28/11/21

DMW
ASSIGNMENT - 4
ASSOCIATION RULES

G00041900571
JUNAID GIRKAR

Q1 Construct FP tree with support = 2

TID	ITEMS
1	{b, a}
2	{b, d, c}
3	{a, d, e}
4	{d, e, a, c}
5	{c, b, a}
6	{a, c, b, d}
7	{a, f}
8	{b, a, c}
9	{b, d, a}
10	{c, e, b}

ANS Individual supports :-

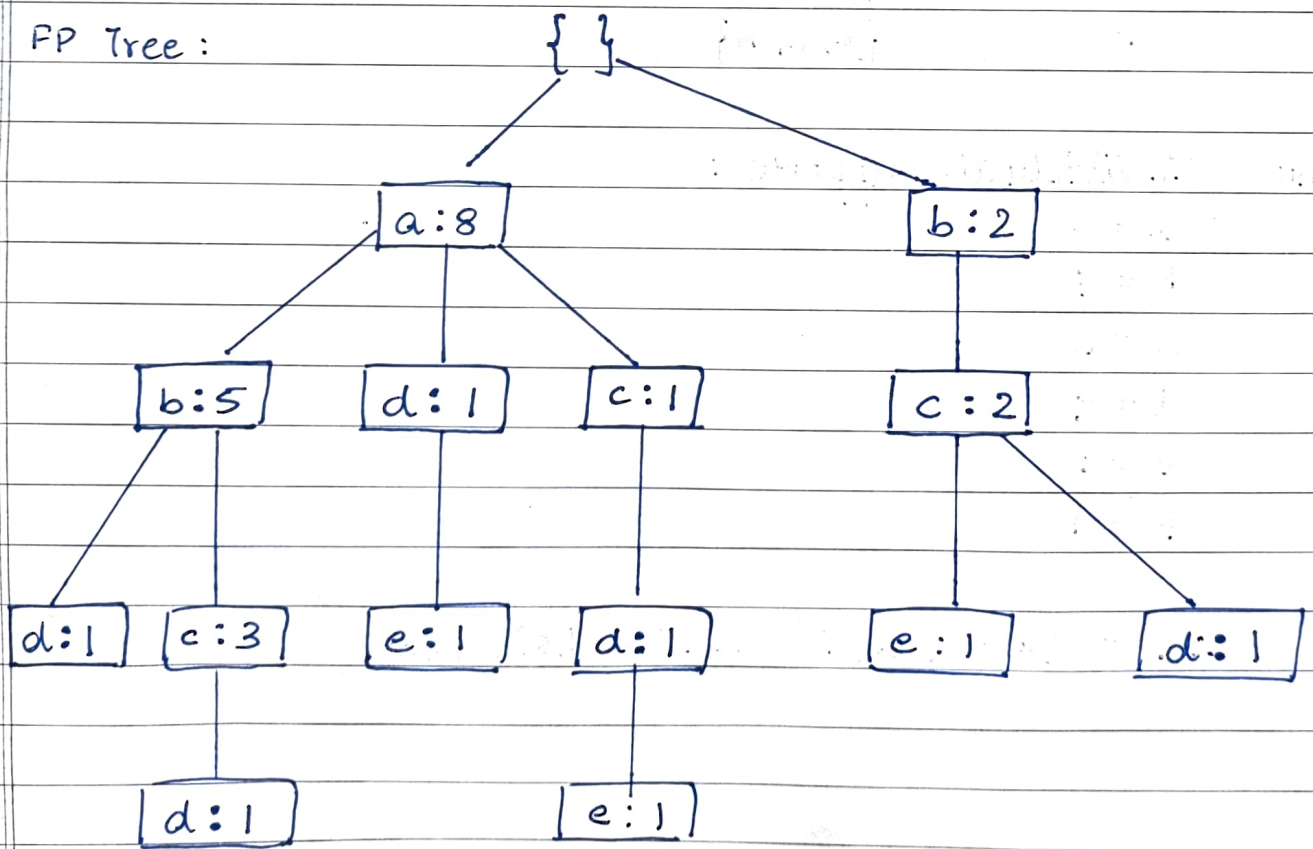
a = 8
b = 7
c = 6
d = 9
e = 3
f = 1

∴ L = (a, 8), (b, 7), (c, 6), (d, 5), (e; 3).

