21/8/19

Association Rule - Correlation between two or more items.
Defines relationship between various items

Market basket analysis - Evaluating market strategies based on the sell and applying association rule mining on the data set.

Support denotes the probability that a transaction contains a item set while confidence denotes the probability that a transaction containing item set 1 also contains item set 2

Basic Concept of Association Rule
Let $I = \{i_1, i_2, i_3 \cdots i_m\}$
D = Task Relevant Data = Transaction having TID
A = Itemset } one or more items.
B = Itemset }
$A \Rightarrow B$ where $A \subset T$, $B \subset T$ & $A \cap B = \phi$

Support = $P(A \cup B)$
Confidence = $P\left(\frac{B}{A}\right)$

Ex: T_1 : A, B, C
T_2 : A, C
T_3 : A, B, D

for $A \Rightarrow B$: Support = 2 (66%) → both A & B present
Confidence = 66% $\left(\frac{2}{3} \times 100\right)$
if A is present then B is present

But when $B \Rightarrow A$

Support = 66%

Confidence = 100%

Classification of Association Rule (AR)

i) Based on the type handled by AR

a) Boolean

0 or 1 value

Ex: age(30) ⇒ buys(Scooter)

b) Quantitative

Ex: age(X, '25..40') ∧ income(X, '2K..40K') ⇒ buys(X, computer)

ii) Dimension of data involved in rule

a) Single dimension

Ex: age(40) ⇒ buys(car)

b) Multi-dimensional

Ex: age(40) ∧ gender(male) ⇒ buys(car)

iii) Level of abstraction

a) Higher level

Ex: occupation(student) ⇒ buys(computer)

b) Lower level

Ex: occupation(student) ⇒ buys(laptop)

iv) Based on various expansion

$T_1 : \{a_1, a_2 \dots a_{50}\}$

$T_2 : \{a_1, a_2 \dots a_{100}\}$

1-itemset	support	2-itemset	
$\{a_1\}$	2	$\{a_1, a_{49}\}$	2
$\{a_2\}$	2	$\{a_1, a_{99}\}$	1
⋮		⋮	
$\{a_{51}\}$	1	$\{a_{51}, a_{99}\}$	1
⋮		⋮	
$\{a_{100}\}$	1	$\{a_{99}, a_{100}\}$	1

(confidence not possible to find)

Total itemset (subsets) $= 2^n - 1 = 2^{100} - 1$

a) Closed pattern (Loss-less)

A pattern (itemset) X is a closed pattern if X is frequent & there exists no super pattern Y with the same support as X.

Here,

		(support)
P1	$\{a_1, a_2 \cdots a_{50}\}$	2
P2	$\{a_1, a_2 \cdots a_{100}\}$	1
	$\{a_1 \cdots a_{49}\}$	1
	$\{a_1 \cdots a_{48}\}$	2

b) Max pattern

A pattern X is a max pattern if X is frequent and there exists no frequent super pattern Y

Here, P1 $= \{a_1, a_2 \cdots a_{100}\} : 1$

$\{a_1 \cdots a_{48}\} : 1$

Thus, lossy representation.

Find AR from large dataset:
Step1: Find frequent ~~dataset~~ Itemset.
Step2: Find a strong association rule from the frequent Itemset

→ If the support of a itemset satisfies the minimum threshold, then it is called a frequent item set.

⇒ If the confidence of a Item set satisfies the minimum confidence threshold then it is called a strong AR

Apriori algorithm
Any subset of a frequent itemset should be frequent. This is the principle of apriori

Ex: Generate AR from given Transaction
Min. support : 50%
Min confidence : 75%.

T_1	Bread, Cheese, Juice, Butter
T_2	Bread, Cheese, Juice
T_3	Bread, Milk, Yogurt
T_4	Bread, Juice, Milk
T_5	Cheese, Juice, Milk

⇒ Candidate creation:

C_1

Item	Support
Bread	4 (80%)
Cheese	3 (60%)

Juice	4	(80%)	given
Butter	1	(20%)	× (Supp cut < 50%)
Milk	3	(60%)	
Yogurt	1	(20%)	×

L1

Item	Support
Bread	4
Cheese	3
Juice	4
Milk	3

C2

Itemset	Support		
{B, C}	2	(40%)	×
{B, J}	3	(60%)	
{B, M}	2	(40)	×
{C, J}	3	(60%)	
{C, M}	1	(20%)	×
{J, M}	2	(40%)	×

L2

Itemset	Support
{B, J}	3
{C, J}	3

We skip creating C3 from L2 because if have itemset having 3 items, the support will be only 1

AR possible from L2

	Confidence
Bread ⇒ Juice	75%
Juice ⇒ Bread	75%
Cheese ⇒ Juice	3/3 = 100%
Juice ⇒ Cheese	3/4 = 75%

∴ All are strong AR.

