**3. Exercise 2: Apriori, Confidence, Itemsets and Association Rules**

(1 point)

**4. Exercise 2-1 Combinatoric Explosion**

1. A database contains transactions over the following items: apples, bananas, and cherries. How many different combinations of these items can exist (i.e., how many different transactions could possibly occur in the database)?

(We do not distinguish whether a transaction contains a fruit once or several times, e.g., regardless if someone bough one apple or several apples would just result in the transaction containing apples).

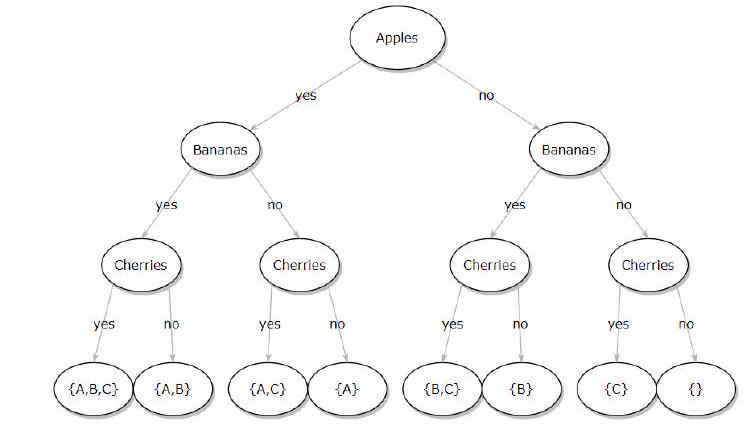
**5. Suggested solution:**

A transaction can either contain apples or not. We have 2 possibilities here.

Each of these possiblities can either contain bananas or not. That is, for each of the 2 previous possibilities, we have 2 possibilities. Therefore we have four overall.

Each of these four possibilities can either contain cranberries or not. Eight possibilities.

For illustration, you could sketch this at the blackboard as a branching pattern (like a binary tree: first layer: apples: yes/no, second layer at each branch: bananas: yes/no, etc.).



1. The database now also contains the items dates, eggplants, figs, and guavas. How many possible transactions do we have now?

**6. Suggested solution:**

It becomes clear that sketching a tree is not convenient anymore. We can explain this in another way, using a table.

Fill in the TID column later, after discussion about the numbers and their relationship to the code we have in each row (binary vs. decimal):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TID |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 7 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 8 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |
| 16 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |
| 127 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

1. How many combinations (possible different transactions) do we have with items?

**7. Suggested solution:**

From the previous discussion it should have become clear now that the set of all possible combinations is the powerset over the set of items, where each item can be either in or out. This property (in or out) can be represented as a binary code, i.e., each element of the powerset can be uniquely mapped to exactly one number in binary representation, and each number , can be uniquely mapped to exactly one element of the powerset.

So we have overall possible combinations (i.e., different transactions), where is the number of items.

We say, the number of possibilities grows exponentially. And this growth rate is quite fast. For we have 1024 , for we have , for we have . 4. How many transactions with exactly two items (i.e., 2-itemsets) can we have when the database contains 3 items? When it contains 5 items? How many -itemsets do we have when the database contains items?

**8. Suggested solution:**

Use the database with 3 items as example: we can have two of the three elements : .

To answer the question with complete enumeration of all possibilities for 5 elements already becomes tiresome, so we will derive the answer from the general solution.

You can view this question as having a collection of elements and drawing of them sequentially without putting an element back.

Let us first assume, we care for the order of drawing, i.e., we distinguish from . Then we have possibilities to draw the first element, possibilities to draw the second, and so on until we have possibilities to draw the -th element. Altogether:

Now we do actually not care for the order, i.e., we do not distinguish from . Therefore we have to divide the result by the number of possible orderings. A set of elements can be ordered/permuted in different ways.

The number of -itemsets out of different items is therefore the expression from before, divided by :

This is also written with the expression

and called the binomial coefficient. Exercise 2-2 Itemsets and Association Rules

(1 point)

Given a set of transactions according to the following table:

|  |  |
| --- | --- |
| Transaction ID | Items in basket |
| 1 | Milk, Eggs, Pasta |
| 2 | Bread, Butter, Milk |
| 3 | Milk, Pasta, Onions |
| 4 | Bread, Butter, Onions |
| 5 | Eggs, Onions, Pasta |
| 6 | Milk, Pasta, Bread, Butter |
| 7 | Bread, Butter, Pasta |
| 8 | Eggs, Pasta |
| 9 | Milk, Pasta, Bread, Butter |
| 10 | Eggs, Onions |

1. What are the support and confidence of Pasta Milk ?

**9. Suggested solution:**

Support is the amount of times they appear together in the table.

Support is

Confidence is the support, divided by the times pasta appears alone in the table.

Confidence is

1. What is the maximum number of size-3 itemsets that can be derived from this data set?

**10. Suggested solution:**

First we need to know the number of items:

Milk, Eggs, Pasta, Bread, Butter, Onions

To choose any 3 of 6, the mathematical term is

1. What is the maximum number of association rules that can be extracted from this dataset (including rules which have zero support)?

**11. Suggested solution:**

From six items, we can generate association rules by having 1 or 2 or ... or 5 items in the antecedent and include all or some of the remaining items in the consequent (we exclude the case of an empty consequent, hence we subtract 1 from the number of elements in the powerset of the remaining items).

Mathematically:

that is for items:

The actual number is therefore:

1. What is the maximum size of frequent itemsets that can be extracted (assuming ?

**12. Suggested solution:**

The maximum frequent itemset occuring in the database has size 4 . We can therefore not find any larger itemset with support .

1. Find an itemset (of size 2 or larger) that has the largest support.

**13. Suggested solution:**

Bread, Butter

1. Find a pair of items , s.t. the rules and have the same confidence.

**14. Suggested solution:**

Exercise 2-3 Apriori Candidate Generation

point)

Given the frequent 3 -itemsets:

list all candidate 4-itemsets following the Apriori joining and pruning procedure.

Suggested solution:

Joining:

Two frequent -itemsets are joined if they are identical in the first items.

Pruning:

Remove any non-frequent itemsets.

Result:

After joining and pruning we have the following itemsets.

Exercise 2-4 The monotonicity of confidence

(1 point)

Theorem 2.1 (lecture) states:

Given:

* itemset

If , then

1. Prove the theorem.

Suggested solution:

Consider the following two rules:

and

where

The confidence of the rules are: and , respectively.

Since , we have: .

Therefore the former rule cannot have a higher confidence than the latter rule.

1. Sketch an algorithm (pseudo code) that generates all association rules with support or above and a minimum confidence of , provided the set of all frequent itemsets (w.r.t. with their support, efficiently using the pruning power of the given theorem. What is the asymptotic time complexity of the algorithm?

Suggested solution:

AssociationRules

foreach do:

while do:

maximal element in

if then

print

else

end if

end while

end foreach