PAGE NO.

Market Basket Analysis

Frequent Patterns — Itemsets {milk, Bread}
— Sequential {Computer, Antivirus}

Market Basket Analysis is a Data-Mining technique or machine learning technique used by Retailers to increase sales by understanding Customer Purchasing Patterns. It involves analyzing large Datasets, such as Purchase history, to reveal Product groupings and Products that are likely to be purchased together.

How does Market Basket Analysis work?

It is modelled on Association Rule Mining, i.e the IF() , THEN() constructs. For example, if a customer buys bread, THEN he is likely to buy butter as well.

These are Represented as {Bread} → {Butter}

{Bread} : antecedent
{Butter} : consequent

→ Antecedent : Items or 'Itemsets' found within data are Antecedents. In simpler words, it's the IF component, written on the left - hand side. In the above example, bread is the Antecedent.

→ Consequent : It is an item or set of items found in combination with the Antecedent. It's the THEN component, written on the Right - Hand Side. In the above example, Butter is the consequent.

→ Association Rule Mining is all about building the Rules.

A → B — Single cardinality

A, B → C
A, B, C → D } cardinality increases

↓

lot of combinations around these data

Measures Association

- Support
- Confidence
- Lift

Support: Identifies how frequently a rule is applied to given dataset.

$$S(P \rightarrow Q) = \frac{\sigma(P \cup Q)}{N}$$ N = transactions

confidence: defines frequent occurence of items of Q in transactions of P.

$$C(P \rightarrow Q) = P(B/A)$$

Finding Frequent Item sets

- Apriori Algorithm
  - candidate generation
- FP-growth Algorithm.
  - FP tree

Support

Apriori Algorithm / Frequent Pattern mining.

It refers to the Algorithm which is used to calculate the Association rules b/w objects. It means how two or more objects are related to each other

→ Apriori Algo. is an association Rule learning that Analyzes that People who bought Product A also bought Product B.

Primary objective:

→ create the Association Rule b/w different objects.
→ It describes how 2 or more objects are related to one Another

Q

Transactions	Itemsets
T1	A, B, C
T2	A, C
T3	A, D
T4	B, E, F

min support = 50%. $\frac{50}{100}$ x 4 (no. of transactions) = 2

min confidence = 50%.

Campus: ..... Course: .....
Name of Faculty: .....
Class/Section: .....
Name of Subject: .....
Date: .....
Code: .....

C_1 = support count

Items	Support
A	3
B	2
C	2
D	1
E	1
F	1

L_1 =

Items	Support
A	3
B	2
C	2

Items set which are frequently repeating using minimum support count calculated

C_2

Items	Support
{A, B}	1
{B, C}	1
{A, C}	2

It's support one count

$A \wedge C \rightarrow D$

$A \wedge D \rightarrow C$.

Confidence

$A \rightarrow C =$ Support / occurance of A

L_2 =	Item	Support
	$\{A, C\}$	2

C_1 will contain 1 Item sets
C_2 will contain 2 Item sets
C_3 will contain 3 Item sets.

Association Rule	Support	Confidence	Confidence %
$A \rightarrow C$	2	$2/3 = 0.66$	66 %
$C \rightarrow A$	2	$2/2 = 1$	100 %

Min confidence = 50 %.
50 % se upar hona chahiye.

Final Rule

$A \rightarrow C$
$C \rightarrow A$.

Advantages:
1) Calculate large itemsets
2) Simple to understand & apply

Disadvantages:
1) Expensive method to find support since the calculation has to pass the whole database
2) Sometimes, you need a huge no. of candidate Rules, so it becomes computationally more expensive

FP - Growth Algorithm / Frequent Pattern Growth Algorithm

The 2 Primary Drawbacks of the Apriori Algo are:
1) At each step, candidate sets have to be built.
2) To build the candidate sets, the Algorithm has to repeatedly scan the database.

