Data Mining Homework 2: Discovery of Frequent Itemsets and Association Rules

Massimo Perini
Samuel Leonardo Gracio

November 2018

1 Short explanation of the program

Our program has been coded in python. At the beginning of the program, you have the different parameters that were asked for the assignment, which are : the threshold for the support length and the confidence c for the second-sub problem. The first part of the code is the implementation of the a-priory algorithm and the second solves the second-sub problem. There is no function or class, you can only change the value of the support, the confidence c and the data set. For the example, we used the "T10I4D100K.dat" available on Canvas. The program still have an important computing time despite the optimization, this is due to the number of combinations to compute the different subsets. Concerning the libraries, we just used "itertools" in order to make the different combinations.

2 Instructions to test the program

In order to run the code, you have different parameters :

- The "threshold" value which is the length of the support S .

- The "confidence_threshold" value which is the c value, i.e the level of confidence you want in order to validate a rule.

- The "path" where you have to put the path of the dataset you want to use, which is in this case, the "T10I4D100K.dat" dataset.

After changing this differents parameters, you just have to run the code and wait a few minutes to compute the differents combinations and different rules. The results will be printed at the end of the program.

3 Results

In order to print the results, we had to think about the values of the different parameters. In the lesson, it is clear that a confidence threshold of 0.5 is sufficient to know if a rule has an interest or not. As a consequence, we've decided to put c = 0.5 to print the results.

For the length of the support, we were wondering what value we could choose. In facts, it is recommended to use 1% of the length of the dataset but this will lead to a value of support = 3170. We've been testing this value and this a part of the results :

Results : frozenset('775'): 3771, frozenset('368'): 7828, frozenset('538'): 3982, frozenset('205'): 3605, frozenset('401'): 3667, frozenset('39'): 4258, frozenset('120'): 4973, frozenset('895'): 3385, frozenset('937'): 4681, frozenset('883'): 4902, frozenset('966'): 3921, frozenset('283'): 4082, frozenset('766'): 6265, frozenset('529'): 7057, frozenset('956'): 3626, frozenset('145'): 4559, frozenset('12'): 3415, frozenset('354'): 5835, frozenset('684'): 5408, frozenset('829'): 6810, frozenset('346'): 3470, frozenset('460'): 4438, frozenset('919'): 3710, frozenset('489'): 3420, frozenset('494'): 5102, frozenset('362'): 4388, frozenset('32'): 4248, frozenset('598'): 3219, frozenset('470'): 4137, frozenset('692'): 4993, frozenset('438'): 4511, frozenset('510'): 3281

What we can figure out with these results is that with 1% of the total length for the support value, it leads to a result with only singletons which has no interest to find some dependance rules. We've decided to reduce the value of support in order to have at least one triplet.

We've choosen support = 1000 in order to have one triplet. In fact, the last triplet is the following : frozenset('39', '704', '825'): 1035. //
This triplet is 39,704,825 and appears 1035 times.

These are the results for 1000 as a support value :

Results : [(frozenset('704'), frozenset('825'), 0.6142697881828316, 1102), (frozenset('704'), frozenset('39'), 0.617056856187291, 1107), (frozenset('227'), frozenset('390'), 0.577007700770077, 1049), (frozenset('704'), frozenset('39', '825'), 0.5769230769230769, 1035), (frozenset('39', '825'), frozenset('704'), 0.8719460825610783, 1035), (frozenset('39', '704'), frozenset('825'), 0.9349593495934959, 1035), (frozenset('825', '704'), frozenset('39'), 0.9392014519056261, 1035)]

These results shows that there a few rules that exists with a certain confidence.