

Apprentissage non supervisé

Joseph Salmon, Nicolas Verzelen

INRA / Université de Montpellier

Plan

Introduction

k -means

Modèles de mélanges gaussiens

Classification hiérarchique ascendante

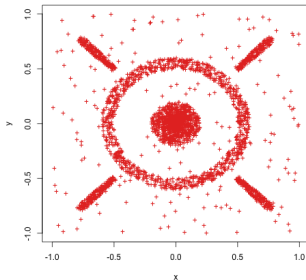
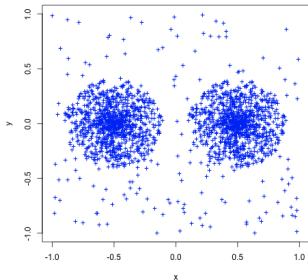
Introduction

Objectifs :

- ▶ Structurer les données.
- ▶ On cherche à regrouper les observations “proches” en classes.

Vocabulaire :

- ▶ partitionner les données (🇬🇧 : *Clustering*)
- ▶ une méthode *non-supervisé* (sans étiquettes, *i.e.*, sans y)



Exemples d'applications

Gestion - Marketing :

- ▶ données : infos client, produits, ...
- ▶ but : segmenter la clientèle, définir des profils

Exemples d'applications

Gestion - Marketing :

- ▶ données : infos client, produits, ...
- ▶ but : segmenter la clientèle, définir des profils

Traitement Naturel du Langage (: *NLP*) :

- ▶ données : texte, email, ...
- ▶ but : grouper automatiquement les textes proches

Exemples d'applications

Gestion - Marketing :

- ▶ données : infos client, produits, ...
- ▶ but : segmenter la clientèle, définir des profils

Traitement Naturel du Langage (: *NLP*) :

- ▶ données : texte, email, ...
- ▶ but : grouper automatiquement les textes proches

Sociologie :

- ▶ données : attributs d'un individu, e.g., revenus, sexe, ...
- ▶ but : former des catégories de population

Exemples d'applications

Gestion - Marketing :

- ▶ données : infos client, produits, ...
- ▶ but : segmenter la clientèle, définir des profils

Traitement Naturel du Langage (: *NLP*) :

- ▶ données : texte, email, ...
- ▶ but : grouper automatiquement les textes proches

Sociologie :

- ▶ données : attributs d'un individu, e.g., revenus, sexe, ...
- ▶ but : former des catégories de population

Analyse génomique :

- ▶ données : gènes
- ▶ but : former des groupes homogènes de gènes.

Notion de proximité

Questions :

- ▶ Comment mesurer la proximité de deux observations ?
- ▶ Comment mesurer la proximité de deux classes ?

Ingrédients :

- ▶ **fonction de dissimilarité** : plus la mesure est faible, plus les objets sont similaires (\approx à une distance)
- ▶ **fonction de similarité** : plus la mesure est grande, plus les objets sont similaires

Distances usuelles entre deux observations x_1 et x_2

- Distance Euclidienne :

$$d^2(x_1, x_2) = \sum_{i=1}^d (x_1^i - x_2^i)^2$$

- Distance de Manhattan :

$$d(x_1, x_2) = \sum_{i=1}^d |x_1^i - x_2^i|$$

- Distance de Minkowski :

$$d(x_1, x_2) = \left(\sum_{i=1}^d |x_1^i - x_2^i|^p \right)^{\frac{1}{p}}$$

- Distance de Mahalanobis (pour une matrice symétrique W)

$$d^2(x_1, x_2) = \sum_{i=1}^d \sum_{j=1}^d W_{i,j} (x_1^i - x_2^i)(x_1^j - x_2^j)$$

Cas des variables discrètes

Distance de Hamming :

$$d(x_1, x_2) = \sum_{i=1}^d \mathbb{1}_{\{x_1^i \neq x_2^i\}}$$

Exemple : donne le nombre d'entrées où les vecteurs diffèrent :

$$x_1 = (0, 1, 2, 1, 2, 1)^\top \text{ et } x_2 = (1, 0, 2, 1, 0, 1)^\top$$

Ainsi,

$$d(x_1, x_2) = 3$$

Distances entre deux classes \mathcal{C}_1 et \mathcal{C}_2

- ▶ plus proche voisin :

$$d(\mathcal{C}_1, \mathcal{C}_2) = \inf \{ \text{dist}(x, y) : x \in \mathcal{C}_1, y \in \mathcal{C}_2 \}$$