Support = Occurance of an item.

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

22/8/19 Confidence of Bread → Juice

Support = 3 Support = 4

∴ Confidence = $\frac{3}{4}$ = 75%

Q. Transaction Items

T_1	I_1, I_2, I_5	Min. Support = 22%
T_2	I_2, I_4	Min. Confidence = 70%
T_3	I_2, I_3	
T_4	I_1, I_2, I_4	Min. Support = 22% of 9 = 2
T_5	I_1, I_3	
T_6	I_2, I_3	
T_7	I_1, I_3	
T_8	I_1, I_2, I_3, I_5	
T_9	I_1, I_2, I_3	

⇒ C1

Item	Support
I_1	6 (66%)
I_2	7 (77%)
I_3	6 (66%)
I_4	2 (22%)
I_5	2 (22%)

L1

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

C2.

Items	Support	
I_1, I_2	4	
I_1, I_3	4	
I_1, I_4	1	×
I_1, I_5	2	
I_2, I_3	4	
I_2, I_4	2	
I_2, I_5	2	
I_3, I_4	0	×
I_3, I_5	1	×
I_4, I_5	0	×

C2.

Item set	Support
$\{I_1, I_2\}$	4
$\{I_1, I_3\}$	4
$\{I_1, I_5\}$	2
$\{I_2, I_3\}$	4
$\{I_3, I_4\}$	2
$\{I_2, I_5\}$	2

C3

Item set	Support	
$\{I_1, I_2, I_3\}$	2	
$\{I_1, I_2, I_5\}$	2	
$\{I_1, I_2, I_4\}$	1	×
$\{I_1, I_3, I_5\}$	1	×
don't write sets with 4 items: $\{I_1, I_3, I_3, I_4\}$		
$\{I_2, I_2, I_3, I_5\}$		
I_1, I_2, I_4, I_5		
$\{I_2, I_3, I_4\}$	0	×
$\{I_2, I_3, I_5\}$	1	×
$\{I_2, I_4, I_5\}$	0	×

L3

Itemset	Support
$\{I_1, I_2, I_3\}$	2
$\{I_1, I_2, I_5\}$	2

Rules:

From L2

$I_1 \Rightarrow I_2$	$4/6 = 66\%$	$I_2 \Rightarrow I_1$	$4/7 = 57\%$
$I_1 \Rightarrow I_3$	$4/6 = 66\%$	$I_3 \Rightarrow I_1$	$4/6 = 66\%$
$I_1 \Rightarrow I_5$	$2/6 = 33\%$	$I_5 \Rightarrow I_1$	$2/2 = 100\%$ ✓
~~$I_2, I_3 \Rightarrow I_1$~~	~~$2/4 = 50\%$~~	~~$I_1, I_2 \Rightarrow I_3$~~	~~$2/4 = 50\%$~~
~~$I_2, I_3 \Rightarrow I_1$~~	~~$2/4 = 50\%$~~	~~$I_1, I_3 \Rightarrow I_5$~~	~~$2/4 = 50\%$~~
$I_1, I_5 \Rightarrow I_2$	$2/2 = 100\%$		

$I_2 \Rightarrow I_3$	$4/7 = 57\%$	$I_3 \Rightarrow I_2$	$4/6 = 66\%$
$I_2 \Rightarrow I_4$	$2/7 = 28\%$	$I_4 \Rightarrow I_2$	$2/2 = 100\%$ ✓
$I_2 \Rightarrow I_5$	$2/7 = 28\%$	$I_5 \Rightarrow I_2$	$2/2 = 100\%$ ✓

From L3:

$\{I_1, I_2\} \Rightarrow I_3$	$2/4 = 50\%$	$I_3 \Rightarrow \{I_1, I_2\}$	$2/6 = 33\%$
$\{I_1, I_3\} \Rightarrow I_2$	$2/4 = 50\%$	$I_2 \Rightarrow \{I_1, I_3\}$	$2/7 = 28\%$
$\{I_2, I_3\} \Rightarrow I_1$	$2/4 = 50\%$	$I_1 \Rightarrow \{I_2, I_3\}$	$2/6 = 33\%$
$\{I_1, I_2\} \Rightarrow I_5$	$2/4 = 50\%$	$I_5 \Rightarrow \{I_1, I_2\}$	$2/2 = 100\%$
$\{I_1, I_5\} \Rightarrow I_2$	$2/2 = 100\%$ ✓	$I_2 \Rightarrow \{I_1, I_5\}$	$2/7 = 28\%$
$\{I_2, I_5\} \Rightarrow I_1$	$2/2 = 100\%$ ✓	$I_1 \Rightarrow \{I_2, I_5\}$	$2/6 = 33\%$

