# Data Mining HW1

## Task1

**1. What transaction you define?**

第一題我依照範例定義(PM2.5,Humidity,Temperature)作為 Transaction，但我所找的 device_id 為 74DA388FF5F6。

**2. What discretization methods you use?**

我所使用的方法為 divide by 10 跟 divide by 20。

**3. What algorithms you use?**

我使用了 apriori 以及 fp-growth。

**4. What rules you discover**

Apriori: (min_Support: 10% , min_Confidence: 50% )(**divide by 10**)

```
erature:10.0'], ['pm2.5:50', 'temperature:20.0'], ['humidity:70',
dity:70', 'temperature:10.0'], ['humidity:70', 'pm2.5:50'], ['humi
'], ['pm2.5:60', 'temperature:10.0'], ['humidity:90', 'temperature
--------------------------------------------------------------------
----------------------FREQUENT 3-ITEMSET-----------------------
[]
--------------------------------------------------------------------
---------------------ASSOCIATION RULES-----------------
RULES    SUPPORT%        CONFIDENCE%
-------------------------------------------------------
Rule#  1 : ['pm2.5:40'] ==> ['temperature:20.0'] 13 76
Rule#  2 : ['humidity:80'] ==> ['temperature:10.0'] 14 54
Rule#  3 : ['pm2.5:50'] ==> ['temperature:20.0'] 12 60
Rule#  4 : ['humidity:70'] ==> ['temperature:20.0'] 18 62
Rule#  5 : ['pm2.5:50'] ==> ['humidity:70'] 11 54
Rule#  6 : ['humidity:60'] ==> ['temperature:20.0'] 13 64
Rule#  7 : ['pm2.5:60'] ==> ['temperature:10.0'] 11 67
Rule#  8 : ['humidity:90'] ==> ['temperature:20.0'] 11 79
-------------------------------------------------------
```

Apriori: (min_Support: 10% , min_Confidence: 50% )(**divide by 20**)

```
[]
------------------------------------------------------------------
--------------------ASSOCIATION RULES-----------------
RULES    SUPPORT%        CONFIDENCE%
------------------------------------------------------------
Rule#  1 : ['pm2.5:40'] ==> ['temperature:20.0'] 26 68
Rule#  2 : ['humidity:80'] ==> ['temperature:20.0'] 23 57
Rule#  3 : ['pm2.5:0'] ==> ['humidity:80'] 10 84
Rule#  4 : ['pm2.5:60'] ==> ['temperature:0.0'] 12 56
Rule#  5 : ['pm2.5:40'] ==> ['humidity:60'] 24 62
Rule#  6 : ['temperature:20.0'] ==> ['humidity:60'] 32 52
Rule#  7 : ['humidity:60'] ==> ['temperature:20.0'] 32 63
Rule#  8 : ['temperature:0.0'] ==> ['humidity:60'] 19 50
Rule#  9 : ['pm2.5:60'] ==> ['humidity:60'] 15 68
Rule#  10 : ['pm2.5:20'] ==> ['temperature:20.0'] 10 57
Rule#  11 : ['humidity:60', 'temperature:20.0'] ==> ['pm2.5:40'] 17 54
Rule#  12 : ['humidity:60', 'pm2.5:40'] ==> ['temperature:20.0'] 17 73
Rule#  13 : ['pm2.5:40', 'temperature:20.0'] ==> ['humidity:60'] 17 67
------------------------------------------------------------
```

Fp-growth:(min_Support: 10% , min_Confidence: 50%)(**divide by 10**)

