10 itemsets out of 100 contain item A, of which 5 also contain B. The rule A ➔ B has:

A. 5% support and 10% confidence
B. 10% support and 50% confidence
C. 5% support and 50% confidence
D. 10% support and 10% confidence

Answer C

The number of itemsets containing both A and B is 5, therefore the support is 5%.

In 5 out of 10 cases the rule is correct, therefore the confidence is 50%.

10 itemsets out of 100 contain item A, of which 5 also contain B. The rule B ➔ A has:

A. unknown support and 50% confidence
B. unknown support and unknown confidence
C. 5% support and 50% confidence
D. 5% support and unknown confidence

Answer D

As in the previous question the support is 5%.

Since we do not know the number of itemsets containing the item B, we cannot determing the confidence of the rule.

Given the frequent 2-itemsets {1,2}, {1,4}, {2,3} and {3,4}, how many 3-itemsets are generated and how many are pruned?

A. 2, 2

B. 1, 0

C. 1, 1

D. 2, 1

Answer C

Only one itemset will be generated, {1,2,4} for the first shared prefix {1}. Since the itemset {2,4} is not frequent, it will be pruned.

After the join step, the number of k+1-itemsets ...

A. is equal to the number of frequent k-itemsets
B. can be equal, lower or higher than the number of frequent k-itemsets
C. is always higher than the number of frequent k-itemsets
D. is always lower than the number of frequent k-itemsets

Answer B

We cannot make any statement on the cardinality of the candidate set generated in the join step.

Examples:

{1,2} {1,3} will generate 1 3-itemset, so the cardinality is lower.

{1,2} {1,3} {1,4} will generate 3 3-itemsets, so the cardinality is equal.

{1,2} {1,3} {1,4} {1, 5} will generate 6 3-itemsets, so the cardinality is higher.

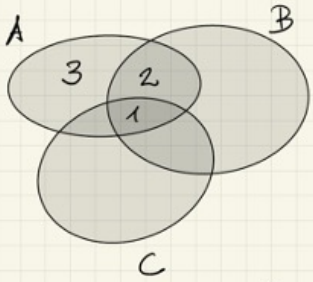
If rule {A,B} ➔ {C} has confidence c_1 and rule {A} ➔ {C} has confidence c_2, then ...

A. $c_2 >= c_1$
B. $c_1 > c_2$ and $c_2 > c_1$ are both possible
C. $c_1 >= c_2$

Answer B

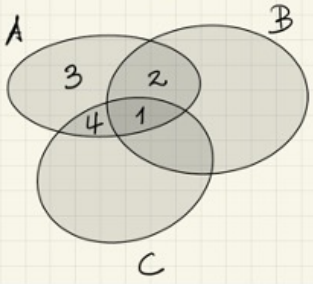
See example on next slide.

Example



$$c_1 = \frac{s(A,B,C)}{s(A,B)} = \frac{1}{3}$$

$$c_2 = \frac{s(A,C)}{s(A)} = \frac{1}{6} = 0 \Rightarrow c_1 > c_2$$



$$c_1 = \frac{s(A,B,C)}{s(A,B)} = \frac{1}{3}$$

$$c_2 = \frac{s(A,C)}{s(A)} = \frac{5}{10} = \frac{1}{2} \Rightarrow c_1 < c_2$$

A false negative in sampling can only occur for itemsets with support smaller than …

A. the threshold s

B. p*s

C. p*m

D. None of the above

Answer D (under reasonable assumptions)

A false negative occurs if an itemset has a support above the threshold s, but in the sample less then p*s transactions are selected that contain the itemset (respectively less than the lowered threshold).

For each of the options in answers A, B, C it is easy to construct an adversarial example in which this is the case. Actually the more difficult case is when the itemset has the larger support count, as fewer remaining transactions remain to select from. Therefore, we conly consider the maximal possible size.

For answer A consider an itemset with support count larger than s. If m > (s – p*s+1) + p*m, then we can select the sample such that it contains at most p*s – 1 of the occurrences of the itemset. This is equivalent to saying that m > s, i.e. the database should have larger size than the support threshold. We never explicitly stated that this should be the case but is an obvious condition to satisfy.

For answer B the itemset has a support strictly smaller than s, and therefore for the same reasoning as in answer A, we can produce the counter-example.

For answer C we can select the occurrences without including the itemset from the (1-p)*m remaining occurrences if m > (m – p*m + 1) + p*m, which is equivalent to m > m-1. Here lowering the threshold could in the extreme case avoid having a false negative.

In the first pass over the database of the FP Growth algorithm

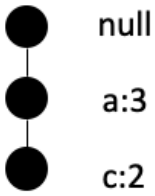
A. Frequent itemsets are extracted
B. A tree structure is constructed
C. The frequency of items is computed
D. Prefixes among itemsets are determined

Answer C

The algorithms proceeds in two steps, where the first step has two passes over the database. The first pass is to compute the frequency of items, and the second to construct the tree structure. The frequent itemsets are extracted in the second step.

The FP tree below is …

A. not valid, b is missing
B. not valid, since count at leaf level larger than 1
C. possible, with 2 transactions {a}
D. possible, with 2 transactions {a,c}



Answer D

Answer A can be excluded, since it is possible to have paths that are not including some elements. It simply means that the corresponding itemsets do not occur.

Answer B can be excluded since leaf counts can be larger than 1, of multiple itemsets corresponding to the path exist.

Answer C is not possible, since in the presence of two transactions with itemset {a} and the tree structure implying that there exist two transactions with itemset {a,c}, the minimal count of node a has to be 4.

Answer D is possible with three transactions {a}, {a,c}, {a,c}