network analysis

Various companies use association rule mining to identify patterns in social media data that can inform the analysis of social networks.

For example, an analysis of Twitter data might reveal that users who tweet about a particular topic are also likely to tweet about other related topics, which could inform the identification of groups or communities within the network.

5. Recommendation systems

Association rule mining can be used to suggest items that a customer might be interested in based on their past purchases or browsing history. For example, a music streaming service might use association rule mining to recommend new artists or albums to a user based on their listening history.

Market Basket Analysis in Data Mining

- A data mining technique that is used to uncover purchase patterns in any retail setting is known as **Market Basket Analysis**. In simple terms Basically, Market basket analysis in data mining is to analyze the combination of products which been bought together.
- This is a technique that gives the careful study of purchases done by a customer in a supermarket. This concept identifies the pattern of frequent purchase items by customers.
- This analysis can help to promote deals, offers, sale by the companies, and data mining techniques helps to achieve this analysis task.
- Example:
  - Data mining concepts are in use for Sales and marketing to provide better customer service, to improve cross-selling opportunities, to increase direct mail response rates.
  - Customer Retention in the form of pattern identification and prediction of likely defections is possible by Data mining.
  - Risk Assessment and Fraud area also use the data-mining concept for identifying inappropriate or unusual behavior etc.

Market basket analysis mainly works with the ASSOCIATION RULE {IF} -> {THEN}.

- **IF** means **Antecedent:** An antecedent is an item found within the data
- **THEN** means **Consequent:** A consequent is an item found in combination with the antecedent.

Like we said above Antecedent is the item sets that are available in data. By formulating from the rules means **{if}** component and from the example is the domain.

Same as Consequent is the item that is found with the combination of Antecedents. By formulating from the rules means **{THEN}** component and from the example is extra plugins/extensions.
With the help of these, we are able to predict customer behavioral patterns. From this, we are able to make certain combinations with offers that customers will probably buy those products. That will automatically increase the sales and revenue of the company.

An association rule has two parts: an antecedent (if) and a consequent (then). An antecedent is an item found within the data. A consequent is an item found in combination with the antecedent.

Types of Market Basket Analysis

There are three types of Market Basket Analysis. They are as follow:

1. **Descriptive market basket analysis**: This sort of analysis looks for patterns and connections in the data that exist between the components of a market basket. This kind of study is mostly used to understand consumer behavior, including what products are purchased in combination and what the most typical item combinations. Retailers can place products in their stores more profitably by understanding which products are frequently bought together with the aid of descriptive market basket analysis.

2. **Predictive Market Basket Analysis**: Market basket analysis that predicts future purchases based on past purchasing patterns is known as predictive market basket analysis. Large volumes of data are analyzed using machine learning algorithms in this sort of analysis in order to create predictions about which products are most likely to be bought together in the future. Retailers may make data-driven decisions about which products to carry, how to price them, and how to optimize shop layouts with the use of predictive market basket research.

3. **Differential Market Basket Analysis**: Differential market basket analysis analyses two sets of market basket data to identify variations between them. Comparing the behavior of various client segments or the behavior of customers over time is a common usage for this kind of study. Retailers can respond to shifting consumer behavior by modifying their marketing and sales tactics with the help of differential market basket analysis.

Benefits of Market Basket Analysis

1. **Enhanced Customer Understanding**: Market basket research offers insights into customer behavior, including what products they buy together and which products they buy the most frequently. Retailers can use this information to better understand their customers and make informed decisions.
2. **Improved Inventory Management**: By examining market basket data, retailers can determine which products are sluggish sellers and which ones are commonly bought together. Retailers can use this information to make well-informed choices about what products to stock and how to manage their inventory most effectively.

3. **Better Pricing Strategies**: A better understanding of the connection between product prices and consumer behavior might help merchants develop better pricing strategies. Using this knowledge, pricing plans that boost sales and profitability can be created.
4. **Sales Growth**: Market basket analysis can assist businesses in determining which products are most frequently bought together and where they should be positioned in the store to grow sales. Retailers may boost revenue and enhance customer shopping experiences by improving store layouts and product positioning.