Strong AR:

$I_5 \Rightarrow I_1$

$I_4 \Rightarrow I_2$

$I_5 \Rightarrow I_2$

$\{I_1, I_5\} \Rightarrow I_2$

$\{I_2, I_5\} \Rightarrow I_1$

$I_5 \Rightarrow \{I_1, I_2\}$

Improving the efficiency of Apriori 5 methods.

i) Shrink the no. of candidates
   a) Hash-base technique
   b) Sampling

ii) Reduce the passes of transaction database scan
   a) Transaction reduction b) Partitioning
   c) Dynamic Itemset counting

→ Hash-base technique

$h(x,y) = [\text{order of } x] * 10 + [\text{order of } y] \bmod 7$

For T_1, 3 item sets with 2 items are possible

$h(I_1, I_2) = (1 \times 10) + 2) \% 7 = 5$ → place in bucket 5
(order)

$h(I_1, I_5) = (1 \times 10) + 5) \% 7 = 1$

$h(I_2, I_5) = ((2 \times 10) + 5) \% 7 = 4$

Bucket address	0	1	2	3	4	5	6
Bucket count	2	2	3	2	2	4	4
Bucket content	I_1, I_4	I_1, I_5	I_2, I_3	I_2, I_4	I_2, I_5	I_1, I_2	I_1, I_3
	I_3, I_5	I_1, I_5	I_2, I_3	I_2, I_4	I_1, I_5	I_1, I_2	I_1, I_3
			I_2, I_3			I_1, I_2	I_1, I_3
						I_1, I_2	I_1, I_3

Now, if min. support > bucket count, the candidates can be discarded.

26/8/19

Transaction Reduction
Discard the trans. which don't have the frequent item sets.

Partitioning
Divide the entire data set into multiple parts and find freq. item sets from each.

Dynamic item set counting
Include n-item sets in the first scan and calculate its support. In subsequent scans, we simply add the pre-calculated support.

Scan part (S1-S2) and list all n itemsets with their support.
On next scan (S2-S3), add the support for corresponding item set.

⇒ when we have 10^4 1-itemset, we get 10^7 2-itemset.

finding freq. itemsets without candidate generation.
Reasons: i) It may need to generate huge no. of candidates.
ii) It may repeatedly require to scan the

entire dataset.

Frequent Pattern Growth (FP Tree)
We can generate 2-itemset using this method useful when $K > 1$

T_1	A_1, A_2, A_5
T_2	A_2, A_4
T_3	A_2, A_3
T_4	A_1, A_2, A_4
T_5	A_1, A_3
T_6	A_2, A_3
T_7	A_1, A_3
T_8	A_1, A_2, A_3, A_5
T_9	A_1, A_2, A_3

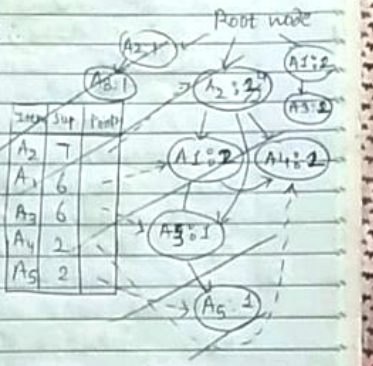
$\Rightarrow$

A_1	6
A_2	7
A_3	6
A_4	2
A_5	2

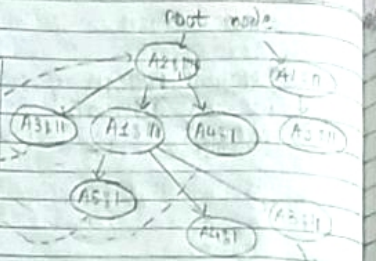
Order
$A_2\ A_1\ A_3\ A_4\ A_5$

Reorder all trans.

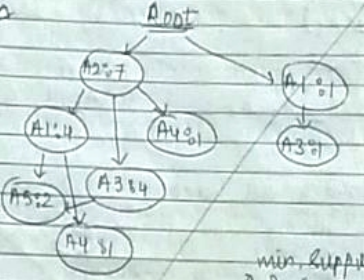
T_1	$A_2\ A_1\ A_5$
T_2	$A_2\ A_4$
T_3	$A_2\ A_3$
T_4	$A_2\ A_1\ A_4$
T_5	$A_1\ A_3$
T_6	$A_2\ A_3$
T_7	$A_1\ A_3$
T_8	$A_2\ A_1\ A_3\ A_5$
T_9	$A_2\ A_1\ A_3$



