Data Mining-HW1

P76061425 林聖軒

主程式: assoc_analysis.py

執行參數:

```
shengxuan@gpuserval-System-Product-Name:~/DM/DM_HW1$ py assoc_analysis.py -h
usage: assoc_analysis.py [-h] [-p POLICY] [-msp MIN_SUPPORT]
                         [-mcf MIN_CONFIDENCE] [-f DATA_FILE] [-ftp FILE_TYPE]

optional arguments:
  -h, --help           show this help message and exit
  -p POLICY            input policy,default=fpg,fpg=(fpGroeth),apr=apriori
  -msp MIN_SUPPORT     input min_support,default=3
  -mcf MIN_CONFIDENCE  input min_confidence,default=0.5
  -f DATA_FILE         input database,default=data.ntrans_1
  -ftp FILE_TYPE       input file type,default=1(IBM-Quest-Data-Generator)
```

Result compare:

Dataset: data.ntrans_1 (IBM Quest Data)
Number_of_transactions:1000

Minimum Support = 3 ,Minimum Confidence = 0.5

執行時間:
Apriori Algorithm:
```
real    1m50.782s
```
FP-Growth:
```
real    0m5.904s
```


Minimum Support = 4 ,Minimum Confidence = 0.5

執行時間:
Apriori Algorithm:
```
real    0m13.639s
```

FP-Growth:

```
real    0m1.349s
```

Minimum Support = 5 ,Minimum Confidence = 0.5

執行時間:

Apriori Algorithm:

```
real    0m0.947s
```

FP-Growth:

```
real    0m0.610s
```

     上面幾個例子可以很明顯的看到，FP-Growth 演算法的執行時間較少，效率明顯高於 Apriori Algorithm，而 minimum support 越低則需要越長的時間來找 association rule。

程式驗證:

與 weka 結果相同

```
Associator output

=== Run information ===

Scheme:     weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.4 -S
Relation:   fp_test_data1
Instances:  4
Attributes: 5
            a
            b
            c
            d
            e
=== Associator model (full training set) ===

FPGrowth found 14 rules

 1. [e=1]: 3 ==> [b=1]: 3    <conf:(1)> lift:(1.33) lev:(0.19) conv:(0.75)
 2. [b=1]: 3 ==> [e=1]: 3    <conf:(1)> lift:(1.33) lev:(0.19) conv:(0.75)
 3. [a=1]: 2 ==> [c=1]: 2    <conf:(1)> lift:(1.33) lev:(0.13) conv:(0.5)
 4. [e=1, c=1]: 2 ==> [b=1]: 2    <conf:(1)> lift:(1.33) lev:(0.13) conv:(0.5)
 5. [c=1, b=1]: 2 ==> [e=1]: 2    <conf:(1)> lift:(1.33) lev:(0.13) conv:(0.5)
 6. [e=1]: 3 ==> [c=1]: 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) conv:(0.38)
 7. [c=1]: 3 ==> [e=1]: 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) conv:(0.38)
 8. [c=1]: 3 ==> [b=1]: 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) conv:(0.38)
 9. [b=1]: 3 ==> [c=1]: 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) conv:(0.38)
10. [c=1]: 3 ==> [a=1]: 2    <conf:(0.67)> lift:(1.33) lev:(0.13) conv:(0.75)
11. [e=1]: 3 ==> [c=1, b=1]: 2    <conf:(0.67)> lift:(1.33) lev:(0.13) conv:(0.75)
12. [c=1]: 3 ==> [e=1, b=1]: 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) conv:(0.38)
13. [b=1]: 3 ==> [e=1, c=1]: 2    <conf:(0.67)> lift:(1.33) lev:(0.13) conv:(0.75)
14. [e=1, b=1]: 3 ==> [c=1]: 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) conv:(0.38)
```

```
shengxuan@gpuserval-System-Product-Name:~/DM/DM_HW1$ py assoc_analysis.py -p fpg -msp 2 -f example_data.txt
policy: fpg
min_support: 2
min_conf: 0.5
data_file: example_data.txt

frequent itemset:
1_itemset_count: 4
2_itemset_count: 4
3_itemset_count: 1
all_itemset_count: 9

association_list:
('b', 'c') -> [('e',)]
('c', 'e') -> [('b',)]
('b', 'e') -> [('c',)]
('c',) -> [('b', 'e'), ('a',), ('b',), ('e',)]
('b',) -> [('c', 'e'), ('c',), ('e',)]
('e',) -> [('c', 'b'), ('c',), ('b',)]
('a',) -> [('c',)]
found 14 rules
```