Homework 2

Pattern Mining and Social Network Analysis

BOUYSSOU Gatien , de POURTALES Caroline, LAMBA Ankit

21 octobre, 2020

Contents

Continuous

Continuous data is data that can take any value while discrete data can take only certain values. with continuous (distance/similarity based) : Silhouette, Dunn, …

Silhouette coefficient

The Silhouette coefficient evaluates the performance of your clustering model on a dataset. This coefficient can take 3 value :

- 1 it means that the cluster is far away from its neighbours
- 0 indicates that it is close from one or multiple clusters
- -1 or negative values in general means that the cluster is allocated to the wrong values

It is possible to compute this coefficient thanks to the following formula :

$$Silhouette\ Score = (b - a)/max(a, b)$$

Where :

- a is the distance between each point within a cluster
- b is the distance between all the clusters

Dunn index

The dunn index is the min distance between two clusters (separation) over the max distance btw the objects of one clusters (diameter).

$$Dunn\ index = \frac{min.separation}{max.diameter}$$

Discrete

with discrete (binary, graph based) : modularity, C measure, ...

Principal Components Analysis

The goal of PCA is to identify which features in the dataset explain the most variability.

Different kinds of PCA

Standard PCA

Incremental PCA

Sparse PCA

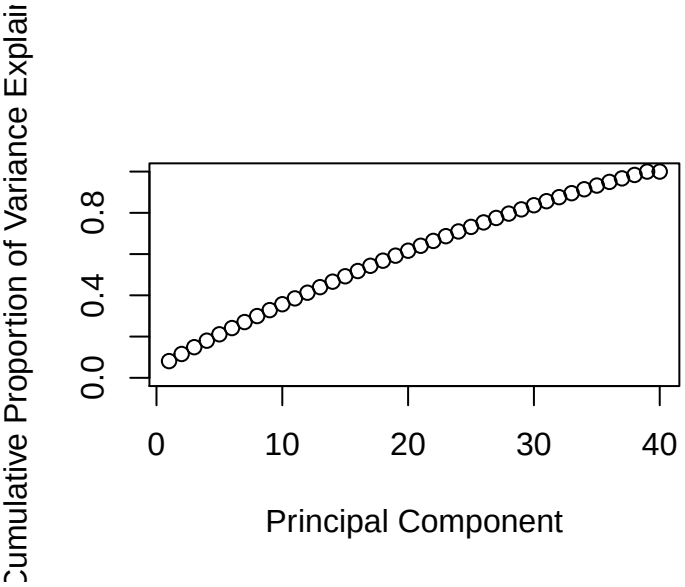
Kernel PCA

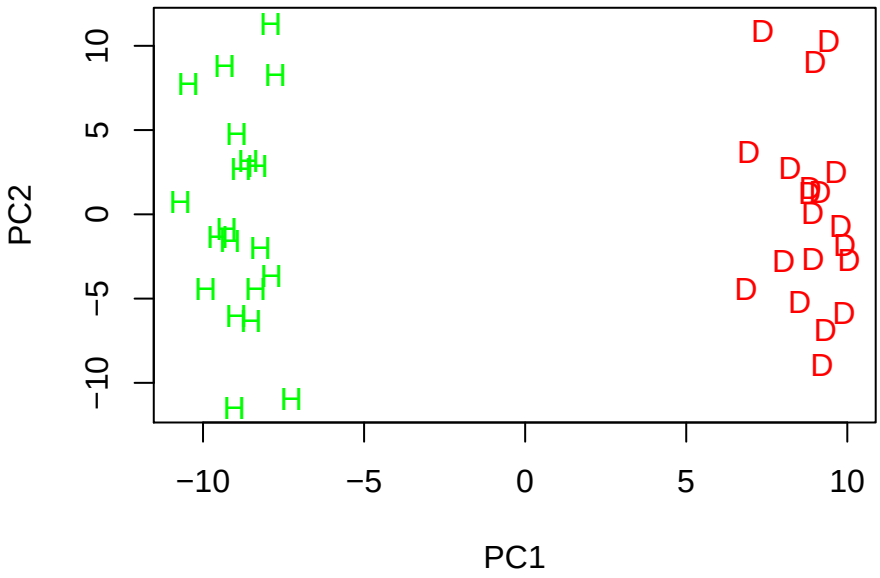
Proportion of variance explained (PVE)

Deciding how many PCs to use

Example

The following dataset consists of 40 tissue samples with measurements of 1,000 genes. The first 20 tissues come from healthy patients (H) and the remaining 20 come from a diseased patient group (D).