+1000:pm2.5

+2000:humidity

+3000:temperature

```
pm2.5:60 humidity:60 temperature:20.0
pm2.5:70 humidity:60 temperature:20.0

min_Support:  0.1  min_Condifance:  0.5
+1000:PM2.5
+2000:Humidity
+3000:Temperature

fp-growth algorithm
X, Y, Support counts, Confidence

(frozenset({1050}), frozenset({2070}), 166, 0.5424836601307189)
(frozenset({1060}), frozenset({3010}), 177, 0.6755725190839694)
(frozenset({2080}), frozenset({3010}), 213, 0.5419847328244275)
(frozenset({1040}), frozenset({3020}), 210, 0.7692307692307693)
(frozenset({1050}), frozenset({3020}), 186, 0.6078431372549019)
(frozenset({2060}), frozenset({3020}), 208, 0.6479750778816199)
(frozenset({2070}), frozenset({3020}), 283, 0.6206140350877193)
(frozenset({2090}), frozenset({3020}), 167, 0.7952380952380952)
```

Fp-growth:(min_Support: 10% , min_Confidence: 50%)(**divide by 20**)

+1000:pm2.5

+2000:humidity

+3000:temperature

```
+2000:Humidity
+3000:Temperature

fp-growth algorithm
X, Y, Support counts, Confidence

(frozenset({3020, 2060}), frozenset({1040}), 267, 0.5437881873727087)
(frozenset({1040, 2060}), frozenset({3020}), 267, 0.7355371900826446)
(frozenset({1040, 3020}), frozenset({2060}), 267, 0.6742424242424242)
(frozenset({1040}), frozenset({2060}), 363, 0.6269430051813472)
(frozenset({1060}), frozenset({2060}), 228, 0.6805970149253732)
(frozenset({1000}), frozenset({2080}), 164, 0.845360824742268)
(frozenset({1060}), frozenset({3000}), 189, 0.564179104477612)
(frozenset({3000}), frozenset({2060}), 286, 0.5053003533568905)
(frozenset({1020}), frozenset({3020}), 158, 0.572463768115942)
(frozenset({1040}), frozenset({3020}), 396, 0.6839378238341969)
(frozenset({2060}), frozenset({3020}), 491, 0.631917631917632)
(frozenset({3020}), frozenset({2060}), 491, 0.5245726495726496)
(frozenset({2080}), frozenset({3020}), 347, 0.5754560530679934)
```

5. **What you have learned, and do some comparisons between different methods you use.**

我同時使用兩個演算法(apriori, fp-growth)來跑同一筆資料，以驗證其正確性。由其中發現的 rules 我可以從中做一些總結跟推測。

一、首先是 divide by 10 時，我們可以從 support 來找最常出現的關係，其中最常出現的是：{濕度 70}=>{溫度 20} (support: 0.18)，代表此 sensor 溼度時常為 70~79 且溫度在 20~29 度之間。再看其 confidence，在濕度 70~79 的條件下有 62%的 transaction 的溫度在 20~29。 Divide by 20 時，資料較容易失真，因為 discretize 的區間較大，例如溫度大概都在十幾二十度附近，但是十幾度時如果 divide by 20 則溫度會跑到 0 度，與真實相差甚大。但是可以在 pm2.5 及濕度這種區間較大的看出大致的 rules，例如 pm2.5 在 0~19 的條件下，大致上濕度都在 80~99 的區間。

二、當資料做 discretization 時，資料會被重新分配區間，更多的數據會被集中到同一個區間，因此可以觀察到在同樣的 min_support 以及 min_confidence 之下，support 有相當大的提升，假設原本在溫度 0~9 有 5 筆資料，10~19 之間有 20 筆資料，在 20~29 之間有 10 筆資料，在 30~39 之間有 5 筆資料，各區間 support 分別為 12.5%、50%、25%、12.5%，但是在 divide by 20 後，在 0~19 有 62.5%，在 20~39 有 37.5%，support 明顯增加。

三、fp-growth 的優點在於不用產生 candidate，不用反覆掃資料，並將 transaction 建成一顆樹。而 apriori 則要產生 candidate，非常耗時間，且要一直掃全部的資料。理論上 fp-growth 會比 apriori 快。

**6. What dataset you use in this homework**

我使用了 before processing 的資料，因為 discretization 可以解決一些原本的問題，因此我使用 before processing 的資料來觀察 discretization 處理資料的效果。

