MARKET BASKET ANALYSIS / ASSOCIATION ANALYSIS / ASSOCIATION RULE MINING

- Association analysis also known as Market Basket Analysis, is a datamining technique used to discover relationships between items in a data set.
- It is commonly used in retail for understanding Customer purchasing behaviour.
- In association analysis mostly used algorithm is Apriori algorithm, which identifies frequent itemsets, and generates association Rules.
- These rules help businessess, understand patterns like "If item A is purchased, then item B is also likely to be Purchased".
- Association analysis is useful for discovering interesting relationships hidden in large data sets.
- From the given set of transactions, find rules that will predict the occurence of an item based on occurence of other item in the transaction.

Market Basket Transactions

Tid	Items
1	Bread, milk
2	Bread, Diapper, Beer, Eggs
3	Milk, Diapper, Beer, Coke
4	Bread, milk, Diapper, Beer
5	Bread, milk, Diapper, coke

Binary Representations :-

Tid	Bread	Milk	Diapers	Beer	Eggs	Coke
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Item set, Support count, support & confidence :-

Item set :-

- In association analysis, a collection of zero (or) more items is formed an itemset.
- If an itemset contains K-items, it is called K- itemset.

Eg :- {Beer, Diaper, milk} is an example of 3-itemset.

Support count :-

- It refers, the no.of times a particular itemset appears in a dataset.
- It is a measure of how frequently a combination of items occurs together.
- Frequency of occurence of an itemset.

Eg :- σ({milk, Bread, Diaper}) = 2.

Scanned with OKEN Scanner

Support :-

- Support is the proportion / fraction of transactions in a dataset that contain specific itemset.
- It is calculated by dividing the no. of transactions containing an itemset by the total no. of transactions.
- An implication expression of the form $\underset{\text{if}}{\overset{\text{antecedent}}{X}} \rightarrow \underset{\text{then}}{\overset{\text{consequent}}{Y}}$, where X & Y are itemsets.

if it is raining, then the streets will be wet.
Antecedent — Consequent

Eg :- {milk, Diaper} → {Beer}

Support, $S(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \Rightarrow \frac{2}{5}$

Note :- High support indicates that the itemset occurs frequently i.e, strong association between the items in the dataset.

confidence :-

Confidence measures the reliability of the inference made by the ~~rule~~ rule. It's the probability of the consequent being true when the antecedent is true.

- confidence is calculated as the ratio of the support count of the combined itemset to the support count of the antecedent alone :

$$\boxed{\text{Confidence, } c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}}$$

where (X) is the antecedent
(Y) is the consequent
(X∪Y) represents the combined itemset of antecedent & consequent.

Note :- A high confidence value indicates that the consequent is often found in transactions containing the antecedent, suggesting a strong association between the two items.

Eg :-

Tid	Items
1	Bread, milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, coke
4	Bread, milk, Diaper, Beer
5	Bread, milk, Diaper, coke.

Support = 0.4
Confidence = 0.5

$\{milk, Diaper\} \Rightarrow Beer$

$$\text{Support (S)} = \frac{\sigma(\{milk, Diaper, Beer\})}{N} = \frac{2}{5} = \underline{0.4}$$

$$\text{confidence (C)} = \frac{\sigma(\{milk, Diaper, Beer\})}{\sigma(\{milk, Diaper\})} = \frac{2}{3} = \underline{0.6}$$

- The goal of association rule mining is to find all rules having

support ≥ minsup threshold
confidence ≥ minconf threshold

- If the rule satisfied support and confidence threshold them that rule is strong rule.

$\boxed{\{milk, Diapers\} \Rightarrow Beer}$

is the strong rule because satisfies minsup & minconf.

