Programming Assignment 1

DSC 440 Data Mining

Meiying Chen

Oct. 17, 2019

Compare Performance between Apriori and FP-growth Algorithms

About Codes

```
Project Structure

|- main.py
|- dataset
  |- adult.data
|- fp_growth.py
|- fptree.py
|- apriori.py
|- dataloader.py
|- README.md

To run code, eval desired part of main.py
```

Overview

Apriori and FP-growth algorithms are both data mining methods for frequent set generation. But they are different in many aspects, and have their own strength and weakness in different situations.

In general, FP-growth is **about a magnitude faster** than Apriori and more widely used in frequent pattern detection. **Time complexity** is what most bothers Apriori. However, Apriori can be used to mine association rules, while it is not true for FP-growth.

The next following part will analyze them into more details.

Implemente Complexity

Commonly speaking, Apriori implementation is easier than FP-growth. The naive implementation is intuitive with the hardest part is just iterations. But in implementing vanilla FP-growth, you will need a Trie tree(Prefix tree) for the fp-tree generation and a linked list to keep track of the same item. FP-growth is basically some **prefix tree + a linked list**.

However, apart from what is written in our text book, there are some advanced improvement with Apriori using parallel matrix calculation and numpy[1]. There are hard to implement, but they speed Apriori algorithm to compare with FP-growth.

Algorithm(Vanilla Implement)	**Code Lines**
Apriori	98
FP-growth	147

[1] rasbt/mlxtend: https://github.com/rasbt/mlxtend

Data Distribution

Using **prefix tree** + a linked list, FP-growth could be viewed as **a special data structure** which **encoding files** and **compressed it into a smaller one**. Like all prefix tree algorithms ,PF-tree is a method **sacrificing space complexity to lower time complexity**. We can see from this understanding that:

1. If the dataset is **too large**, so that the fp-tree is too large to store in memory, fp-growth will not working as expected.
2. If the transactions has fewer in common, **the compressing** ratio will not be as good.

In other situations, FP-growth has a good change outperforming the Apriori.

Data Size

As Apriori need to go over the whole dataset every iteration, data size really matters when comparing these two methods. However, ways of implementation seems more influential.

Using the most advanced improvements, test on UCI dataset, these two algorithm achieved same result and their time consumption are as follows:

```
#transactions = 32561
min_sipport = 0.5
     support                                  itemsets
0   1.000000                                      ( 0)
1   0.759190                                  ( <=50K)
2   0.669205                                   ( Male)
3   0.697030                                ( Private)
4   0.895857                          ( United-States)
5   0.854274                                  ( White)
6   0.759190                               ( 0,  <=50K)
7   0.669205                                ( 0,  Male)
8   0.697030                             ( Private,  0)
9   0.895857                       ( 0,  United-States)
10  0.854274                                ( 0,  White)
11  0.544609                          ( Private,  <=50K)
12  0.675624                    ( <=50K,  United-States)
13  0.635699                             ( <=50K,  White)
14  0.598507                     ( Male,  United-States)
15  0.588864                              ( Male,  White)
16  0.618378                 ( Private,  United-States)
17  0.595928                           ( Private,  White)
18  0.786862                   ( United-States,  White)
19  0.544609                     ( Private,  0,  <=50K)
20  0.675624               ( 0,  <=50K,  United-States)
21  0.635699                        ( 0,  <=50K,  White)
22  0.598507                ( 0,  Male,  United-States)
23  0.588864                         ( 0,  Male,  White)
24  0.618378            ( Private,  0,  United-States)
25  0.595928                      ( Private,  0,  White)
26  0.786862                ( 0,  United-States,  White)
27  0.580971           ( <=50K,  United-States,  White)
28  0.542152            ( Male,  United-States,  White)
29  0.544455        ( Private,  United-States,  White)
30  0.580971     ( 0,  <=50K,  United-States,  White)
31  0.542152      ( 0,  Male,  United-States,  White)
32  0.544455  ( Private,  0,  United-States,  White)
```

Algorithm(Mostly Improved)	Time(s)
Apriori	7.9192399978637695
FP-growth	6.660172939300537

Hyperparameter Setting

For both algorithms, the larger the min support is, the less time it needs.

FP-growth is always faster than Apriori.

Also, as the min support grows bigger, the more FP-growth outperforms Apriori.

Algorithm(Mostly Improved)	Min support	Time(s)
FP-growth	0.1	7.5122649669647 22
Apriori	0.1	20.823474884033203
FP-growth	0.3	6.891494274139404
Apriori	0.3	7.337347745895386
FP-growth	0.5	6.660172939300537
Apriori	0.5	7.9192399978637695