## Task2

**1. What transaction you define?**

這次我選擇(PM2.5,PM1,PM10)來當作 transaction 的 column，device_id 則是跟 Task1 一樣為 74DA388FF5F6。

**2. What discretization methods you use?**

我使用 **divide by 10**、**qcut** 以及 **kmeans**。

qcut 是將 dataframe 照頻率切成區間。之後我再將其區間取平均做為值，放回 dataframe。

```
:
import pandas as pd
import numpy as np
from orangecontrib.recommendation import *
import pyfpgrowth

df = pd.read_csv('201703_Taiwan.csv')
df = df[df['device_id']=='74DA388FF5F6']
df = df.reset_index(drop=True)

a = pd.qcut(df['PM2.5'],10,labels=False,retbins=True)
b = pd.qcut(df['PM1'],10,labels=False,retbins=True)
c = pd.qcut(df['PM10'],10,labels=False,retbins=True)
#print(a[0])
print(a[1])
print(b[1])
print(c[1])

for i in range(0,a[0].size):
    index1 = a[0][i]
    index2 = b[0][i]
    index3 = c[0][i]
    df['PM2.5'][i] = (a[1][index1]+a[1][index1+1])/2
    df['PM1'][i] = (b[1][index2]+b[1][index2+1])/2
    df['PM10'][i] = (c[1][index3]+c[1][index3+1])/2
df
```

```
[   2.    16.1   30.    39.    44.    50.    56.    60.    64.    75.   167. ]
[   2.   14.   24.   29.   32.   36.   40.   43.   46.   66.  156.]
[   2.    17.1   32.    44.    53.    62.    69.    75.    81.    88.   173. ]
```

Kmeans 則是將所有的點分成 20 群，並且我將它們都改成其 mean 的 center 坐標，因此只會有 20 種不同的結果且這 20 條 confidence 都會是 100，但是其 support 則要看靠近的點有多少個。

```
from sklearn.cluster import KMeans
import numpy as np
import pandas as pd
df = pd.read_csv('201703_Taiwan.csv')
df = df[df['device_id']=='74DA388FF5F6']
df = df.reset_index(drop=True)
df = df[['PM2.5','PM1','PM10']]
X = df.as_matrix()

kmeans = KMeans(n_clusters=20, random_state=0).fit(X)
kmeans.labels_

for i in range(0,len(df)):
    tmp = kmeans.labels_[i] #0~9
    df['PM2.5'][i] = kmeans.cluster_centers_[tmp][0]
    df['PM1'][i] = kmeans.cluster_centers_[tmp][1]
    df['PM10'][i] = kmeans.cluster_centers_[tmp][2]


print(kmeans.cluster_centers_)
print(df)
for j in df.index:
    print('pm2.5:',df['PM2.5'][j],' pm1:',df['PM1'][j],' pm10:',df['PM10'][j],sep='')
```

```
[[  52.59722222   37.59027778   65.57638889]
 [ 144.          133.          148.42857143]
 [  19.16883117   16.07792208   20.72727273]
 [  92.45         79.55         101.4       ]
```

### 3. What algorithms you use?

apriori 以及 fp-growth。

### 4. What rules you discover

Apriori:(min_support: 10%, min_confidence: 90%)(**divide by 10**)

```
]]
------------------------------------------------------------
----------------------FREQUENT 3-ITEMSET--------------------
[['pm10:50', 'pm1:30', 'pm2.5:40']]
------------------------------------------------------------
----------------------FREQUENT 4-ITEMSET--------------------
[]
------------------------------------------------------------
---------------------ASSOCIATION RULES----------------------
RULES    SUPPORT%        CONFIDENCE%
------------------------------------------------------------
Rule#  1 : ['pm2.5:30'] ==> ['pm1:20'] 10 91
Rule#  2 : ['pm2.5:40'] ==> ['pm1:30'] 17 96
Rule#  3 : ['pm10:50'] ==> ['pm2.5:40'] 12 99
Rule#  4 : ['pm10:50'] ==> ['pm1:30'] 12 98
Rule#  5 : ['pm10:70'] ==> ['pm1:40'] 14 90
Rule#  6 : ['pm10:50'] ==> ['pm1:30', 'pm2.5:40'] 12 98
Rule#  7 : ['pm10:50', 'pm1:30'] ==> ['pm2.5:40'] 12 100
Rule#  8 : ['pm10:50', 'pm2.5:40'] ==> ['pm1:30'] 12 99
------------------------------------------------------------
```

