Byoungjin Oh
0030411229

CS37300 Homework 5

Due date: Wednesday December 4, 11:59pm.

Assignment

1. (5 pts) The *Apriori* algorithm uses a generate-and-count strategy for deriving frequent itemsets. Candidate itemsets of size $k + 1$ are created by joining a pair of frequent itemsets of size k. A candidate is discarded if any one of its subsets is found to be infrequent during the pruning step. Suppose the algorithm is applied to data set below with *minsup* = 30%:

Trans ID	*Items*
1	{a,b,d,e}
2	{b,c,d}
3	{a,b,d,e}
4	{a,c,d,e}
5	{b,c,d,e}
6	{b,d,e}
7	{c,d}
8	{a,b,c}
9	{a,d,e}
10	{b,d}

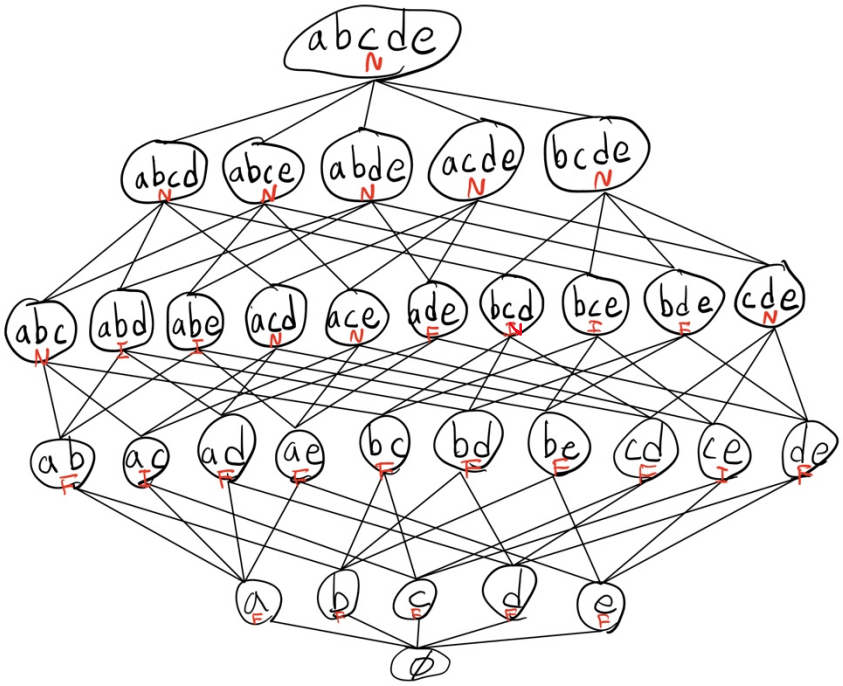
(a) Draw an itemset lattice representing the data above (with {∅} at the bottom and at {*a,b,c,d,e*} at the top). Label each node of the lattice with the following letter(s):

(i) **N**: If the itemset is not considered to be a candidate itemset by the *Apriori* algorithm. Note that there are two reasons for an itemset to not be considered as a candidate: (1) it is not generated at all during the candidate generation step, or (2) it is generated during the candidate generation step but is subsequently removed during the candidate pruning step because one of its subsets is found to be infrequent.

(ii) **F**: If the candidate is found to be frequent by the *Apriori* algorithm.

(iii) **I**: If the is found to be infrequent after support counting.

Byoungjin Oh
0030411229



(b) What is the percentage of frequent itemsets (with respect to all itemsets in thelattice)?

The percentage of frequent itemset is 15/32 (0.46875) 46.9%

(c) What is the pruning ratio of the *Apriori* algorithm on this data set? (The pruning ratio is defined as the percentage of all itemsets that are not considered to be candidates, either because they are not generated or because they are pruned before support is measured.)

The pruning ratio is 11/32 (0.34375) 34.4%

(d) What is the false alarm rate (i.e., percentage of candidate itemsets that are found to be infrequent after performing support counting?)

The false alarm rate is 5/32 (0.15625) 15.6%

Byoungjin Oh
0030411229

2. (5 pts) Consider the yelp5.csv data. In your programming assignment, you will construct itemsets based on the unique attribute values in each of the 11 discrete attributes. Given this:

   (i) How many frequent itemsets are possible?

   The number of possible frequent itemsets is (2^141). Make the discrete attributes to binary then the number of attributes is 141.

   (ii) How many association rules are possible?

   The number of possible association rules is (3^141) – (2^142) + 1.

