Association Analysis

Support:

The support of an association pattern is the percentage of relevent data transaction for which the pattern is true i.e

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

$$= \frac{\text{No. of tuples containing } A \& B}{\text{total data set}}$$

Confidence:-

Confidence is defined as the measure of certainty or trust associated with each discovered pattern i.e.

$$\text{Confidence}(A \Rightarrow B) = P(A \cap B)$$

$$= \frac{\text{No. of tuple containing } A \& B}{\text{No. of tuple containing } A}$$

Itemset:-

A set of item is referred as itemset. An itemset containing 'k' is called k-itemset.

An itemset satisfies minimum support then it is called fr itemset.

Association Rule Mining:->

Mining of association rule involves

(i) frequent item generation

(ii) Rules generation.

Given a set of transactions, the goal of association rule mining i find all rules having

(i) support $\geq$ minimum support threshold.

(ii) confidence $\geq$ minimum confidence threshold.

Approaches of Rule Mining:-

(A) **Brute force Approach:**

(i) List all possible association rules.

(ii) Compute the support and confidence for each rule

(iii) Pure rules that fail the minimum support and minimum confidence threshold

(B) Apriori Approach:-

If an itemset is frequent then all of its subsets must al[so] be frequent. or, superset of non-frequent itemset is also non-freque[nt]

Apriori algorithm is an influential algorithm for mining freque[nt] itemset

→ It use a level wise search i.e K item sets are used to explo[re] K+1 itemset.

→ At first the set is found at level 1, ~~and so on until no freque[nt]~~ ~~itemset at~~ level 1 is used to find frequent itemset at lev[el] 2 and so on until no frequent itemset is found.

Algorithm:-

Step 1 → Read the transaction database and get support from each itemset.

Step 2 → compute the support with minimum support to generate ~~as~~ of frequent itemset at level 1.

Step 3 → use joint to generate a set of candidate K item set at next level.

Step 4 → Generate frequent itemset at next level using minimum support.

Step 5 → Repeat 2 and 3 until no frequent Itemset can be generated

Step 6 → Generate rules from frequent item sets from level 2 onward using minimum confidence.

Example:

T_{id}	Items
1	A, C, D
2	B, C, E
3	A, B, C, E
4	B, E.
5	A, C, E

Let minimum support $= 33\% = \frac{33}{100} \times 5 = 1.65 \approx 2$ (25 to 45% Normally)

minimum confidence = 70% (65 to 80% normally) → Assume if not given

At level 1,
candidate item set

$C_1 =$

Item	count
A	3
B	3
C	4
D	1
E	4

frequent items at level 1

$F_1 =$

Item	count
A	3
B	3
C	4
E	4
~~D~~	~~1~~

D is removed as it is less frequent

At level 2,
candidate $C_2 =$

item	count
AB	1
AC	3
AE	2
BC	2
BE	3
CE	3

$F_2 =$

itemset	count
AC	3
AE	2
BC	2
BE	3
CE	3

At level 3,

$C_3 =$

item	count
ACE	2
~~ABC~~	1
~~ABE~~	1
BCE	2

$F_3 =$

item	count
ACE	2
BCE	2

At level 4

$C_4 =$

item	count
ABCE	1

(No frequent item is generated So, end.

Now, Generating rules from level 2:

$A \Rightarrow C$, confidence $(A \Rightarrow C) = \frac{3}{3} = 100\%$ (3 → occurance of AC, 3 → occurance of A)

$C \Rightarrow A$, confidence $(C \Rightarrow A) = \frac{3}{4} = 75\%$

$A \Rightarrow E$; confidence $(A \Rightarrow E) = 2/3 = 66.67\%$

$E \Rightarrow A$; confidence $(E \Rightarrow A) = 2/4 = 50\%$

Similarly other confidence can be generated

∴ Rules are

$A \Rightarrow C$
$C \Rightarrow A$ } Since minimum confidence is 70%

from Levels:

$A \Rightarrow CE$, confidence $(A \Rightarrow CE) = 2/3 = 66.67\%$

$CE \Rightarrow A$, confidence $(CE \Rightarrow A) = 2/3 = 66.67\%$

$AC \Rightarrow E$, confidence $(AC \Rightarrow E) = 2/3 = 66.67\%$

$E \Rightarrow AC$, confidence $(E \Rightarrow AC) = 2/4 = 50\%$

$C \Rightarrow AE$,

$AE \Rightarrow C$,

⋮

Mining frequent Item sets without candidate generation.

Frequent pattern Growth (FP-Growth)

→ It is divide and conqure strategy

→ It compresses the database representing frequent items into a frequent pattern tree (FP-tree), which retain itemset association information.

→ Divides the compressed database into a set of conditional database each associated with one frequent item or pattern fragment and mines each such database seperately.

FP-tree Algorithm

step 1: creat root node of tree labeled with null

step 2: Scan the complete dataset

step 3: The items in each transactions are processed in sorted order (ascending ~~to decending~~) and branch is created for each transaction.

