Clustering

Julien JACQUES

Université Lyon 2

Introduction

Clustering

The **goal of clustering** is to create homogeneous group of obsevations, s.t.:

- observations within a group are as similar as possible
- groups are as different as possible from each other

The groups are called **clusters**.

Use of clustering

- Clustering is an unsupervised technique.
- It aims to explore the data and to discover some typical pattern.
- It is often used as a preliminary step between supervised approach.

Notations

- individuals (observations) are described by a set of p features
- $\mathbf{X}_i = (X_{i1}, \ldots, X_{ip})$ is the set of features for individual i $(1 \leq i \leq n)$
- we have to assign each individual to on of the K clusters : $Z_i \in \{1, \ldots, K\}$ is the cluster number of individual i
- the set $(Z_1, \ldots, Z_n)$ is a partition of the n individuals into K groups.

Distance

Historical methods are based on the notion of distance d_{ij} between two observations X_i and X_j.

$D = (d_{ij})_{1 \leq i \leq n, 1 \leq j \leq n}$ is a matrix of distance if:

- $d_{ii} = 0$
- $d_{ij} = d_{ji} \geq 0$ for all $i \neq j$
- $d_{ij} \leq d_{ik} + d_{kj}$

Examples:

- Euclidean distance:

$$d_{ij} = \left(\sum_{\ell=1}^{p} (x_{i\ell} - x_{j\ell})^2\right)^{1/2}$$

- Manhattan distance :

$$d_{ij} = \sum_{\ell=1}^{p} |x_{i\ell} - x_{j\ell}|$$

- Mahalanobis distance (when variables are of different scales):

$$d_{ij} = \left(\sum_{\ell=1}^{p} \frac{1}{\sigma_\ell^2} (x_{i\ell} - x_{j\ell})^2\right)^{1/2}$$

where σ_ℓ^2 is the variance of variable ℓ

Clustering interpretation

One the clustering is performed, results are analyzed by:

- extracting one representative per cluster (typically the cluster's means)
- comparing the features values among clusters

Plotting the data (thanks to PCA, MDS, t-SNE) in different colour according to their cluser membership is often helpful

Clustering validation

- Clustering is a unsupervised technique
- No validation data set exists
- If the interpretation of clustering results improve the knowledge about the data, clustering is successful

Comparing clustering results

For comparing two partitions $\mathbf{Z}_1 = (Z_{11}, \ldots, Z_{1n})$ and $\mathbf{Z}_2 = (Z_{21}, \ldots, Z_{2n})$ (resulting for instance from two clustering algo.), we use the **Rand index**:

$$R = \frac{a+d}{a+b+c+d} = \frac{a+d}{\binom{2}{n}} \in [0,1]$$

where, among the $\binom{2}{n}$ pairs of individuals:

- a: number of pairs of individuals which are in the same cluster in both $\mathbf{Z}_1$ and $\mathbf{Z}_2$
- b: number of pairs of individuals which are in the same cluster in $\mathbf{Z}_1$ and in different clusters in $\mathbf{Z}_2$
- c: number of pairs of individuals which are in different clusters in $\mathbf{Z}_1$ but in the same cluster in $\mathbf{Z}_2$
- d: number of pairs of individuals which are in different clusters in both $\mathbf{Z}_1$ and $\mathbf{Z}_2$

Exercice

Let's compute the Rand index for $(\mathbf{Z}_1, \mathbf{Z}_2)$ and $(\mathbf{Z}_1, \mathbf{Z}_3)$:

- $\mathbf{Z}_1 = \{1, 1, 2, 2, 2\}$
- $\mathbf{Z}_2 = \{1, 2, 2, 1, 2\}$
- $\mathbf{Z}_3 = \{2, 2, 1, 1, 1\}$

Comparing clustering results

The **adjusted Rand index** (ARI), which is the corrected-for-chance version of the Rand index, is often prefered.

```
library(mclust)
?adjustedRandIndex
```

k-means

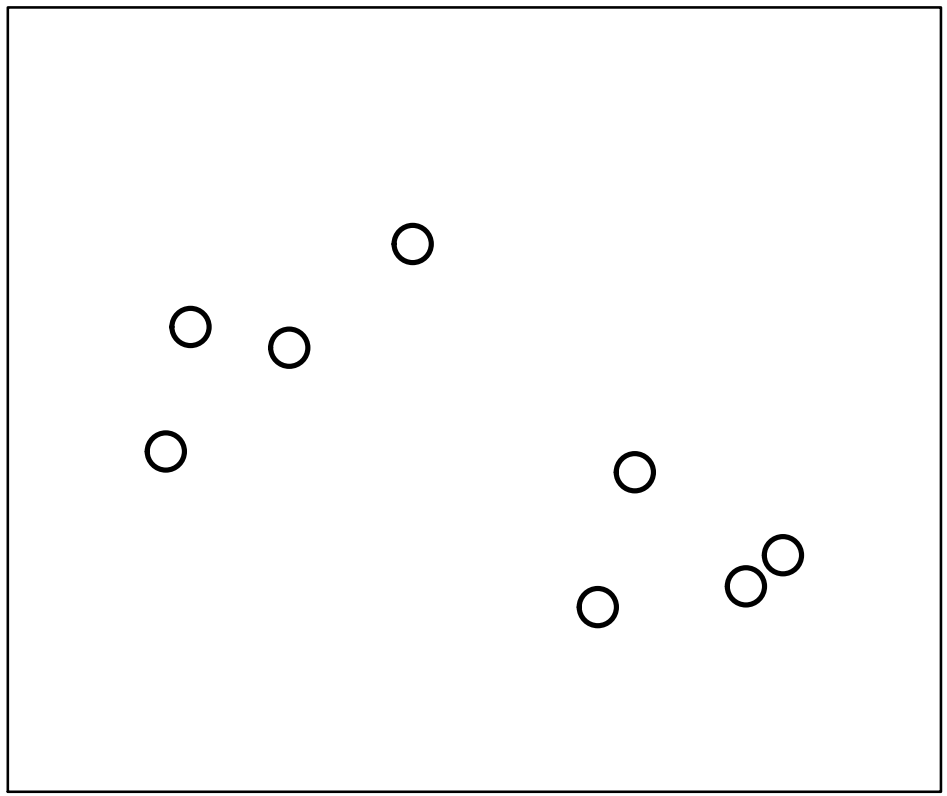
k-means algorithm

We assume $\mathbf{X} \in \mathbb{R}^p$, d is the Euclidean distance, K is known.