Applications of Market Basket Analysis

1. **Retail**: Market basket research is frequently used in the retail sector to examine consumer buying patterns and inform decisions about product placement, inventory management, and pricing tactics. Retailers can utilize market basket research to identify which items are sluggish sellers and which ones are commonly bought together, and then modify their inventory management strategy accordingly.
2. **E-commerce**: Market basket analysis can help online merchants better understand the customer buying habits and make data-driven decisions about product recommendations and targeted advertising campaigns. The behaviour of visitors to a website can be examined using market basket analysis to pinpoint problem areas.
3. **Finance**: Market basket analysis can be used to evaluate investor behaviour and forecast the types of investment items that investors will likely buy in the future. The performance of investment portfolios can be enhanced by using this information to create tailored investment strategies.
4. **Telecommunications**: To evaluate consumer behaviour and make data-driven decisions about which goods and services to provide, the telecommunications business might employ market basket analysis. The usage of this data can enhance client happiness and the shopping experience.
5. **Manufacturing**: To evaluate consumer behaviour and make data-driven decisions about which products to produce and which materials to employ in the production process, the manufacturing sector might use market basket analysis. Utilizing this knowledge will increase effectiveness and cut costs.

Apriori Algorithm

- **Apriori algorithm** is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule.
- *Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.*
- To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called *Apriori property* which helps by reducing the search space.

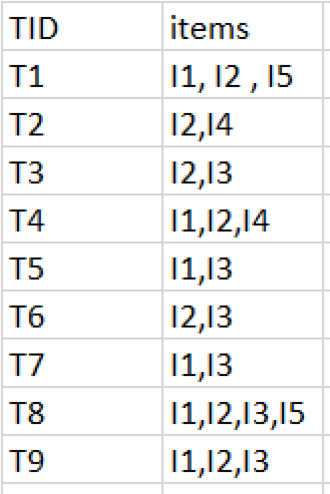
Apriori Property –

- All non-empty subset of frequent itemset must be frequent. The key concept of Apriori algorithm is its anti-monotonicity of support measure. Apriori assumes that

All subsets of a frequent itemset must be frequent (Apriori property).

If an itemset is infrequent, all its supersets will be infrequent.

Consider the following dataset and we will find frequent itemsets and generate association rules for them.

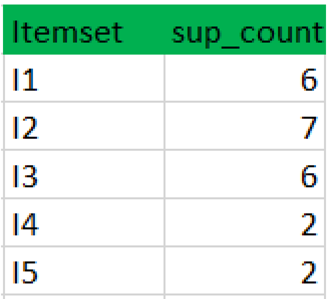


TID	items
T1	I1, I2 , I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

minimum support count is 2
minimum confidence is 60%

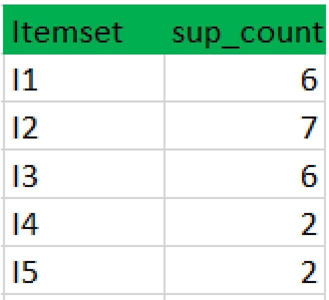
Step-1: K=1

(I) Create a table containing support count of each item present in dataset – Called **C1(candidate set)**



Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

(II) compare candidate set item's support count with minimum support count(here min_support=2 if support_count of candidate set items is less than min_support then remove those items). This gives us itemset L1.

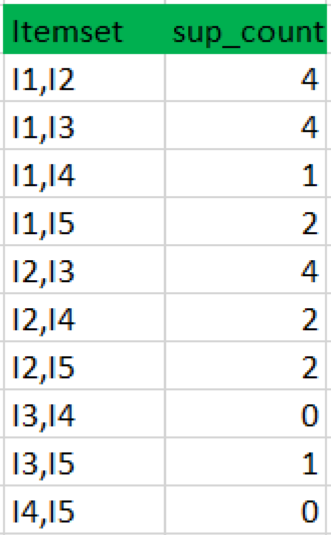


Itemset	sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

Step-2: K=2

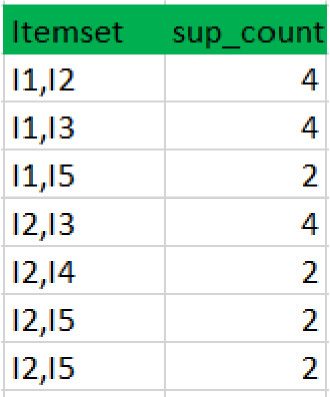
- Generate candidate set C2 using L1 (this is called join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common.

