Co-Occurrences Frequent Item Tree

Ying Xu 徐莹
yx2@cs.ualberta.ca

1 Introduction

- Association rule mining
- FP Growth
- COFI tree mining

(COFI-tree Mining: A New Approach to Pattern Growth with Reduced Candidacy Generation Hajj, Zaiane)

2 FP-tree

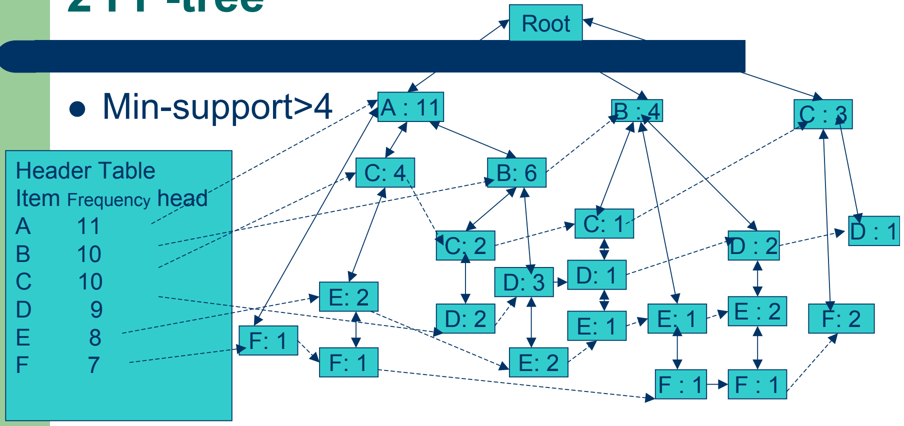
- Frequent item header that contains item names and pointer to the first node in FP tree.
- Prefix tree
- Each node contains the item name, frequency and pointer to another node of the same kind.

2 FP-tree

TID	Items	TID	Items
T1	A G D C B	T10	C F G R
T2	B C H E D	T11	A D B H I
T3	B D E A M	T12	D E B K L
T4	C E F A N	T13	M D C G
T5	A B N	T14	C F
T6	A C G	T15	B D E F I
T7	A C H I G	T16	J E B A D
T8	L E F K B	T17	A K E F C

2 FP-tree

- Min-support>4

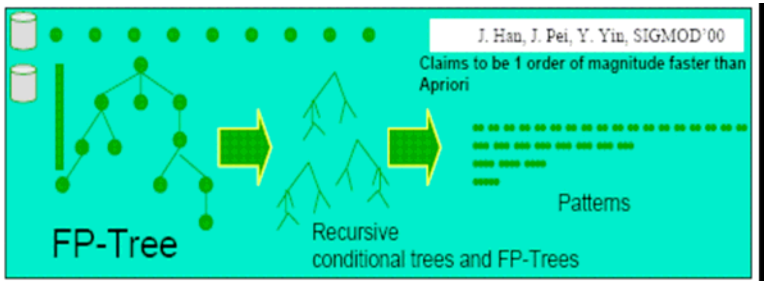


Header Table

Item	Frequency	head
A	11	
B	10	
C	10	
D	9	
E	8	
F	7	

2 FP-tree

- Mining



J. Han, J. Pei, Y. Yin, SIGMOD'00

Claims to be 1 order of magnitude faster than Apriori

2 FP-tree

- Drawback:

memory space usage

3. COFI-tree

- Prunning
- **global frequent/local non-frequent property:**

the itemset that is global frequent but not local frequent with respect to the item A of the A-COFI-tree

It is an anti-monotone property

3. COFI-tree

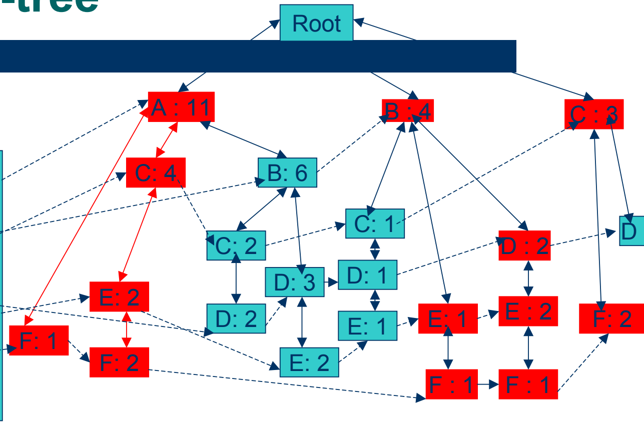
- Frequent item header that contains items names which are frequent with respect to the specific item ascending ordered by global frequency.
- Prefix tree
- Each node contains the item name, frequency, participation counter and pointer to another node of the same kind.