Item	Support	Pointer
A_2	7	→
A_1	6	→
A_3	6	→
A_4	2	→
A_5	2	→



Ex Given



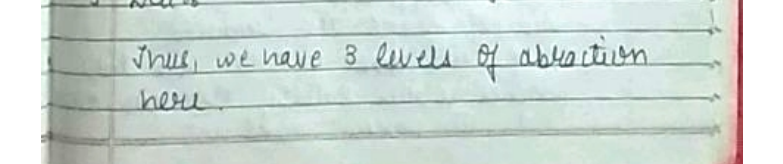
min. support (given) = 2
& $\{A_2\}$ has support 3

Item	Conditional Pattern Base	Conditional Tree	Frequent Item
A_5	$\{A_2, A_1\}$ $\{A_2, A_3\}$ $\{A_2, A_1, A_3\}$	$\{A_2\}$ $\{A_2\}$ $\{A_1, A_3\}$ $\{A_3\}$	
A_4	$\{A_2, A_1\}$ $\{A_2\}$		
A_3	$\{A_2, A_1\}$ $\{A_1\}$ $\{A_2\}$		
A_1	$\{A_2\}$		

From

	Conditional Pattern Base	Conditional FP Tree	Frequent
A_5	$\{A_2, A_1\} : 1$ $\{A_2, A_1, A_3\} : 2$ Paths leading upto A5 & the count of A5 node	$\{A_1\} : 2$ $\{A_2\} : 2$ $\{A_1, A_2\} : 2$ Freq item set (min support=2)	$\{A_1, A_5\} : 2$ $\{A_2, A_5\} : 2$ $\{A_1, A_2, A_5\} : 2$ A2 & A5 will get obtained in CFPT
A_4	$\{A_2\} : 1$ $\{A_2, A_1\} : 1$	$\{A_2\} : 2$	$\{A_2, A_5\} : 2$
A_3	$\{A_2\} : 2$ $\{A_2, A_1\} : 2$ $\{A_1\} = 2$	$\{A_2\} : 4$ $\{A_1\} = 4$ $\{A_1, A_2\} : 2$	$\{A_2, A_5\} = 4$ $\{A_1, A_5\} = 4$ $\{A_1, A_2, A_5\} = 2$
A_1	$\{A_2\} : 4$	$\{A_2\} : 4$	$\{A_2, A_5\} : 4$
A_2	-		

Mining multi-level association rules from transactional databases



Thus, we have 3 levels of abstraction here.

buys(X, "computer") ⇒ buys(X, "printer")
support = 50 (Higher level AR)

buys(X "laptop") ⇒ buys(X, "color")
support = 30 (Lower level AR)

buys(X, "Dell") ⇒ buys(X, "Sony")
support = 17

Approaches to mine multi-level AR

i) Using uniform min. support for all the levels
- Need to declare only one global min. support
- The items at lower level can be missed

ii) Using the reduced min. support at lower level
- Decreasing the min support at every level

iii) Level by level independence

b) Level cross filtering by single item
- The child nodes are scanned only if the parent node satisfies the min. support
- No. of scans are reduced
- We can miss out on the child nodes which satisfy the min. support but, its parent does not

(b) Level cross filtering by k item

(c) Controlled level cross filtering
The concept of Passage min support is used

min support ≤ Passage min support ≤ min support of parent of child

Then for every node in the level, we check for the passage min support

⇒ buys(X, computer) [s=20, c=70] ⇒ buys(X, Printer)
buys(X, laptop) [s=7, c=72] ⇒ buys(X, Printer)
are redundant AR
Thus, we can discard the lower level rule if it gives the expected output

Expected values:
From first rule laptop accounts for almost 1/3rd of total sale. Therefore an expected support for laptop would be $\frac{20}{3}$ (6.66) approx.

and confidence near to 70 which is satisfied. Thus, second rule can be discarded.

Multidimensional rules
finding Multidimensional AR from RDB & DW
age(X, 25-30) ^ buys(X, "computer") ⇒ buys(X, printer)

is a hybrid multidimensional AR as 'buys' predicate dimension is repeated

age(X, 25-30) ∧ income(X, 25K-55K) ⇒ buys(X, "laptop")

(No predicate dimension repeated)

Methods for mining MDAR

i) Quantitative attributes are discretized using concept hierarchy.

ii) Quantitative attributes are discretized into fields based on the distribution of data

iii) Quantitative attributes are discretized to capture the semantic meaning of such data

Purchase of IBM PC : 50

min support = 14

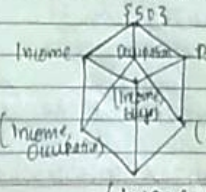
age : 12 × (based on min. support)

Income : 14

Gender : 6 ×

Occupation : 18

∴ Total 3 predicates : Income + Occupation + buy



Mining Quantitative AR (Binning method)

1) Binning
2) finding frequent predicate set
3) clustering the association rule