CS 412 – Introduction to Data Mining

Assignment 3

Problem 1:

Consider the following transaction database:

Transaction ID	Items
1	b, d, f, g, l
2	f, g, h, l, m, n
3	b, f, h, k, m
4	a, f, h, j, m
5	d, f, g, j, m

(a) Given Minimum support = 0.4

To compute the ordered list of frequent items (length 1 patterns): We list the count of each length 1 pattern and compute its Relative Support = $\frac{Count}{Total\ no.of\ transaction}$

Item	Count	Relative Support
a	1	0.2
b	2	0.4
d	2	0.4
f	5	1
g	3	0.6
h	3	0.6
j	2	0.4
k	1	0.2
l	2	0.4
m	4	0.8
n	1	0.2

Since Minimum Relative Support = 0.4, we prune the items with support < 0.4, hence we get the following list of frequent items:

(Absolute Support count = 0.4*5 = 2, hence we prune items with support count < 2).

Item	Count	Relative Support
b	2	0.4
d	2	0.4
f	5	1
g	3	0.6
h	3	0.6
j	2	0.4
l	2	0.4
m	4	0.8

Note that a, k and n got pruned.

Next, we arrange the items in decreasing order of the support count values to get the ordered frequent items list as:

Item	Count	Relative Support
f	5	1
m	4	0.8
g	3	0.6
h	3	0.6
b	2	0.4
d	2	0.4
j	2	0.4
l	2	0.4

Hence, the resulting ordered frequent items list is

L = {{f: 5}, {m: 4}, {g: 3}, {h: 3}, {b: 2}, {d: 2}, {j: 2}, {l: 2}}

(b)

To compute the FP tree, we need to compute the ordered list of frequent items for each transaction based on the above obtained list L = {{f: 5}, {m: 4}, {g: 3}, {h: 3}, {b: 2}, {d: 2}, {j: 2}, {l: 2}} as follows:

Next, we create the ordered frequent items list for each transaction: (Each item in transaction is ordered according to L and it does not contain pruned items)

Table A:

Transaction ID	Items	Ordered frequent items
1	b, d, f, g, l	f, g, b, d, l
2	f, g, h, l, m, n	f, m, g, h, l
3	b, f, h, k, m	f, m, h, b
4	a, f, h, j, m	f, m, h, j
5	d, f, g, j, m	f, m, g, d, j

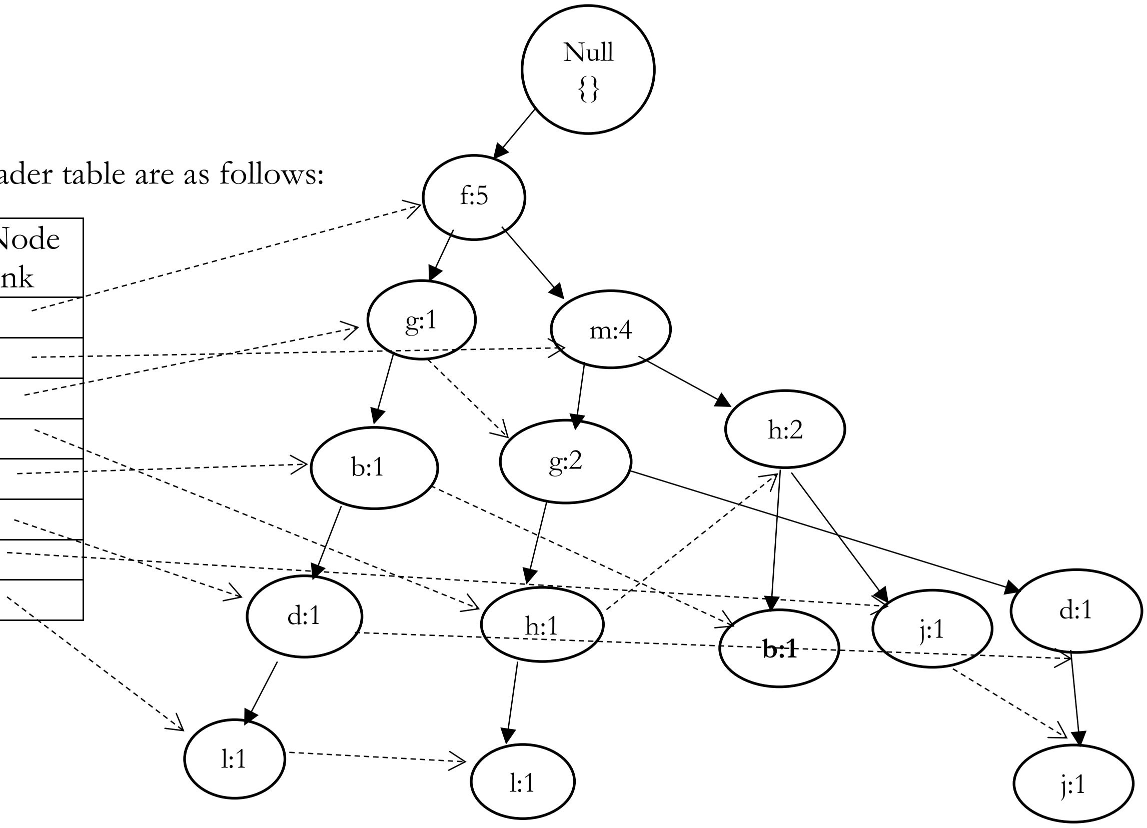
To compute the FP Growth tree, we next consider the header table where each item points to its occurrences in the tree via node links.