Re-Arranged transactions :-

TID	ITEMS
1	{a, b}
2	{b, c, d}
3	{a, d, e}
4	{a, c, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

FP Tree :



2

ITEM	CONDITIONAL PATTERN BASE	CONDITIONAL FP TREE	FP GENERATED
e	(a → c → d = 1), (a → d = 1), (b → c = 1)	{a : 2, d : 2, c : 2}	{a, e}, {d, e}, {a, d, e}, {c, e}
d	(a → c = 1), (a → b → c = 1), (a → b = 1), (a → = 1), (b → c = 1)	{a:4, b:2}, {a:4, c:2} {b:2, c:2}	{a, d}, {b, d}, {c, d}, {a, b, d} {a, c, d}, {b, c, d}
c	(a → = 1), (a → b = 3), (b → 2)	{a:4, b:5}	{a, c}, {b, c} {a, b, c}
b	(a → = 5)	{a : 5}	{a, b}

Q2 A database has five transactions. let min_sup = 60% and min_conf = 80%

TID	Items
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

support = 0.6 × 5
= 3

(i) Using FP tree

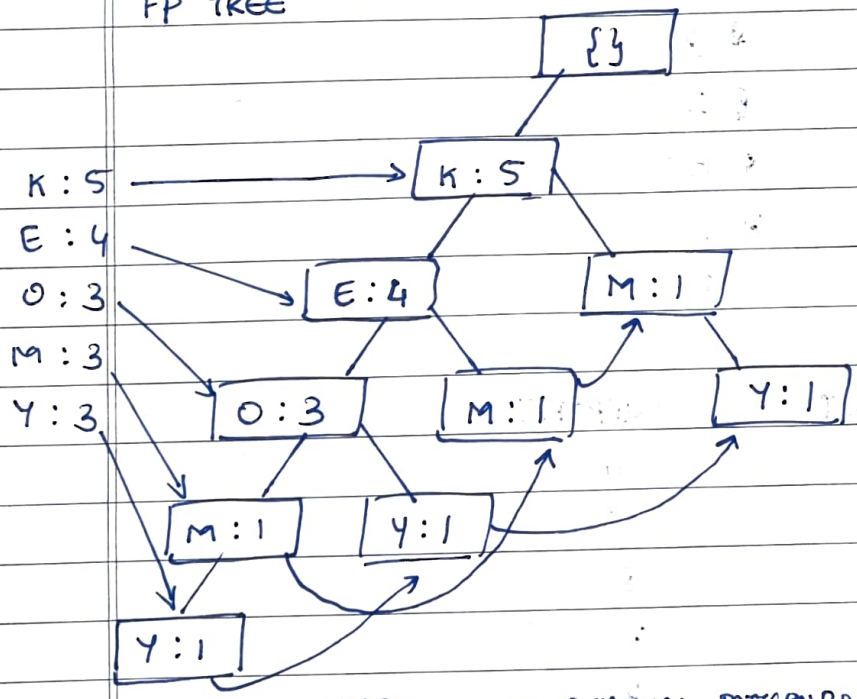
Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
U	1
Y	3

L = { {K:5}, {E:4}, {O:3}, {M:3}, {Y:3}

Arranging transactions in order of L

TID	Items
t100	{K, E, O, M, Y}
t200	{K, E, O, Y}
t300	{K, E, M}
t400	{K, M, Y}
t500	{K, E, O}

FP TREE



ITEM	CONDITIONAL PATTERN BASE	CONDITIONAL FP TREE	FP GENERATED
Y	(K → E → O → M = 1), (K → E → O = 1), (K → M = 1)	{K : 3}	{(K, Y)}
M	(K → E → O = 1), (K → E = 1), (K → = 1)	{K : 3}	{K, M}
O	{K → E = 3}	{K : 3, E : 3}	{K, O}, {E, O}, {K, E, O}
E	{K → = 4}	{K : 4}	{K, E}

(2) Using Apriori

C1 :

ITEMSET	SUPPORT
A	1
B	2
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
V	1
Y	3

L1 :

ITEMSET	SUPPORT
E	4
K	5
M	3
O	3
Y	3

4

C2 :

ITEMSET	SUPPORT
E, K	4
E, M	2
E, O	3
E, Y	2
K, M	3
K, O	3
K, Y	3
M, O	1
M, Y	2
O, Y	2

L2 :

ITEMSET	SUPPORT
E, K	4
E, O	3
K, M	3
K, O	3
K, Y	3

C3 :

ITEMSET	SUPPORT
E, K, O	3
E, K, M	2
E, K, Y	2
K, M, O	1
K, M, Y	2
K, O, Y	2

L3 :

ITEMSET	SUPPORT
E, K, O	3

∵ NO more combinations can be formed

∴ L = {E, K, O}

Subsets : {K}, {E}, {O}, {O, K}, {O, E}, {K, E}

Association rules using Apriori :

1 {K} → {O, E} = 3/5 (confidence) = 60%

2 {O, E} → {K} = 3/3 (confidence) = 100%

3 $\{O\} \rightarrow \{K, E\}$ = 3/3 (confidence) = 100%

4 $\{K, E\} \rightarrow \{O\}$ = 3/4 (confidence) = 75%

5 $\{E\} \rightarrow \{O, K\}$ = 3/4 (confidence) = 75%

6 $\{O, K\} \rightarrow \{E\}$ = 3/3 (confidence) = 100%

∵ min_confidence = 80%

∴ Strong association rules are :

$\{O, E\} \rightarrow \{K\}$,

$\{O\} \rightarrow \{K, E\}$,

$\{O, K\} \rightarrow \{E\}$.

