

科目：1071_資料探勘 DATA MINING

學生：楊沛霖

學號：Q36071156

日期：2018-10-23

Topic 1:

先以老師上課資料確認程式運行是否正確。

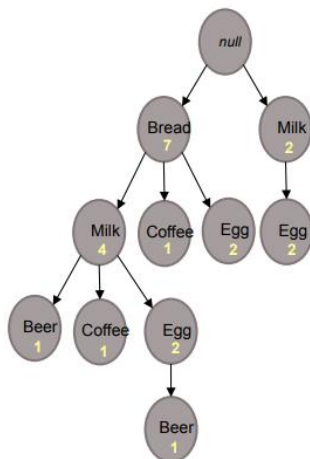
Min_sup : 2

執行時間：1.1854257

使用資料(IBM Quest Data Generator):

FP-growth:

老師講義上的圖(圖一)與這次程式實作出來的圖(圖二)



圖二

```
Null Set 1
  Bread 7
    Milk 4
      Beer 1
      Coffee 1
      Egg 2
        Beer 1
    Coffee 1
    Egg 2
      Milk 2
      Egg 2
```

圖一

Association analysis:

分別為自己寫的程式(圖 四)與 WEKA 的 FP-tree(圖 三)的關聯顯示,可以看到兩邊答案相同。

對應 function 為 start.py 中的 `associate(freqItems, simpDat)`

```
{'Coffee'} >>> {'Bread'}
conf : 1.0
{'Beer'} >>> {'Milk'}
conf : 1.0
{'Beer'} >>> {'Milk', 'Bread'}
conf : 1.0
{'Beer'} >>> {'Bread'}
conf : 1.0
{'Milk', 'Beer'} >>> {'Bread'}
conf : 1.0
{'Beer', 'Bread'} >>> {'Milk'}
conf : 1.0
{'Milk'} >>> {'Egg'}
conf : 0.6666666666666666
{'Milk'} >>> {'Bread'}
conf : 0.6666666666666666
{'Milk', 'Bread'} >>> {'Beer'}
conf : 0.5
{'Milk', 'Bread'} >>> {'Egg'}
conf : 0.5
{'Egg'} >>> {'Milk'}
conf : 0.6666666666666666
{'Egg'} >>> {'Bread'}
conf : 0.6666666666666666
{'Milk', 'Egg'} >>> {'Bread'}
conf : 0.5
{'Egg', 'Bread'} >>> {'Milk'}
conf : 0.5
{'Bread'} >>> {'Milk'}
conf : 0.5714285714285714
{'Bread'} >>> {'Egg'}
conf : 0.5714285714285714
rules_count : 16
```

圖 四

```
Scheme:      weka.associations.FPGrowth -P 2 -I -1 -N 30 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 2.0
Relation:     weka_IBM
Instances:    9
Attributes:   5
              Bread
              Milk
              Beer
              Coffee
              Egg

=== Associator model (full training set) ===

FPGrowth found 16 rules (displaying top 16)

1. [Coffee=T]: 2 ==> [Bread=T]: 2 <conf:(1)> lift:(1.29) lev:(0.05) conv:(0.44)
2. [Beer=T]: 2 ==> [Bread=T]: 2 <conf:(1)> lift:(1.29) lev:(0.05) conv:(0.44)
3. [Beer=T]: 2 ==> [Milk=T]: 2 <conf:(1)> lift:(1.5) lev:(0.07) conv:(0.67)
4. [Beer=T]: 2 ==> [Bread=T, Milk=T]: 2 <conf:(1)> lift:(2.25) lev:(0.12) conv:(1.11)
5. [Bread=T, Beer=T]: 2 ==> [Milk=T]: 2 <conf:(1)> lift:(1.5) lev:(0.07) conv:(0.67)
6. [Milk=T, Beer=T]: 2 ==> [Bread=T]: 2 <conf:(1)> lift:(1.29) lev:(0.05) conv:(0.44)
7. [Milk=T]: 6 ==> [Bread=T]: 4 <conf:(0.67)> lift:(0.86) lev:(-0.07) conv:(0.44)
8. [Egg=T]: 6 ==> [Bread=T]: 4 <conf:(0.67)> lift:(0.86) lev:(-0.07) conv:(0.44)
9. [Milk=T]: 6 ==> [Egg=T]: 4 <conf:(0.67)> lift:(1) lev:(0) conv:(0.67)
10. [Egg=T]: 6 ==> [Milk=T]: 4 <conf:(0.67)> lift:(1) lev:(0) conv:(0.67)
11. [Bread=T]: 7 ==> [Milk=T]: 4 <conf:(0.57)> lift:(0.86) lev:(-0.07) conv:(0.58)
12. [Bread=T]: 7 ==> [Egg=T]: 4 <conf:(0.57)> lift:(0.86) lev:(-0.07) conv:(0.58)
13. [Bread=T, Milk=T]: 4 ==> [Egg=T]: 2 <conf:(0.5)> lift:(0.75) lev:(-0.07) conv:(0.44)
14. [Bread=T, Egg=T]: 4 ==> [Milk=T]: 2 <conf:(0.5)> lift:(0.75) lev:(-0.07) conv:(0.44)
15. [Milk=T, Egg=T]: 4 ==> [Bread=T]: 2 <conf:(0.5)> lift:(0.64) lev:(-0.12) conv:(0.3)
16. [Bread=T, Milk=T]: 4 ==> [Beer=T]: 2 <conf:(0.5)> lift:(2.25) lev:(0.12) conv:(1.04)
```