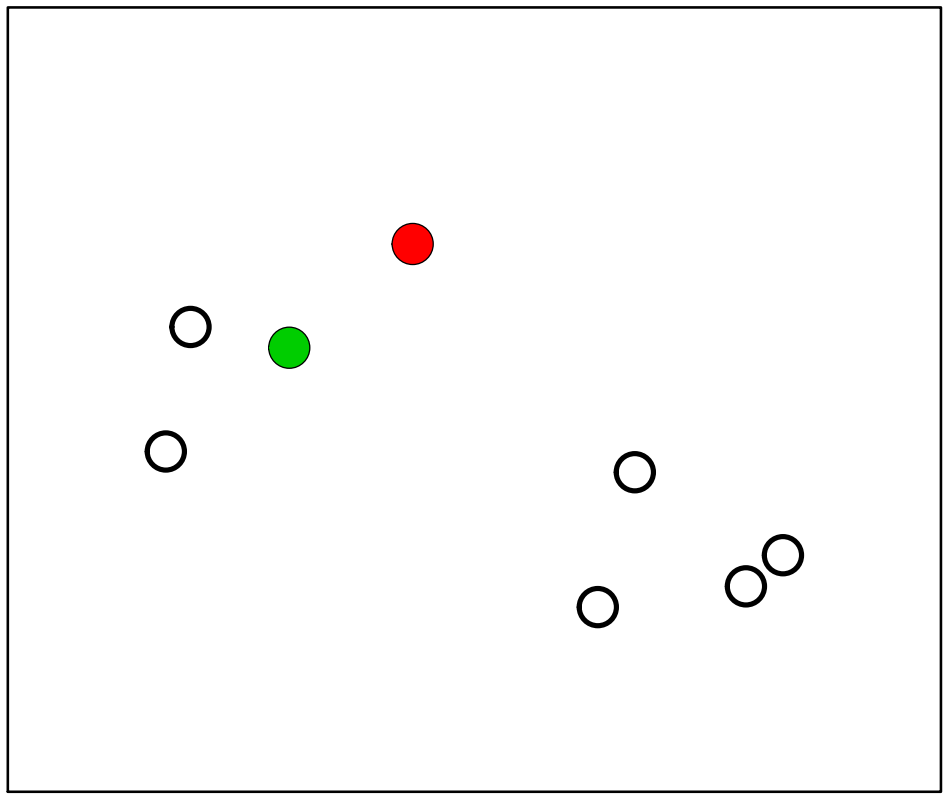
Lloyd k-means algorithm

- init.: randomly choose K *centres* μ_k among the n observations
- while parition not stable:
  - assign each observation to the cluster whose center is closest
  - update the cluster means μ_k