- ▶ diamètre maximum :

$$d(\mathcal{C}_1, \mathcal{C}_2) = \sup \{ \text{dist}(x, y) : x \in \mathcal{C}_1, y \in \mathcal{C}_2 \}$$

- ▶ distance moyenne :

$$d(\mathcal{C}_1, \mathcal{C}_2) = (\#\mathcal{C}_1)^{-1}(\#\mathcal{C}_2)^{-1} \sum_{x \in \mathcal{C}_1, y \in \mathcal{C}_2} \text{dist}(x, y)$$

- ▶ distance des barycentres :

$$d(\mathcal{C}_1, \mathcal{C}_2) = \text{dist}(\mu_1, \mu_2)$$

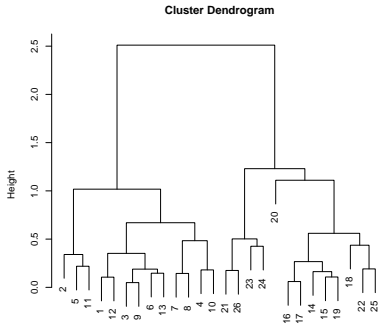
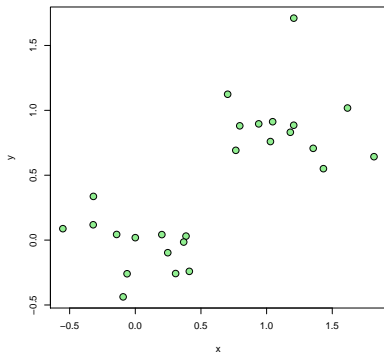
- ▶ distance de Ward :

$$d(\mathcal{C}_1, \mathcal{C}_2) = \left(\frac{\#\mathcal{C}_1 \#\mathcal{C}_2}{\#\mathcal{C}_1 + \#\mathcal{C}_2} \right)^{\frac{1}{2}} \text{dist}(\mu_1, \mu_2)$$

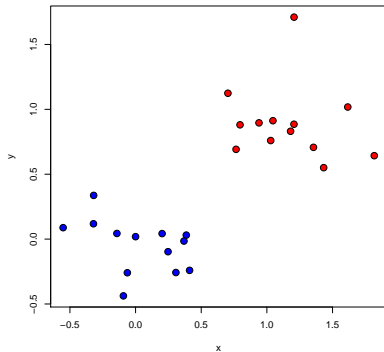
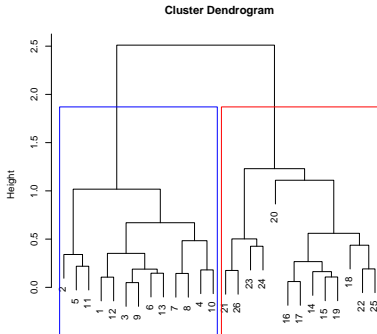
Panorama de clustering

