

### Organisational Rules for Exam Admission

The lecture is accompanied by exercise sheets. At the beginning of each exercise lesson, the exercises are *voted for*. By voting for an exercise, one expresses one's willingness to present something about it. (Suggestions for a solution will be discussed, they need not be correct right away.) To the exam will be admitted, who

1. voted for *at least* half of the exercises *and*
2. presented something for *at least* two exercises.

### Exercise Sheet 1

#### Exercise 1 Use of Frequent Patterns

- a) How could information in the form of frequent item sets or association rules be used for marketing purposes etc.?
- b) What are potential applications of frequent item set mining?  
Try to find others than those mentioned on the motivation slide of the lecture!
- c) Does its frequency already make a pattern or a rule interesting?  
If not, why not? Try to find counterexamples!
- d) What are advantages of representing frequent patterns as association rules compared to representing them as frequent item sets? What are disadvantages?

#### Exercise 2 Basic Notions

- a) Why are transactions databases represented as vectors/arrays?  
Why can't we use simple sets? What are alternative representations?  
What are advantages/disadvantages of alternative representations?
- b) Why is the item base often not given explicitly? How can it be obtained?
- c) How can absolute and relative (minimum) support be translated into each other?  
What additional information is needed for the transformation?
- d) In frequent item set mining we consider only whether an item is present in or absent from a transaction. However, in market basket data, items have multiplicities (a customer buys 3 cartons of milk, 5 apples, 2 cans of soup etc.).  
How can this be handled? Can it be handled by preprocessing? If yes, how?

#### Exercise 3 Properties of Support

- a) What does it mean that support is anti-monotone or downward closed?  
How can this be expressed as a formula?
- b) What relation of item set covers corresponds to the anti-monotonicity of support?
- c) What is the apriori property? How is it stated formally? How does it follow?  
What does the contraposition of the apriori property say? What does it suggest?

#### Exercise 4 Properties of Support

- a) Demonstrate that the support of an item set and one of its supersets can be equal! (That is, the support of the superset need not be actually smaller.) How can one describe the condition under which this holds? Can there be multiple different supersets that have the same support?
- b) Let  $I$  be an item set and  $a$  and  $b$  two items that are not in  $I$ . Let  $s(I) = s(I \cup \{a\}) = s(I \cup \{b\})$ . What is the support of  $I \cup \{a, b\}$ ?

#### Exercise 5 Properties of Support

Find an example where one can determine from the support of (several) item sets  $I_k$  that  $\bigcup_k I_k$  must be frequent w.r.t.  $s_{\min} \geq 3$  without accessing the transaction database (you may, however, use the size  $n$  of the transaction database). Do *not* use the solution/assumptions of Exercise 4b!

#### Exercise 6 Searching for Frequent Patterns

- a) What is the size of the search space of frequent item set mining? How does it grow with the number of items?
- b) Why is an exhaustive search through all possible item sets not advisable? What is the (asymptotic) time complexity of an exhaustive search? Is the pruned/reduced search fundamentally more efficient? If yes, in what way? If no, why may it be useful nevertheless?

#### Exercise 7 Searching for Frequent Patterns

- a) How many subsets of size  $k - 1$  does an item set with  $k$  items have? Why is this a problem for the candidate generation?
- b) Show why any frequent item set of size  $k + 1$  can be formed in  $j = (k + 1)k/2$  possible ways with the original version of the Apriori algorithm!
- c) In how many ways can an *infrequent* item set of size  $k + 1$  be generated? Does this number differ from the answer for frequent item sets? Why?

#### Exercise 8 Searching for Frequent Patterns

- a) What structure of the search space are we exploiting for the search? Why is this structure advantageous given the properties of support?
- b) What is a Hasse diagram? What does it represent? For what are we using Hasse diagrams?
- c) How is the Hasse diagram transformed to improve the search? What is the purpose of this transformation?
- d) What do we achieve by assigning unique parent item sets? Could we also assign unique child item sets?
- e) Are there any item sets that always have a unique parent item set, even in the unmodified Hasse diagram?