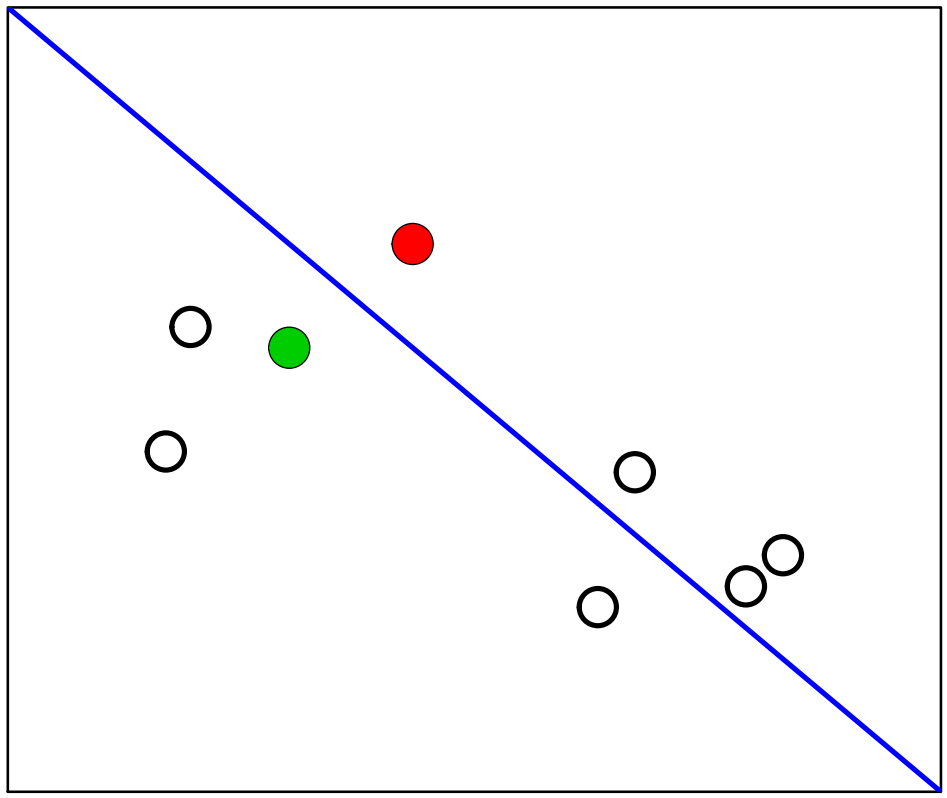
k-means algorithm



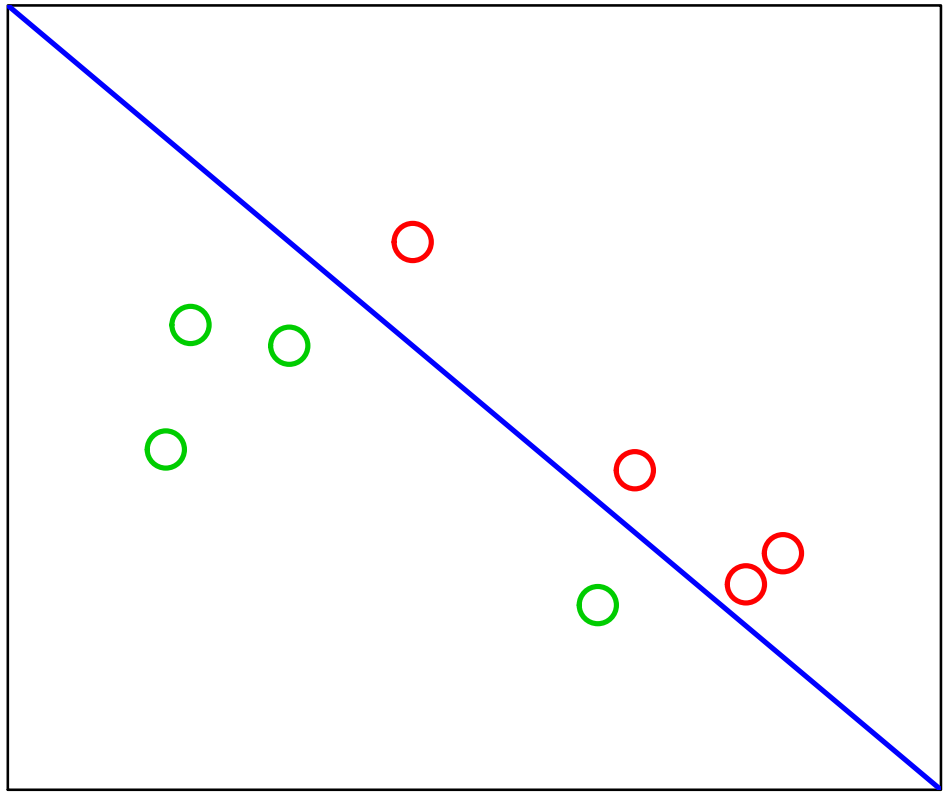
k-means algorithm



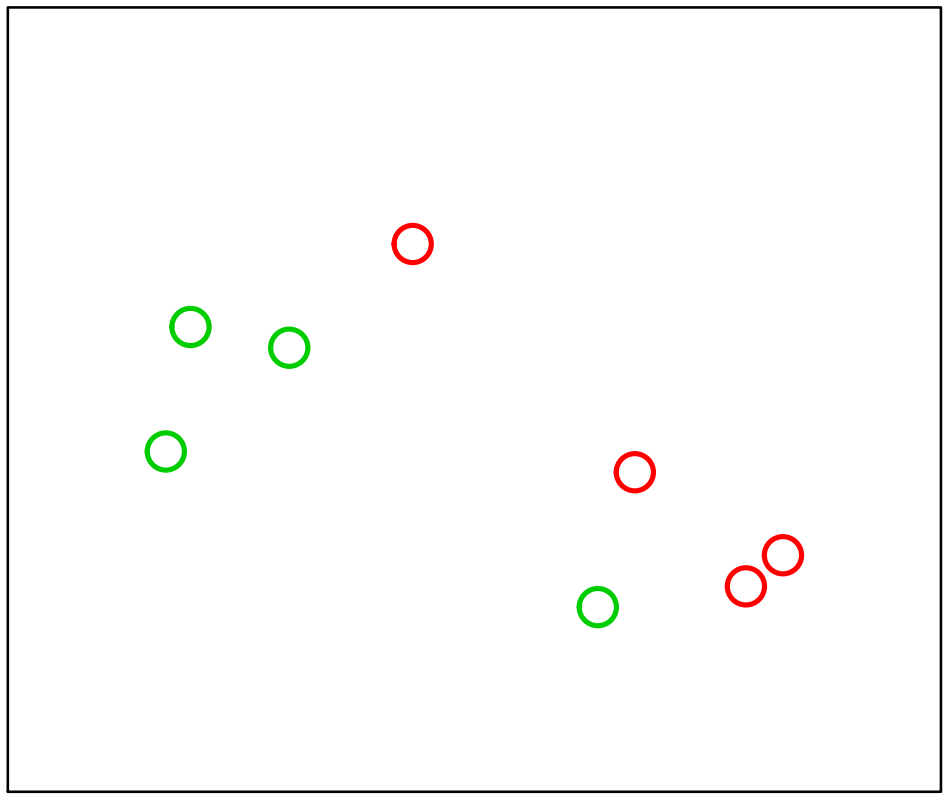
k-means algorithm



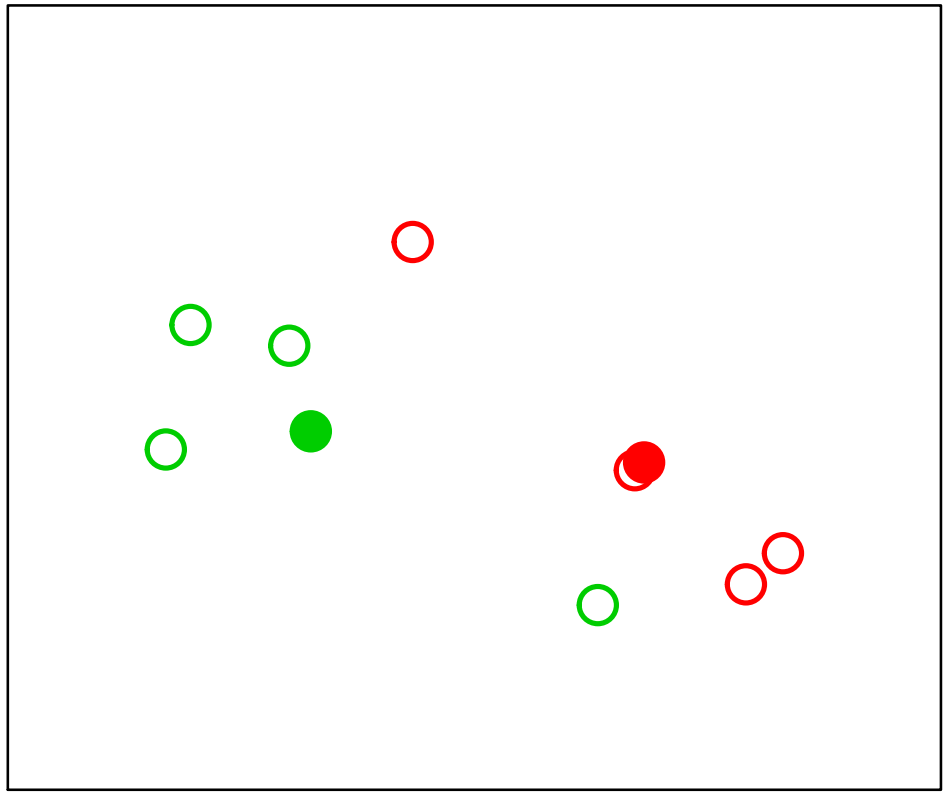
k-means algorithm



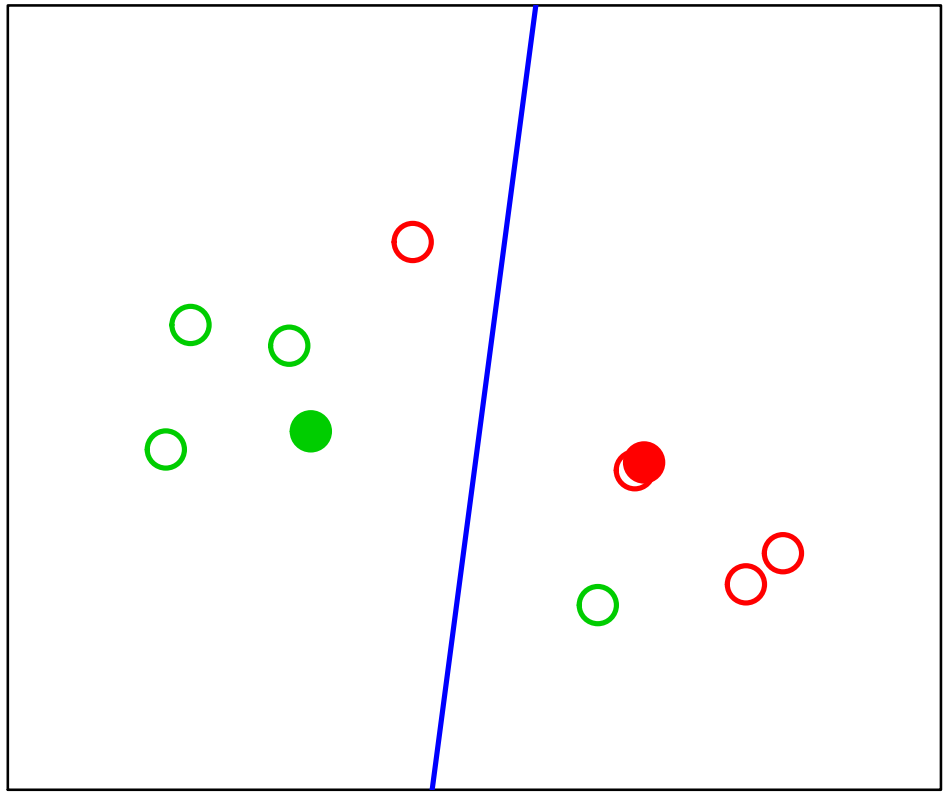
k-means algorithm



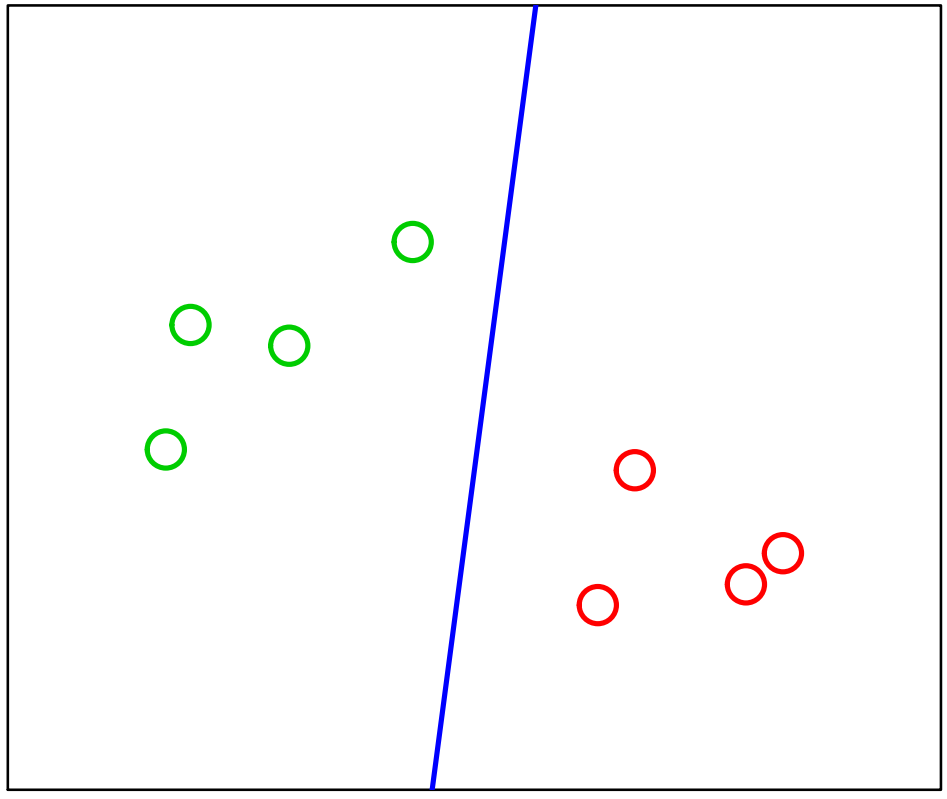
k-means algorithm



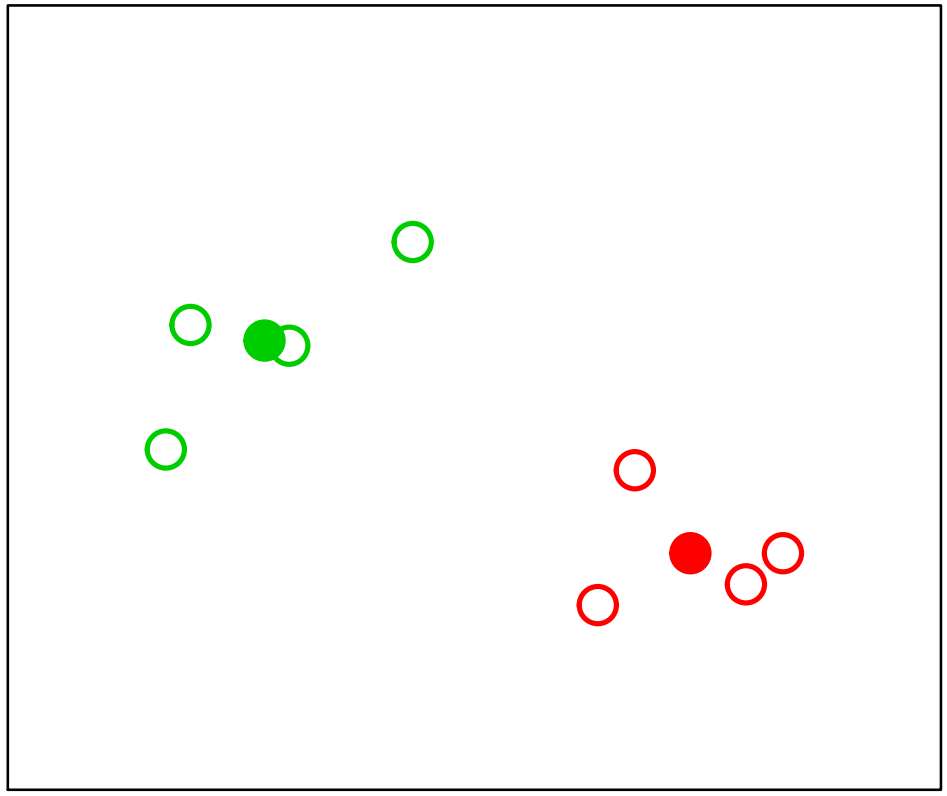
k-means algorithm



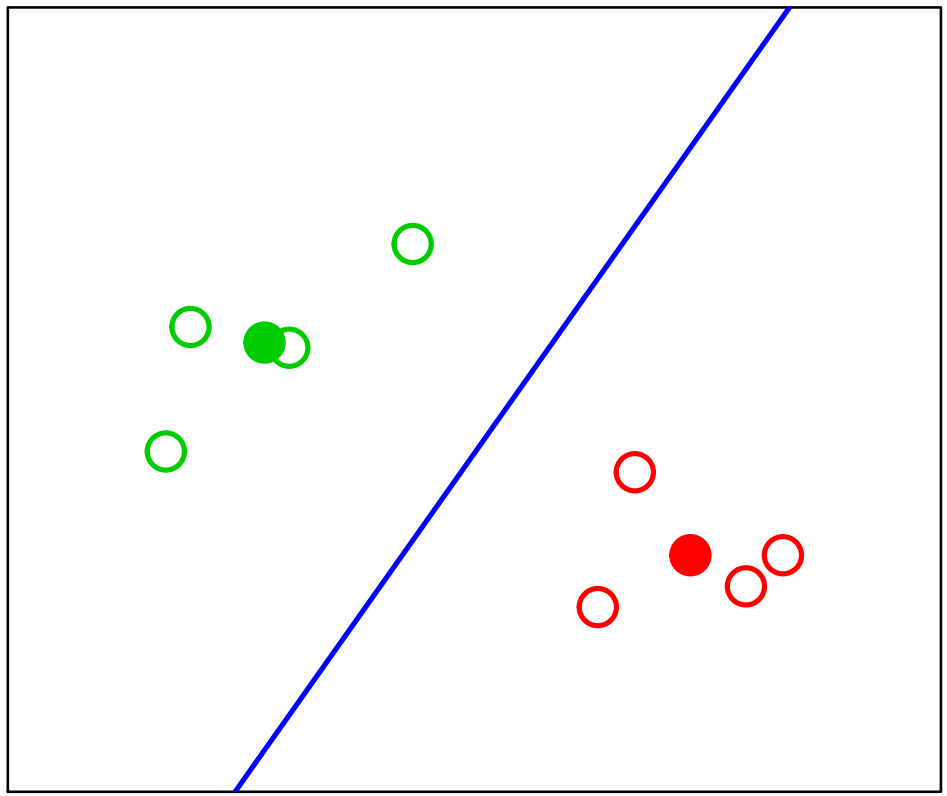