Tid	Item sets
1	a, a, c, d, g, m, p
2	a, b, c, f, l, m, o
3	b, f, h, o
4	b, k, c, p.
5	a, f, c, l, p, m, n.

min support : 3

[a, b, c, d, f, g, k, l, m, h, o, p]

Item	Support
a	3
b	3
c	4
d	1
f	4
g	1
k	1
l	2
m	3
n	1
o	2
p	3

DETAILED LECTURE NOTES

PAGE NO. .....

Item	Support
f	4
c	4
a	3
b	3
m	3
p	3

↑ pattern

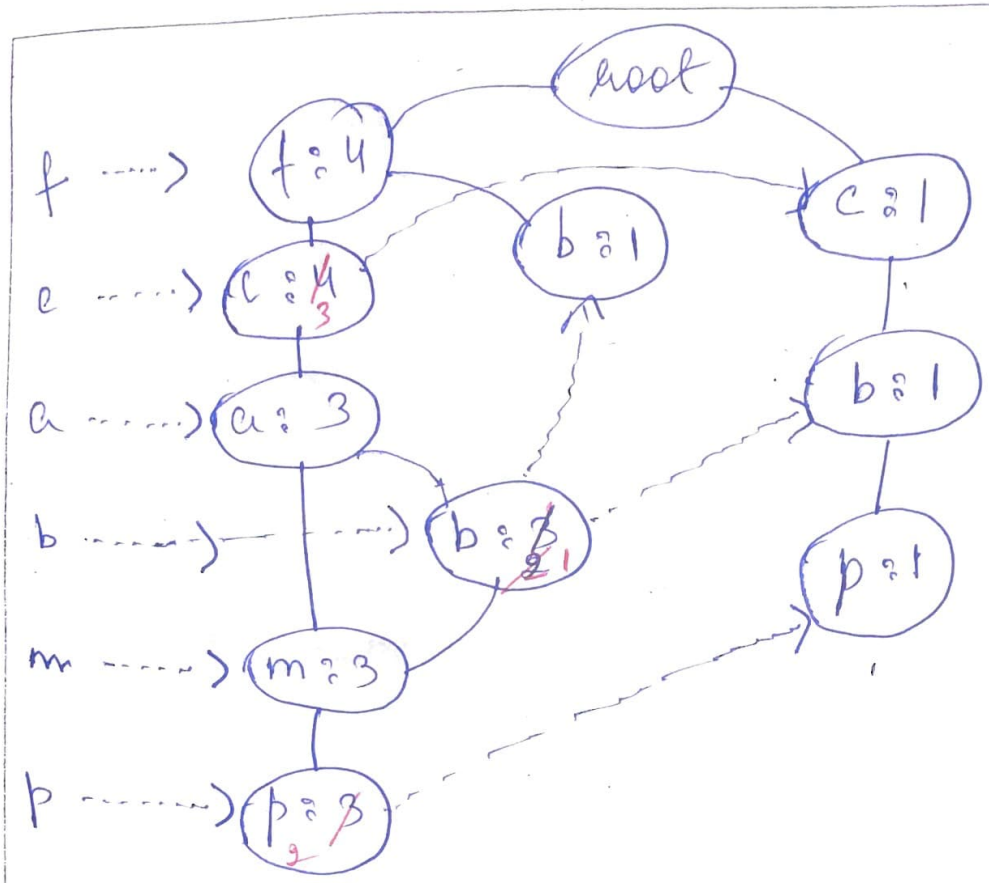
[f, c, a, b, m, p) — Pattern & this pattern would be followed

↓

create a new table

Id	Item subs
1	f, a, c, d, g, m, p
2	a, b, c, f, l, m, o
3	b, f, h, o
4	b, k, c, p
5	a, f, c, l, p, m, n

Ordered Items
t, c, a, m, p, p
f, c, a, b
t, b
c, b, p
f, c, a, m, p.


root
f : 4
c : 1
b : 1
c : 4
3
b : 1
a : 3
b : 3
2 1
p : 1
m : 3
p : 3
2
f
e
a
b
m
p