Apriori Principle:-

Reduce the number of candidate itemsets and to reduce the number of Comparisons we can use Aprior principle

Aprior Principle:-for frequent itemsets

* If an itemset is frequent then all of its Subsets must also be frequent
* Aprior principle holds due to the following property of the Support measure

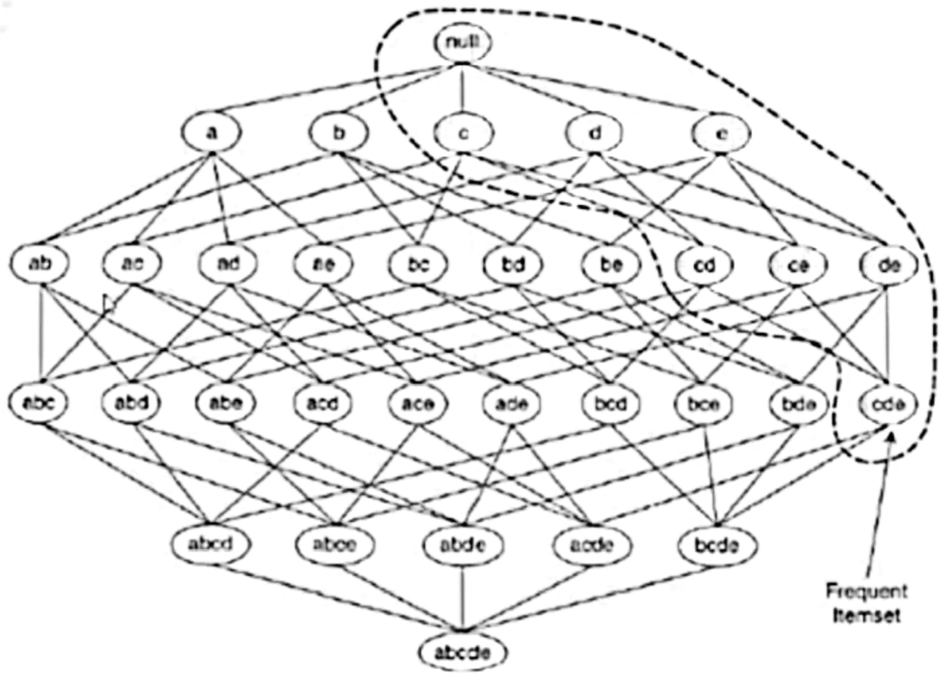
$$\forall X, Y : (X \subseteq Y) \Rightarrow S(X) \geq S(Y)$$

* Support of an itemset never exceeds the Support of it sub sets

* This is known as anti-monotone property of Support

From the below lattice diagram If [c,d,e] is frequent then all subsets i.e [cd, ce, dc, c, d, e] also be frequent

An illustration of the Apriori principle. If {c,d,e} is frequent then all subsets of this itemset are frequent.