Fp-growth:(min_Support: 10% , min_Confidence: 90%)(**divide by 10**)

+1000:pm2.5

+2000:pm1

+3000:pm10

```
pm2.5:60 pm1:40 pm10:70
pm2.5:70 pm1:50 pm10:80

min_Support:  0.1  min_Condifance:  0.9
+1000:PM2.5
+2000:PM1
+3000:PM10

fp-growth algorithm
X, Y, Support counts, Confidence

(frozenset({3050, 2030}), frozenset({1040}), 184, 1.0)
(frozenset({1040, 3050}), frozenset({2030}), 184, 0.9945945945945946)
(frozenset({3050}), frozenset({1040, 2030}), 184, 0.989247311827957)
(frozenset({1030}), frozenset({2020}), 157, 0.9181286549707602)
(frozenset({1040}), frozenset({2030}), 264, 0.967032967032967)
(frozenset({3050}), frozenset({1040}), 185, 0.9946236559139785)
(frozenset({3050}), frozenset({2030}), 184, 0.989247311827957)
(frozenset({3070}), frozenset({2040}), 219, 0.9087136929460581)
```

Apriori:(min_support: 10%, min_confidence: 90%)(**qcut into 10 part**)

```
----------------------------------------------------
----------------------FREQUENT 1-ITEMSET------------
[['pm1:26'], ['pm1:34'], ['pm10:57'], ['pm2.5:53'],
'pm1:19'], ['pm2.5:23'], ['pm10:24'], ['pm2.5:9'], [
----------------------------------------------------
----------------------FREQUENT 2-ITEMSET------------
[['pm10:24', 'pm2.5:23'], ['pm10:9', 'pm2.5:9']]
----------------------------------------------------
----------------------FREQUENT 3-ITEMSET------------
[]
----------------------------------------------------
---------------------ASSOCIATION RULES--------------
RULES     SUPPORT%         CONFIDENCE%
----------------------------------------------------
Rule#  1 : ['pm10:24'] ==> ['pm2.5:23'] 10 98
Rule#  2 : ['pm2.5:23'] ==> ['pm10:24'] 10 95
Rule#  3 : ['pm2.5:9'] ==> ['pm10:9'] 10 100
Rule#  4 : ['pm10:9'] ==> ['pm2.5:9'] 10 100
----------------------------------------------------
```

Fp-growth: (min_support: 10%, min_confidence: 90%)(**qcut into 10 part**)

```
min_Support:  0.1  min_Condifance:  0.9
+1000:PM2.5
+2000:PM1
+3000:PM10

fp-growth algorithm
X, Y, Support counts, Confidence

(frozenset({3009}), frozenset({1009}), 151, 1.0)
(frozenset({1009}), frozenset({3009}), 151, 1.0)
(frozenset({1023}), frozenset({3024}), 152, 0.9559748427672956)
(frozenset({3024}), frozenset({1023}), 152, 0.9806451612903225)
```