- Check all subsets of an itemset are frequent or not and if not frequent remove that itemset. (Example subset of{I1, I2} are {I1}, {I2} they are frequent. Check for each itemset)
- Now find support count of these itemsets by searching in dataset.



Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

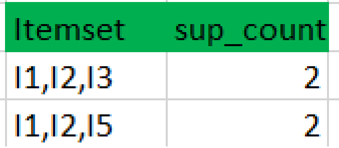
(II) compare candidate (C2) support count with minimum support count (here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L2.



Itemset	sup_count
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I2,I5	2

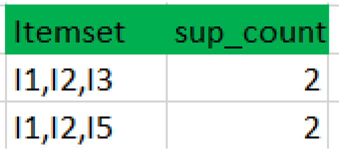
Step-3:

- Generate candidate set C3 using L2 (join step). Condition of joining L_{k-1} and L_{k-1} is that it should have (K-2) elements in common. So here, for L2, first element should match. So itemset generated by joining L2 is {I1, I2, I3}{I1, I2, I5}{I1, I3, i5}{I2, I3, I4}{I2, I4, I5}{I2, I3, I5}
- Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset. (Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
- find support count of these remaining itemset by searching in dataset.



Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

(II) Compare candidate (C3) support count with minimum support count (here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L3.



Itemset	sup_count
I1,I2,I3	2
I1,I2,I5	2

Step-4:

- Generate candidate set C4 using L3 (join step). Condition of joining L_{k-1} and L_{k-1} (K=4) is that, they should have (K-2) elements in common. So here, for L3, first 2 elements (items) should match.
- Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So, no itemset in C4
- We stop here because no frequent itemsets are found further.

Thus, we have discovered all the frequent item-sets. Now generation of strong association rule comes into picture. For that we need to calculate confidence of each rule.

Confidence –

A confidence of 60% means that 60% of the customers, who purchased milk and bread also bought butter.

Confidence(A->B)=Support_count(A∪B)/Support_count(A)

So here, by taking an example of any frequent itemset, we will show the rule generation.

Itemset {I1, I2, I3} //from L3
SO rules can be
[I1^I2]=>[I3] //confidence = sup(I1^I2^I3)/sup(I1^I2) = 2/4*100=50%
[I1^I3]=>[I2] //confidence = sup(I1^I2^I3)/sup(I1^I3) = 2/4*100=50%
[I2^I3]=>[I1] //confidence = sup(I1^I2^I3)/sup(I2^I3) = 2/4*100=50%
[I1]=>[I2^I3] //confidence = sup(I1^I2^I3)/sup(I1) = 2/6*100=33%
[I2]=>[I1^I3] //confidence = sup(I1^I2^I3)/sup(I2) = 2/7*100=28%
[I3]=>[I1^I2] //confidence = sup(I1^I2^I3)/sup(I3) = 2/6*100=33%

So, if minimum confidence is 50%, then first 3 rules can be considered as strong association rules.

Limitations of Apriori Algorithm

- ➢ Apriori Algorithm can be slow. The main limitation is time required to hold a vast number of candidates sets with much frequent itemsets, low minimum support or large itemsets i.e., it is not an efficient approach for large number of datasets.

- For example, if there are 10^4 from frequent 1- itemsets, it needs to generate more than 10^7 candidates into 2-length which in turn they will be tested and accumulate.
- Furthermore, to detect frequent pattern in size 100 i.e., v1, v2… v100, it has to generate 2^100 candidate itemsets that yield on costly and wasting of time of candidate generation.
- So, it will check for many sets from candidate itemsets, also it will scan database many times repeatedly for finding candidate itemsets.
- Apriori will be very low and inefficiency when memory capacity is limited with large number of transactions.

FP Growth Algorithm in Data Mining

- The FP-Growth Algorithm proposed by ***Han in***. This is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree).
- In his study, Han proved that his method outperforms other popular methods for mining frequent patterns, e.g., the Apriori Algorithm and the TreeProjection.
- In some later works, it was proved that FP-Growth performs better than other methods, including ***Eclat*** and ***Relim***.
- The popularity and efficiency of the FP-Growth Algorithm contribute to many studies that propose variations to improve its performance.