PC2
PC1
10
5
0
−5
−10
−10
−5
0
5
10
H
D

Clustering

K-means

The objective of clustering is to distinct groups from the datatest. With k-means we want to distinct k groups. The algorithm will assign each observation to exactly one of the cluster. It optimizes the groups by minimizing the within-cluster variation such that the sum of the with-cluster variations across all the clusters is the smallest possible.

Within-cluster variation (squared Euclidean distance)

If μ_k is the center of the cluster k. The total with-cluster variation is TW :

$$TW = \Sigma_{j=1}^{k} W_j = \Sigma_{j=1}^{k} \Sigma_{x_i \in C_j} (x_i - \mu_k)^2$$

K-means algorithm

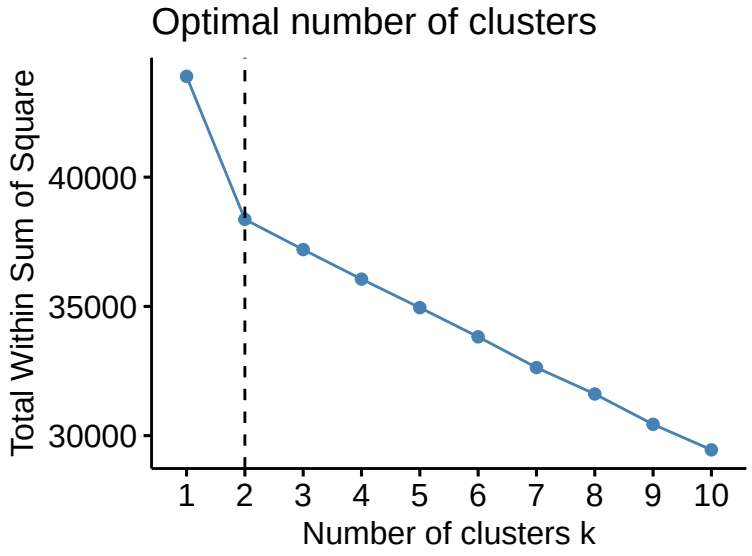
The first step when using k-means clustering is to indicate the number of clusters (k) that will be generated in the final solution. The algorithm starts by randomly selecting k objects from the data set to serve as the initial centers for the clusters. The selected objects are also known as cluster means or centroids.

Choice of k

We compute k-means clustering using different k, then we choose the number of cluster according to the location of a bend on the graph representing the Within-cluster variation according to k.

Example

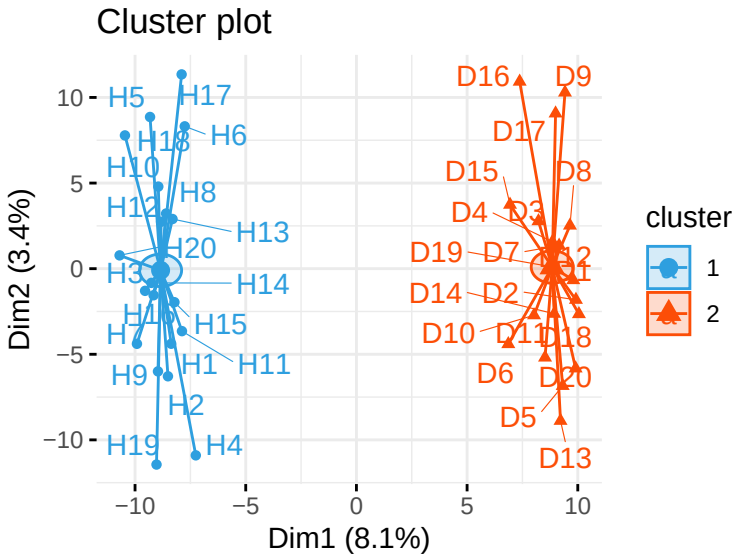
On R According to this graph, we should choose k=2 (it makes sense since we have Healthy and non healthy patients).