k-means algorithm



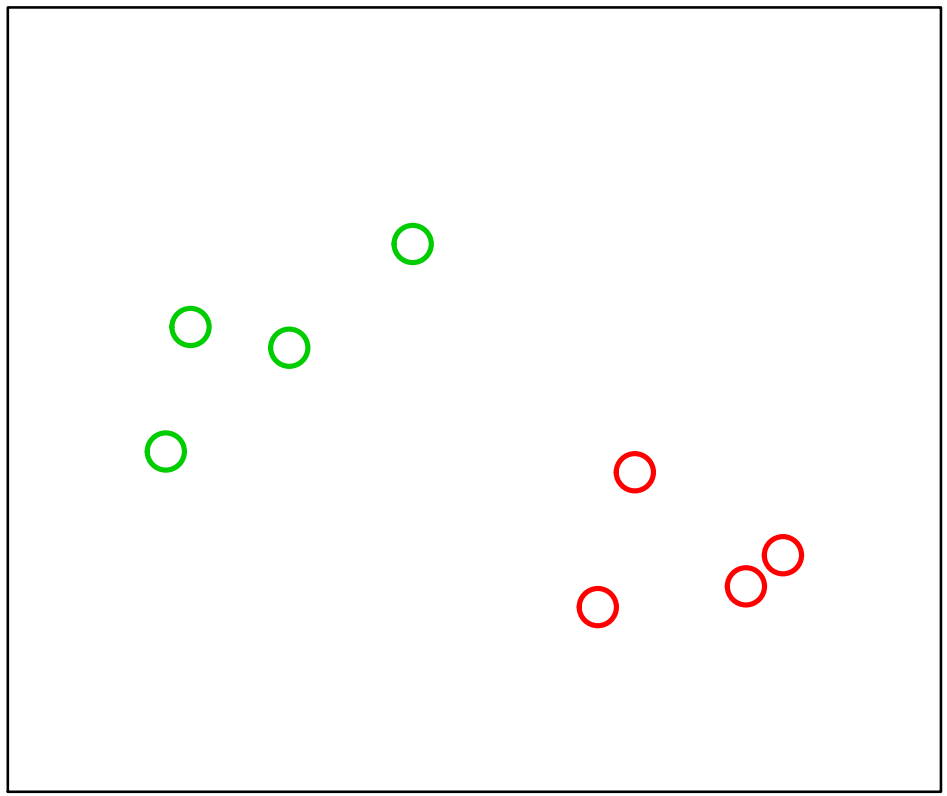
k-means algorithm



k-means algorithm



k-means algorithm



Sum-of-squares decomposition

The total sum-of-squares (T) can be decompose as follow:

$$\underbrace{\sum_{i=1}^{n} d^2(\mathbf{X}_i, \mu)}_{T} = \underbrace{\sum_{k=1}^{K} \sum_{i=1, n: Z_i = k} d^2(\mathbf{X}_i, \mu_k)}_{W(\mathbf{Z})} + \underbrace{\sum_{k=1}^{K} n_k d^2(\mu_k, \mu)}_{B(\mathbf{Z})}$$

where:

