Chapter 5
Association Analysis

Er. Shiva Ram Dam

Assistant Professor

Gandaki University

Content:

1. Basic Concepts
2. Association Rules and Analysis
3. Apriori Algorithm
4. FP-Growth Algorithm

5.1 Basic Concept

5.1 Association Analysis

- Association analysis is a fundamental concept in data mining, particularly used to discover relationships between variables in large datasets.
- The main objective is to identify patterns, correlations, or associations among items in transactional databases.
- Association rule learning is a type of **unsupervised learning** technique that **checks for the dependency of one data item on another data item** and maps accordingly so that it can be more profitable.
- These associations can reveal important insights into consumer behavior, market basket analysis, and many other applications.
- For example, the following rule can be extracted from the data set shown in Table 1:

$$\{Diapers\} \rightarrow \{Beer\}$$

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Table 1: Market basket transactions of five customers

Basic concepts involved in Association analysis:

- Below are the basic concepts involved in association analysis:
  1. Association Rules
  2. Itemset
  3. Support
  4. Support count
  5. Minimum Support
  6. Frequent Itemset
  7. Confidence
  8. Minimum confidence

1. Association Rules

- Association rules describe relationships between items in datasets. A typical association rule has the form: A ⇒ B
- where:
  - **A** is the antecedent (left-hand side) — the condition or itemset.
  - **B** is the consequent (right-hand side) — the resulting item or event.
- Example:

  {milk} → {bread}

- implies that if a customer buys milk, they are likely to buy bread. But, not the reverse.

Transaction ID	Items Purchased
T1	{milk, bread, butter}
T2	{milk, bread}
T3	{bread, butter}
T4	{milk, butter}
T5	{milk, bread}

2. Itemset

- An **itemset** refers to a collection of one or more items from a dataset that are considered together. It is a set of items that can be found in a particular transaction or a collection of transactions.
  - **Single Itemset**: An itemset containing just one item. For example, {milk} is a single-item itemset.
  - **Multi-itemset**: An itemset containing multiple items. For example, {milk, bread} is a two-item itemset, which refers to a combination of two items appearing together in transactions.
- Examples:
  - **Single-item itemsets**: {milk}, {bread}, {butter}
  - **Two-item itemsets**: {milk, bread}, {milk, butter}, {bread, butter}
  - **Three-item itemsets**: {milk, bread, butter}

Transaction ID	Items Purchased
T1	{milk, bread, butter}
T2	{milk, bread}
T3	{bread, butter}
T4	{milk, butter}
T5	{milk, bread}

3. Support

- The **support** of an itemset is the proportion of transactions in the dataset that contain the itemset.
- It is a measure of how often the itemset appears in the dataset.

$$\text{Support}(X) = \frac{\text{Number of transactions containing } X}{\text{Total number of transactions}}$$

- Example:
- For the itemset {milk, bread}, it appears in transactions T1, T2, and T5. Therefore, the support of {milk, bread} is:

$$\text{Support}(\{milk, bread\}) = \frac{3}{5} = 0.6$$

Transaction ID	Items Purchased
T1	{milk, bread, butter}
T2	{milk, bread}
T3	{bread, butter}
T4	{milk, butter}
T5	{milk, bread}

4. Support count

- It is the frequency of occurrence of an itemset, i.e number of transactions it appears.
- Example:
  - Support count of {milk, bread} = 3

Transaction ID	Items Purchased
T1	{milk, bread, butter}
T2	{milk, bread}
T3	{bread, butter}
T4	{milk, butter}
T5	{milk, bread}

5. Minimum support (minsup)

- **Minimum support** (minsup) refers to the threshold value that determines whether an itemset is considered **frequent** or not.
- It is a user-defined parameter.
- It is a parameter supplied to the Apriori algorithm in order to prune candidate rules by specifying a minimum lower bound for the Support measure of resulting association rules.

Transaction ID	Items Purchased
T1	{milk, bread, butter}
T2	{milk, bread}
T3	{bread, butter}
T4	{milk, butter}
T5	{milk, bread}

6. Frequent Itemsets

- A **frequent itemset** in the context of **association rule mining** refers to a set of items that appear together in a dataset (such as transaction data) with a frequency or **support** greater than or equal to a predefined minimum threshold.
- Now, let's say the **minimum support threshold** is 40%, or 2 transactions (since there are 5 transactions in total).
- For the itemset {milk, bread}, it appears in transactions T1, T2, and T5. Therefore, the support of {milk, bread} is:

$$\text{Support}(\{milk, bread\}) = \frac{3}{5} = 0.6$$

- Since 0.6 (60%) is greater than the minimum support of 40%, {milk, bread} is a **frequent itemset.**
- On the another way, the support count is 3, which is greater than or equal to 2, so {milk, bread} is a **frequent itemset.**

Transaction ID	Items Purchased
T1	{milk, bread, butter}
T2	{milk, bread}
T3	{bread, butter}
T4	{milk, butter}
T5	{milk, bread}

7. Confidence

- For an association rule **A ⇒ B**, the **confidence** is the conditional probability of observing item B given that item A has already occurred. It is calculated as:

$$\text{Confidence}(A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

- Where:
  - **Support(A ∪ B)** is the transactions containing both A and B.
  - **Support(A)** is the transactions containing A.
- Example: **Confidence of the association rule {milk} ⇒ {bread}** is:

$$\text{Confidence} = \frac{\text{Support}(\{milk, bread\})}{\text{Support}(\{milk\})}$$

$$= \frac{\text{number of transactions containing both milk and bread}}{\text{Total numbers of transactions only } milk}$$

$$= \frac{3}{4} = 0.75 = 75\%$$

- This means that there is a 75% chance that a customer who buys milk will also buy bread.