3. COFI-tree

- FP-tree

Header Table

Item	Frequency	head
A	11	
B	10	
C	10	
D	9	
E	8	
F	7	



3. COFI-tree

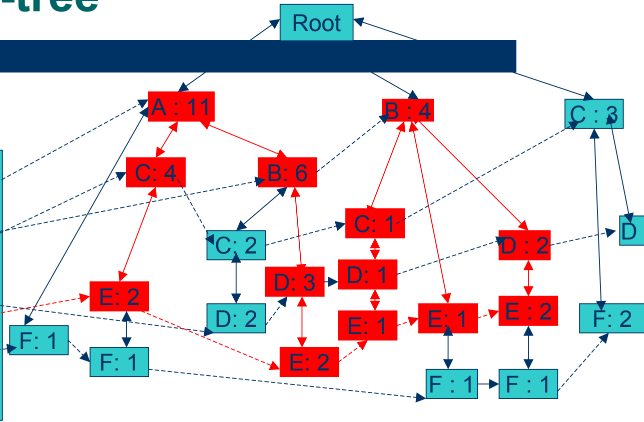
E	4
D	2
C	4
B	2
A	3

F (7 0)

3. COFI-tree

Header Table

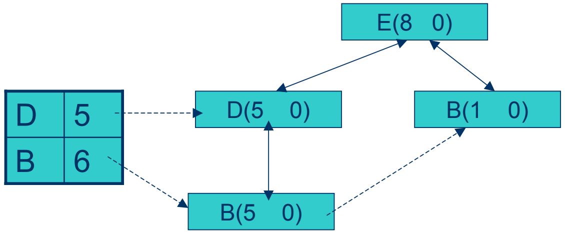
Item	Frequency	head
A	11	
B	10	
C	10	
D	9	
E	8	
F	7	



3. COFI-tree

- E-COFI-tree (Support>4)

