DATA MINING ASSIGNMENT - 5

Parampreet Singh

1(a) Let s be a frequent itemset,

min_sup be the minimum support

D be the task relevant data, a set of database transactions

|D| be the no. of transactions in D

Since s is a frequent itemset support_count(s) = min_sup × |D|

So, s' be any non empty set of s

then any transaction containing itemset s will also contain itemset s'. Therefore support_count(s')

$$support\_count(s') \geq support\_count(s) = min\_sup \times |D|$$

Thus, s' is also a frequent itemset

(b) Let D be the task relevant data, a set of database transactions, |D| be the no. of transactions in D.

So, $support(S) = \frac{support\_count(S)}{|D|}$ S be the itemset

Let s' be the non-empty subset of s

$$support(s') = \frac{support\_count(s')}{|D|}$$

we know that support(s') ≥ support(s)

therefore the support of non-empty subset s' of itemset s will be greater than the support of s.

c) Given frequent itemset l and subset s of l,

$$\text{confidence}(s \to (l-s)) = \frac{\text{Support}(l)}{\text{support}(s)}$$

let s' be the any non-empty subset s then,

$$\text{confidence}(s' \to (l-s')) = \frac{\text{Support}(l)}{\text{Support}(s')}$$

we already know that support(s') ≥ Support(s)

therefore, confidence $(s' \to (l-s')) \leq$ confidence $(s \to (l-s))$
that is confidence of the rule $s' \to l-s'$ cannot be more than the confidence of the rule $s \to (l-s)$

2. Given database has 5 transactions,
let min_sup = 60%
min conf = 80%

TID	Items_bought
T100	{M, O, N, K, E, Y}
T200	{D, O, N, K, E, Y}
T300	{M, A, K, E}
T400	{M, U, C, K, Y}
T500	{C, O, O, K, I, E}

(a) Apriori Algorithm :-

So, no. of transactions = 5

min support count = $\frac{60}{100} \times$ (total no. of transactions)

$= \frac{60}{100} \times 5 = 3$

min confidence count = $\frac{80}{100} \times$ (total no. of transactions)

$= \frac{80}{100} \times 5 = 4$

C_k = candidate itemset
L_k = frequent itemset

Scanned with CamScanner

we will generate the candidates itemset and also the frequent itemset by scanning the DB scans.

→ Need to remove the repeated item in same transaction based on association rule mining

C1

{M} → 1+1+1 = 3
{O} → 1+1+1 = 3
{N} → 1+1 = 2
{K} → 1+1+1+1+1 = 5
{E} → 1+1+1+1 = 4
{Y} → 1+1+1 = 3
{D} → 1 = 1
{A} → 1 = 1
{U} → 1 = 1
{C} → 1+1 = 2
{I} → 1 = 1

L1

{M} → 3
{O} → 3
{K} → 5
{E} → 4
{Y} → 3

C2

{M, O} → 1 = 1
{M, K} → 1+1+1 = 3
{M, E} → 1+1 = 2
{M, Y} → 1+1 = 2
{O, E} → 1+1+1 = 3

L2

{M, K} - 3
{O, E} - 3
{O, K} - 3
{K, E} - 4
{K, Y} - 3

{O, K} → 3
{O, Y} → 2
{K, E} → 4
{K, Y} → 3
{E, Y} → 2

C_3
{M, K, O} ⇒ 1 = 1
{M, K, E} → 1+1 = 2
{M, K, Y} → 1+1 = 2
{K, O, E} → 1+1+1 = 3
{K, O, Y} → 1+1 = 2
{K, E, Y} → 1+1 = 2
{O, E, Y} → 1+1 = 2

$\underline{L_3}$
{K, O, E} - 3

So the total frequent itemsets are
{M, O, K, E, Y, MK, OE, OK, KE, KY, KOE}

FP growth Algorithm:-
minimum support = $\frac{60}{100} \times 5 = 3$

Item	sup
M	~~∅~~ 3
O	3
N	2
K	5

ordered frequent itemset
K = 5
E = 4
M = 3
O = 3
Y = 3
} ≥ min-sup

E - 4
Y - 3
D - 1
A - 1
U - 1
C - 2
I - 1

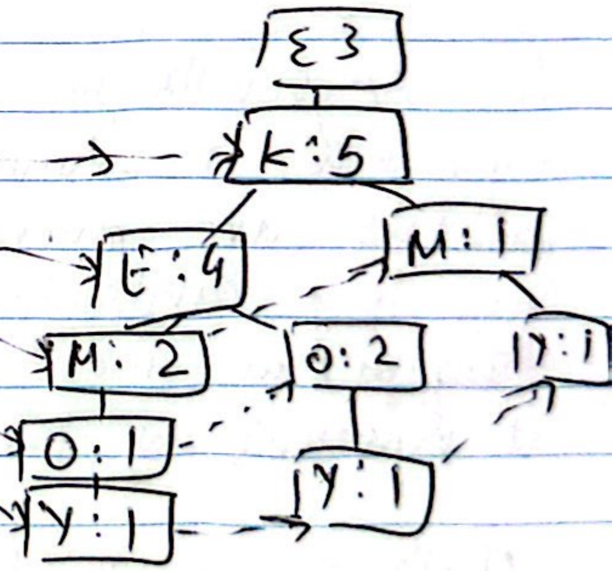
f list : K - E - M - O - Y

TID	Items_bought	ordered_frequent_item
T100	{M, O, N, K, E, Y}	{K, E, M, O, Y}
T200	{D, O, N, K, E, Y}	{K, E, O, Y}
T300	{M, A, K, E}	{K, E, M}
T400	{M, U, C, K, Y}	{K, M, Y}
T500	{C, O, O, K, I, E}	{K, E, O}