Transaction ID	Items Purchased
T1	{milk, bread, butter}
T2	{milk, bread}
T3	{bread, butter}
T4	{milk, butter}
T5	{milk, bread}

8. Minimum Confidence:

- Just like **minimum support, minimum confidence** is often set as a threshold in association rule mining.
- If the confidence of a rule is greater than or equal to the minimum confidence, the rule is considered strong and is kept.

Transaction ID	Items Purchased
T1	{milk, bread, butter}
T2	{milk, bread}
T3	{bread, butter}
T4	{milk, butter}
T5	{milk, bread}

Approaches for Association Rules Mining

1. Apriori
2. FP-Tree

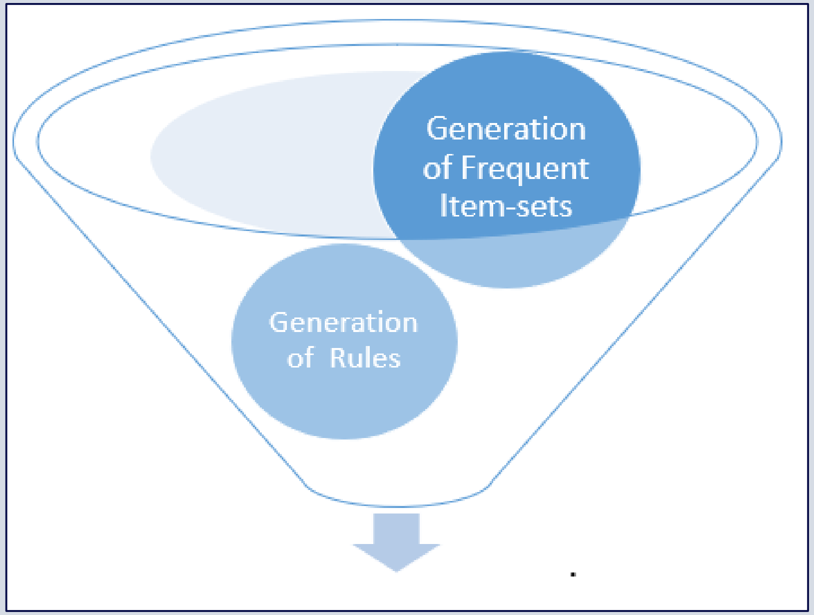
1. Apriori Algorithm

- The **Apriori algorithm** is a classical algorithm used in **association rule mining** to identify frequent itemsets in a transaction dataset and generate association rules based on those itemsets.
- It is particularly used for **market basket analysis** and has been widely applied in various domains to find relationships between items that are frequently purchased together.
- The Apriori algorithm works on the **bottom-up** approach, iteratively identifying larger itemsets based on smaller, frequent itemsets.
- It uses the **Apriori property**: **if an itemset is frequent, all of its subsets must also be frequent**.
- This property helps the algorithm efficiently prune the search space.

Association Rules Mining: Process

1. Frequent itemset generation

- Itemsets whose **support is greater than the minimum_support** are called ***frequent itemset.***
- The main objective is to find all the itemsets that satisfy the minimum support threshold. These itemsets are called ***frequent itemsets.***
- All other item-sets are filtered out.

