README AND OUTPUT – Sahar Hajiseyednasir

Please note that I read my data.txt in a python module, and did all the discretization that is needed and then save my data again in data_discretization.csv file. This step took a while because of the large amount of data, therefore I used data_discretization.csv file for my Apriori and FP-growth algorithms.

The answers for both FP-growth and improved Apriori Algorithm are the same with the same minimum support value.

Run Apriori Algorithm Improved:

Please note that this is Apriori improved algorithm. As an improvement I saved all the supports for k-1 each subset, so that it will not calculate the supports over and over again.

1) Make sure that the data_discretization.csv file is in the same directory of the apriori_algorithm.py file
2) When the apriori_algorithm.py is open and it is run, it will ask you for the minimum support value
3) Then it will run and print the frequent item set and also the amount of run-time.

Run FP-growth Algorithm:

1) Make sure that the data_discretization.csv file is in the same directory of the fp-growth_algorithm.py file
2) When the fp-growth_algorithm.py is open and it is run, it will ask you for the minimum support value
3) Then it will run and print the frequent item set, the amount of run-time, and also the FP-tree which is nodes and edges.

OUTPUT: Result of running both algorithms with two different minimum supports:

As it can be seen FP-growth is much faster than Apriori Algorithm.

Run FP-growth Algorithm with minimum support of 22000:

Enter the minimum support: 22000
The run-time for FP-growth Algorithm is: 7.523008823394775

The Frequent items are:

The weighted nodes are:

[['capital-gain', 23116], ['education', 23115], ['age', 23115], ['capital-loss', 23115], ['hours-per-week', 23115], ['fnlwgt', 23115], ['education-num', 23115], ['marital-status', 23114], ['income', 23114], ['race', 23114], ['relationship', 23114], ['sex', 23114], ['native-country', 22701]]

The FP-Tree is:

The edges are:

{('education', 'age'), ('hours-per-week', 'fnlwgt'), ('marital-status', 'income'), ('race', 'relationship'), ('capital-loss', 'hours-per-week'), ('capital-gain', 'education'), ('income', 'race'), ('fnlwgt', 'education-num'), ('relationship', 'sex'), ('education-num', 'marital-status'), ('sex', 'native-country'), ('age', 'capital-loss')}

Run FP-growth Algorithm with minimum support of 3:

Enter the minimum support: 3
The run-time for FP-growth Algorithm is: 6.449647426605225

The Frequent items are:

The weighted nodes are:

[['capital-gain', 23116], ['education', 23115], ['age', 23115], ['capital-loss', 23115], ['hours-per-week', 23115], ['fnlwgt', 23115], ['education-num', 23114], ['marital-status', 23114], ['income', 23114], ['race', 23114], ['relationship', 23114], ['sex', 23113], ['native-country', 22700], ['workclass', 21819], ['occupation', 21817]]

The FP-Tree is:
The edges are:

{('education', 'age'), ('hours-per-week', 'fnlwgt'), ('marital-status', 'income'), ('race', 'relationship'), ('sex', 'workclass'), ('capital-loss', 'hours-per-week'), ('capital-gain', 'education'), ('native-country', 'workclass'), ('income', 'race'), ('fnlwgt', 'education-num'), ('relationship', 'sex'), ('education-num', 'marital-status'), ('education-num', 'workclass'), ('sex', 'native-country'), ('workclass', 'occupation'), ('age', 'capital-loss')}

Run Apriori Algorithm with minimum support of 22000:

Enter the minimum support: 22000
The Frequent Item Set is: {0: ['education', 'capital-loss', 'hours-per-week', 'fnlwgt', 'native-country', 'marital-status', 'race', 'income', 'education-num', 'capital-gain', 'age', 'relationship', 'sex']}
The run-time for Apriori Algorithm is: 95.14037036895752

Run Apriori Algorithm with minimum support of 3:

Enter the minimum support: 3
The Frequent Item Set is: {0: ['education', 'capital-loss', 'hours-per-week', 'fnlwgt', 'workclass', 'native-country', 'marital-status', 'occupation', 'income', 'education-num', 'relationship', 'capital-gain', 'age', 'race', 'sex']}
The run-time for Apriori Algorithm is: 398.98909521102905