What is FP Growth Algorithm?

- The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance.
- For so much, it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information.

This algorithm works as follows:

- First, it compresses the input database creating an FP-tree instance to represent frequent items.
- After this first step, it divides the compressed database into a set of conditional databases, each associated with one frequent pattern.
- Finally, each such database is mined separately.

Using this strategy, the FP-Growth reduces the search costs by recursively looking for short patterns and then concatenating them into the long frequent patterns.

In large databases, holding the FP tree in the main memory is impossible. A strategy to cope with this problem is to partition the database into a set of smaller databases (called projected databases) and then construct an FP-tree from each of these smaller databases.

FP-Tree

- The frequent-pattern tree (FP-tree) is a compact data structure that stores quantitative information about frequent patterns in a database. Each transaction is read and then mapped onto a path in the FP-tree.
- This is done until all transactions have been read. Different transactions with common subsets allow the tree to remain compact because their paths overlap.
- A frequent Pattern Tree is made with the initial item sets of the database. The purpose of the FP tree is to mine the most frequent pattern. Each node of the FP tree represents an item of the item set.
- The root node represents null, while the lower nodes represent the item sets. The associations of the nodes with the lower nodes, that is, the item sets with the other item sets, are maintained while forming the tree.

Han defines the FP-tree as the tree structure given below:

1. One root is labelled as "null" with a set of item-prefix subtrees as children and a frequent-item-header table.
2. Each node in the item-prefix subtree consists of three fields:
    - Item-name: registers which item is represented by the node;
    - Count: the number of transactions represented by the portion of the path reaching the node;
    - Node-link: links to the next node in the FP-tree carrying the same item name or null if there is none.
3. Each entry in the frequent-item-header table consists of two fields:
    - Item-name: as the same to the node;
    - Head of node-link: a pointer to the first node in the FP-tree carrying the item name.

Additionally, the frequent-item-header table can have the count support for an item.

Algorithm by Han

The original algorithm to construct the FP-Tree defined by Han is given below:

Algorithm 1: FP-tree construction

***Input*:** A transaction database DB and a minimum support threshold?

***Output*:** FP-tree, the frequent-pattern tree of DB.

***Method*:** The FP-tree is constructed as follows.

1. The first step is to scan the database to find the occurrences of the itemsets in the database. This step is the same as the first step of Apriori. The count of 1-itemsets in the database is called support count or frequency of 1-itemset.
2. The second step is to construct the FP tree. For this, create the root of the tree. The root is represented by null.
3. The next step is to scan the database again and examine the transactions. Examine the first transaction and find out the itemset in it. The itemset with the max count is taken at the top, and then the next itemset with the lower count. It means that the branch of the tree is constructed with transaction itemsets in descending order of count.
4. The next transaction in the database is examined. The itemsets are ordered in descending order of count. If any itemset of this transaction is already present in another branch, then this transaction branch would share a common prefix to the root. This means that the common itemset is linked to the new node of another itemset in this transaction.
5. Also, the count of the itemset is incremented as it occurs in the transactions. The common node and new node count are increased by 1 as they are created and linked according to transactions.
6. The next step is to mine the created FP Tree. For this, the lowest node is examined first, along with the links of the lowest nodes. The lowest node represents the frequency pattern length 1. From this, traverse the path in the FP Tree. This path or paths is called a conditional pattern base.
   A conditional pattern base is a sub-database consisting of prefix paths in the FP tree occurring with the lowest node (suffix).
7. Construct a Conditional FP Tree, formed by a count of itemsets in the path. The itemsets meeting the threshold support are considered in the Conditional FP Tree.
8. Frequent Patterns are generated from the Conditional FP Tree.

Using this algorithm, the FP-tree is constructed in two database scans. The first scan collects and sorts the set of frequent items, and the second constructs the FP-Tree.

Example

Support threshold=50%, Confidence= 60%

Table 1:

Transaction	**List of items**

T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Solution: Support threshold=50% => 0.5*6= 3 => min_sup=3

Table 2: Count of each item

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

Table 3: Sort the itemset in descending order.

