RWTH Aachen
Lehrstuhl für Informatik 9
Prof. Dr. T. Seidl

November 9th, 2015

Exercise 3 for the lecture

Data Mining Algorithms

WS 2015/2016

Hand in your solutions on November 16th before the lecture. The tutorial for this exercise will be held on November 20th. Solutions of groups with less than 3 or more than 4 students will not be graded.

Note: All commands for the R-exercises are required to be provided with comments, indicating which task the commands belong to. All R script files should contain a comment-line with the names and matriculation numbers of all group-members. Send all R-files to yifeng.lu@informatik.rwth-aachen.de. The subject of the mail must start with "[DMA1]".

Exercise 3.1) Frequent itemsets — 4 = 2+2 points

The *Apriori*-algorithm makes use of prior knowledge of subset support properties. Let I be the set of all items. Give proofs or counterexamples for the following claims:

a) Let $S \subseteq I$ be a frequent itemset. Then every non-empty subset $S' \subseteq S$ must also be frequent.
b) Let $S \subseteq I$ be an arbitrary itemset. Then support$(S') \geq$ support(S) holds for any non-empty subset $S' \subseteq S$.

Exercise 3.2) Mining frequent itemsets — 9 = 4+5 points

Let D be a database that contains four transactions. In addition let *min_sup* = 60% and *min_conf* = 80%.

TID	date	items_bought
T1	10-15-99	{K, A, D, B}
T2	10-15-99	{D, A, C, E, B}
T3	10-19-99	{C, A, B, E}
T4	10-22-99	{B, A, D}

a) Find all frequent itemsets using the Apriori algorithm.
b) Find all frequent itemsets using the FP-growth algorithm.

Exercise 3.3) Hash tree **5 = 3+2 points**

a) Construct a hash tree from the following candidates for frequent 3-itemsets with a maximum of 2 transactions per node and use the hash function $h(x) = x \bmod 3$: (1,4,5), (2,3,4), (3,6,8), (1,2,5), (4,5,8), (3,4,5), (1,2,4), (1,3,6), (3,5,6), (4,5,7), (3,5,7), (1,5,9), (6,8,9), (3,6,7), (5,6,7).

b) Search for all the candidate 3-itemsets that appear in (1,3,7,8,9) with the hash tree constructed in a). Show which nodes of the hash tree have to be visited. Count how many candidates have to be checked, compare this to the effort needed without using this hash tree.