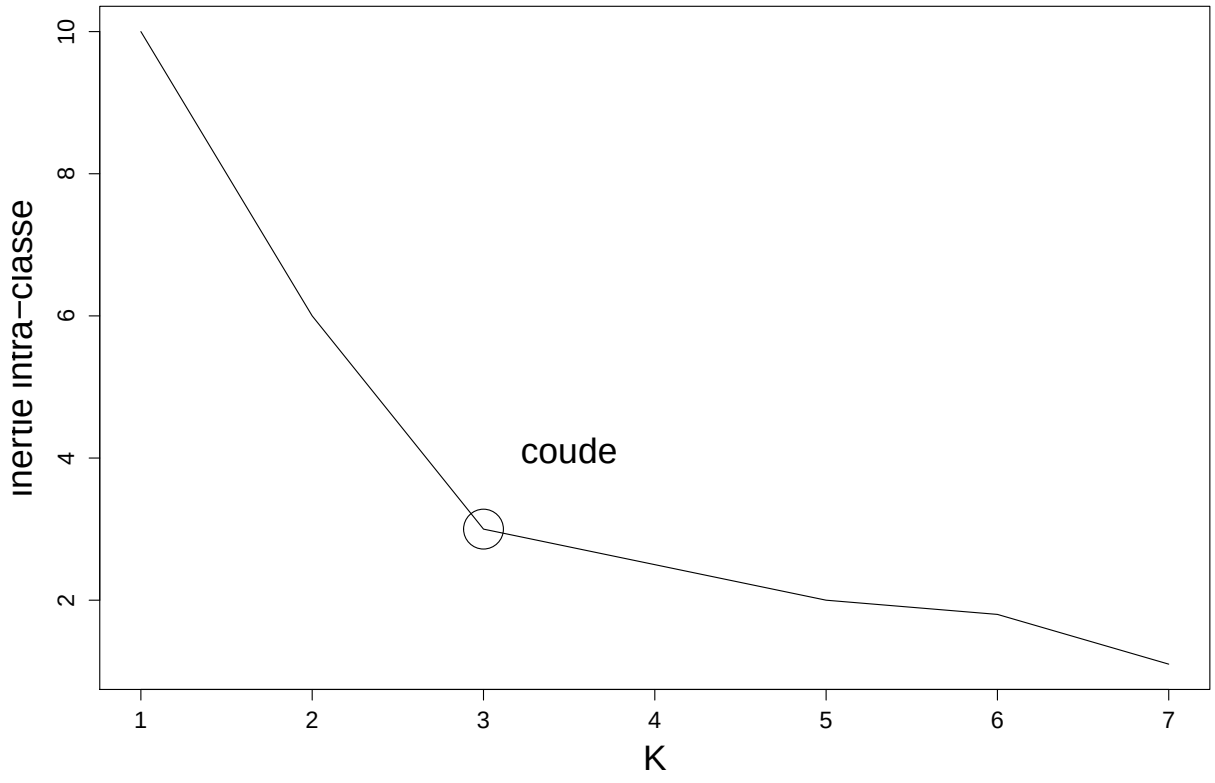
- $W(\mathbf{Z})$: within sum-of-squares
- $B(\mathbf{Z})$: between sum-of-squares

k-means properties

- The k-means algorithm converges
- The **k-means algorithm minimizes** $W(\mathbf{Z})$ (and consequently maximize $B(\mathbf{Z})$)
- But it can leads to a local minimum: indeed, k-means is a stochastic algorithm and its solution can depend on the initialization: $\Rightarrow$ multiple initialization has to be used

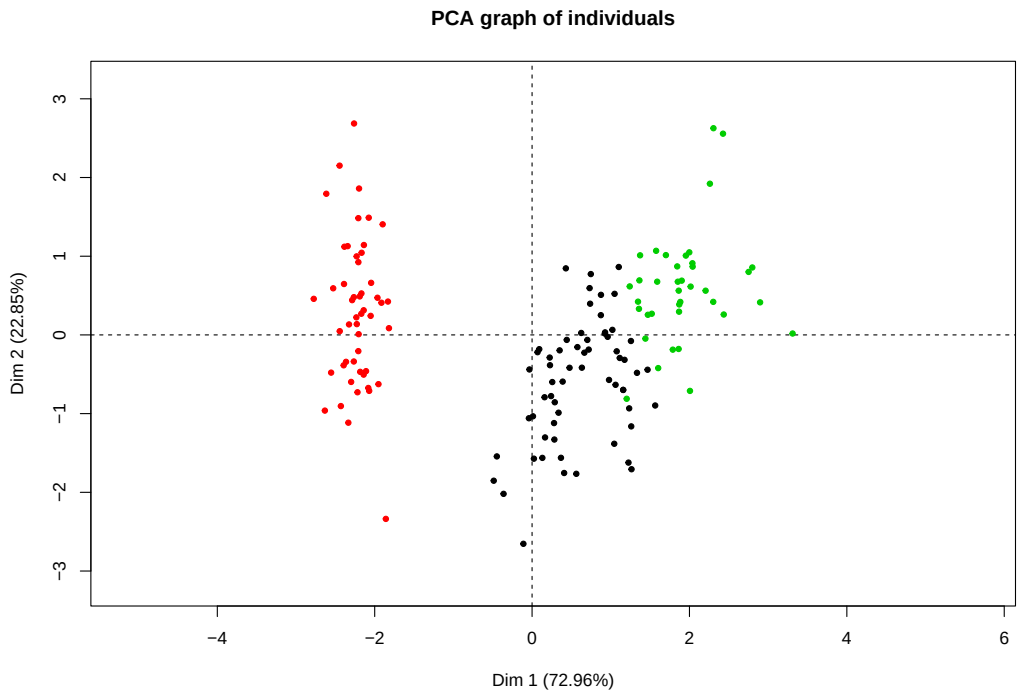
Choosing K

- within-sum-of-square decrease with K
- we seek for an elbow in the wihtinss plots:



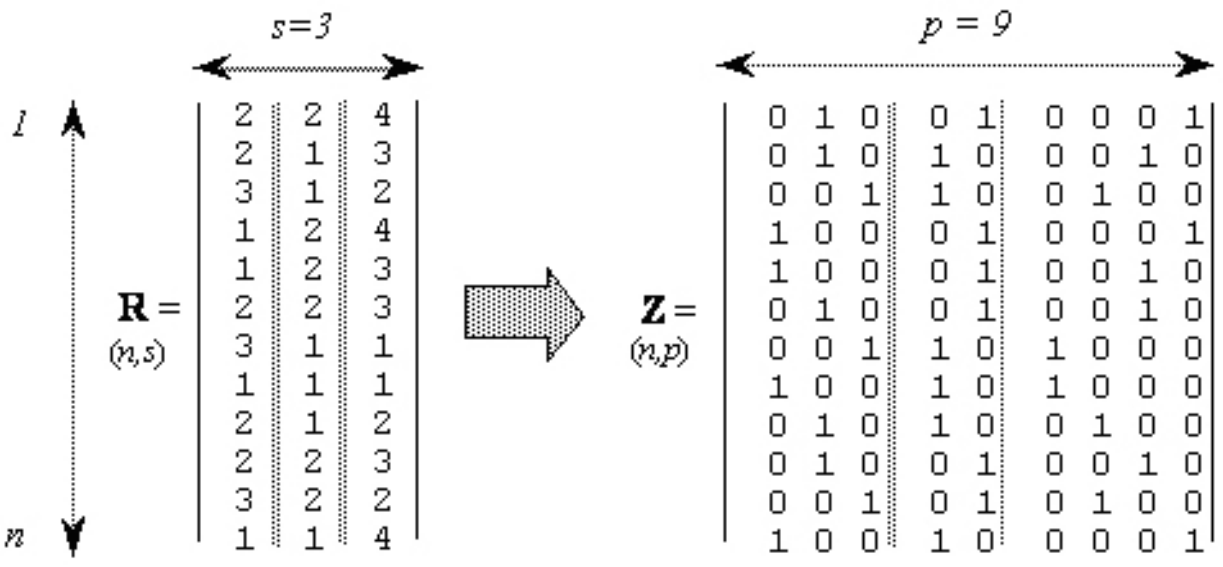
k-means in R

```
clus=kmeans(iris[,1:4],centers=3,nstart=5)
library("FactoMineR")
res.pca <- PCA(iris[,-5],graph = F)
plot(res.pca,choix="ind",col.ind=clus$cluster,
     graph.type = "classic",label='none')
```