<http://scikit-learn.org/stable/modules/clustering.html>

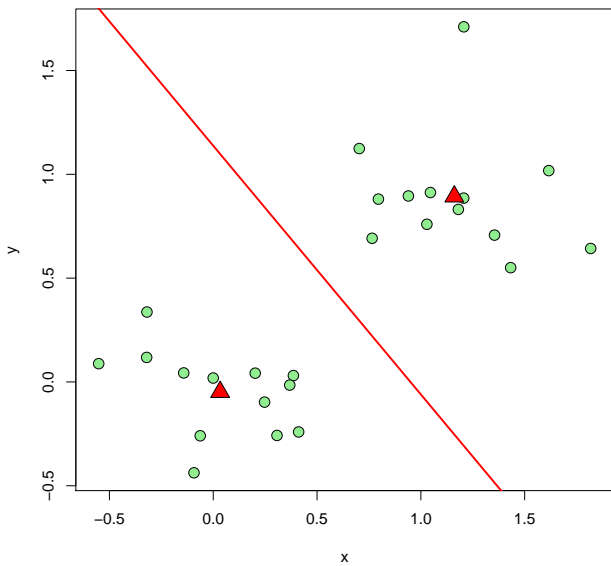
Classification ascendante hiérarchique



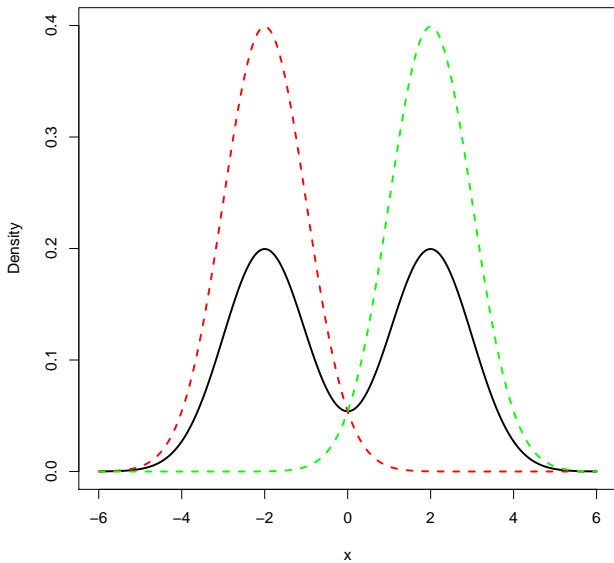
Classification ascendante hiérarchique



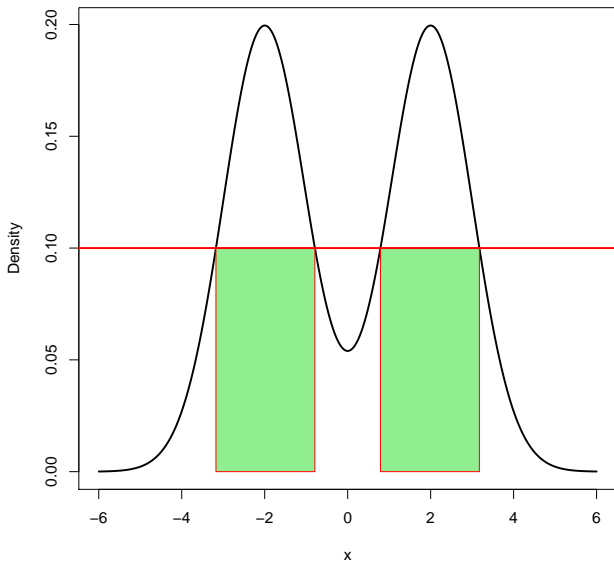
k -means



Modèle de mélange



Approche par densité — Modes



Qualité d'une partition

Soit $\mathcal{C}_1 \cup \dots \cup \mathcal{C}_K$ une partition de $\llbracket 1, n \rrbracket$, tq.

$|\mathcal{C}_1| = N_1, \dots, |\mathcal{C}_K| = N_K$, et de "centres" μ_1, \dots, μ_K

Inertie intra-cluster

$$I_w = \sum_k \sum_{i \in \mathcal{C}_k} d^2(x_i, \mu_k)$$

Inertie inter-cluster

$$I_b = \sum_{k=1}^K N_k d^2(\mu_k, \bar{x}_n), \text{ où } \bar{x}_n = \frac{1}{n} \sum_{i=1}^n x_i$$

Stratégie :

- (i) Minimiser l'inertie intra-cluster ;
- (ii) Maximiser l'inertie inter-cluster.

Plan

Introduction

k-means

Introduction

Propriétés

Modèles de mélanges gaussiens

Classification hiérarchique ascendante

La méthode du k -means

Contexte : On dispose d'un échantillon, supposé i.i.d., de taille n : x_1, \dots, x_n à valeurs dans \mathbb{R}^d

Principe :

- ▶ Chaque groupe k : représenté par un **centroïde** $\mu_k \in \mathbb{R}^d$
- ▶ Formation des groupes : affecter chaque donnée au centroïde le plus proche

Heuristique :

Déterminer K centroïdes μ_1, \dots, μ_K minimisant un **critère de distorsion** :

$$\mathcal{E}_n(\mu_1, \dots, \mu_K) = \frac{1}{n} \sum_{i=1}^n \min_{1 \leq k \leq K} d(x_i, \mu_k)^2$$

Algorithme de Lloyd : optimisation alternée

Objectif : minimiser le critère de distorsion $\mathcal{E}_n(\mu_1, \dots, \mu_K)$

