ID2222 Data Mining

Niklas Barth and Yu Gao

**Lab Report of Discovery of Frequent Itemsets and Association Rules**

**Abstract**

In this lab, we use Support (Support for itemset I is the number of baskets containing all items in I) to find frequent itemsets which with Support at least s in datasets. Then we find association rules between itemsets we find above.

**Code Structure**

**Class APrioriAlgorithm**

This class to implement the A-Priori algorithm for finding frequent itemsets with support at least s in a dataset of sales transactions. Support of an itemset is the number of transactions containing the itemset. Generally, there are three main functions in this Class: *createCandidates, getFrequentItems* and *runApriori.*

Function *createCandidates* is designed to build higher dimension sets based on low dimension sets. That is: if first k-1(k is the length of sets) elements between two low dimension sets are equal, merge the sets that have a shared intersection which creates all the candidates union sets without repeating.

Function *getFrequentItems* aims to calculate supports of itemsets and compare the result with support threshold s. If the support of an itemset is bigger than s, it means this itemset is frequent in the dataset.

Function *runApriori* is main function of this Class. It takes 1-dimension item as initial candidate for getFrequentItems test and call *createCandidates* to build high dimension itemsets based on frequent itemsets of last step. It return a list of all the frequent itemsets and ranged from low dimension to high dimension.

**Class AssociationRules**

This class solves the second sub-problem, i.e., develop and implement an algorithm for generating association rules between frequent itemsets discovered using the A-Priori algorithm in a dataset of sales transactions. The rules must have the support of at least s and confidence of at least c, where s and c are given as input parameters. Generally, there are three main functions in this Class: *getPartitioning, countsupport and getAssociationRules*

Function *getPartitioning* divides the frequent itemsets we get into a single list. For example, the frequent itemsets is [[{a}, {b}, {c}], [{a,b}, {a,c}]], we will get [{a}, {b}, {c}, {a,b}, {a,c}] as the new item list.

Function *countsupport* calculates the value of support for each itemset, but different from Function getFrequentItems in Class APrioriAlgorithm, it returns the value of support rather than itemset.

Function getAssociationRules is designed to find association rules between itemsets. Each itemset X will be paired with another itemset Y. If the intersection between them is null and the proportion of their union’s support to which of the first itemset is bigger than the value of confidence, we can say there is an association rule is an implication X → Y.

**Class TransactionCollection**

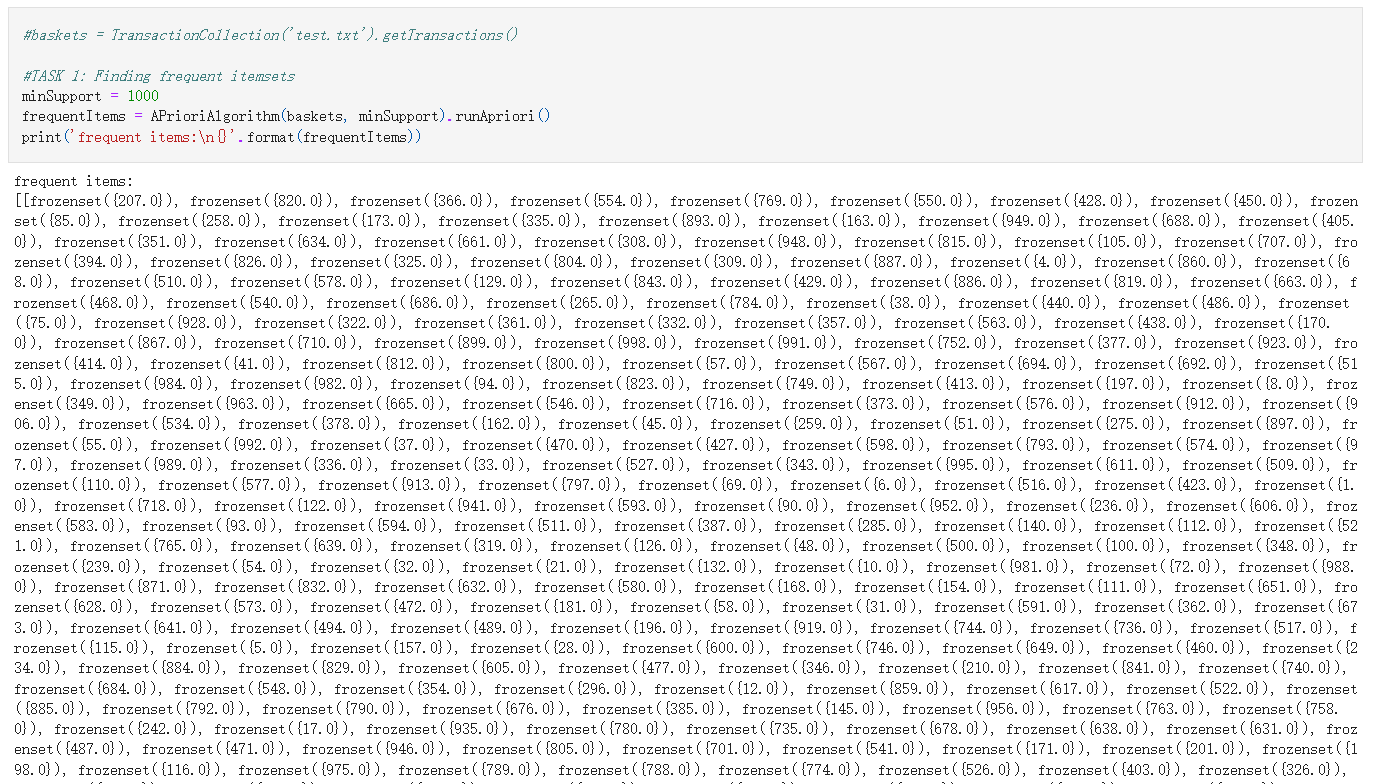
This Class is designed to read and load dataset files(e.g. csv) into our project. Itemsets in datasets will be read as single element of a list where the the number of elements equals to the number of itemset lines.

