CMPT 459 E100 Data Mining

Assignment 3

Report

Prof: Martin Ester

Student Name: Zeyong Jin

SFU ID: 301353174

Computing ID: zeyongj@sfu.ca

Date: April 12th, 2021

Contents

Part A Closed Itemsets

After implementing a method that takes the itemsets and returns a list of all closed itemsets, the running time is around 0.0909 seconds.

The first several lines of returned closed itemsets are as follows (generated by Google Colab):

```
{1: {('203729',): 1324,
  ('203733',): 755,
  ('253633',): 981,
  ('55267',): 3766,
  ('55271',): 2284,
  ('55283',): 2093,
  ('55287',): 1477,
  ('55291',): 2003,
  ('55295',): 1834,
  ('55307',): 1056,
  ('55315',): 2248,
  ('55319',): 2497,
  ('55323',): 3417,
  ('55327',): 2150,
  ('55335',): 1227,
  ('55339',): 1046,
  ('55343',): 1293,
  ('55347',): 787,
  ('55351',): 2249,
  ('55367',): 1296,
  ('55387',): 1608,
  ('55543',): 1434,
  ('55547',): 725,
  ('55551',): 1423,
  ('55555',): 786,
  ('55559',): 1009,
  ('55563',): 703,
  ('55831',): 1291,
  ('55835',): 953,
  ('55843',): 1032,
  ('55859',): 1027,
  ('55867',): 757,
  ('55871',): 1031,
```

Figure 1: First several lines of returned closed itemsets.

The number of closed frequent itemsets is 157.

The number of 1-itemsets is 43.

The number of 2-itemsets is 56.

The number of 3-itemsets is 49.

The number of 4-itemsets is 9.

Part B Maximal Itemsets

After implementing a method that takes the itemsets and returns a list of all closed itemsets, the running time is around 0.0711 seconds.

The first several lines of returned maximal itemsets are as follows (generated by Google Colab):

```
{1: {('197025',): 840,
  ('222439',): 882,
  ('222639',): 430,
  ('222643',): 452,
  ('228795',): 395,
  ('229179',): 388,
  ('239699',): 631,
  ('244119',): 880,
  ('244135',): 419,
  ('244251',): 495,
  ('244351',): 629,
  ('248237',): 457,
  ('250267',): 760,
  ('250271',): 1019,
  ('250283',): 507,
  ('250487',): 391,
  ('250515',): 400,
  ('250539',): 791,
  ('270057',): 966,
  ('271663',): 445,
  ('285525',): 761,
  ('305405',): 658,
  ('305409',): 464,
  ('305425',): 438,
  ('305505',): 526,
  ('306069',): 397,
  ('55275',): 1105,
  ('55331',): 775,
  ('55355',): 567,
  ('55363',): 609,
  ('55379',): 571,
  ('55403',): 416,
```

Figure 2: First several lines of returned maximal itemsets.

The number of maximal frequent itemsets is 251.

The number of 1-itemsets is 127.

The number of 2-itemsets is 64.

The number of 3-itemsets is 39.

The number of 4-itemsets is 19.

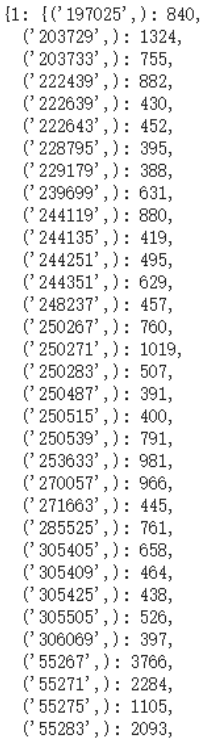
The number of 5-itemsets is 2.

Part C Statistics

1 Apriori Algorithm

Average Time Used: 80.2670 seconds.

The first several lines of returned all itemsets are as follows (generated by Google Colab):



```
{1: {('197025',): 840,
  ('203729',): 1324,
  ('203733',): 755,
  ('222439',): 882,
  ('222639',): 430,
  ('222643',): 452,
  ('228795',): 395,
  ('229179',): 388,
  ('239699',): 631,
  ('244119',): 880,
  ('244135',): 419,
  ('244251',): 495,
  ('244351',): 629,
  ('248237',): 457,
  ('250267',): 760,
  ('250271',): 1019,
  ('250283',): 507,
  ('250487',): 391,
  ('250515',): 400,
  ('250539',): 791,
  ('253633',): 981,
  ('270057',): 966,
  ('271663',): 445,
  ('285525',): 761,
  ('305405',): 658,
  ('305409',): 464,
  ('305425',): 438,
  ('305505',): 526,
  ('306069',): 397,
  ('55267',): 3766,
  ('55271',): 2284,
  ('55275',): 1105,
  ('55283',): 2093,
```

Figure 3: First several lines of returned all itemsets.

2 All Frequent Itemsets

The number of all frequent itemsets is 408.

The number of 1-itemsets is 170.

The number of 2-itemsets is 120.

The number of 3-itemsets is 88.

The number of 4-itemsets is 28.

The number of 5-itemsets is 2.

Average Time Used: 0.0488 seconds.

3 Closed Frequent Itemsets

The number of closed frequent itemsets is 157.

The number of 1-itemsets is 43.

The number of 2-itemsets is 56.

The number of 3-itemsets is 49.

The number of 4-itemsets is 9.

Average Time Used: 0.0909 seconds.

4 Maximal Frequent Itemsets

The number of maximal frequent itemsets is 251.

The number of 1-itemsets is 127.

The number of 2-itemsets is 64.

The number of 3-itemsets is 39.

The number of 4-itemsets is 19.

The number of 5-itemsets is 2.

Average Time Used: 0.0711 seconds.

5 Instruction

The values of average running times are based on the following configuration, different computer's values may differ:

Intel Core i7 – 7700 HQ @2.80 GHz with 16.0 GB RAM.

Part D Insights

According to the statistics before, it is obvious that the number of maximal frequent itemsets (251) is larger than the number of closed frequent itemsets (157). And the running time of maximal frequent itemsets mining and closed frequent itemsets mining are close.

Also, the number of all itemsets (408) equal to the sum of the number of closed itemsets (157) and the number of maximal itemsets (251).

Given that the minimum support is 0.005, which is relatively small, if we use all frequent item sets mining, the output has relatively a large number of itemsets.

From the statistics, it is clear that after using closed frequent itemsets mining and maximal frequent itemsets mining, the number of items gets significantly reduced.

And we can also indicate that the set of all maximal frequent itemsets is a subset of the set of all closed frequent itemsets. To be precise, maximal frequent itemsets efficiently provide a compact expression of frequent itemsets. In other words, the maximal frequent itemsets can form the smallest sets of itemsets of all frequent itemsets that can be derived.

Given that the dataset is from an e-commerce website, the space complexity of frequent itemsets may be O (2^n). In this case, it is impossible to find out all the frequent items, but maximal frequent itemsets provide a valuable expression. Maximal frequent itemsets are sufficient for store owners to determine all often-bought goods in this store if the minimum support and minimum confidence are reasonable.

Part E ReadMe

The algorithms of finding closed and maximal frequent itemsets are from the website: https://blog.csdn.net/u013007900/article/details/54743395.

The ideas of insights are from an online video, which is available on https://www.bilibili.com/video/BV1BA411e73h?p=28.

To run the codes, the following libraries are needed:

1. efficient_apriori

2. time

To execute the codes, you can just type “python 301353174.py” on the terminal or command line.

Thank you.

April 12, 2021