Categorical features

- k-means is based on the Euclidean distance, and then is devoted to (normalized) quantitative features
- for categorical features, the simplest way to work with is to transform them into one-hot encoding



- another alternative is to use Multiple Correspondence Analysis to embbed the categorical features into a quantitative space
- their exists also in the literature some extensions of k-means for categorical data

k-means++ initialization

Idea: enforce distant cluster centers from the start.

It often lead to a dramatic improvement in practice.

k-means++

- choose first center μ_1 at random among the data points
- for $j = 2$ to K,repeat:
  - compute (for each points not already chosen):
    $D_i^j = \min_{\ell<j} ||X_i - \mu_\ell||$
  - choose $\mu_j = X_i$ with probability proportional to D_i^j
- once the K centers have been chosen, perfom usual k-means

k-medoids

- the k-means centers being the cluster's mean, they can be sensible to outliers
- the k-medoids version assign as cluster center the cluster medoids: the points which is the closest to all the cluster points of the cluster

$$\mu_k = argmin_{\mathbf{X}_i} \sum_{\mathbf{X}_j \in \mathcal{C}_k} ||\mathbf{X}_i - \mathbf{X}_j||$$

where $\mathcal{C}_k$ is the cluster k.

Exercice

- Implement you own k-means algorithm:
  - with random or k-means++ initialization
  - with k-medoids variant as an option
- Compare the complexity (in computation time) of the algorithms

Hierarchical clustering

Hierarchical Cluster Analysis

Require to choose:

- distance (or dissimilarity) between observations
- distance between clusters

Dissimilarity

$D = (d_{ij})_{1 \leq i \leq n, 1 \leq j \leq n}$ is a matrix of dissimilarity if:

- $d_{ij} = d_{ji} \geq d_{ii}$

Dissimilarity are especially useful for binary variables:

- Jaccard dissimilarity:

$$1 - \frac{a_{ij}}{p - d_{ij}}$$

  where:
  - $0 \leq a_{ij} \leq p$ is the number variables equal to 1 for individuals i and j
  - $0 \leq d_{ij} \leq p$ is the number variables equal to 0 for individuals i and j
- Concordance dissimilarity: $1 - \frac{a_{ij}+d_{ij}}{p}$
- Dice dissimilatiry: $1 - \frac{2a_{ij}}{a_{ij}+p-d_{ij}}$

Distance between clusters

Distance between clusters (A, B):

- single linkage

$$D(A, B) = \min\{d(\mathbf{X}, \mathbf{Y}), \mathbf{X} \in A, \mathbf{Y} \in B\}$$

- complete linkage

$$D(A, B) = \max\{d(\mathbf{X}, \mathbf{Y}), \mathbf{X} \in A, \mathbf{Y} \in B\}$$

- mean distance

$$D(A, B) = \sum_{\mathbf{X} \in A} \sum_{\mathbf{Y} \in B} \frac{d(\mathbf{X}, \mathbf{Y})}{\#A\#B}$$

- Ward

$$D(A, B) = \frac{\#A\#B}{\#A + \#B} d^2(\mu_A, \mu_B)$$

where μ_A and μ_B are centers of clusters A and $B\}$

Hierarchical Cluster Analysis

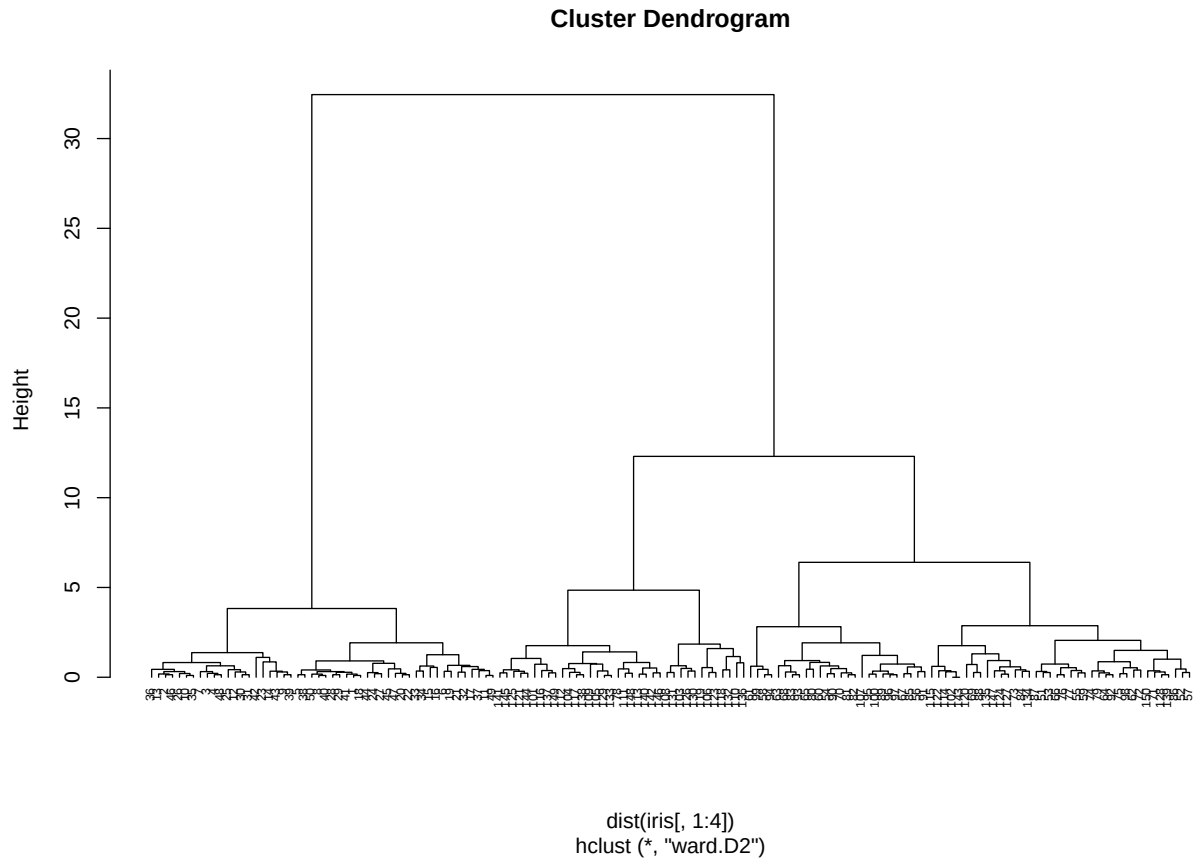
Aglomerative Hierarchical Cluster Analysis algorithm

- init.: each observation is its own cluster
- while more than one cluster:
  - compute the distances between any pair of clusters
  - merge the 2 closest ones

Thus, a set hierarchical partitions is build, from n clusters to 1 cluster

Hierarchical Cluster Analysis

```
clus=hclust(dist(iris[,1:4]),method ="ward.D2")
plot(clus, hang = -1,cex=.6)
```




```
cluster=cutree(clus,k=3)
```