**Instructions for use**

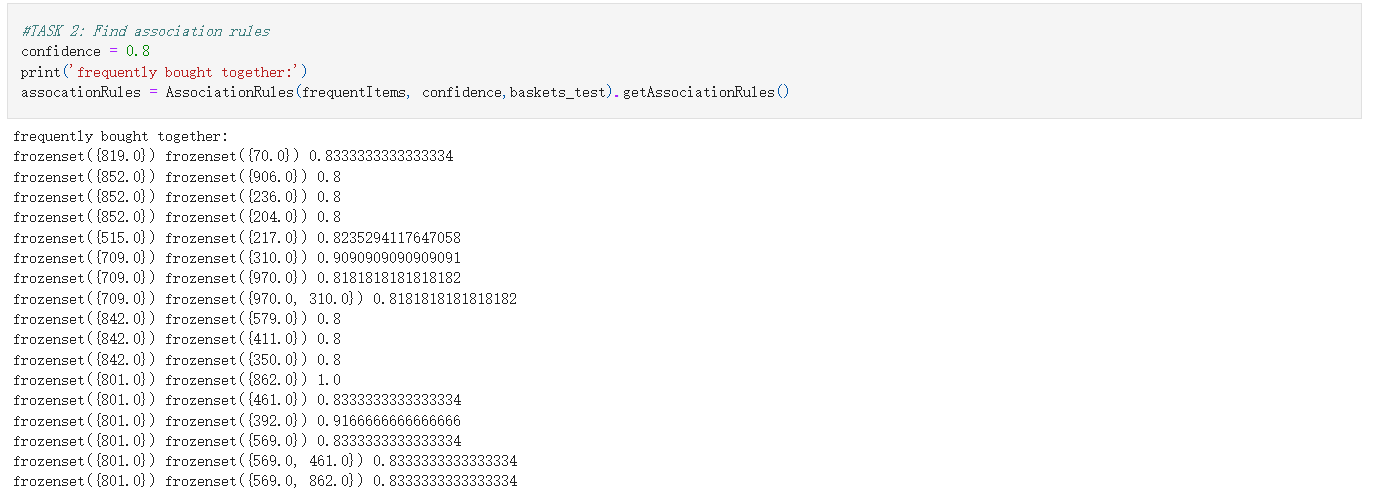
1. Put your test documents as a csv file within the same folder of project
2. Start the main class
3. Check the length of csv file you have and decide the number of documents you want to test
4. Decide the value of s, c
5. Run the main function

The Class APrioriAlgorithm would shows a list of all the frequent itemsets and ranged from low dimension to high dimension. While the Class AssociationRules will print the itemset pairs have association rules with the value of fraction of transactions containing X ⋃Y in all transactions that contain X.

**Result**



*Figure 1 Frequent Itemset*



*Figure 2 Itemset Pairs with association rules*