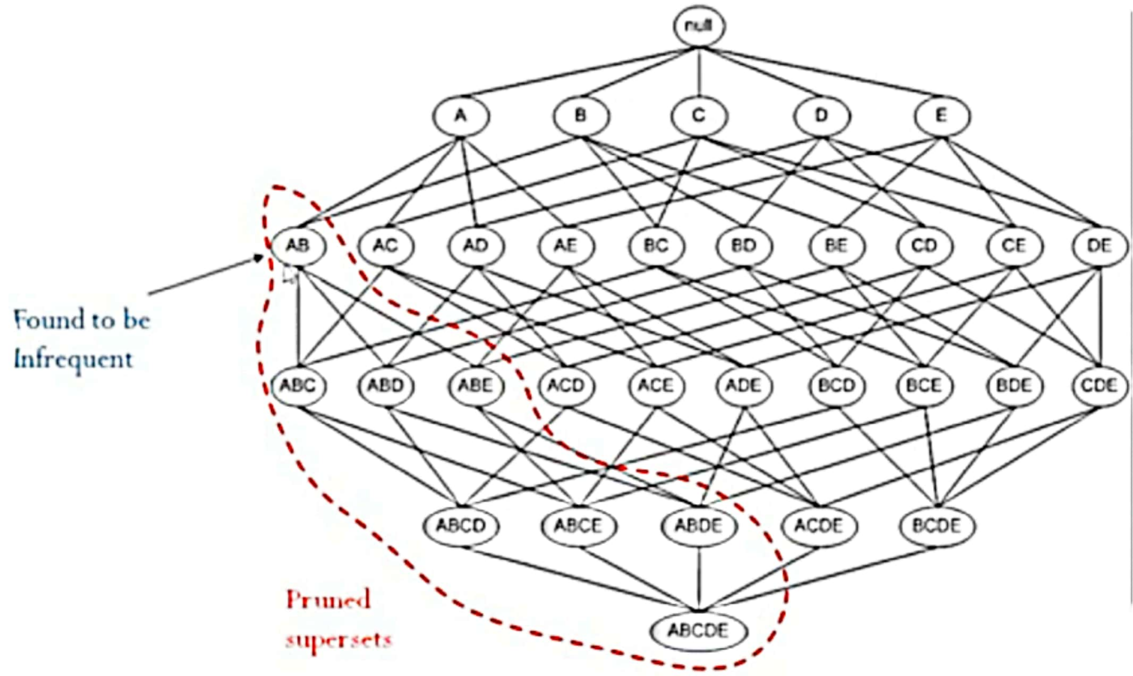


Aprior Principle for infrequent itemsets:

* If an itemset is infrequent, then all of its Supersets must also be infrequent

* If AB is infrequent, then all of its Super Sets i.e {ABC, ABD, ABE, ABCD, ABCE, ABDE, ABCDE} also be frequent

An illustration of support-based pruning. If {a ,b} is infrequent then all supersets {a ,b} are infrequent.



Apriori Algorithm

1) Let k=1

Generate frequent item sets of length 1.

Repeat until no new frequent itemsets are identified

2) Generate length (k+1) candidate itemsets from length k that are frequent

③ Count the support count of each candidate by scanning the given dataset

④ Eliminate candidates that are infrequent having only that are frequent.

Example:

TID	items
T1	1,3,4
T2	2,3,5
T3	1,2,3,5
T4	2,5
T5	1,3,5

Min support count=2

min confidence = 60%

Step 1: create 1-candidate frequent item sets (or) create itemsets as size 1

C1

Itemset	Support count
[1]	3
[2]	3
[3]	4
[4]	1 eliminated
[5]	4

⇒

F1

Itemset	Support count
[1]	3
[2]	3
[3]	4
[5]	4

Step 2:- Create 2-candidate frequent itemset using F_1

(or)

Create items as size 2 using F_1

Item set	Support count
[1,2]	1
[1,3]	3
[1,5]	2
[2,3]	2
[2,5]	3
[3,5]	3

$\Rightarrow$

F_2

Itemset	Support count
[1,3]	3
[1,5]	2
[2,3]	2
[2,5]	3
[3,5]	3

Step 3: Create 3-candidate itemsets using F_2

(or)

Create itemset as size 3 using F_2

Itemset	Support count
[1,3,5]	2
[1,2,3]	1
[1,2,5]	1
[2,3,5]	2

eliminate.

$\Rightarrow$

F_3

Itemset	Support count
[1,3,5]	2
[2,3,5]	2

Step 4:- Create 4-candidate itemsets using F_3

Create itemset as size 4 using F_3

Item set	Support count
[1,2,3,5]	1

Eliminated

So, itemsets of size 3 items considered as Frequent item sets

Frequent itemsets are : [1,3,5]

[2,3,5].

From the freequent item set {1, 3, 5} and {2, 3, 5}
we can find out 2-item, frequent item sets based on apriori principle.

If {1, 3, 5} is frequent itemset then all of sub sets {(1, 3), (1, 5), (3, 5), (1, 3, 5)} also frequent.

Likewise {2, 3, 5} ⟹ {(2, 3) (2, 5), (3, 5), (2, 3, 5)} also frequent.