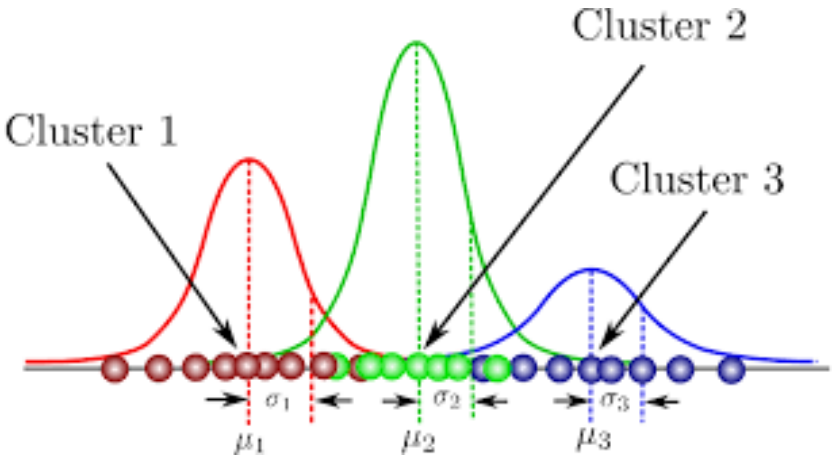
Hierarchical Cluster Analysis properties

- no need to specify the number of clusters
- it can be selected by looking for the highest gap in the dendogram
- Ward distance merges the two clusters by minimizing within sum-of-squares ; but it is sub-optimal in comparison with k-means since we can only merge to clusters at each step

Mixture model

Mixture model

- *Modern* appraoches for clustering consider probabilistic framework rather than working with distances
- A cluster is define as a set of data generated by a same (univariate) probability distribution
- The goal of clustering is then to estimate a mixture of distribution



Src: https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

R packages:

```
library(mclust)
library(Rmixmod)
```

DBscan

DBscan

- DBscan = density-based spatial clustering of applications with noise
- parameters: radius ϵ and minimal cluster size *minPts*

DBscan Repeat as long as at least one point has not been visited:

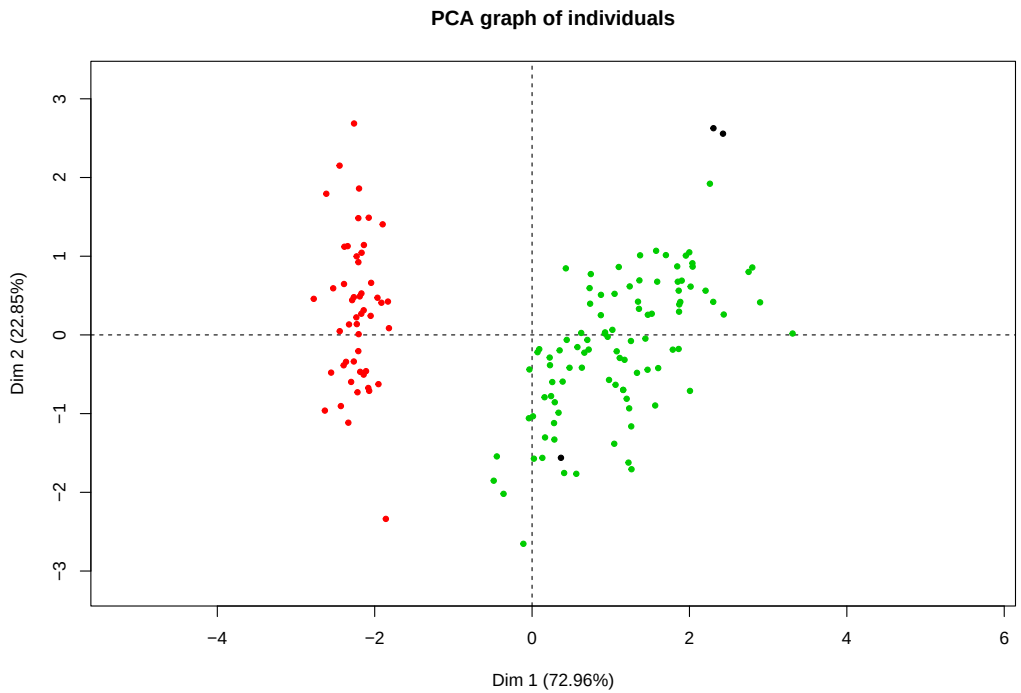
- pick an unvisited point $\mathbf{X}_i$ at random
- if it has less than *minPts* at a distance less than ϵ, mark it as outlier
- other, form the cluster of all points that can be reached by jumps of at most ϵ starting from $\mathbf{X}_i$

DBscan properties

- no need to specify number of clusters
- sensitive to the choice of parameters (ϵ, *minPts*)
- choosing ϵ, *minPts* is hard. In practice:
  - choice of ϵ: such that the proportion of outliers is at most 10%
  - choice of *minPts*: such that at least 90% have at least *minPts* neighbors

DBscan in R

```
library(dbscan)
clus=dbscan(iris[,1:4], eps = .7, minPts = 5)
res.pca <- PCA(iris[,-5],graph = F)
plot(res.pca,choix="ind",col.ind=clus$cluster+1,
     graph.type = "classic",label='none')
```



Spectral clustering

Spectral clustering

- Clustering is performed by **embedding the data into the subspace of the eigenvectors of an similarity matrix**
- The goal is to reduce the dimension of the space in which to perform clustering
- Clustering is then perform with a standard clustering method
- If S is a similarity matrix, for instance $S_{ij} = -||\mathbf{X}_i - \mathbf{X}_j||^2$, the Laplacian matrix is defined by:

$$L = D - S$$

  where D is a diagonal matrix with $D_{ii} = \sum_j S_{ij}$.
- L is normalized such that the diagonal elements be all unit. Different normalization exists, among which the Shi–Malik ones:

$$L = I_d - D^{-1/2} S D^{-1/2}$$

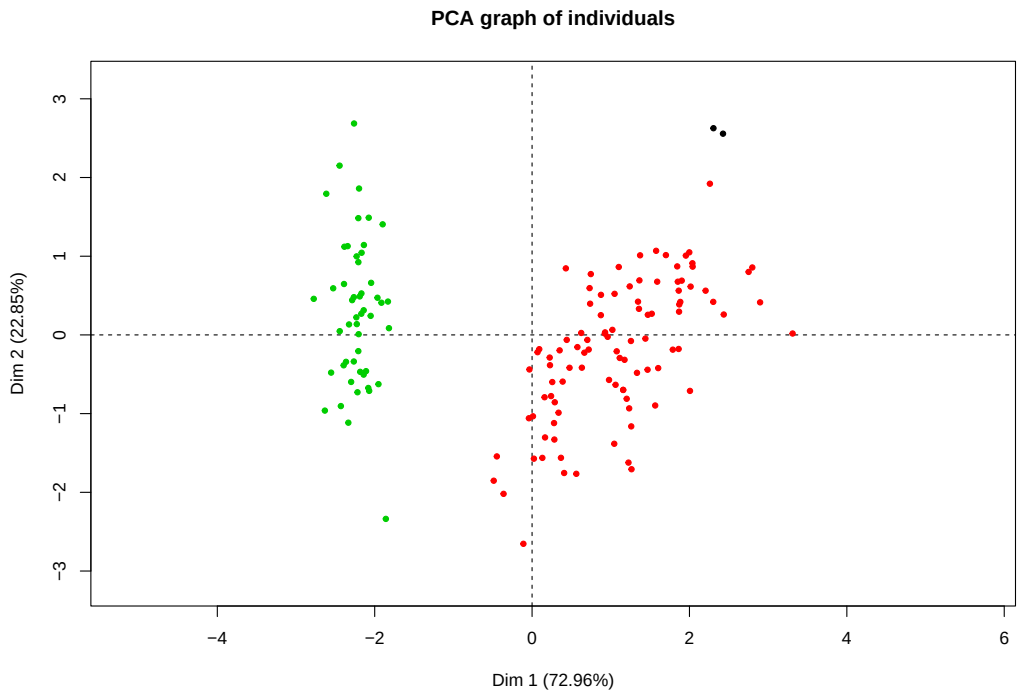
Spectral clustering

Basic spectral clustering algorithm

- Calculate the (normalized) Laplacian matrix L
- Compute the k eigenvectors corresponding to the k smallest eigenvalues
- Consider the matrix of these k eigenvectors as features for the n points
- Perform standard clustering algorithm on these matrix