FP-tree Mining:

→ start from each frequent length pattern as an initial suffix pattern.

→ construct of conditional pattern base (A conditional pattern base is a sub database which consists of the set of prefix paths in the FP-tree co-occuring with suffix pattern)

→ construct each FP-tree and perform mining recursively on such tree.

Example:

Tid	List of items
1.	I_1, I_2, I_5
2.	I_2, I_4
3.	I_2, I_3
4.	I_1, I_2, I_4
5.	I_1, I_3
6.	I_2, I_3
7.	I_1, I_3
8.	I_1, I_2, I_3, I_5
9.	I_1, I_2, I_3

Let, minimum support = 2

Item	count
I_1	6
I_2	7
I_3	6
I_4	2
I_5	2

Sorted order → descending

Item	count
I_2	7
I_1	6
I_3	6
I_4	2
I_5	2

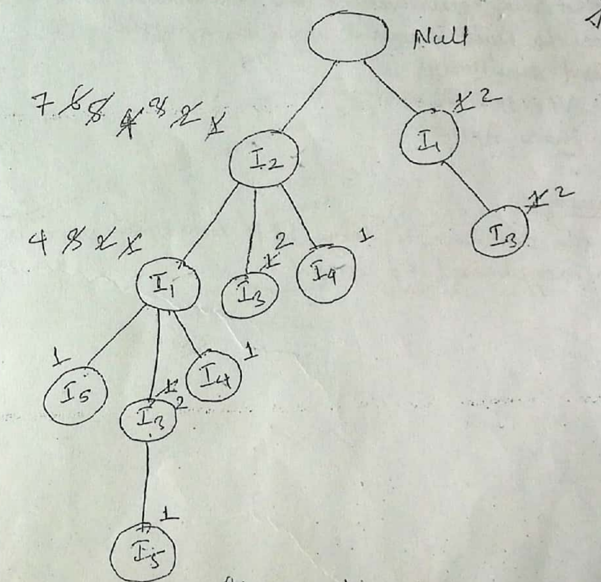


fig: FP-tree.

item	conditional pattern	Conditional FP tree	frequent pattern
I_5	$\{I_2, I_1 : 1\}, \{I_2, I_1, I_3 : 1\}$	$(I_2 : 2)(I_1 : 2)$	$(I_2, I_5 : 2)(I_1, I_5 : 2)(I_1, I_2, I_5 : 2)$
I_4	$\{I_2, I_1 : 1\}, \{I_2 : 1\}$	$(I_2 : 2)$	$(I_2, I_4 : 2)$
I_3	$\{I_2, I_1 : 2\}, \{I_2 : 2\}, \{I_1 : 2\}$	$(I_2 : 4)(I_1 : 2)(I_1 : 2)$	$(I_2, I_3 : 4)(I_1, I_3 : 4)(I_1, I_2, I_3 : 4)$
I_1	$\{I_2 : 4\}$	$(I_2 : 4)$	$(I_2, I_1 : 4)$

(path to an item I_1) (I_2 start first i.e same branch so add) (different branch write seperately) (makecombination with item)

$F_3 =$

item	count
I_1, I_2, I_5	2
$I_1 I_2 I_3$	4

$F_2 =$

item	count
$I_1 I_5$	2
$I_2 I_5$	2
$I_2 I_4$	2
$I_2 I_3$	4
$I_1 I_3$	4
$I_2 I_1$	4

Advantages of FP-growth.

(1) It transform the problem of finding long frequent pattern to searching for shorter ones recursively then concatenating the suffix.
(2) It uses the least frequent item as a suffix.
(3) Has good sensitivity
(4) Reduce the search cost
(5) faster than Apriori

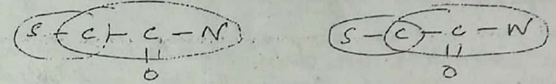
Disadvantages:-

(1) when the database is large, it is sometime unrealistic to construct a main memory based Fp-tree.

Sequence pattern:-

Eg. a d ca b ca e d ca

Subgraph pattern:-



Infrequent pattern:

→ Not common items
→ can hold significant information
→ useful in scientific research prediction
→ useful in rare item identification.

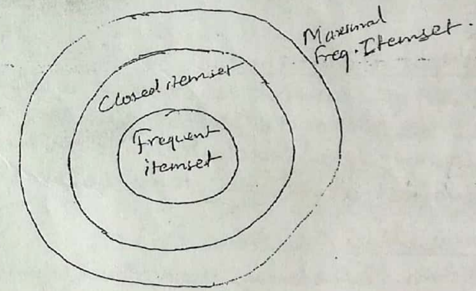
Maximal frequent Itemset:-

An itemset is maximal if non of its immediate superset is freque

(super set बराबर)

Closed item set:

An itemset is closed if non of its immediate superset has the sam as of the item set.



Bind