FP growth tree

Header table

Item	frequency	head
K	5	
E	4	
M	3	
O	3	
Y	3	



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

Frequent patterns generated using the F-P growth tree

fp growth tree
{{K, Y : 3}, {O, K : 3}, {O, E : 3}, {O, K, E : 3} {M, K : 3}, {E, K : 4}, {E : 4}, {K : 4}, {M : 3}, {Y : 3}, {O : 3}}

Apriori algorithm generates candidate itemsets may be large in no. if the itemset in the database is huge, It needs multiple scans of the database and this makes very slow...

In FP growth the problems in the Apriori algorithm is solved. In this algorithm frequent patterns are generated without the need of generating candidate itemsets. FP growth algorithm need only two db scans and it represents the database in the form of fp tree.

That's the reason fp growth algorithm is more efficient than a Apriori Algorithm

b) Meta Rule : $\forall x \in$ transactions,
buys (X, $item_1$) $\wedge$ buys(X, $item_2$) => buys (X, $item_3$)

X -> variable representing customers,
$item_i$ -> denotes variables representing items (Eg:- "A", "B", --- etc)

Association rules based on the above meta rule
$\forall X \in$ transaction, buys(X, E) $\wedge$ buys(X, O) $\Rightarrow$ buys(X,
$\forall X \in$ transaction, buys (X, K) $\wedge$ buys(X, O) $\Rightarrow$ buys(X, E

3. (a) At the granularity of item_category (Eg: milk give the rules of the template)

The largest frequent itemset for largest values of k=3 is {milk, Bread, cheese}

① Buys(X, milk) $\wedge$ buys(X, Bread) $\Rightarrow$ buys(X, cheese)

$$\text{Support} = \frac{\sigma(\text{Milk, Bread, cheese})}{\text{Total transactions}} = \frac{3}{4} \times 100 = 75\%$$

$$\text{Confidence} = \frac{\sigma(\text{Milk, Bread, Cheese})}{\sigma(\text{Cheese})} = \frac{3}{3} \times 100 = 100\%$$

[75%, 100%]

② buys(X, milk) ∧ buys(X, cheese) ⇒ buys (X, Bread)

$$\text{Support} = \frac{\sigma(\text{Milk, cheese, bread})}{\text{Total transactions (T)}} = \frac{3}{4} \times 100 = 75\%$$

[75%, 100%]

③ Buys (X, Bread) ∧ buys (X, cheese) ⇒ buys (X, Milk)

$$\text{Support} = \frac{\sigma(\text{Bread, cheese, Milk})}{\text{Total transactions (T)}} = \frac{3}{4} \times 100 = 75\%$$

$$\text{confidence} = \frac{\sigma(\text{Bread, cheese, Milk})}{\sigma(\text{Bread, cheese})} = \frac{3}{3} \times 100 = 100\%$$

[75%, 100%]

b) The largest frequent itemset for k=3 is
{ {wonder_bread, sunset_milk, Dairyland_cheese},
{Dairyland_Milk, Tasty-pie, wonder_Bread} }

4. For the wine association rules the algorithm is as follows:-

→ firstly we need to find the local frequent itemsets in each store and here suppose, |F| be the union of all local frequent itemsets in the four stores, so in each store we need to find the local absolute support for each itemset in |F|

And then we must determine the global (absolute) support for each itemset in 1FI, this can be done by summing up for each itemset, the local support of that itemset in the four stores. Doing this for each itemset in 1FI will give us their global supports. Itemset whose global supports pass the support threshold are good frequent itemsets.

Then lastly need to also derive strong association rules from the global frequent itemsets.
Thus these are the rules for the mine global association rules in an algorithm.

5. Given

	hotdogs	hotdogs(NOT)	Σrow
Hamburgers	2000	500	2500
Hamburgers NOT	1000	1500	2500
Σcol	3000	2000	5000

(a) Given rule:

hotdogs ⇒ hamburgers

min_sup = 25%

min_confidence = 50%

$$\text{Support} = \frac{2000}{5000} = 40\%$$

$$\text{confidence} = \frac{2000}{3000} = 66.7\%$$

Support 40%, confidence 66.7% is greater than the given minimum support and minimum confidence.

∴ The given rule association is strong

b) correlation(hotdog, hamburger)

$$= \frac{P(\{hotdog, hamburgers\})}{P\{hotdog\} P\{hamburger\}}$$

$$cor(A, B) = \frac{P(A \cap B)}{P(A) . P(B)} = \frac{(2000/5000)}{\left(\frac{3000}{5000}\right)\left(\frac{2500}{5000}\right)}$$

$$= \frac{0.4}{0.6 \times 0.5} = 1.33 \; (>1)$$

So the correlation value is greater than 1

Therefore the purchase of hotdog is not independent of the purchase of hamburgers and there is a positive correlation.