Apriori:(min_support: 10%, min_confidence: 100%)(**kmeans**)

```
Rule#  1 : ['pm1:41'] ==> ['pm2.5:57'] 10 100
Rule#  2 : ['pm2.5:57'] ==> ['pm1:41'] 10 100
Rule#  3 : ['pm10:72'] ==> ['pm2.5:57'] 10 100
Rule#  4 : ['pm2.5:57'] ==> ['pm10:72'] 10 100
Rule#  5 : ['pm10:72'] ==> ['pm1:41'] 10 100
Rule#  6 : ['pm1:41'] ==> ['pm10:72'] 10 100
Rule#  7 : ['pm2.5:61'] ==> ['pm1:44'] 11 100
Rule#  8 : ['pm1:44'] ==> ['pm2.5:61'] 11 100
Rule#  9 : ['pm2.5:61'] ==> ['pm10:78'] 11 100
Rule#  10 : ['pm10:78'] ==> ['pm2.5:61'] 11 100
Rule#  11 : ['pm10:78'] ==> ['pm1:44'] 11 100
Rule#  12 : ['pm1:44'] ==> ['pm10:78'] 11 100
Rule#  13 : ['pm10:72'] ==> ['pm1:41', 'pm2.5:57'] 10 100
Rule#  14 : ['pm1:41'] ==> ['pm10:72', 'pm2.5:57'] 10 100
Rule#  15 : ['pm2.5:57'] ==> ['pm10:72', 'pm1:41'] 10 100
Rule#  16 : ['pm10:72', 'pm2.5:57'] ==> ['pm1:41'] 10 100
Rule#  17 : ['pm10:72', 'pm1:41'] ==> ['pm2.5:57'] 10 100
Rule#  18 : ['pm1:41', 'pm2.5:57'] ==> ['pm10:72'] 10 100
Rule#  19 : ['pm2.5:61'] ==> ['pm10:78', 'pm1:44'] 11 100
```

5. **What you have learned, and do some comparisons between different methods you use.**

一、使用 qcut 做 discrtization 時，在 pm2.5 為 9 時 pm10 的值 100%為 9，且 pm10 為 9 時 pm2.5 的值也 100%為 9。在這組資料中，他們 100%伴隨彼此出現。

二、使用 divide by 10 的作為 discretization 時，在 pm10 為 50 且 pm1 為 30 的情況下，pm2.5 的值 100%為 40，但在 pm10 為 50 且 pm2.5 為 40 的情況下，pm1 的值 99%為 30。由此可見，雖然有相同的 support 但是 confidence 卻不一定相同。

三、使用 qcut 再取上下界做平均的 discretization 方法相較於使用 divide by 10，qcut 所切出來的值比較靠近真實的值。

四、使用 qcut 做為 discretization，算出來 rules 的 support 幾乎都在 9 點多跟 10 點多之間，非常均勻，數值比較不同的是 confidence。

五、使用 kmeans 的 confidence 都會是 100%是因為我使用中心當作一組的數值，但是靠近每個中心的數目不一樣，所以 support 會有差異。

6. **What dataset you use in this homework**

我使用的是 before processing 的資料。

## Task3

1. **What transaction you define?**

Columns: (PM1,PM10,Humidity)，device_id = '74DA388FF5F6'。

2. **What discretization methods you use?**

qcut(into 5 part)，kmean(into 15 cluster)。

3. **What algorithms you use?**

Apriori and fp-growth

4. **What rules you discover**

Apriori:(min_support: 10%, min_confidence: 90%)(qcut into 5 part)

```
----------------------FREQUENT 2-ITEMSET--------------------
[['pm10:42', 'pm1:28'], ['humidity:94', 'pm1:13'], ['humidity
1:43'], ['pm10:127', 'pm1:101'], ['humidity:60', 'pm10:42'],
------------------------------------------------------------
----------------------FREQUENT 3-ITEMSET--------------------
[['humidity:94', 'pm10:17', 'pm1:13']]
------------------------------------------------------------
----------------------FREQUENT 4-ITEMSET--------------------
[]
------------------------------------------------------------
---------------------ASSOCIATION RULES-----------------
RULES    SUPPORT%        CONFIDENCE%
------------------------------------------------------
Rule#  1 : ['pm10:42'] ==> ['pm1:28'] 18 92
Rule#  2 : ['pm1:28'] ==> ['pm10:42'] 18 93
Rule#  3 : ['pm1:13'] ==> ['pm10:17'] 19 96
Rule#  4 : ['pm10:17'] ==> ['pm1:13'] 19 97
Rule#  5 : ['humidity:94', 'pm1:13'] ==> ['pm10:17'] 10 98
Rule#  6 : ['humidity:94', 'pm10:17'] ==> ['pm1:13'] 10 96
------------------------------------------------------
```