圖 三

比較結果:

程式經過老師範例的洗禮過後,能正常運行並且和分析工具 WEKA 所顯示的資料相同,因此可以加入 IBM 資料去處理了。

Topic 2:

實際運行 IBM Quest Data Generator 所生成的資料

Min_sup : 2

執行時間 : 2.124941

使用資料(IBM Quest Data Generator):

使用參數 :

```
lit -ntrans 1 -ascii -tlen 5 -nitems 10 -npats 3
```

圖 五為 IBM Quest Data Generator ， 表一 為切換成老師投影片的形式。

1	1	247
1	1	506
1	1	578
2	2	247
2	2	506
2	2	578
3	3	247
3	3	506
4	4	247
4	4	506
4	4	578
5	5	247
5	5	506
5	5	578
6	6	247
6	6	506
6	6	578

圖 五

TID	ItemSet
1	{247,506,578}
2	{247,506,578}
3	{247,506,247}
4	{247,506,578}
5	{247,506,578}
6	{247,506,578}

表 一

FP-growth:

有程式生成的 FP-growth(圖 六)

```
[[ '247', '506', '578'], [ '247', '506', '578'], [ '247', '506'], [ '247', '506', '578'], [ '247', '506', '578']]
Null Set 1
  247 5
    506 5
      578 4
```

圖 六

Association analysis:

分別為自己寫的程式(圖 八)與 WEKA 的 FP-tree(圖 七)的關聯顯示,可以看到兩邊答案相同。

對應 function 為 start.py 中的 associate(freqItems, simpDat)

```
{'578'} >>> {'247'}
conf : 1.0
{'578'} >>> {'506'}
conf : 1.0
{'578'} >>> {'247', '506'}
conf : 1.0
{'506', '578'} >>> {'247'}
conf : 1.0
{'247', '578'} >>> {'506'}
conf : 1.0
{'247'} >>> {'578'}
conf : 0.8
{'247'} >>> {'506', '578'}
conf : 0.8
{'247'} >>> {'506'}
conf : 1.0
{'506'} >>> {'578'}
conf : 0.8
{'506'} >>> {'247', '578'}
conf : 0.8
{'506'} >>> {'247'}
conf : 1.0
{'247', '506'} >>> {'578'}
conf : 0.8
rules_count : 12
```

圖 八

```
=== Run information ===

Scheme:          weka.associations.FPGrowth -P 2 -I -1 -N 20 -T 0 -C 0.1 -D 0.05 -U 1.0 -M 0.1
Relation:         weka_IBM
Instances:        5
Attributes:       3
                  247
                  506
                  578

=== Associator model (full training set) ===

FPGrowth found 12 rules (displaying top 12)

1. [506=T]: 5 ==> [247=T]: 5    <conf:(1)> lift:(1) lev:(0) conv:(0)
2. [247=T]: 5 ==> [506=T]: 5    <conf:(1)> lift:(1) lev:(0) conv:(0)
3. [578=T]: 4 ==> [506=T]: 4    <conf:(1)> lift:(1) lev:(0) conv:(0)
4. [578=T]: 4 ==> [247=T]: 4    <conf:(1)> lift:(1) lev:(0) conv:(0)
5. [578=T]: 4 ==> [506=T, 247=T]: 4    <conf:(1)> lift:(1) lev:(0) conv:(0)
6. [506=T, 578=T]: 4 ==> [247=T]: 4    <conf:(1)> lift:(1) lev:(0) conv:(0)
7. [247=T, 578=T]: 4 ==> [506=T]: 4    <conf:(1)> lift:(1) lev:(0) conv:(0)
8. [506=T]: 5 ==> [578=T]: 4    <conf:(0.8)> lift:(1) lev:(0) conv:(0.5)
9. [247=T]: 5 ==> [578=T]: 4    <conf:(0.8)> lift:(1) lev:(0) conv:(0.5)
10. [506=T]: 5 ==> [247=T, 578=T]: 4    <conf:(0.8)> lift:(1) lev:(0) conv:(0.5)
11. [247=T]: 5 ==> [506=T, 578=T]: 4    <conf:(0.8)> lift:(1) lev:(0) conv:(0.5)
12. [506=T, 247=T]: 5 ==> [578=T]: 4    <conf:(0.8)> lift:(1) lev:(0) conv:(0.5)
```

圖 七

比較結果:

無論資料量大小皆與 WEKA 的分析結果一致,不過在自己 associate 的寫法中沒辦法像 WEKA 能夠把除了 True 或 False 的資料顯示出來,只能簡單的用存不存在去做對比。