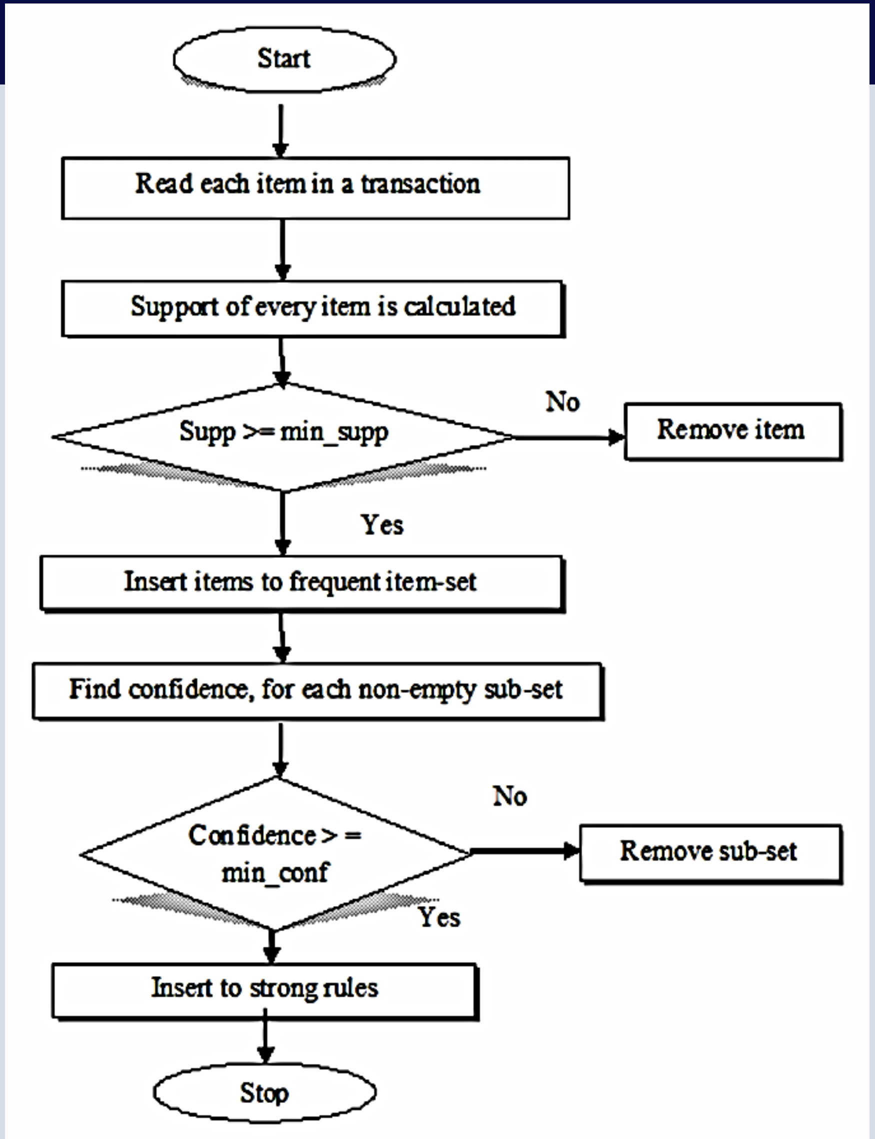


2. Rule generation

- Association Rules are **generated from the frequent itemset.**
- **Confidence** for each of these association rules are calculated first.
- Association Rules that meets the minimum confidence thresholds are our required Association rules.

Apriori Algorithm

1. Input : A dataset, minimum support and minimum confidence threshold.
2. Generate Frequent 1-itemsets and count the support.
   - If the support >= min sup, keep the itemset and make new candidate itemset
   - Else, prune it.
3. Repeat step 2 increasing the size of the itemsets (eg: 2-itemsets, 3-itemsets and so on) until no more frequent itemsets can be generated,
4. Generate association rules considering the confidence.



Solved Example 1: Apriori Algorithm

- Given the following transaction set, find the frequent itemset using Apriori Algorithm. Assume Minimum support=2, confidence = 75%. Also, evaluate the Association rules.

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Given dataset is:

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Step 1: Frequent ItemSet Generation

Frequent 1-itemsets:

Itemset	Support Count	Total Transactions	Support (%)	Remarks
{1}	2	4	50%	
{2}	3	4	75%	
{3}	3	4	75%	
{4}	1	4	25%	Pruned
{5}	3	4	75%	
Frequent 1-itemsets	{1}, {2}, {3}, {5}			

Frequent 2-itemsets:

Itemset	Support Count	Total Transactions	Support (%)	Remarks
{1, 2}	1	4	25%	Pruned
{1, 3}	2	4	50%	
{1, 5}	1	4	25%	Pruned
{2, 3}	2	4	50%	
{2, 5}	3	4	75%	
{3, 5}	2	4	50%	
Frequent 2-itemsets	{1, 3}, {2, 3}, {2, 5}, {3, 5}			

Frequent 3-itemsets:

Itemset	Support Count	Total Transactions	Support (%)	Remarks
{1, 2, 3}	1	4	25%	Pruned
{1, 2, 5}	1	4	25%	Pruned
{1, 3, 5}	1	4	25%	Pruned
{2, 3, 5}	2	4	50%	
Frequent 3-itemsets	{2, 3, 5}			

- The final “frequent” item sets are those remaining in Frequent Itemset 2 and Frequent Itemset 3. However, {2,3}, {2,5}, and {3,5} are all contained in the larger item set {2, 3, 5}.
- Thus, the final group of item sets reported by Apriori are **{1,3}** and **{2,3,5}.** These are the only item sets from which we will generate association rules.

Step 2: Generation of Association rules

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

We have:

Confidence (A→B) = $\frac{\text{number of transactions containing both A and B}}{\text{Total numbers of transactions only A}}$

- **Confidence (1→3)**
  $= \frac{\text{number of transactions containing both 1 and 3}}{\text{Total numbers of transactions only 1}}$
  $= \frac{2}{2}$
  = 1.0
- **Confidence (3→1)**
  $= \frac{\text{number of transactions containing both 3 and 1}}{\text{Total numbers of transactions only 3}}$
  $= \frac{2}{3}$
  = 0.67
- **Confidence ({2,3}→5)**
  $= \frac{\text{number of transactions containing both \{2,3\} and 5}}{\text{Total numbers of transactions only \{2.3\}}}$
  $= \frac{2}{2}$
  = 1.0

Candidate rules for {1,3}		Candidate rules for {2,3,5}			
Rule	Conf.	Rule	Conf.	Rule	Conf.
{1}→{3}	2/2 = 1.0	{2,3}→{5}	2/2 = 1.00	{2}→{5}	3/3 = 1.00
{3}→{1}	2/3 = 0.67	{2,5}→{3}	2/3 = 0.67	{2}→{3}	2/3 = 0.67
		{3,5}→{2}	2/2 = 1.00	{3}→{2}	2/3 = 0.67
		{2}→{3,5}	2/3 = 0.67	{3}→{5}	2/3 = 0.67
		{3}→{2,5}	2/3 = 0.67	{5}→{2}	3/3 = 1.00
		{5}→{2,3}	2/3 = 0.67	{5}→{3}	2/3 = 0.67

Assuming a min. confidence of 75%, the final set of rules reported by Apriori are: {1}→{3}, {2,3}→{5}, {3,5}→{2}, {5}→{2} and {2}→{5}

Assignment:

1. Given the following transaction set, find the frequent itemset using Apriori Algorithm. Assume Minimum support=33%, confidence = 60%. Also, evaluate the Association rules.