Initialisation des K centres μ_1, \dots, μ_K (au hasard, ou kmeans++⁽¹⁾)

Affectation de chaque observation au centre le plus proche

Mise à jour des centres, en calculant la moyenne empirique des observations dans chaque classe

Itérer sur 2 et 3 jusqu'à convergence

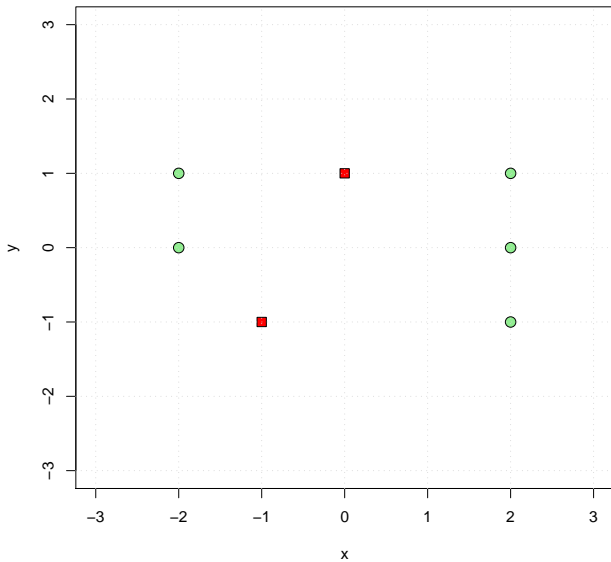
Rem:

- ▶ Convergence vers un minimum local seulement.
- ▶ En pratique : faire tourner plusieurs fois, avec différentes initialisations (choisir le meilleur sur le critère)

(1). D. ARTHUR et S. VASSILVITSKII. "k-means++ : The advantages of careful seeding". In : *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial et Applied Mathematics. 2007, p. 1027-1035.

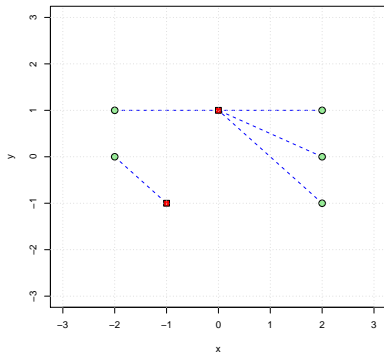
Exemple (1/3)

Centres initiaux

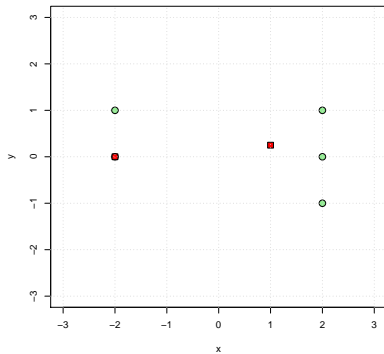


Exemple (2/3)

Groupes initiaux

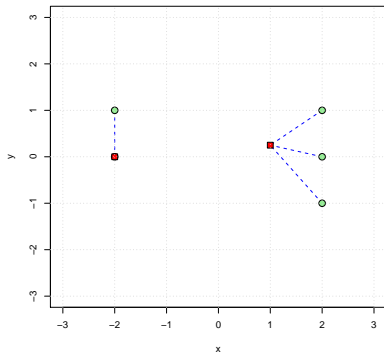


Centres Itération 1

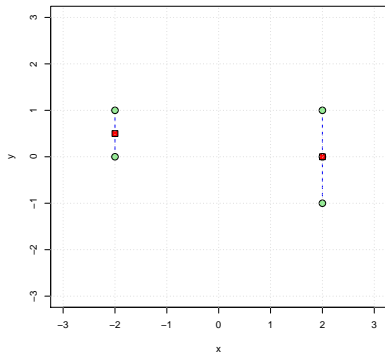


Exemple (3/3)

