March 2001

1. Apriori algoritm (2p+1p+1p+1p+1p=6p)

a. Run the Apriori algorithm on the following transactional database with minimum support equal to one transaction. Explain step by step the execution.

Transaction id	Items
1	A, B, C
2	A, B, D

b. Repeat the exercise 1a with the following additional constraint: Find the frequent itemsets that contain the items A and B (both!). Explain step by step the execution. Make clear when and how the constraint is used. Incorporate the constraint into the algorithm, i.e. do not simply run the algorithm and afterwards consider the constraint.

c. Repeat the exercise 1a with the following additional constraint: Find the frequent itemsets that do not contain the item A or the item B. Explain step by step the execution. Make clear when and how the constraint is used. Incorporate the constraint into the algorithm, i.e. do not simply run the algorithm and afterwards consider the constraint.

d. Sketch a proof of the correctness of the Apriori algorithm.

e. Run the Simple Algorithm on the transactional database below to produce association rules for the itemset XYZ. Use a minimum confidence threshold equal to 100 %. Explain step by step the execution.

Transaction id	Items
1	X, Y, Z
2	X, Y
3	Y, Z

Solution:

a.

C1: A, B, C, D

L1: A, B, C, D

C2: AB, AC, AD, BC, BD, CD

CD does not have minimum support, then

L2: AB, AC, AD, BC, BD

C3: ABC, ABD

Note that ACD and BCD are not in C3 because they are produced by self-join of L2 but they don't pass the subset checking step, i.e. ACD's and BCD's subset CD is not in L2.

L3: ABC, ABD

C4: empty

Note again that ABCD is not in C4 because it is produced by self-join of L3 but it doesn't pass the subset checking step, i.e. ABCD's subset BCD is not in L3.

b.

The constraint is monotone and, thus, it avoids having to check the constraint sometimes.

C1: A, B, C, D

L1: A, B, C, D

Note that none of the itemsets in L1 is in the output, because none of them satisfies the constraint. Note that the constraint has to be checked for every itemset in L1.

C2: AB, AC, AD, BC, BD, CD

CD does not have minimum support, then

L2: AB, AC, AD, BC, BD

Note that the only itemset in L2 that is in the output is AB. Note that the constraint has to be checked for every itemset in L2.

C3: ABC, ABD

Note that ACD and BCD are not in C3 because they are produced by self-join of L2 but they don’t pass the subset checking step, i.e. ACD’s and BCD’s subset CD is not in L2.

L3: ABC, ABD

Note that both itemsets in L3 are in the output. Note that the constraint does not need to be checked for any of the itemsets in L3, because both of them are supersets of AB and we have checked above that AB satisfies the constraint. This advantage is due to the fact that the constraint is monotone.

C4: empty

Note again that ABCD is not in C4 because it is produced by self-join of L3 but it doesn’t pass the subset checking step, i.e. ABCD’s subset BCD is not in L3.

c.

The constraint is antimonotone and, thus, it helps to prune the search space.

C1: A, B, C, D

L1: A, B, C, D

Note that all the itemsets in L1 are in the output, because all of them satisfy the constraint (i.e. the itemsets A, C and D do not contain the item B, and the itemset B does not contain the item A).

C2: AB, AC, AD, BC, BD, CD

CD does not have minimum support and AB does not satisfy the constraint, then

L2: AC, AD, BC, BD

Note that all the itemsets in L2 are in the output.

C3: empty

Note that ACD and BCD are not in C3 because they are produced by self-join of L2 but they don’t pass the subset checking step, i.e. ACD’s and BCD’s subset CD is not in L2.

d.

Seen in the classroom. See also the following article (available in the course website):

R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules, Proc. of the 20th Int. Conf. on Very Large Databases, Santiago, Chile, September 1994. Expanded version available as IBM Research Report RJ9839, June 1994.

e.

The algorithm first considers all the subsets of XYZ of size 2 as antecedent to the rule. The rest of the items are in the consequent. That is,

XY -> Z, confidence=50 %

YZ -> X, confidence=50 %

XZ -> Y, confidence=100 %

In the first two cases, the algorithm does not apply recursion because none of the rules whose antecedent is a subset of XY or YZ can give rise to a rule with confidence higher than 50 %. In the last case, the algorithm applies recursion to consider the rules

X -> YZ, confidence=50 %

Z -> XY, confidence=50 %

So, the only rule in the output is XZ -> Y, confidence=100 %.

2. FP algorithm (2p+1p+1p+1p=5p)

a. Run the FP algorithm on the transactional database in exercise 1a with minimum support equal to 1 transaction. Explain step by step the execution.

b. How do you incorporate a monotone constraint in the FP grow algorithm ?

c. How do you incorporate an antimonotone constraint in the FP grow algorithm ?

d. What is the main advantage that the FP grow algorithm has over the Apriori algorithm ?

Solution:

a.

f-list: A,B,C,D
No reordering of the items in the transactions is needed

FP tree:

```
          {}
          |
          A, 2
          |
          B, 2
          /   \
     C, 1     D, 1
```

Output: A:2, B:2, C:1, D:1

C-conditional database: AB:1
f-list: A,B
No reordering of the items in the transactions is needed
FP tree:

```
{}
|
A, 1
|
B, 1
```

Output: Since the FP tree has a single branch, the output contains all the combinations of the items in the tree, i.e. AC:1, BC:1, ABC:1

D-conditional database: AB:1
Proceed similarly to the C-conditional database.
Output: AD:1, BD:1, ABD:1

B-conditional database: A:2. Output: AB:2.

A-conditional database: empty

b.

Check course slides.

c.

Check course slides.

d.

No candidate is generated, which has several advantages: No storage is needed, no self-join is needed, no subset checking is needed, no pattern matching is needed when counting support, etc.

3. Constraints (1p+1p+1p=3p)

a. Given a transactional database, we want to find all the itemsets whose support is exactly 10. Specify the minimun support and constraints (if any) that are needed to run the Apriori or FP grow algorithm and find the desired itemsets.

b. Let C1 be a monotone constraint. Let us define another constraint C2 so that C2 holds for an itemset X if and only if C1 does not hold for X. Is C2 monotone, antimonotone, both, none, or we simply cannot know ? Explain your answer.

c. Give an example of a constraint that is both convertible monotone and convertible antimonotone. Explain your answer.

Solution:

a.

Minimum support=10. Constrain: The support of the itemset must be equal or smaller than 10. Note that the constraint is monotone and, thus, we know how to incorporate it in the Apriori and FP grow algorithms.

b.

If C2 is true for an itemset X, then C1 is false for X. Consider any subset Y of X. Since C1 is monotone, then C1 is false for the itemset Y. Thus, C2 is true for Y. Consequently, C2 is antimonotone.

c.

Check the course slides.