1	Milk	Egg	Bread	Butter
2	Milk	Butter	Egg	Ketchup
3	Bread	Butter	Ketchup	
4	Milk	Bread	Butter	
5	Bread	Butter	Cookies	
6	Milk	Bread	Butter	Cookies
7	Milk	Cookies		
8	Milk	Bread	Butter	
9	Bread	Butter	Egg	Cookies
10	Milk	Butter	Bread	
11	Milk	Bread	Butter	
12	Milk	Bread	Cookies	Ketchup

2. Use the Apriori algorithm using candidate generation for finding frequent itemset and then evaluate the valid association rules:

TID	List of items
T100	A,C,D
T200	B,C,E
T300	A,B,C,E
T400	B,E

3. Given the following transaction set, find the frequent itemset using Apriori algorithm. (Minimum support =2).

Transaction	Items
T1	{pasta, lemon, bread, orange}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{pasta, lemon, orange, cake}

Reference: https://www.youtube.com/watch?v=43CMKRHdH30

5.3 FP-Growth and FP Tree

4.3 FP-Growth

- **Frequent Pattern Growth Algorithm**
- This algorithm is an **improvement to the Apriori method.**
- A frequent pattern is generated without the need for candidate generation.
- FP growth algorithm **represents the database in the form of** a tree called a frequent pattern tree or **FP tree.**
- This tree structure will maintain the association between the itemsets.
- The database is fragmented using one frequent item. This fragmented part is called “pattern fragment”.
- The itemsets of these fragmented patterns are analyzed.
- Thus, with this method, the search for frequent itemsets is reduced comparatively.

FP tree

- Frequent Pattern Tree is a tree-like structure that is made with the initial itemsets 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 itemset.
- The root node represents null while the lower nodes represent the itemsets.
- The association of the nodes with the lower nodes that is the itemsets with the other itemsets are maintained while forming the tree.

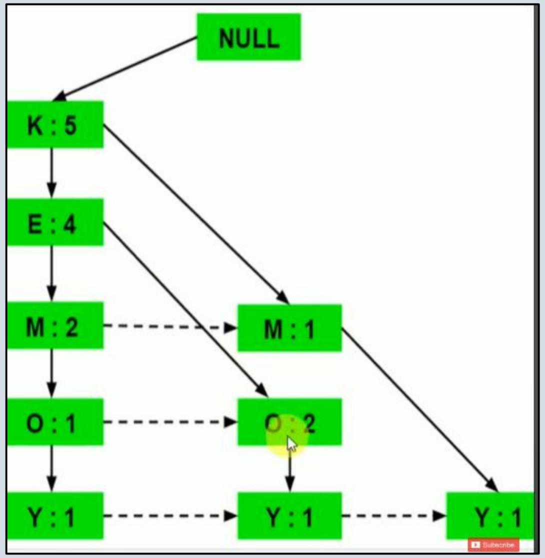


Figure: A sample FP-tree

Advantages and Disadvantages of FP algorithm

- **Advantages :**
  1. This algorithm **needs to scan the database only twice** when compared to Apriori which scans the transactions for each iteration.
  2. **Faster** since the pairing of items is not done in this algorithm.
  3. The database is stored in a **compact version in memory.**
  4. It is **efficient and scalable** for mining both long and short frequent patterns.

- **Disadvantages:**
  1. FP Tree is **more cumbersome and difficult** to build than Apriori.
  2. It may be **expensive**.
  3. When the **database is large, the algorithm may not fit** in the shared memory.

Frequent Pattern Algorithm Steps

1. **Compute the *frequencies*** of the itemsets in the database.
2. **Arrange the itemsets in *descending order*** after applying Minimum_support.
3. **Generate *ordered_itemset.***
4. **Construct *FP-tree*.**
5. **Compute *Conditional pattern base*.**
6. **Generate *Conditional Frequent Patterns Tree***
7. Generate *Frequent Pattern* from the Conditional FP Tree, considering the **minimum_support.**

Refernce: https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/#:~:text=FP%20growth%20algorithm%20represents%20the,is%20called%20%E2%80%9Cpattern%20fragment%E2%80%9D

Solved Example 1:

- Construct FP-tree for below hypothetical dataset of transactions with each letter representing an item. Let the minimum support be 3. Assuming confidence is 75%, find association rules.

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

Reference: https://www.geeksforgeeks.org/ml-frequent-pattern-growth-algorithm/

Reference Video: https://www.youtube.com/watch?v=7oGz4PCp9jI

Solution:

Given dataset is:

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

Step 1: Compute the frequencies of the itemsets in the database

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

Step 2: Build FP set (L) in descending order after applying Minimum_support.