The header table is given by:

Item	Count
f	5
m	4
g	3
h	3
b	2
d	2
j	2
l	2

The FP tree and header table are as follows:

Item	Support Count	Node link
f	5	
m	4	
g	3	
h	3	
b	2	
d	2	
j	2	
l	2	



We first create the root of FP Tree called Null. The, we scan the database and go through items in each transaction in the order of list L i.e. we process items in each transaction in the order shown in Table A above and keep on creating branches for the transactions.

Considering the first transaction, the ordered frequent item list is {f, g, b, d, l}. Hence, a branch in the tree with nodes {f: 1}, {g: 1}, {b: 1}, {d: 1}, {l: 1} is created. A link is established between these nodes where 'f' is linked to the root node, 'g' is linked to 'f', 'b' to 'g', 'd' to 'b' and 'l' to node 'd'.

Now in the next transaction {f, m, g, h, l}, a branch of the tree will be created with nodes f, m, g, h and l, where f is linked to the root node, m to f, g to m, h to g and l to h. Whereas, this branch shares node 'f' with the previous branch of the FP tree for transaction ID 1. Hence, the count for 'f' node is incremented by 1 making the node as {f: 2} and new nodes {m: 1}, {g: 1}, {h: 1}, {l: 1} are created and linked as discussed.

So, whenever we are creating a new branch for a transaction, the counts of common nodes are incremented and the new nodes appearing are created and linked appropriately.

Also, each item using the header table node links points to its' occurrences in the FP tree. All occurrences of an item in the FP tree are linked together via a chain of these node links (dotted lines in the FP tree above).

(c)

To construct the conditional pattern bases and FP tree, we start with the last item in L out of {m, h, b j} which is j.

Now, from the FP tree above, j occurs from two different branches namely {f, m, g, d, j: 1} and {f, m, h, j:1}. Now, keeping j as the suffix, we get two prefix paths namely {f, m, g, d: 1} and {f, m, h:1}. These correspond to the conditional pattern base for item j.

Now using these two conditional pattern bases, we construct the j- conditional FP tree with only {f, m: 2} as the path. 'g', 'd', and 'h' are not included because their support count is 1 which is less than the minimum support count of 2.

Now, {f, m: 2} generates all combinations of frequent patterns namely {f, m, j: 2}, {f, j: 2}, {m, j: 2}. Hence the frequent patterns generated using j are {f, m, j: 2}, {f, j: 2}, {m, j: 2}.

Similar for item 'b', from the branches of FP tree we get the conditional pattern base as

{f, m, h: 1} and {f, g: 1}. Now, from these pattern bases, we get a single node conditional FP tree {f: 2}. This conditional FP tree generates all the frequent patterns using b, namely {f, b: 2}.

Similar for item ‘h’, from the branches of FP tree we get the conditional pattern base as

{{f, m, g: 1}, {f, m: 2}}. Now, from these pattern bases, we get {f: 3, m: 3} as the conditional FP tree. This conditional FP tree generates all the frequent patterns using h, namely {f, h: 3}, {m, h: 3} and {f, m, h: 3}.

Similar for item ‘h’, from the branches of FP tree we get the conditional pattern base as

{{f, m, g: 1}, {f, m: 2}}. Now, from these pattern bases, we get {f: 3, m: 3} as the conditional FP tree. This conditional FP tree generates all the frequent patterns using h, namely {f, h: 3}, {m, h: 3} and {f, m, h: 3}.

Item	Conditional Pattern Base	Conditional FP tree	Frequent Patterns generated
j	{{f, m, g, d: 1}, {f, m, h: 1}}	{f: 2, m: 2}	{f, m, j: 2}, {f, j: 2}, {m, j: 2}
b	{{f, m, h: 1}, {f, g:1}}	{f: 2}	{f, b: 2}
h	{{f, m, g: 1}, {f, m: 2}}	{f: 3, m: 3}	{f, h: 3}, {m, h: 3}, {f, m, h: 3}
m	{{f:4}}	{f: 4}	{f, m: 4}

(d)

We order the items in each transaction in the decreasing frequency before constructing the FP tree because if the items in transactions repeat, in that scenario, the branches in the tree overlap and we do not need to create another node for such cases when repetition of prefix occurs, rather we can simply increment the count of previous nodes of the tree. Hence, fixing a particular order before constructing the FP tree helps in overlap of the branches of the FP tree in case of item repetitions in transactions. So, in the scenario when different transactions share certain items or prefix, the counts of the items (nodes) in the tree already present are incremented.

Ordering in decreasing order of the frequency of items helps in overlap of nodes of the tree when the paths overlap, or different transactions share some items.