   (iii) Describe how the support and confidence thresholds will help to prune this space.

   The support threshold is a rule that the minimum frequency of an itemset in the transaction and it will help to prune the frequent itemsets. We only consider the itemsets that has higher frequency than the support threshold. The confidence threshold is a rule that the minimum accuracy of an association rule in the frequency itemsets and it will help to prune the association rules. We only consider the association rules that has higher accuracy than the confidence threshold.

   (iv) Which threshold will have a larger impact on the efficiency of the association rule algorithm?

   Confidence threshold will have a larger impact on the efficiency of the association rule algorithm.

3. (20 pts) Implement the Apriori association rule algorithm You only need to consider rules with a single variable in the consequent. Track how many frequent itemsets and association rules are discovered during the algorithm's search for patterns.

4. (10 pts) Apply the Apriori association rule algorithm to the yelp5.csv data. Use cutoff thresholds of *minsup* = 25% and *minconf* = 75%.

   (i) List the 20 discovered association rules with goodForGroups=1 or goodForGroups=0 as a consequent, and largest support values, to characterize the data. Report the rules themselves in an interpretable form (e.g., refer to the original attribute name and value), along with their numerical scores (i.e., support and confidence).

['casual']->['goodForGroups'] sup: 0.9663 conf: 0.9066
['open']->['goodForGroups'] sup: 0.8783 conf: 0.9084
['casual','open']->['goodForGroups'] sup: 0.8498 conf: 0.9071
['noise_average']->['goodForGroups'] sup: 0.6808 conf: 0.9237
['noise_average','casual']->['goodForGroups'] sup: 0.6627 conf: 0.9225
['waiterService']->['goodForGroups'] sup: 0.6296 conf: 0.9167
['casual','waiterService']->['goodForGroups'] sup: 0.6070 conf: 0.9154
['noise_average','open']->['goodForGroups'] sup: 0.6025 conf: 0.9236
['noise_average','casual','open']->['goodForGroups'] sup: 0.5873 conf: 0.9226
['open','waiterService']->['goodForGroups'] sup: 0.5525 conf: 0.9160
['casual','open','waiterService']->['goodForGroups'] sup: 0.5341 conf: 0.9152
['price_range_2']->['goodForGroups'] sup: 0.4919 conf: 0.9439
['caters']->['goodForGroups'] sup: 0.4882 conf: 0.9155
['price_range_2','casual']->['goodForGroups'] sup: 0.4880 conf: 0.9438
['casual','caters']->['goodForGroups'] sup: 0.4783 conf: 0.9145
['alcohol_none']->['goodForGroups'] sup: 0.4578 conf: 0.8436
['casual','alcohol_none']->['goodForGroups'] sup: 0.4564 conf: 0.8435
['price_range_1']->['goodForGroups'] sup: 0.4462 conf: 0.8647
['price_range_1','casual']->['goodForGroups'] sup: 0.4453 conf: 0.8644
['noise_average','waiterService']->['goodForGroups'] sup: 0.4369 conf: 0.9330

(ii) Given your prior knowledge of the patterns in this data from previous homeworks, discuss whether any of the discovered rules are *interesting*. Why or why not?

My prior knowledge is goodForGroups is associated with alcohol and my expectation is none-alcohol is not goodForGroups. However from the discovered rules, there is a rule ['alcohol_none']->['goodForGroups'] with 0.4462 support and 0.8644 confidence. This is interesting to me because this turns out my expectation is wrong.

5. (10 pts) Evaluate the efficiency of the algorithm.

Counting the number of frequent itemsets and associaion rules from k = 2.

(i) Report how many frequent itemsets ($|I|$) and association rules ($|R|$) are discovered when the algorithm is run on the full yelp5.csv data with cutoff thresholds of *minsup* = 25% and *minconf* = 75%.

143 frequent-itemsets and 227 associaiton rules with minsup 25% and minconf = 75%.

Byoungjin Oh
0030411229

(ii) Now keep *minconf* = 75% and rerun the algorithm while varying *minsup* = [10%,30%,50%]. Report the values of |*I*|,|*R*| for each of the different *minsup* values.

minconf = 75%

minsup	# frequent-itemsets	# association-rules
10%	710	1265
30%	87	144
50%	17	30

(iii) Now keep *minsup* = 25% and rerun the algorithm while varying *minconf* = [40%,60%,80%]. Report the values of |*I*|,|*R*| for each of the different *minconf* values.

minsup = 25%

minconf	# frequent-itemsets	# association-rules
40%	143	422
60%	143	318
80%	143	210