Then applying kmeans with 2 clusters we observe that the 20 first individuals (healthy) are not in the same cluster than the 20 others (non healthy).

```
##  H1  H2  H3  H4  H5  H6  H7  H8  H9 H10 H11 H12 H13 H14 H15 H16 H17 H18 H19 H20
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  D1  D2  D3  D4  D5  D6  D7  D8  D9 D10 D11 D12 D13 D14 D15 D16 D17 D18 D19 D20
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
```

Since we have a multi-dimensional dataset, we apply dimensionality reduction with the use of PCA to plot the clusters. On the x axis, it is the first PCA, on the y axis, it is the second PCA.



On python with scikit-learn By applying Kmeans (with 2 clusters) from scikit-learn on the gene dataset, we have the following assgnation to clusters. The 20 first individuals (Healthy) are well separated from the 20 last individuals since there are not in the same cluster.

```
## array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
##        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int32)
```

k-medoids algorithm

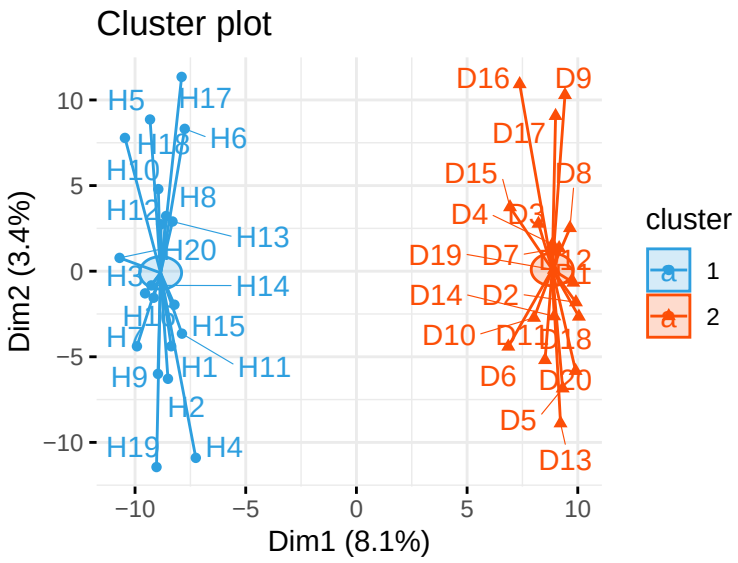
Principle

The k-medoids algorithm is a clustering approach related to k-means clustering. In k-medoids clustering, each cluster is represented by one of the data point in the cluster.

The most common k-medoids clustering methods is the PAM.

PAM algorithm (Partitioning Around Medoids)

```
##  H1  H2  H3  H4  H5  H6  H7  H8  H9 H10 H11 H12 H13 H14 H15 H16 H17 H18 H19 H20
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  D1  D2  D3  D4  D5  D6  D7  D8  D9 D10 D11 D12 D13 D14 D15 D16 D17 D18 D19 D20
##   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
```


Cluster plot
Dim2 (3.4%)
Dim1 (8.1%)
10
5
0
−5
−10
−10
−5
0
5
10
H5
H17
H6
H10
H8
H12
H13
H20
H3
H14
H15
H7
H1
H11
H9
H2
H19
H4
D16
D9
D17
D15
D8
D4
D19
D7
D14
D2
D10
D18
D6
D5
D13
cluster
1
2