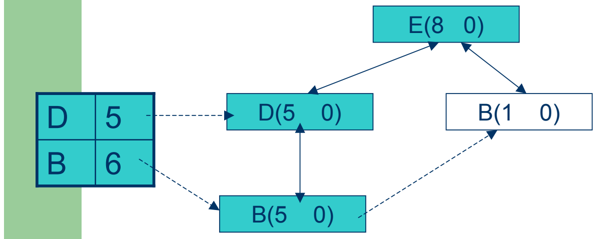
D	5
C	3
B	6
A	4



D	5
B	6

3. COFI-tree

- Mining



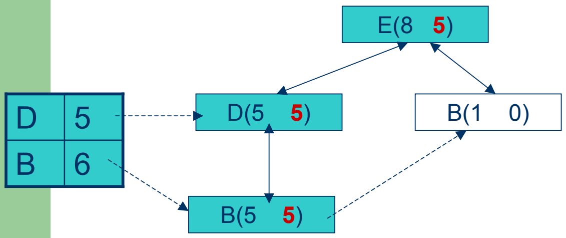
D	5
B	6

Pattern
E D B 5

E D 5
E B 5
E D B 5

3. COFI-tree

- Mining



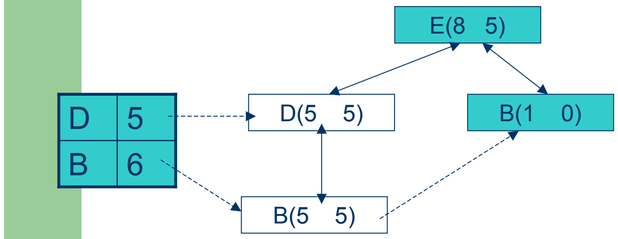
D	5
B	6

Pattern
E D B 5

E D 5
E B 5
E D B 5

3. COFI-tree

- Mining



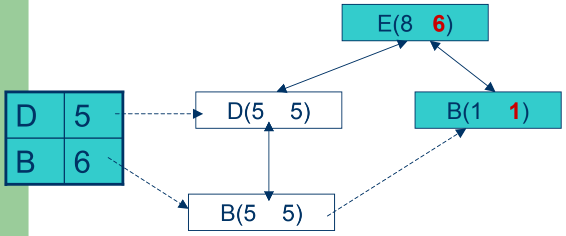
D	5
B	6

Pattern
E B 1

E D 5
E B 6
E D B 5

3. COFI-tree

- Mining

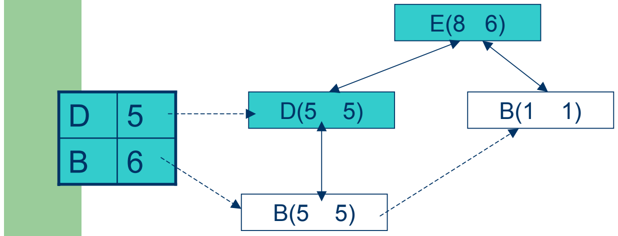


Pattern
E B 1

E D 5
E B 6
E D B 5

3. COFI-tree

- Mining



Pattern
E D 0

4 Algorithm

- Algorithm COFI:

```
Input: modified FP-Tree, a minimum support threshold
Output: Full set of frequent patterns
Method:
1. A = the least frequent item on the header table of FP-Tree
2. While (There are still frequent items) do
   2.1 count the frequency of all items that share item (A) a path.
         Frequency of all items that share the same path are the same as of
          the frequency of the (A) items
   2.2 Remove all non-locally frequent items for the frequent list
         of item (A)
   2.3 Create a root node for the (A)-COFI-tree with both frequency-count
         and participation-count = 0
         2.3.1 C is the path of locally frequent items in the path of item A to
```

4 Algorithm

- Algorithm COFI:

```
         2.3.2 Items on C form a prefix of the (A)-COFI-tree.
         2.3.3 If the prefix is new then Set frequency-count= frequency of (A)
               node and participationcount= 0 for all nodes in the path
         Else
         2.3.4 Adjust the frequency-count of the already exist part of the path.
         2.3.5 Adjust the pointers of the Header list if needed
         2.3.6 find the next node for item A in the FP-tree and go to 2.3.1
   2.4 MineCOFI-tree (A)
   2.5 Release (A) COFI-tree
   2.6 A = next frequent item from the header table
3. Goto 2
```

4 Algorithm

- Function: MineCOFI-tree (A)

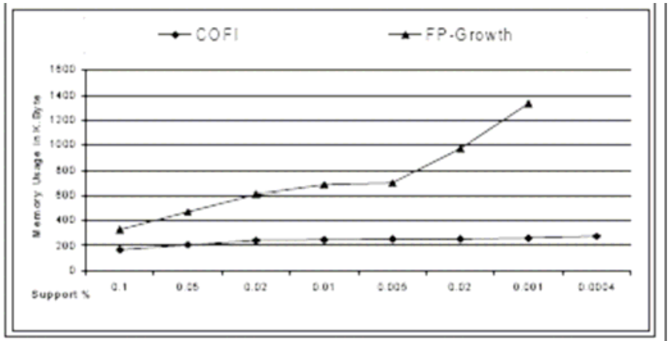
```
1. nodeA = select next node //Selection of nodes starts with the node of
   most globally frequent item and following its chain, then the next less
   frequent item with its chain, until we reach the least frequent item in the
   Header list of the (A)-COFI-tree
2. while there are still nodes do
   2.1 D = set of nodes from nodeA to the root
   2.2 F = nodeA.frequency - nodeA.participationCount
   2.3 Generate all Candidate patterns X from items in D.
         Patterns that do not have A will be discarded.
   2.4 Patterns in X that do not exist in the A-Candidate List will be added to
         it with frequency = F otherwise just increment their frequency with F
   2.5 Increment the value of participationCount by F for all items in D
   2.6 nodeA = select next node
```

4 Algorithm

- Function: MineCOFI-tree (A)

```
3. Goto 2
4. Based on support threshold  remove non-frequent patterns
   from A Candidate List.
```

5 Experimental Studies



(B) Total Memory requirement

Questions?