Fp-growth: (min_support: 10%, min_confidence: 90%)(qcut into 5 part)

```
pm1:43 pm10:75 humidity:60
pm1:101 pm10:75 humidity:60
pm1:101 pm10:75 humidity:60
pm1:101 pm10:127 humidity:60

min_Support:  0.1  min_Condifance:  0.9
+1000:PM1
+2000:PM10
+3000:Humidity

fp-growth algorithm
X, Y, Support counts, Confidence

(frozenset({1013, 3094}), frozenset({2017}), 159, 0.9814814814814815)
(frozenset({2017, 3094}), frozenset({1013}), 159, 0.9695121951219512)
(frozenset({1013}), frozenset({2017}), 297, 0.9674267100977199)
(frozenset({2017}), frozenset({1013}), 297, 0.9705882352941176)
(frozenset({1028}), frozenset({2042}), 276, 0.9324324324324325)
(frozenset({2042}), frozenset({1028}), 276, 0.9261744966442953)
```

Apriori:(min_support: 10%, min_confidence: 100%)(**kmeans into 15 clusters**)

```
----------------------------------------------------------
--------------------ASSOCIATION RULES-------------------
RULES     SUPPORT%         CONFIDENCE%
----------------------------------------------------------
Rule#  1 : ['pm10:68'] ==> ['pm1:39'] 13 100
Rule#  2 : ['pm1:39'] ==> ['pm10:68'] 13 100
Rule#  3 : ['pm1:39'] ==> ['humidity:77'] 13 100
Rule#  4 : ['pm10:68'] ==> ['humidity:77'] 13 100
Rule#  5 : ['pm1:44'] ==> ['humidity:70'] 12 100
Rule#  6 : ['pm10:77'] ==> ['humidity:70'] 12 100
Rule#  7 : ['pm10:77'] ==> ['pm1:44'] 12 100
Rule#  8 : ['pm1:44'] ==> ['pm10:77'] 12 100
Rule#  9 : ['pm10:68'] ==> ['humidity:77', 'pm1:39'] 13 100
Rule#  10 : ['pm1:39'] ==> ['humidity:77', 'pm10:68'] 13 100
Rule#  11 : ['humidity:77', 'pm1:39'] ==> ['pm10:68'] 13 100
Rule#  12 : ['humidity:77', 'pm10:68'] ==> ['pm1:39'] 13 100
Rule#  13 : ['pm10:68', 'pm1:39'] ==> ['humidity:77'] 13 100
Rule#  14 : ['pm10:77'] ==> ['humidity:70', 'pm1:44'] 12 100
Rule#  15 : ['pm1:44'] ==> ['humidity:70', 'pm10:77'] 12 100
Rule#  16 : ['humidity:70', 'pm1:44'] ==> ['pm10:77'] 12 100
```

5. **What you have learned, and do some comparisons between different methods you use.**

一、當區間數值間隔狹小時，如果切太多塊，會造成 support 大幅下降，因此這次資料才會採取 qcut 成 5 個部分。

二、在 qcut 作為 discretizaion 時，pm1 的值時常為 13，且當 pm1 為 13 時，pm10 有非常高的機率為 17。而 pm10 也時常為 17 且當 pm10 為 17 時，pm1 有非常高機率為 13。且當濕度為 94 時，上述這些規則也是符合的。

三、在做 kmean 時也跟 qcut 一樣，要是 cluster 數目太多時，則會造成 support 大大的降低，但 cluster 太少時，也會造成資料失真嚴重。

6. **What dataset you use in this homework**

我使用的是 before processing 的資料。

Reference:


1. https://github.com/biolab/orange3-associate
2. https://github.com/nalinaksh/Association-Rule-Mining-Python