Considering the above example, since f appeared most times, it is the first element in the ordered frequent item list for each transaction. And keeping ‘f’ just after the root node helped decrease the memory effort, since that is going to comparatively overlap most frequently compared to other nodes.

Since the ‘f’ node overlaps the most since its’ most frequent and keeping it at top helps manage memory efficiently.

Clearly, using this ordering, if the paths overlap, the memory usage is reduced, and hence this ordering helps FP tree to be a memory efficient algorithm.

(e)

The frequent patterns obtained are:

{f, m, j: 2}, {f, j: 2}, {m, j: 2}

{f, b: 2}

{f, h: 3}, {m, h: 3},

{f, m, h: 3}

{f, m: 4}

A pattern X is closed if there exists no proper super pattern of X with the same support as X.

- {f, m, j: 2} is closed since there exist no proper super pattern of the same with support 2.
- {f, j: 2} is not closed since there exist a proper super pattern namely {f, m, j: 2} with same support as {f, j: 2}.
- {m, j: 2} is not closed since there exist a proper super pattern namely {f, m, j: 2} with same support as {m, j: 2}.
- {f, b: 2} is closed since there exist no proper super pattern of the same with support 2.
- {f, h: 3} is not closed since there exist a proper super pattern namely {f, m, h: 3} with same support as {f, h: 3}.
- {m, h: 3} is not closed since there exist a proper super pattern namely {f, m, h: 3} with same support as {m, h: 3}.
- {f, m, h: 3} is closed since there exist no proper super pattern of the same with support 3.
- {f, m: 4} is closed since there exist no proper super pattern of the same with support 4.

A pattern X is maximal if there exists no proper super pattern of X which is frequent.

- {f, m, j: 2} is maximal since there exist no proper super pattern of the same with support greater than or equal to 2 (frequent).
- {f, j: 2} is not maximal since there exist a proper super pattern namely {f, m, j: 2} which is frequent.
- {m, j: 2} is not maximal since there exist a proper super pattern namely {f, m, j: 2} which is frequent.
- {f, b: 2} is maximal since there exist no proper super pattern of the same with support greater than or equal to 2 (frequent).
- {f, h: 3} is not maximal since there exist a proper super pattern namely {f, m, h: 3} which is frequent.
- {m, h: 3} is not maximal since there exist a proper super pattern namely {f, m, h: 3} which is frequent.
- {f, m, h: 3} is maximal since there exist no proper super pattern of the same with support greater than or equal to 2 (frequent).
- {f, m: 4} is not maximal since there exist a proper super pattern namely {f, m, h: 3} which is frequent.

  Hence, the closed frequent patterns are {f, m, j: 2}, {f, b: 2}, {f, m, h: 3} and {f, m: 4}.
  The maximal frequent patterns are {f, m, j: 2}, {f, b: 2}, {f, m, h: 3}.

(f)

Consider the frequent pattern {f, m, h: 3}. Then the association rules we may consider for the same are as follows:

f, m ⇒ h

f, h ⇒ m

m, h ⇒ f

f ⇒ m, h

h ⇒ f, m

m ⇒ f, h

Then for each rule, the support is same as the support count of {f, m, h} namely 3.

- And confidence of rule f, m ⇒ h = Support ({f, m, h})/ Support ({f, m}) = ¾ = 0.74

which is greater than minimum confidence level 0.6, hence the rule f, m ⇒ h is an association rule.

- Similarly, confidence of rule f, h ⇒ m = Support ({f, m, h})/ Support ({f, h}) = 3/3 = 1

which is greater than minimum confidence level 0.6, hence the rule f, h ⇒ m is an association rule.

- Similarly, confidence of rule m, h ⇒ f = Support ({f, m, h})/ Support ({m, h}) = 3/3 = 1

which is greater than minimum confidence level 0.6, hence the rule m, h ⇒ f is an association rule.

Hence, the computed support and confidence for each of the rules is tabulated below:

Rules	Support	Confidence
f, m ⇒ h	3	0.75
f, h ⇒ m	3	1
m, h ⇒ f	3	1
f ⇒ m, h	3	0.6
h ⇒ f, m	3	1
m ⇒ f, h	3	0.75

Hence, the association rules with minimum confidence = 0.6 are:

{f, m ⇒ h}, {f, h ⇒ m}, {m, h ⇒ f}, {f ⇒ m, h}, {h ⇒ f, m}, {m ⇒ f, h}.

(a)

Continued…

To compute the FP tree, we need to compute the ordered list of frequent items for each transaction based on the above obtained list L = {{f: 5}, {m: 4}, {g: 3}, {h: 3}, {b: 2}, {d: 2}, {j: 2}, {l: 2}} as follows: (Each item in transaction is ordered according to L and it does not contain pruned items)

Table A:

Transaction ID	Items	Ordered frequent items
1	b, d, f, g, l	f, g, b, d, l

2	f, g, h, l, m, n	f, m, g, h, l
3	b, f, h, k, m	f, m, h, b
4	a, f, h, j, m	f, m, h, j
5	d, f, g, j, m	f, m, g, d, j

Submitted by:

Rachneet Kaur, Net ID: rk4