Hierarchical clustering

Dissimilarity function

Euclidean distance

```
##       H1   H2   H3   H4   H5   H6   H7   H8   H9  H10
## H1   0.0 45.2 44.5 45.6 45.9 45.2 45.0 45.8 45.0 46.0
## H2  45.2  0.0 45.7 44.6 44.4 45.5 44.8 44.6 44.4 44.5
## H3  44.5 45.7  0.0 43.9 46.0 44.9 45.7 45.2 44.1 45.3
## H4  45.6 44.6 43.9  0.0 47.4 45.1 45.6 45.7 44.0 44.4
## H5  45.9 44.4 46.0 47.4  0.0 45.4 44.6 45.6 45.7 44.3
## H6  45.2 45.5 44.9 45.1 45.4  0.0 44.9 43.9 44.7 43.3
## H7  45.0 44.8 45.7 45.6 44.6 44.9  0.0 45.3 43.5 44.4
## H8  45.8 44.6 45.2 45.7 45.6 43.9 45.3  0.0 44.0 44.1
## H9  45.0 44.4 44.1 44.0 45.7 44.7 43.5 44.0  0.0 44.2
## H10 46.0 44.5 45.3 44.4 44.3 43.3 44.4 44.1 44.2  0.0
```

Correlation-based distance Correlation-based distance considers two observations to be similar if their features are highly correlated, even though the observed values may be far apart in terms of Euclidean distance.

```
##      H1 H2  H3  H4  H5  H6  H7  H8  H9 H10
## H1  0.0  1 1.0 1.0 1.0 1.0 1.0 1.1 1.0 1.1
## H2  1.0  0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
## H3  1.0  1 0.0 0.9 1.0 1.0 1.0 1.0 1.0 1.0
## H4  1.0  1 0.9 0.0 1.1 1.0 1.0 1.0 1.0 1.0
## H5  1.0  1 1.0 1.1 0.0 1.0 1.0 1.0 1.0 1.0
## H6  1.0  1 1.0 1.0 1.0 0.0 1.0 1.0 1.0 0.9
## H7  1.0  1 1.0 1.0 1.0 1.0 0.0 1.0 0.9 1.0
## H8  1.1  1 1.0 1.0 1.0 1.0 1.0 0.0 1.0 1.0
## H9  1.0  1 1.0 1.0 1.0 1.0 0.9 1.0 0.0 1.0
## H10 1.1  1 1.0 1.0 1.0 0.9 1.0 1.0 1.0 0.0
```

Linkage

The linkage function takes the distances and groups pairs of objects into clusters based on their similarity. These clusters are then linked to each other to create bigger clusters and the linkage continues until all the data are linked together in a hierarchical tree.

Maximum or complete linkage The distance between two clusters is defined as the maximum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce more compact clusters.

Minimum or single linkage The distance between two clusters is defined as the minimum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce long, "loose" clusters.