L = {K : 5, E : 4, M : 3, O : 3, Y : 3}

Step 3: Generate ordered_itemset.

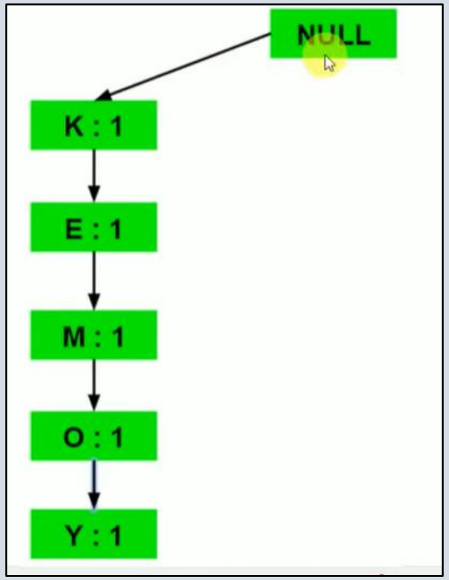
Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}

Step 4: Construct FP-tree.

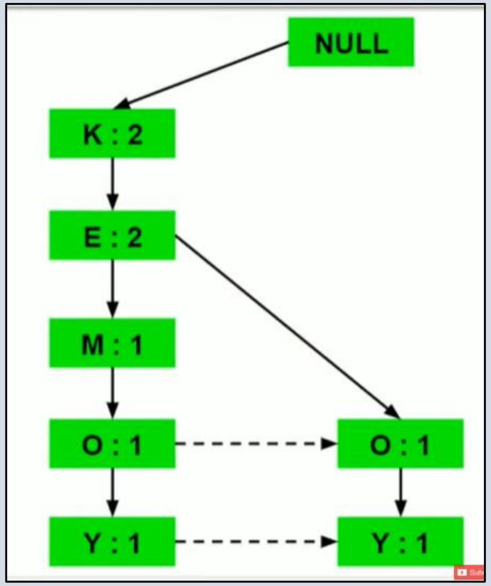
Reference

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

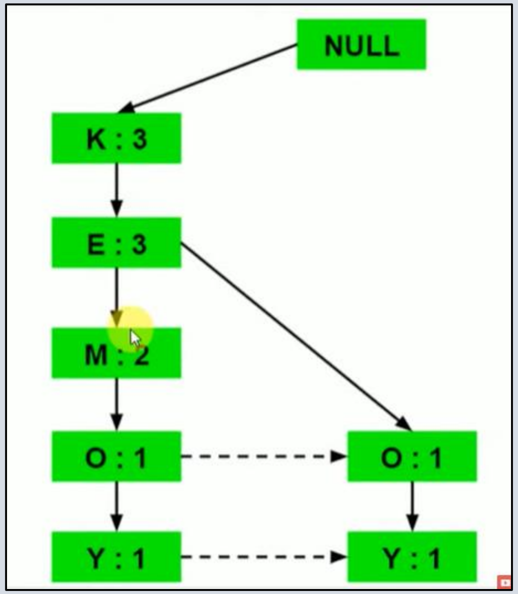
a) Inserting the set {K, E, M, O, Y}:



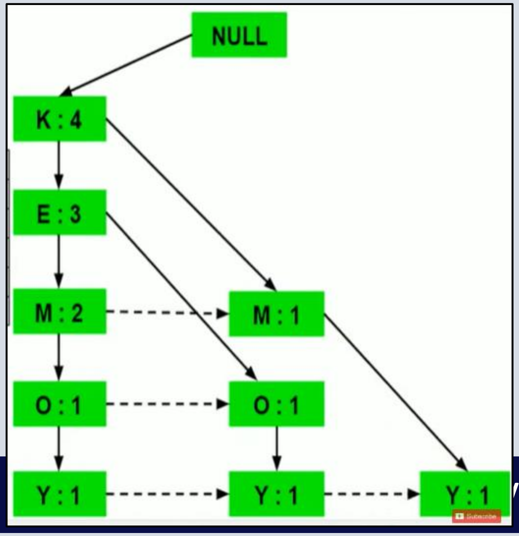
b) Inserting the set {K, E, O, Y}:



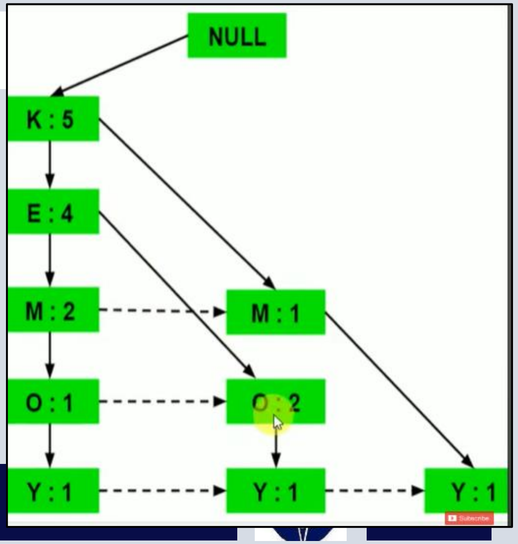
c) Inserting the set {K, E, M}:



d) Inserting the set {K, M, Y}:

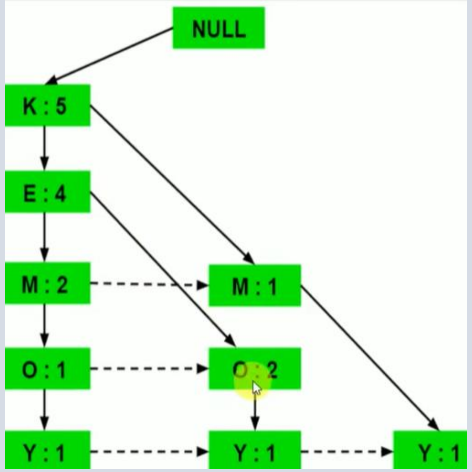


e) Inserting the set {K, E, O}:



Reference

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



Step 5: Compute *Conditional pattern base* from the FP tree.

Now, for each item, the **Conditional Pattern Base** is computed which is path labels of all the paths which lead to any node of the given item in the frequent-pattern tree.

Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	

Step 6 :Generate Conditional Frequent Patterns Tree

Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

Step 7 :Generate Frequent Patterns Rules

- From the Conditional Frequent Pattern tree, the **Frequent Pattern rules** are generated by pairing the items of the Conditional Frequent Pattern Tree set to the corresponding to the item as given in the below table.

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<K, E: 4}
K	

Step 8 : Find the confidence for each rule and apply the confidence threshold.

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<K, E: 4}
K	

Step 8 : Find the confidence for each rule and apply the confidence threshold (Assume confidence=75%).

For Confidence (A→B) = number of transac~~tions containing~~ both A and B / Total number of transactions only A

Candidate rules for {K,Y}	Confidence
{K}→{Y}	3/5 = 0.6
{Y}→{K}	3/3 = 1

Candidate rule for {K,O}, {E,O}, {E,K,O}

~~{K}→{O}~~	Confidence.	~~Confidence~~	Confidence
{K}→{O}	3/5 = 0.6	{O}→{E,K}	3/3 = 1
{O}→{K}	3/3 = 1	{E}→{K,O}	3/4 = 0.75
{E}→{O}	3/4 = 0.75	{K}→{E,O}	3/5 = 0.6
{O}→{E}	3/3 = 1		
{E,K}→{O}	3/4 = 0.75		
{K,O}→{E}	3/3 = 1		
{E,O}→{K}	3/3 = 1		

Candidate rule for (K,M)	Confidence
{K}→{M}	3/5 = 0.6
{M}→{K}	3/3 = 1

Candidate rule for {K,E}	Confidence
{K}→{E}	4/5 = 0.8
{E}→{K}	4/3 = 1.33

Assuming a min confidence of 75% of the final set of rules reported by {Y}→{K}, {O}→{K}, {E}→{O}, ~~{O}~~ {O}→{K}, {E,K}→{O}, {K,O}→{E}, {E,O}→{K}, {O}→{E,K}, {E}→{K,O}, {M}→{K}, and {E}→{K}.

Example 2:

Consider the following transaction data sets. And construct the FP tree and find the FP rules.

Support threshold=50%, Confidence= 60%

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:

Given dataset is:

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

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

Step 1: Compute the frequencies of the itemsets in the database

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

Step 2: Build FP set (L) in descending order after applying Minimum_support.

Item	Count
I2	5
I1	4
I3	4
I4	4

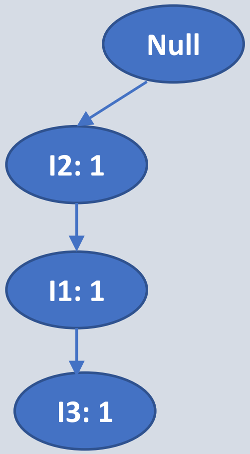
Step 3: Generate ordered_itemset.

Transaction	Items	Ordered-Item set
T1	I1, I2,I3	I2, I1,I3
T2	I2, I3, I4	I2, I3, I4
T3	I4, I5	I4
T4	I1, I2, I4	I2, I1, I4
T5	I1, I2, I3, I5	I2, I1, I3
T6	I1, I2, I3, I4	I2, I1, I3, I4

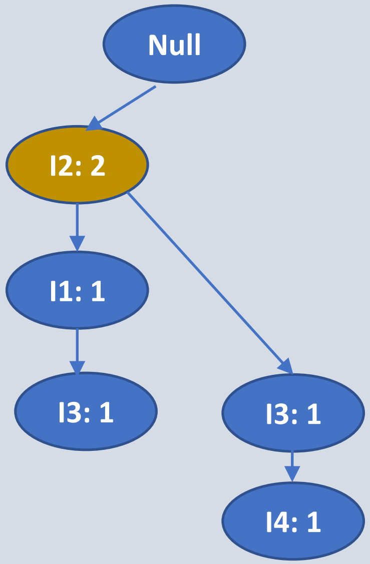
Step 4: Construct FP-tree.

