Markiyan Varhola

9016820171

Apriori Algorithm Implementation

The program starts out by reading the input file and placing each transaction into a list. From the list, the support count of each item is calculated and stored in a dictionary as a key-value pair, the key being the item ID and the value being the support count. The dictionary is then scanned then replaced with another dictionary only containing items that have a support count greater than the minimum support required. From this base dictionary, the candidate sets are generated until the max size of a candidate set has the maximum amount of possible unique items. The dictionary is then cleaned up by sorting the keys and therefore removing the duplicates ({8,1,3} and {1,8,3} become {1,3,8}).

After this, the set of rules are generated. I take advantage of the permutations function inside the Python itertools library to generate all permutations of item set → associative set rules. Each rule is then placed into the rule dictionary, with the key-value relationship containing the itemset as keys and associative set as rules.

Finally, for each rule in the dictionary, the script calculate the support and confidence and outputs the properly formatted lines to the output file.

Instructions on running script:

The script was tested on Python 2.7.3, and running it on Python 3> will result in an error. In order to run the script, use the format as outlined in the original assignment document.

To run the script, use the following format:

python apriori.py {minimum support} {input file} {output file}

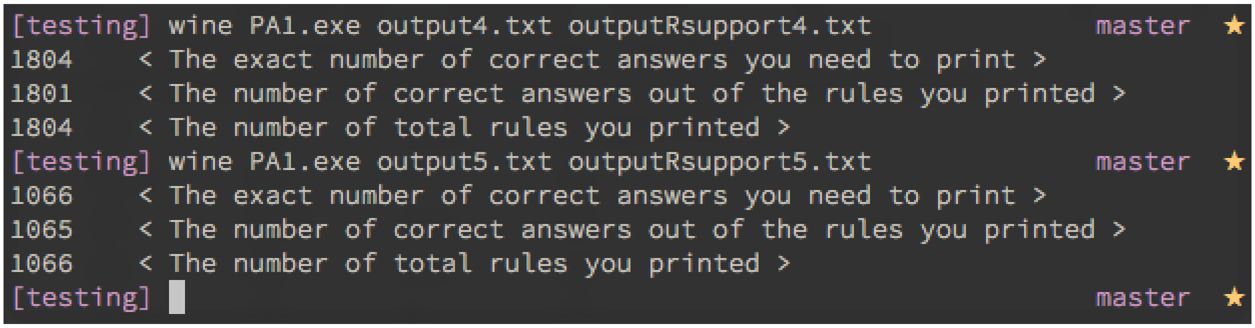
For example:

python apriori.py 5 input.txt output.txt

This command will produce a file called output.txt with the properly formatted data. Use output file as an input for the testing program to view the results.

Current script output:

**There is a bug in the script which misprints some of the lines, I really wish that there was more debug info in your testing program so I could fix the error.



```
[testing] wine PA1.exe output4.txt outputRsupport4.txt              master  ★
1804    < The exact number of correct answers you need to print >
1801    < The number of correct answers out of the rules you printed >
1804    < The number of total rules you printed >
[testing] wine PA1.exe output5.txt outputRsupport5.txt              master  ★
1066    < The exact number of correct answers you need to print >
1065    < The number of correct answers out of the rules you printed >
1066    < The number of total rules you printed >
[testing]                                                           master  ★
```