FP-Growth Algorithm

The Fp-growth Algorithm is a popular method in data mining For finding frequent patterns in transectional databases

* It uses tree structure called the Fp-tree to efficiently discover Frequent itemsets without generating candidate sets explicity

Ex:- Given support count = 2

Transectional dataset

TID	Items
T_1	a, b, e
T_2	b, d
T_3	b, c
T_4	a, b, d
T_5	a, c
T_6	b, c
T_7	a, c
T_8	a, b, c, e
T_9	a, b, c

Step 1:- Find Support count For every item and arrange items in descending order based on support count

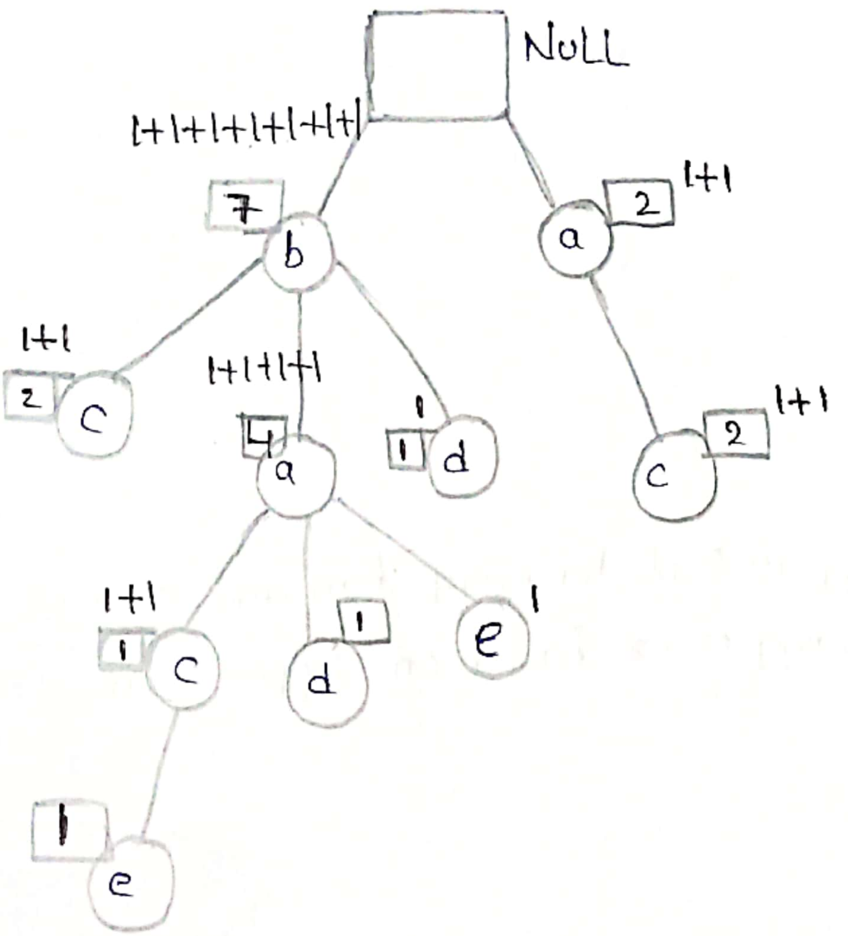
a : 6
b : 7
c : 6
d : 2
e : 2

Descending order

b	a	c	d	e
7	6	6	2	2

TID	descending order
T1	b, a, e
T2	b, d
T3	b, c
T4	b, a, d
T5	a, c
T6	b, c
T7	a, c
T8	b, a, c, e
T9	b, a, c

Step 2:- Create Fp-Tree with root as NULL



Step 3: Construct a table with frequent item sets

Item	Conditional pattern	Conditional Fp-tree	Freq pattern Generate
e	(a:1, b:1) (a:1, b:1, ~~c~~:1)	a:2, b:2 (a:2, b:2)	e:2, ea:2, eb:2, eab:2
d	(b:1) (~~a~~:1, b:1)	b:2	d:2, bd:2
c	(b:2) (a:2, b:2) (a:2)	a:4, b:4 (a:2, b:2)	c:6, ac:4, bc:4, bac:2
b	—	—	b:7
a	(a:4)	b:4	a:6, ba:3

Step 4:-

Itemset	Final Frequency Itemsets
e	e, ea, eb, eab
d	d, bd
c	c, ac, bc, bac
b	b
a	a, ba.