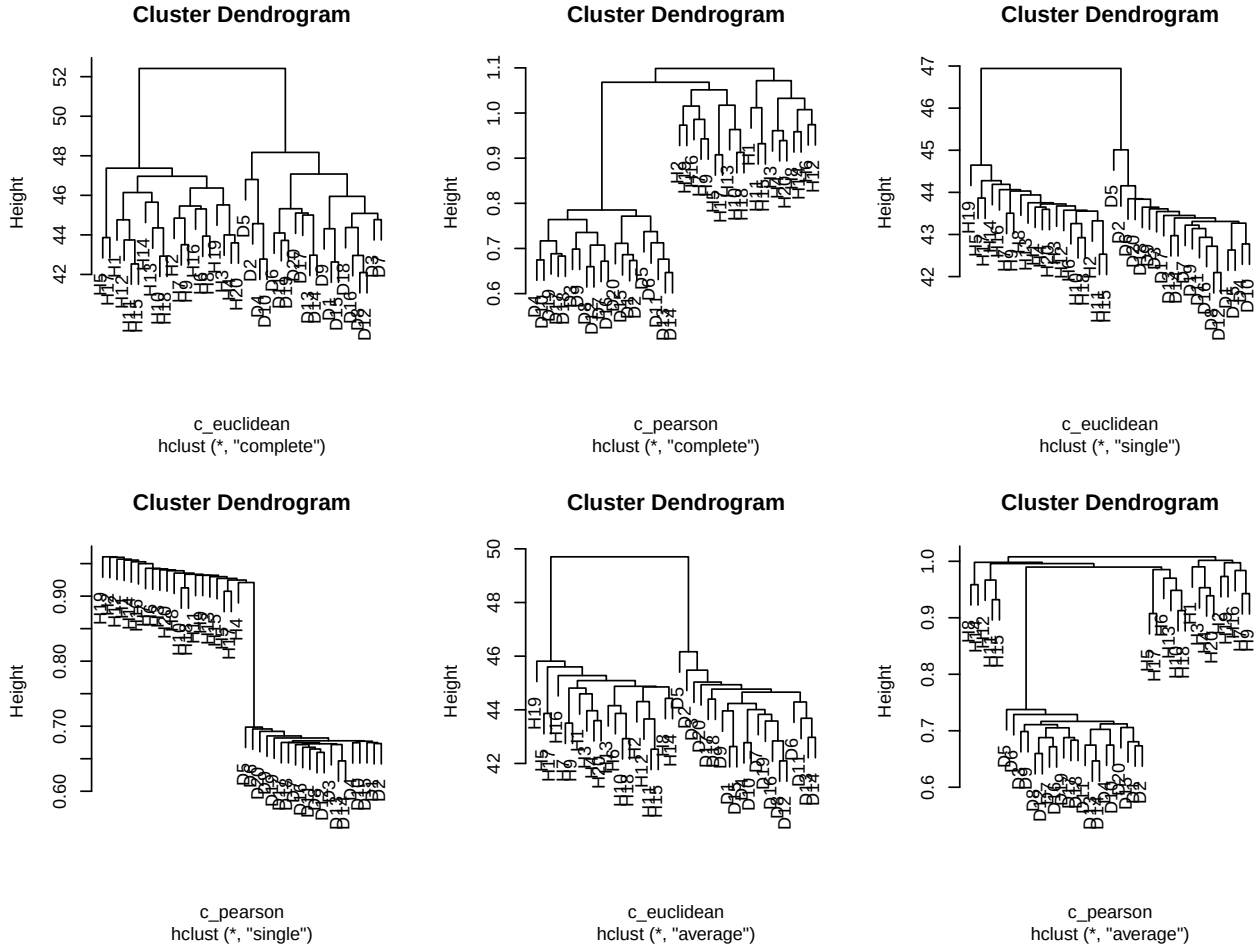
Mean or average linkage The distance between two clusters is defined as the average distance between the elements in cluster 1 and the elements in cluster 2.

Centroid linkage The distance between two clusters is defined as the distance between the centroid for cluster 1 (a mean vector of length p variables) and the centroid for cluster 2.

Ward's minimum variance method It minimizes the total within-cluster vari- ance. At each step the pair of clusters with minimum between-cluster distance are merged.

Example

On R



We can see that the use of euclidean distance in the three methods (complete, single, average) gives good results (no missclassification) but the use of correlation-distance gives very bad results.

Furthermore all methods, except Average with correlation-distance, divide the graph in two groups (healthy and non-healthy) which is very good.

On python with scikit-learn Doing hiearchical clustering with python gives the following dendogramm which shows that individual between 0 and 19 and individuals between 20 and 39 are well separated.

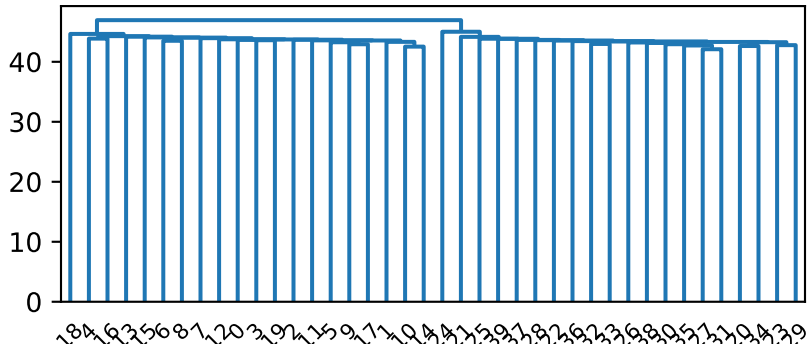
```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt

linked = linkage(r.genematrix, 'single')

plt.figure(figsize=(5, 2))
dendrogram(linked)
```

```
## {'icoord': [[15.0, 15.0, 25.0, 25.0], [55.0, 55.0, 65.0, 65.0], [105.0, 105.0, 115.0, 115.0], [95.0,
```

```
plt.show()
```



Validation techniques

Bootstrapping

Example

On R

On python with scikit-learn