CS390DM Homework 6
Association Rules Report

Tingjun Li
li400@purdue.edu

1. Calculate the size of the pattern space.

Number of binary features I constructed for the Automobile data: 62

I treated high riskiness and low riskiness as two different features, and there are 60 other regular features.

Maximum possible frequent itemsets numbers: $\sum_{k=1}^{62}$C(62,k), length of k

Maximum possible association rules numbers size of k: $\sum_{k=2}^{62}\sum_{j=1}^{k-1}$C(62,k)C(k,j) , length of both size k, consequent size j

The support threshold can control the numbers of frequent itemsets. Since association rules are generated from the frequent itemsets, so the support threshold can not only limit the numbers of frequent itemsets as well as the generated association rules candidates numbers.

The confidence threshold can directly control the number of association rules, but it cannot control the number association rules candidates.

Comparing the two threshold, the support threshold has a larger impact on the efficiency of the association rule algorithm. This is because the support threshold control the frequent itemsets number and the generation of the association rules candidate.

2. Implement the Apriori association rule algorithm.

Output for my program.
Training set: the whole data set.
minsup: 0.25 minconf: 0.75

```
moore05 57 $ python association-rules.py train-set.dat 0.25 0.75
FREQUENT-ITEMS  2   70
FREQUENT-ITEMS  3   136
FREQUENT-ITEMS  4   129
FREQUENT-ITEMS  5   63
FREQUENT-ITEMS  6   16
FREQUENT-ITEMS  7   2
TOTAL FREQUENT-ITEMS  416
ASSOCIATION-RULES  2   64
ASSOCIATION-RULES  3   252
ASSOCIATION-RULES  4   373
ASSOCIATION-RULES  5   250
ASSOCIATION-RULES  6   81
ASSOCIATION-RULES  7   12
TOTAL ASSOCIATION-ITEMS  1032
```

3. Apply the Apriori association rule algorithm to the Automobile data.

20 association rules with high or low as a consequent:

```
support:  82 score:  0.92 two-door   =>  High Riskiness
```

support: 80 score: 0.93 gas two-door => High Riskiness
support: 79 score: 0.92 two-door front => High Riskiness
support: 77 score: 0.93 gas two-door front => High Riskiness
support: 70 score: 0.93 std two-door => High Riskiness
support: 68 score: 0.93 gas std two-door => High Riskiness
support: 67 score: 0.93 std two-door front => High Riskiness
support: 65 score: 0.93 gas std two-door front => High Riskiness
support: 64 score: 0.97 two-door four-cylinders => High Riskiness
support: 64 score: 0.97 two-door front four-cylinders => High Riskiness
support: 62 score: 0.97 gas two-door four-cylinders => High Riskiness
support: 62 score: 0.97 gas two-door front four-cylinders => High Riskiness
support: 60 score: 0.86 hatchback => High Riskiness
support: 60 score: 0.87 gas hatchback => High Riskiness
support: 60 score: 0.86 hatchback front => High Riskiness
support: 60 score: 0.87 gas hatchback front => High Riskiness
support: 57 score: 0.95 two-door hatchback => High Riskiness
support: 57 score: 0.9 two-door ohc => High Riskiness
support: 57 score: 0.95 gas two-door hatchback => High Riskiness
support: 57 score: 0.95 two-door hatchback front => High Riskiness

All of the rules discovered in this algorithm are interesting. They fitted our typical stereotype of high risk car (two-door). And some characters(gas, front engines) are because most cars shared that characters. They might not have to do with the high riskiness. The results also fitted our previous homework results. Therefore I think the result of this algorithm is decent.

4. Evaluate the stability of the constructed rule set.

Using randomized dataset, I got the following results:

support: 23 score: 0.96 two-door => High Riskiness
support: 23 score: 0.96 two-door front => High Riskiness
support: 22 score: 0.96 gas two-door => High Riskiness
support: 22 score: 0.96 gas two-door front => High Riskiness
support: 21 score: 0.95 std two-door => High Riskiness
support: 21 score: 0.95 std two-door front => High Riskiness
support: 20 score: 0.95 gas std two-door => High Riskiness
support: 20 score: 0.95 gas std two-door front => High Riskiness
support: 19 score: 0.76 gas fwd ohc => High Riskiness
support: 19 score: 0.76 gas fwd front ohc => High Riskiness
support: 18 score: 0.9 hatchback => High Riskiness
support: 18 score: 0.9 gas hatchback => High Riskiness
support: 18 score: 0.9 hatchback front => High Riskiness
support: 18 score: 0.9 gas hatchback front => High Riskiness
support: 17 score: 0.94 two-door hatchback => High Riskiness
support: 17 score: 0.94 two-door four-cylinders => High Riskiness
support: 17 score: 0.94 gas two-door hatchback => High Riskiness
support: 17 score: 0.94 two-door hatchback front => High Riskiness
support: 17 score: 0.94 two-door front four-cylinders => High Riskiness
support: 17 score: 0.94 gas two-door hatchback front => High Riskiness

As you can see from the randomized results, they are basically similar to the previous results. The top rules are all featuring the character “two-door”.

We can conclude that the stability of this algorithm is reasonably high.

5. Evaluate the efficiency of the algorithm.

minsup	minconf	|I|	|R|
25%	75%	416	1032
5%	75%	4928	14193
15%	75%	1136	3164
25%	75%	416	1032
35%	75%	145	321
45%	75%	60	139
25%	50%	416	1355
25%	65%	416	1176
25%	75%	416	1032
25%	85%	416	781
25%	95%	416	421