Data Mining 2: Frequent Pattern Mining Summer 2018
Christian Borgelt from 18.06.2018

Exercise Sheet 7

Exercise 25 Carpenter Algorithm

a) How does the Carpenter algorithm exploit the alternative characterization of closed item sets (i.e., the one based on a closure operator)? What is the core problem of exploiting this characterization? (Hint: minimum support)

b) What data structure does the Carpenter algorithm use?
Could it also use a horizontal transaction representation?
What might be reasons why it is *not* implemented that way?

c) Execute the Carpenter algorithm for the transaction database shown on the right! Use a simple set as the repository, not a prefix tree as described in the lecture!

$a\ d\ e$
$b\ c\ d$
$a\ c\ e$
$b\ c\ d$
$a\ e$

d) How can one prune the search with minimum support in the Carpenter algorithm?

Exercise 26 Carpenter Algorithm

a) Can perfect extension pruning be used in the Carpenter algorithm? If yes, how? In what way is this analogous to perfect extension pruning in item set enumeration algorithms (like Eclat, FP-Growth etc.)?

b) Can one use the Carpenter algorithm to find the maximal (frequent) item sets?

c) Can one use the Carpenter algorithm to find *all* frequent item sets?

d) Why is the Carpenter algorithm not advisable for market basket data?
What kind of data is it recommendable for? Why?
(Hint: Argue with the time complexity in the parameters of the database.)

Exercise 27 IsTa Algorithm

a) What recursive relationship does the IsTa algorithm exploit to find closed frequent item sets? Where lies the complexity?

b) Suppose the IsTa algorithm completed its work and we have the final prefix tree representing the repository of found closed frequent item sets. How are these closed item sets actually reported from this repository? What conditions need to be checked when traversing the prefix tree?

c) How could one report the *maximal* item sets from this repository?
What conditions need to be checked in this case?

d) Why is the IsTa algorithm not advisable for market basket data?
What data is it recommendable for? Why?
(Hint: Argue with the time complexity in the parameters of the database.)