Spectral clustering in R

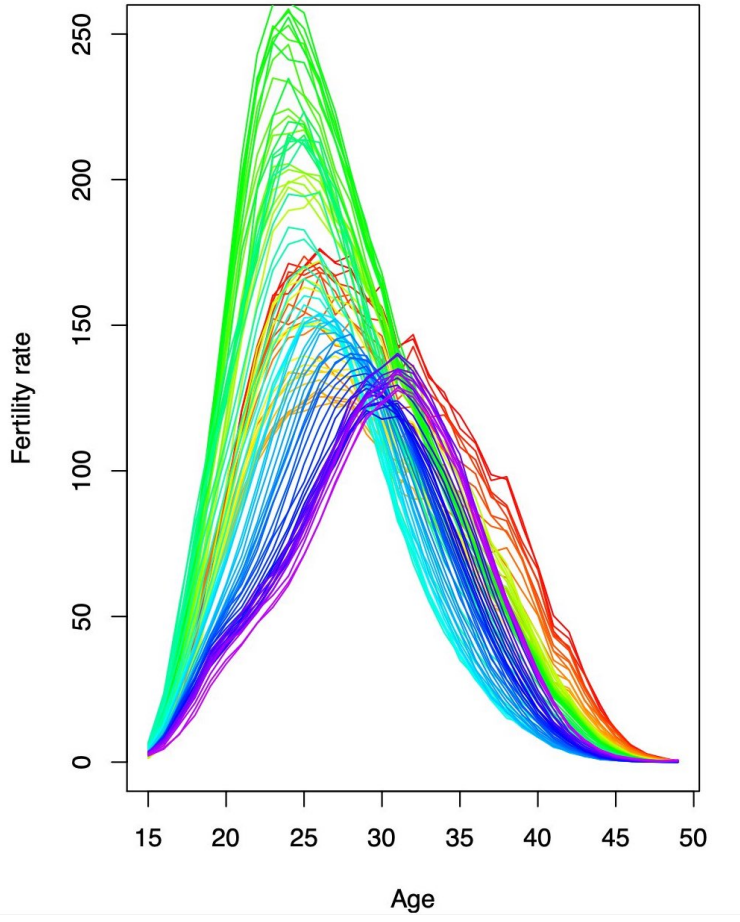
```
library(kernlab)
clus=specc(as.matrix(iris[,1:4]), centers=3)
res.pca <- PCA(iris[,-5],graph = F)
plot(res.pca,choix="ind",col.ind=clus,
     graph.type = "classic",label='none')
```



Functional clustering

Functional clustering

Clustering when data are functions (time series. . .)



R packages:

```
library(funHDDC)
```

Functional clustering

To go further:

A. Schmutz, J. Jacques, C. Bouveyron, L. Chèze and P. Martin (2020). Clustering multivariate functional data in group-specific functional subspaces, Computational Statistics, 35, 1101-1131.

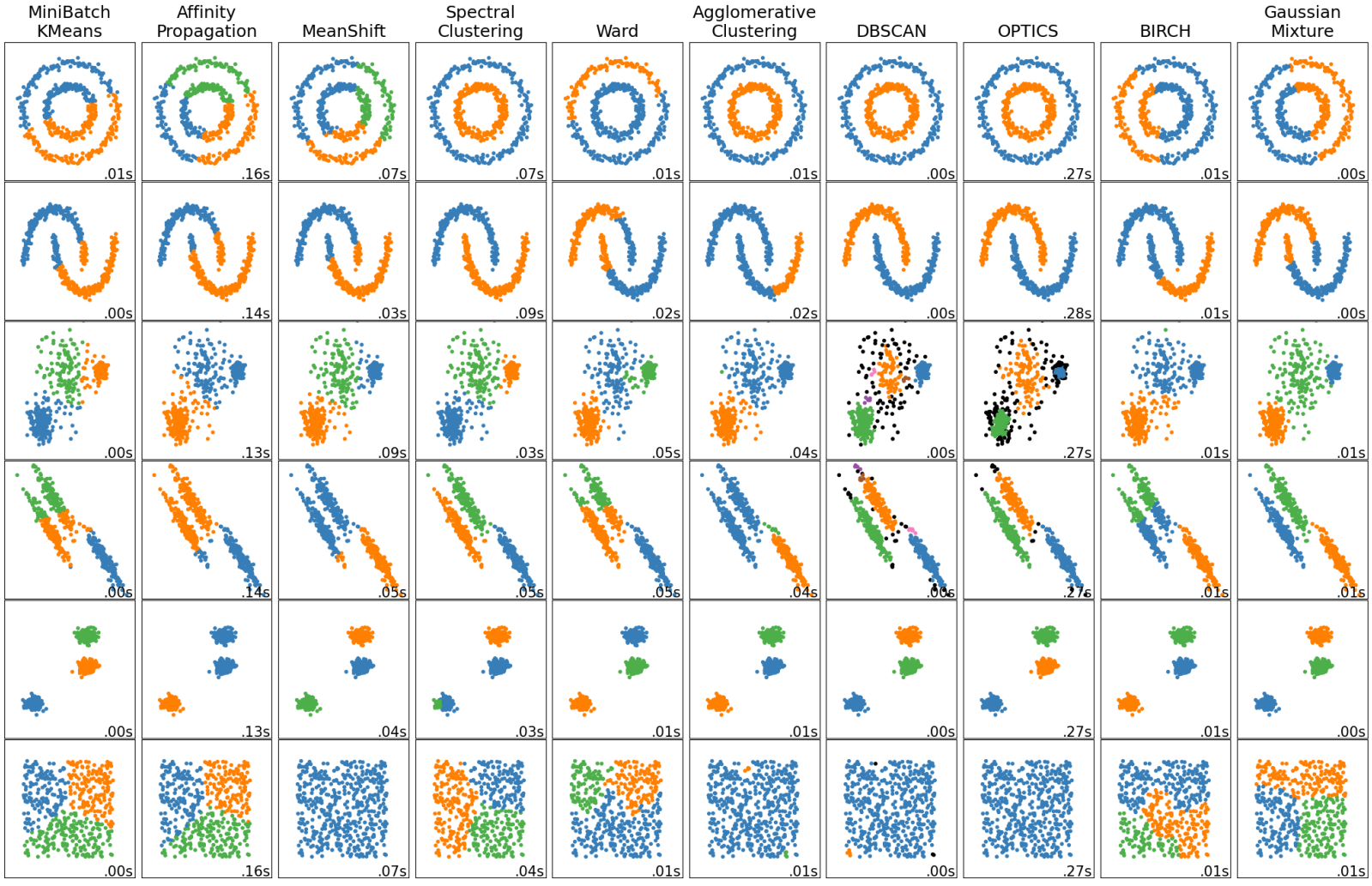
J.Jacques and C.Preda (2014), Functional data clustering: a survey, Advances in Data Analysis and Classification, 8[3], 231-255.

C. Bouveyron, E. Côme and J. Jacques (2015), The discriminative functional mixture model for the analysis of bike sharing systems, Annals of Applied Statistics, 9[4], 1726-1760.

J.Jacques and C.Preda (2014), Model-based clustering of multivariate functional data, Computational Statistics and Data Analysis, 71, 92-106.

Conclusion

Which method to choose



Src: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html