CIS735 - Homework 1-2

Kai Li

1. Each instance has 4 attributes, the first attribute is the 'file size' with integer type, the second is document type with categorical type, the third is 'possibility of change' with a scale from un-likely to definitely likely, and the last is file attribute with categorical type. For each attribute, the distance function will be the following:

   File size: Normalized Euclidean Distance
   Document type: Gower's generalized similarity
   Possibility of change: Normalized rank transform
   File attribute: Gower's generalized similarity

   Based on the above distance measures, the distance of the two instances to XYZ file template is:

   Distance(instance1, XYZ) = $\sqrt{(54k - 52k/52k)^2 + 1^2 + 0^2 + 0^2}$ = 1.00074

   Distance(instance2, XYZ) = $\sqrt{(52k - 5k/52k)^2 + 1^2 + (1/4)^2 + 1^2}$ = 1.6969

   The distance results show instance1 is more like XYZ file template, while instance 2 is not.

2. The distance measures are as follows.

   Cosine: (0*5 + 1*0 + 0*1 + 1*0 + 2*2 + 1*2) / $\sqrt{(1 * 1 + 1 * 1 + 2 * 2 + 1 * 1)}$ * $\sqrt{(5 * 5 + 1 * 1 + 2 * 2 + 2 * 2)}$ = 5 / (2.646*5.831) = 0.324

   Correlation: avg(x) = 5/6, avg(y) = 5/3

   $$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad \text{(Eq.3)}$$

   = -0.33

   Euclidean: $\sqrt{(5 * 5 + 1 * 1 + 1 * 1 + 1 * 1 + 0 + 1 * 1)}$ = 5.385

   Jacckard: |{0,1,2}| / |{0,1,2,5}| = ¾ = 0.75

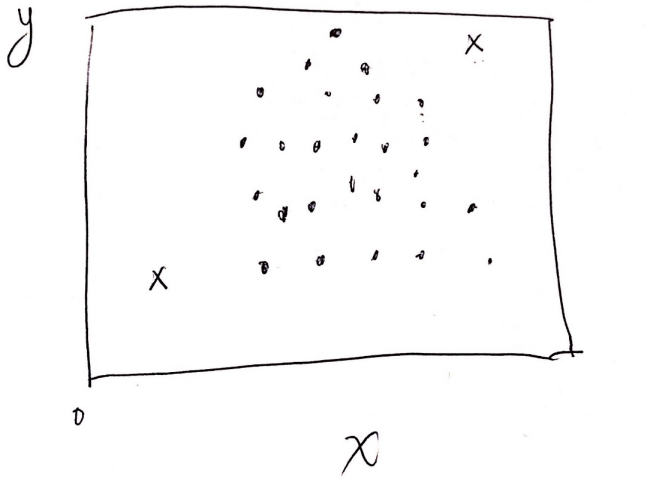
   Cosine distance doesn't satisfy the triangle inequality. Suppose A = (0,1), B = ($\sqrt{2}$/2, $\sqrt{2}$/2), and C = (1,0). Then we have:

   cos-sim(A,C) = 0, cos-sim(A,B) = $\sqrt{2}$/2, and cos-sim(B,C) = $\sqrt{2}$/2. While, according to the triangle inequality,

1+cos(A,C)>cos-sim(A,B)+cos-sim(B,C). However, in the given example, 1+0 < $\sqrt{2}$/2 + $\sqrt{2}$/2.

3. Features to classify computer files that have been altered can be: size in bytes, width, height, file type, date when created, date when last modified, date when last accessed. Features like size, type, width, and the height of a file are correlated(i.e they vary together), as width and height increase, the size of the file also increases.

4. Conside the following diagram, where multiple data points are plotted into the X-Y two dimensions. The outlier are identified as the two crossed data points. However, if we only consider the Y-Y dimension, the two outlier data points will be classified as inlier.

5. 

distance	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	**0**	5	7.21	3.61	5.83	5	8.06	2.24	**1**
x4	3.61	4.24	**6.71**	**0**	**3**	**1.41**	7.21	**1.41**	3.16
x7	8.06	**3.16**	13.89	7.21	5	7.07	**0**	7.62	7.07

The highlighted cell is the centroids that the data point is closest to. Therefore, the three clusters are {X1, X9}, {X3, X4, X5, X6, X8} and {X2, X7}.