Groupes Itération 1

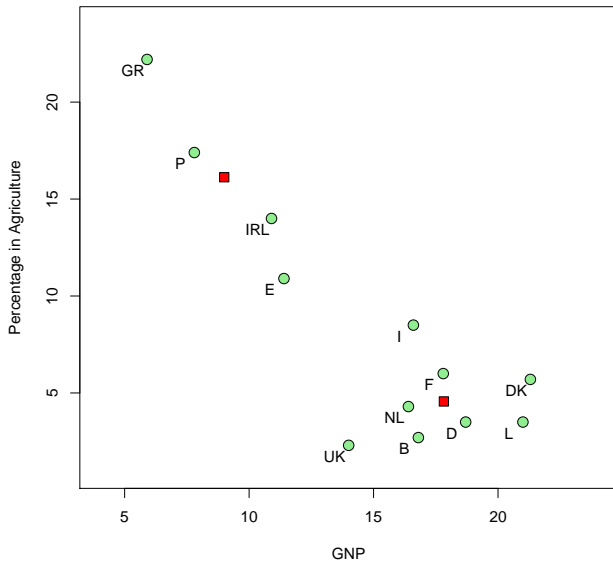


Groupes et Centres finaux



Exemple

EU in 1993



Le k -means en quantification

Quantification vectorielle :

- ▶ L'objet de la quantification est de **remplacer** un ensemble de données par une **représentation compacte**, sous la forme de **centroïdes** μ_1, \dots, μ_K .
- ▶ Une mesure de perte, ou de **distorsion**, est l'erreur quadratique moyenne.
- ▶ L'algorithme des k -means permet de sélectionner les centroïdes minimisant le critère quadratique de distorsion.

Application imagerie :

- ▶ Compression d'images ou de signaux

Les algorithmes combinatoires

Un **encodeur** $C : \{1, \dots, n\} \rightarrow \{1, \dots, K\}$ associe la $i^{\text{ème}}$ donnée au groupe $C(i)$. Un algorithme combinatoire a pour objet de minimiser, par rapport à C , l'inertie :

$$I_w(C) = \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} d^2(x_i, x_{i'}).$$



complexité calculatoire élevée k -means : alternative réaliste, correspondant à une suite particulière d'encodeurs

Géométrie des classes

Définition : Partition de Voronoi

Les K centres μ_1, \dots, μ_K induisent une partition de \mathbb{R}^d appelé la **partition de Voronoi** V_1, \dots, V_K , où :

- ▶ $V_k = \{x \in \mathbb{R}^d : \|x - \mu_k\| \leq \min_{\ell \neq k} \|x - \mu_\ell\|\}$
- ▶ $V_1 \cup \dots \cup V_K = \mathbb{R}^d$
- ▶ $V_k \cap V_\ell = \emptyset$, pour $k \neq \ell$ (aux bords près...).

Les V_k sont appelées **cellule** (de Voronoi)

Affectation des classes :

- ▶ la donnée x_i est affectée à la k -ième classe si $\|x_i - \mu_k\| \leq \min_{\ell \neq k} \|x_i - \mu_\ell\|$
- ▶ dans ce cas, x_i appartient à la **cellule** V_k

Rem: les cellules de Voronoi sont **convexes**

Convergence

Si la loi P est connue, alors il est possible de définir K centroïdes optimaux μ_1^*, \dots, μ_K^* tels que

$$\mathcal{E}(\mu_1^*, \dots, \mu_K^*) = \inf_{\mu_1, \dots, \mu_K} \mathcal{E}(\mu_1, \dots, \mu_K),$$

où

$$\mathcal{E}(\mu_1, \dots, \mu_K) = \mathbb{E}_P \left[\min_{1 \leq k \leq K} \|X - \mu_k\|^2 \right]$$

Théorème

Supposons que \mathcal{E} admette un minimum unique en $(\mu_1^*, \dots, \mu_K^*)$ (à une permutation d'indice près). Notons $(\hat{\mu}_{1,n}, \dots, \hat{\mu}_{K,n})$ un choix de centroïdes minimisant \mathcal{E}_n . Alors, pour tout $1 \leq k \leq K$, et à une permutation des indices près, on a $\hat{\mu}_{k,n} \longrightarrow \mu_k^*$, p.s.

Un cas simple : $\mathcal{U}([0; 1])$ et $K = 2$

- ▶ On considère la loi uniforme sur l'intervalle $[0; 1]$, et $K = 2$ classes, de centroïdes a et b .