Transaction	Items	Ordered-Item set
T1	I1, I2,I3	I2, I1, I3
T2	I2, I3, I4	I2, I3, I4
T3	I4, I5	I4
T4	I1, I2, I4	I2, I1, I4
T5	I1, I2, I3, I5	I2, I1, I3
T6	I1, I2, I3, I4	I2, I1, I3, I4

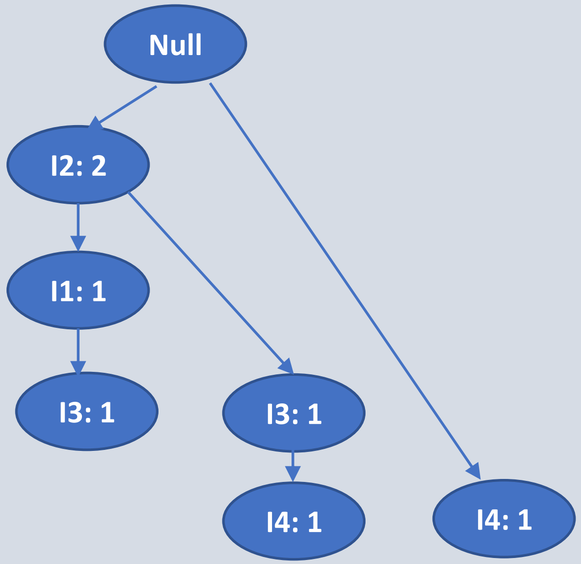
a) Inserting {I2, I1, I3}



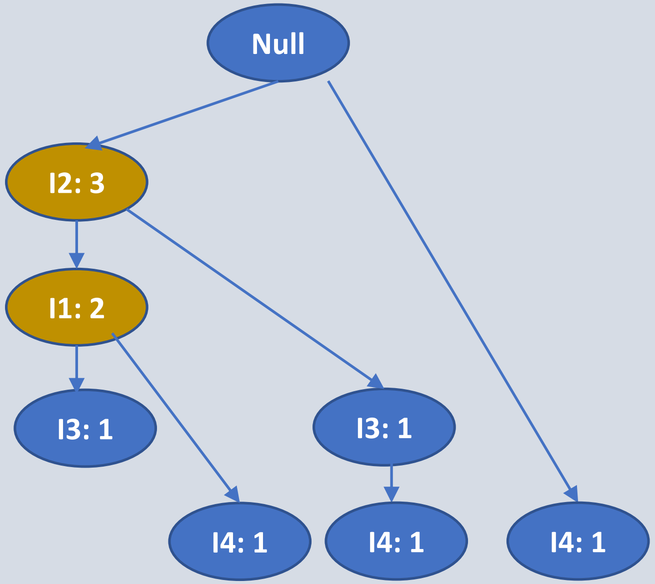
b) Inserting {I2, I3, I4}



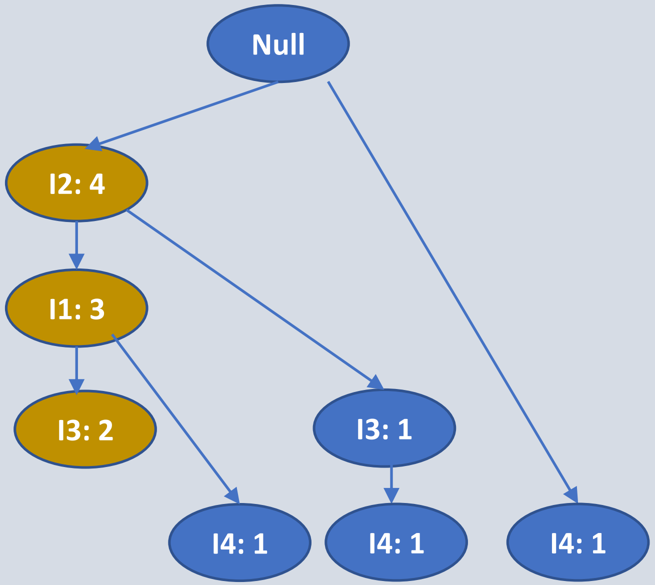
c) Inserting {I4}



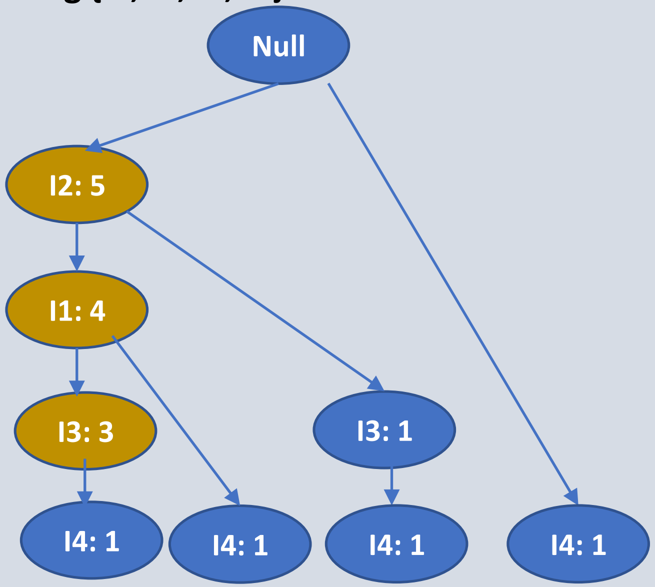
d) Inserting {I2, I1, I4}



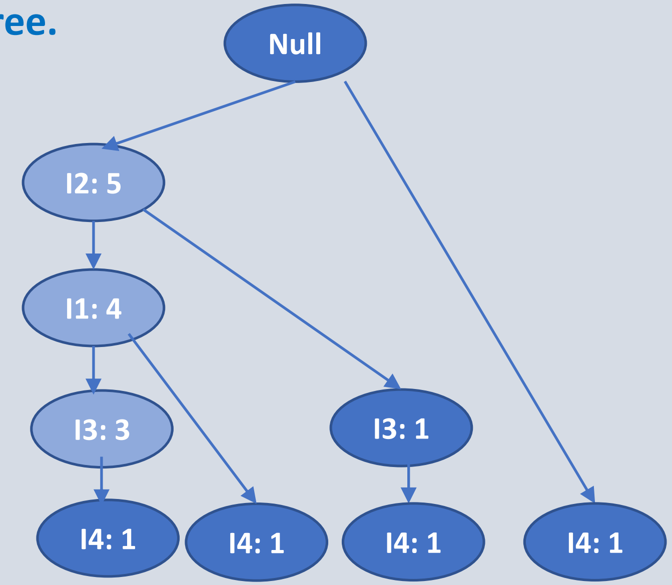
e) Inserting {I2, I1,I3}



f) Inserting {I2, I1, I3, I4}



Step 5: Compute *Conditional pattern base* from the FP tree.



Items	Conditional Patten Base
I4	{I2,I1,I3:1},(I2,I1:1}, {I2,I3:1}
I3	{I2,I1:3}, {I2 : 1}
I1	{I2: 4}
I2	

Step 6 :Generate Frequent Patterns Tree

Items	Conditional Patten Base	Conditional FP Tree set
I4	{I2,I1,I3:1}, {I2,I1:1}, {I2,I3:1}	{I2: 3}
I3	{I2,I1:3}, {I2 : 1}	{I2: 4}, {I1 :3}
I1	{I2: 4}	{I2: 4}
I2		

Step 7 :Generate Frequent Patterns Rules

- From the Conditional Frequent Pattern tree, the **Frequent Pattern rules** are generated by pairing the items of the Conditional Frequent Pattern Tree set to the corresponding to the item as given in the below table.

Items	Conditional FP Tree set	Frequent pattern
I4	{I2: 3}	{ I2, I4 : 3}
I3	{I2: 4}, {I1 :3}	{ I2, I3 : 4 }, {I1, I3 : 3}, {I2, I1, I3 : 3}
I1	{I2: 4}	[I2, I1 : 4 }
I2		

Step 8 : Find the confidence for each rule and apply the confidence threshold 60%.

Transaction	Items	Ordered-Item set
T1	I1, I2,I3	I2, I1, I3
T2	I2, I3, I4	I2, I3, I4
T3	I4, I5	I4
T4	I1, I2, I4	I2, I1, I4
T5	I1, I2, I3, I5	I2, I1, I3
T6	I1, I2, I3, I4	I2, I1, I3, I4

Items	Conditional FP Tree set	Frequent pattern
I4	{I2: 3}	{ I2, I4 : 3}
I3	{I2: 4}, {I1 :3}	{ I2, I3 : 4 }, {I1, I3 : 3}, {I2, I1, I3 : 3}
I1	{I2: 4}	[I2, I1 : 4 }
I2		

Candidate rules for {I2, I4}		Candidate rules for {I2, I3}		Candidate rules for {I1, I3}		Candidate rules for {I2, I1}		Candidate rules for {I2, I1,I3}			
