SYMBOLYC DATA MINING

László Kiss

December 2018

1 Introduction

Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for Boolean association rule. Name of algorithm is Apriori is because it uses prior knowledge of frequent itemset properties. We apply a iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.

2 The Algorithm

Input

The market base transaction dataset.

Procedure

1. The first pass of the algorithm counts item occurrences to determine large 1-itemsets.
2. This process is repeat until no new large 1-itemsets are identified.
3. (k+1) length candidate itemsets are generated from length k large itemsets.
4. Candidate itemsets containing subsets of length k that are are not large are pruned.
5. Support of each candidate itemset is counted by scanning the database.
6. Eliminate candidate itemsets that are small.

Output

Itemsets that are "large" and qualify the min support and min confidence thresholds

Candidate Set Generation Pruning

The biggest improvement in performance in the Apriori algorithm comes form reduction in candidate set generation. In the first pass all the large 1-itemsets are generated. For all the later passes only those itemsets are considered as candidate itemsets which were found to be large in the previous pass. The main idea is that a subset of a large itsemset would itself be large. Thus two generate large itemsets of size k, all that is required is to join itemsets of size (k-1). In this way a large number of itemsets do not have to be considered for generating candidate itemsets as was the case with previous algorithms.

3 Implementation

I have intentions to implement this algorithm as a command line application in python.
First of all I created a generator which able to make a proper input to an algorithm. It has three command line arguments number of rows, number of columns and density in this order. The output of this generator is a text file with rcf file exzensions. It contains the actual data set and some meta information about itself. The file called "Laszlo_Kiss_apriori_generator.py" and you can download from here:

Apriori input generator
Call this way : python Laszlo_Kiss_apriori_generator.py 10 8 0.6
where the first parameter - 10 - is the number of the rows and the second - 8 - number of the columns in the dataset. The third parameter is the density of this dataset.

The first step to realize the algorithm was the read the data from the output file. There is a "find_str" function in my application it can find the actual dataset in the input file. After this the dataset was put to a dataframe for further process and all of the possible candidates were made. In the next the application determined the frequency for all itemset with "count_freq" function. Finally the dataset was filtered by a min_sup value and we can see the set of the frequent itemset.

4 Example

Input dataset

	a	b	c	d	e
0	0	1	1	1	0
1	1	0	1	1	1
2	0	0	1	0	1
3	0	1	1	1	1
4	1	0	0	0	1

The frequent itemsets with minimum support of 3:

itemset	freq
c	4
d	3
e	4
cd	3
ce	3

The application generates an rcf file in the current directory with the results called valid_itemsets.rcf

The source code - file called LaszloKiss_apriori.py - can download from here:

Apriori in python

Call this way: python LaszloKiss_apriori.py out.rcf 3
where the first parameter - out.rcf - is the input dataset and the second - 3 - is the value of the minimum support.

5 References

http://www.columbia.edu/ rd2537/docu/apriori(abstract).pdf