- Association rules using FP tree :

L1 : {K, Y}, L3 : {K, O} L5 : {K, E, O}

L2 : {K, M}, L4 : {E, O} L6 : {K, E}

(i) L7 : {K, Y}, subsets : {K}, {Y}

RULES :- CONFIDENCE

$\{K\} \rightarrow \{Y\}$ = 3/5 = 60% ✗

$\{Y\} \rightarrow \{K\}$ = 3/3 = 100% ✓

∴ Strong association rule : $\{Y\} \rightarrow \{K\}$

(ii) L2 : {K, M}, subsets : {K, M}

RULES: CONFIDENCE

$\{K\} \rightarrow \{M\}$ = 3/5 = 60% ✗

$\{M\} \rightarrow \{K\}$ = 3/3 = 100% ✓

∴ Strong association rule : $\{M\} \rightarrow \{K\}$

5

(iii) L3 : {K, O} , subsets : {K}, {O}

RULES	CONFIDENCE	
{K} → {O}	= 3/5 = 60%	✗
{O} → {K}	= 3/3 = 100%	✓

∴ strong association rule : {O} → {K}

(iv) L4 : {E, O} , subsets : {E}, {O}

RULES	CONFIDENCE
{E} → {O}	= 3/4 = 75%
{O} → {E}	= 3/3 = 100%

∴ strong association rule : {O} → {E}

(v) L5 : {K, E, O} , subsets : {K}, {E}, {O}, {K, E}, {O, K}, {O, E}.

RULES	CONFIDENCE
{K} → {O, E}	= 3/5 = 60%
{O, E} → {K}	= 3/3 = 100%
{O} → {K, E}	= 3/3 = 100%
{K, E} → {O}	= 3/4 = 75%
{E} → {O, K}	= 3/4 = 75%
{O, K} → {E}	= 3/3 = 100%

strong association rules :

{O, E} → {K}

{O} → {K, E}

{O, K} → {E}

(vi) LG : {K, E}, subsets : {K}, {E}

RULES	CONFIDENCE
{K} → {E}	= 4/5 = 80%
{E} → {K}	= 4/4 = 100%

Strong association rules are : {K} → {E}, {E} → {K}

Strong Association rules using FP tree are :-

1. {Y} → {K}
2. {M} → {K}
3. {O} → {K}
4. {O} → {E}
5. {O, E} → {K}
6. {O} → {K, E}
7. {O, K} → {E}
8. {K} → {E}
9. {E} → {K}

b Compare the efficiency of the two mining processes.

ANS The resulting frequent patterns are similar for both FP tree and the Apriori algorithm.

However, in terms of overall algorithm efficiency for finding frequent patterns among dataset FP tree algorithm is better than the apriori algorithm as it doesn't require candidate generation thus saving time and space. Also, FP tree algorithm is able to mine

in the conditional pattern basis which may substantially reduce the size of the dataset to be searched. In certain conditions where the dataset to be mined is small, it is seen that the efficiency of the apriori algorithm increases.

Q3 Explain the various techniques to improve efficiency of Apriori algorithm in brief.

ANS Some techniques to improve the apriori algorithm are :-

1 HASH BASED METHOD : - It can be used to reduce the size of the candidate K-itemsets. C_K for $K > 1$
- When scanning each transaction in database to generate L_1 from C_1, we can generate all the two-itemsets for each transaction and hash them into different buckets of a hash-table structure and increase corresponding bucket counts
- A 2-itemset with corresponding bucket count below support threshold cannot be frequent and thus should be removed from the candidate set.
- Such a hash based technique may substantially reduce the number of K-itemsets (candidate) examined.

2 TRANSACTION REDUCTION : - Reducing number of transactions scanned in future epochs
- A transaction that doesn't contain any frequent K-itemsets cannot contain any frequent (K+1) itemsets.
- Thus such transactions can be removed or marked.