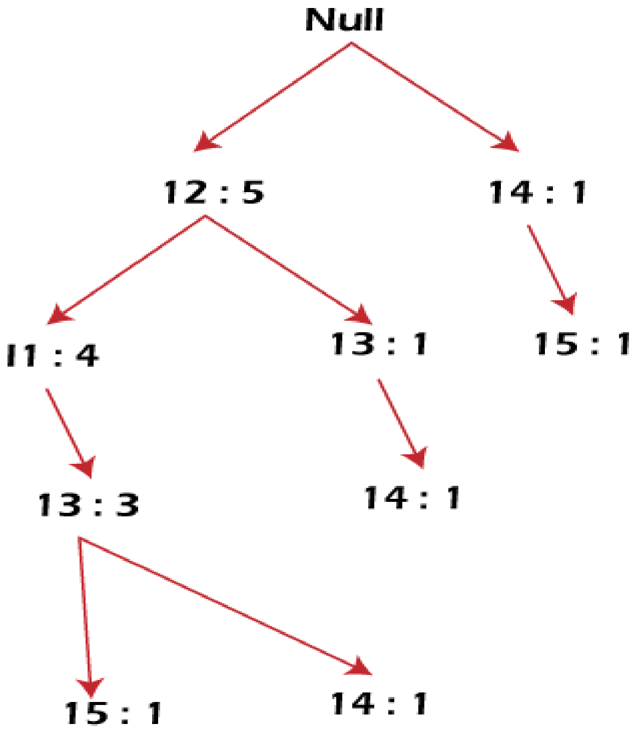
Item	Count
I2	5
I1	4
I3	4
I4	4

Build FP Tree

Let's build the FP tree in the following steps, such as:

1. Considering the root node null.

2. The first scan of Transaction T1: I1, I2, I3 contains three items {I1:1}, {I2:1}, {I3:1}, where I2 is linked as a child, I1 is linked to I2 and I3 is linked to I1.
3. T2: I2, I3, and I4 contain I2, I3, and I4, where I2 is linked to root, I3 is linked to I2 and I4 is linked to I3. But this branch would share the I2 node as common as it is already used in T1.
4. Increment the count of I2 by 1, and I3 is linked as a child to I2, and I4 is linked as a child to I3. The count is {I2:2}, {I3:1}, {I4:1}.
5. T3: I4, I5. Similarly, a new branch with I5 is linked to I4 as a child is created.
6. T4: I1, I2, I4. The sequence will be I2, I1, and I4. I2 is already linked to the root node. Hence it will be incremented by 1. Similarly, I1 will be incremented by 1 as it is already linked with I2 in T1, thus {I2:3}, {I1:2}, {I4:1}.
7. T5:I1, I2, I3, I5. The sequence will be I2, I1, I3, and I5. Thus {I2:4}, {I1:3}, {I3:2}, {I5:1}.
8. T6: I1, I2, I3, I4. The sequence will be I2, I1, I3, and I4. Thus {I2:5}, {I1:4}, {I3:3}, {I4 1}.



Advantages of FP Growth Algorithm

Here are the following advantages of the FP growth algorithm, such as:

- This algorithm needs to scan the database twice when compared to Apriori, which scans the transactions for each iteration.
- The pairing of items is not done in this algorithm, making it faster.
- The database is stored in a compact version in memory.
- It is efficient and scalable for mining both long and short frequent patterns.

Disadvantages of FP-Growth Algorithm

This algorithm also has some disadvantages, such as:

- FP Tree is more cumbersome and difficult to build than Apriori.
- It may be expensive.
- The algorithm may not fit in the shared memory when the database is large.

Difference between Apriori and FP Growth Algorithm

Apriori and FP-Growth algorithms are the most basic FIM algorithms. There are some basic differences between these algorithms, such as:

Apriori	FP Growth
Apriori generates frequent patterns by making the itemsets using pairings such as single item set, double itemset, and triple itemset.	FP Growth generates an FP-Tree for making frequent patterns.
Apriori uses candidate generation where frequent subsets are extended one item at a time.	FP-growth generates a conditional FP-Tree for every item in the data.
Since apriori scans the database in each step, it becomes time-consuming for data where the number of items is larger.	FP-tree requires only one database scan in its beginning steps, so it consumes less time.
A converted version of the database is saved in the memory	A set of conditional FP-tree for every item is saved in the memory
It uses a breadth-first search	It uses a depth-first search.