6. The code to compute the Pearson coefficient between each feature and the User is presented as follows.

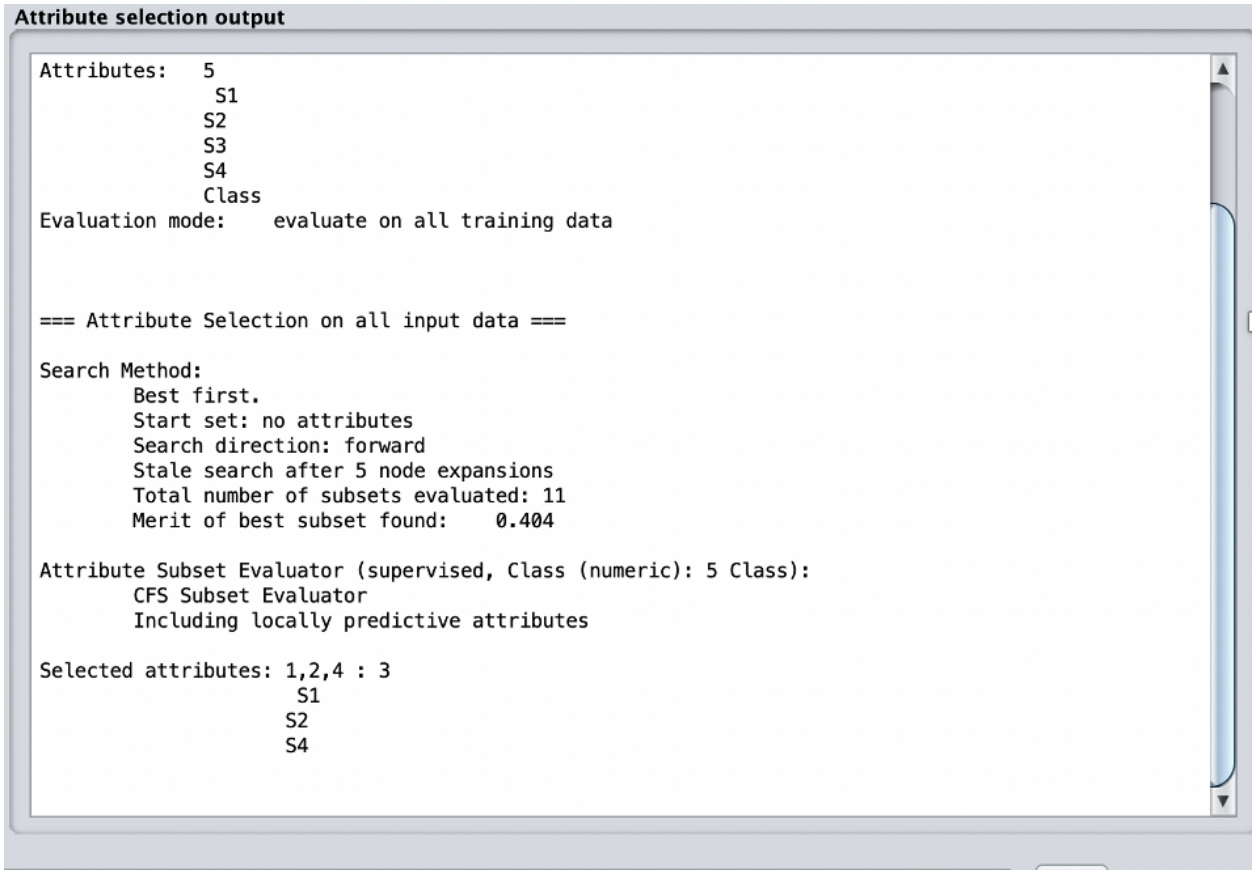
```
from scipy import stats
import numpy as np

F1=[2.3, 5.8, 5.7, 5.4, 5.1, 5.7, 7.0, 6.4, 6.9, 5.5, 5.8, 6.7, 6.7]
F2=[3.0, 4.0, 4.4, 3.5, 3.5, 3.8, 3.2, 3.2, 3.1, 2.3, 3.2, 3.3, 3.0]
F3=[3.1, 1.2, 1.2, 1.3, 1.4, 1.7, 4.7, 4.5, 4.9, 4.0, 2.9, 5.7, 5.2]
F4=[7.0, 3.0, 2.0, 1.5, 1.9, 2.0, 2.0, 3.0, 3.9, 2.3, 4.8, 5.0, 3.3]
F5=[0.1, 0.2, 0.4, 0.4, 0.3, 0.3, 1.4, 1.5, 1.5, 1.3, 2.3, 2.5, 2.3]
Users=[1,1,1,1,1,1,2,2,2,2,3,3,3]

Features=[F1, F2, F3, F4, F5]

index = 1
for feature in Features:
    print(index, stats.pearsonr(feature, Users))
    index +=1
```

The output is:

```
(1, (0.5377607223655454, 0.05802370018411562 3))
(2, (-0.5120315616476586 2, 0.07364014061787765 9))
(3, (0.7953491458323757, 0.00115184243715515 21))
(4, (0.3342447616968408 6, 0.2643375818167420 5))
(5, (0.9919970512715488, 3.078220120986875 6e-11))
```

Pearson coefficient varies between -1 and +1, with 0 implying no correlation. Correlations of -1 or +1 imply an exact linear relationship. The result shows, among the five features, F5, F3 and F1 has the largest absolute value, which indicates these three features have a much higher relevance to the user than the other two features (F2, F4).

7. For this task, I refer to Weka to choose the two most important features. The result shows S1 and S2 are selected. Weka supports correlation based feature selection with the CorrelationAttributeEval technique that requires use of a Ranker search method. The key idea is to compute the pearson coefficient of each sensor to the target class and rank each sensor's coefficient (in absolute), from which, the two sensors that have the highest value are selected.


Attribute selection output
Attributes: 5
S1
S2
S3
S4
Class
Evaluation mode: evaluate on all training data
=== Attribute Selection on all input data ===
Search Method:
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 11
Merit of best subset found: 0.404
Attribute Subset Evaluator (supervised, Class (numeric): 5 Class):
CFS Subset Evaluator
Including locally predictive attributes
Selected attributes: 1,2,4 : 3
S1
S2
S4