3 PARTITIONING: - This technique requires just 2 database scans to mine frequent itemsets.
- It is a two phase algorithm.
- In phase 1, the algorithm logically divides the database into 'n' non-overlapping positions. The partitions formed are considered one at a time as one database and large itemsets in each pattern are generated.
- At the end of this phase, all large itemsets are merged to form superset. This superset is a candidate set with respect to the original database.
- In phase 2, the original support is considered for the itemsets and the global (large) frequent itemsets are identified.

4 SAMPLING: - In this, we pick random samples S of DB and search for frequent itemsets in S instead of DB.
- Here we trade of between accuracy and efficiency as large datasets within sample are found while the ones outside the sample may not be found.
- As samples are small, S can fit in main memory so that only scan of S is required overall.
- To avoid missing global frequent itemsets, we reduce support compound to original support.
- DB except S is used to find actual frequencies of each itemsets in S. If S contains all frequent itemsets then only one scan is required otherwise one more scan is performed to find the missing frequent itemset.

(v) DYNAMIC ITEMSET COUNTING : - DB is partitioned into blocks marked by start points.

- New candidate can be added at any start point unlike Apriori where candidates can be added immediately only before the DB scan.
- It uses count-so-far as lower boundary of actual count. If count-so-far passes support then the itemset is added to frequent itemset collection and used to generate longer candidates.
- This leads to fewer DB scans compared to Apriori.

Q4 What are the advantages of frequent pattern mining over apriori

ANS
- Execution time of FP tree is lesser than apriori due to the absence of candidates
- FP tree algorithm scans the database twice whereas apriori does multiple scans
- FP tree requires small amount of memory space due to compact structure and no candidate generation
- FP tree uses frequent pattern tree, conditional and conditional pattern base which satisfy minimum support
- It is efficient and scalable for mining both long and short frequent patterns.

Q5 Explain the following with examples:

1 Itemsets :

- A collection of zero or more items present in a dataset form an itemset.
- An itemset with zero items is known as a null set
- An itemset with K items is called a K-itemset
- E.g: {A, B, C} → 2-itemset.

2 SUPPORT :

- Support is the frequency of an itemset or how frequently an itemset appears in a dataset
- It is defined as fraction of transaction T that contains the itemset
- support = $\frac{\text{Frequency (Itemset)}}{T}$
- E.g: Consider a database :

1	{A, B, C}
2	{A, C}
3	{A, D, C}
4	{A, E}
5	{E, D}

$\therefore$ Support (A) = $\frac{\text{Freq (A)}}{T} = \frac{4}{5} = 0.8$ //

3 CONFIDENCE :

- It indicates how often the rule has found to be true or how often the itemsets x and y have been bought together when the occurence of x is already given.
- Confidence (x) = $\frac{\text{Freq } (X \cup Y)}{\text{Freq } (X)}$

- Suppose for the above used example, the frequent pattern generated is {A, C, D}

- Rule : {C} → {A, D}
- Confidence = $\frac{\text{Frequency}(\{A, C, D\})}{\text{Frequency}(C)}$

$= \frac{1}{3}$

$= 0.33$ //

4 FREQUENT ITEMSETS :

- A frequent itemset is one that occurs frequently in the dataset.
- An itemset is considered to be frequent if its support is greater than equal to the threshold support.
- Consider the previously used dataset, let the candidate set in second iteration be :-

C2	ITEMSET	SUPPORT
	{A, B}	1
	{A, C}	3
	{A, D}	1
	{A, E}	1
	{B, C}	1
	{B, D}	0
	{B, E}	0
	{C, D}	1
	{C, E}	0
	{D, E}	1

Let the minimum support be 1

∴ The frequent itemsets (L2) for C2 will be :-

L2

ITEMSET	SUPPORT
{A, B}	1
{A, C}	3
{A, D}	1
{A, E}	1
{B, C}	1
{C, D}	1
{D, E}	1

These are the frequent itemsets for iteration 1. These are used to generate the next frequent itemsets.

5 ASSOCIATION RULES :

- Association rules are an implication expression of the form $X \rightarrow Y$ where X and Y are any two itemsets from the subset of the final large itemset.
- It implies that whenever X is bought, Y is also bought together.
- An association has two parts : An antecedent (if) and a consequent (then).
- Strong association rules are those with confidence above threshold.
- Consider the above dataset example. sup = 1, conf = 0.75, L = {A, C, D}

ASSOCIATION RULES		CONFIDENCE		
{A} → {C, D}	=	1/4 = 0.25	✗	
{C, D} → {A}	=	1/1 = 1	✓	STRONG
{D} → {A, C}	=	1/2 = 0.5	✗	
{A, C} → {D}	=	1/3 = 0.33	✗	
{C} → {A, D}	=	1/3 = 0.33	✗	
{A, D} → {C}	=	1/1 = 1	✓	STRONG //