Rule	Confidence	Rule	Confidence	Rule	Confidence	Rule	Confidence	Rule	Confidence	Rule	Confidence
I2→ I4		I2→ I3		I1→ I3		I2→ I1		{I2, I1} → I3		I1→ I2	
I4 →I2		I3→I2		I3→ I1		I1 → I2		{I1, I3} → I2		I2→I1	
								{I3, I2} → I1		I2→ I3	
								I3→ {I2, I1}		I3→ I2	
								I2 → {I1, I3}		I1→ I3	
								I1→ {I3, I2}		I3→I1	

Assignment:

1. **Consider the following transaction data sets. And construct the FP tree and find the FP rules. Support threshold=50%, Confidence= 60%**

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

2. **Construct the FP tree for the following transactions. [BIM 2018, Group B]**

TID	Items
100	{a,b,c}
200	{b,c,d}
300	{a,c,e}
400	{b,d,f}
500	{a,b,e}
600	{b,c,f}

3. **Consider the following transaction data sets. And construct the FP tree. [BIM 2021, Group B]**

TID	Items
1	E, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D
5	D
6	D, B
7	A, D, E
8	B, C

Apriori Vs FP Growth

Difference between Apriori and FP Growth Algorithm

Apriori	FP Growth
1. Apriori **generates frequent patterns** by making the itemsets using pairings such as single item set, double itemset, and triple itemset.	1. FP Growth **generates an FP-Tree** for making frequent patterns.
2. Apriori **uses candidate generation** where frequent subsets are extended one item at a time.	2. FP-growth **generates a conditional FP-Tree for every item** in the data.
3. Since apriori scans the database in each step, it becomes **time-consuming** for data where the number of items is larger.	3. FP-tree requires only one database scan in its beginning steps, so it **consumes less time.**
4. A **converted version of the database is saved in the memory**	4. A **set of conditional FP-tree for every item is saved in the memory**
5. It **uses breadth-first search**	5. It **uses depth-first search.**

End of Chapter 4