- ▶ En supposant $a \leq b$, on a

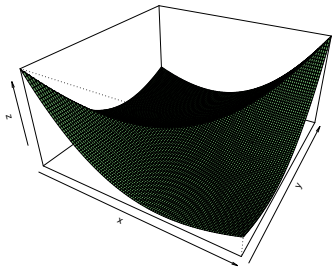
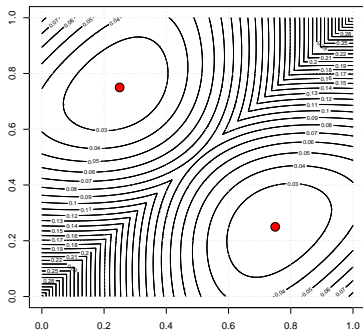
$$\mathcal{E}(a, b) = \frac{1}{3}a^3 + \frac{1}{3}(1 - b)^3 + \frac{1}{12}(b - a)^3.$$

- ▶ Il est facile de montrer que \mathcal{E} admet un minimum unique en

$$(a^*, b^*) = \left(\frac{1}{4}, \frac{3}{4}\right).$$

Exercice: prouver ce résultat

Un cas simple : Fonction de distorsion \mathcal{E}



Plan

Introduction

k -means

Modèles de mélanges gaussiens

Mélange de lois

Estimation des paramètres

Classification hiérarchique ascendante

Définition

Un mélange de lois gaussiennes est une loi dont la densité s'écrit :

$$f(x) = \sum_{m=1}^M \alpha_m \phi(x; \mu_m, \Sigma_m), \quad x \in \mathbb{R}^d,$$

où :

- (i) α_m sont les coefficients du mélange : $\alpha_m \geq 0$ et $\sum_{m=1}^M \alpha_m = 1$.
- (ii) $\phi(\cdot; \mu_m, \Sigma_m)$ est la densité de la loi gaussienne, de moyenne μ_m , et de matrice de covariance Σ_m .

Rem: autres familles de lois possibles (Cauchy, Laplace, t-student,)

Estimation des paramètres du modèle

Paramètres à estimer :

- ▶ les coefficients α_m
- ▶ les moyennes μ_m
- ▶ les matrices de covariance Σ_m ,
- ▶ (souvent) le nombre de composantes du mélange, M

Problème. L'estimation par maximum de vraisemblance est ardue.
Sur l'échantillon x_1, \dots, x_n , la vraisemblance s'écrit :

$$L(\theta) = \prod_{i=1}^n f(x_i; \theta) = \prod_{i=1}^n \left(\sum_{m=1}^M \alpha_m \phi(x_i; \mu_m, \Sigma_m) \right).$$

→ Pas de formule analytique pour $\hat{\mu}_m$ et $\hat{\Sigma}_m$ si $M > 1$.

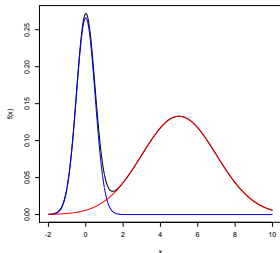
Utilisation pour le clustering

Maximum a posteriori :

Une fois le modèle ajusté : affecter x_i au groupe \hat{m}_i défini par

$$\hat{m}_i = \arg \max_m \hat{p}_{im} := \frac{\hat{\alpha}_m \phi(x_i; \hat{\mu}_m, \hat{\Sigma}_m)}{\sum_{r=1}^M \hat{\alpha}_r \phi(x_i; \hat{\mu}_r, \hat{\Sigma}_r)}$$

Exemple :



$$f(x) = \frac{1}{3} \phi(x; \mu_1, \sigma_1^2) + \frac{2}{3} \phi(x; \mu_2, \sigma_2^2),$$

avec $\mu_1 = 0$, $\sigma_1 = 0.5$, $\mu_2 = 5$, et
 $\sigma_2 = 2$.

Maximum a-posteriori et variables cachés

Variable cachée /variable latente :

Notons G_i la variable aléatoire donnant le groupe auquel la donnée x_i appartient, c'est une **variable cachée**, i.e., non observé, que l'on souhaite reconstruire

La probabilité que l'observation x_i soit dans le groupe m s'écrit :

$$\mathbb{P}(G_i = m | X_i = x_i) = \frac{\alpha_m \phi(x_i; \mu_m, \Sigma_m)}{\sum_{r=1}^M \alpha_r \phi(x_i; \mu_r, \Sigma_r)}.$$

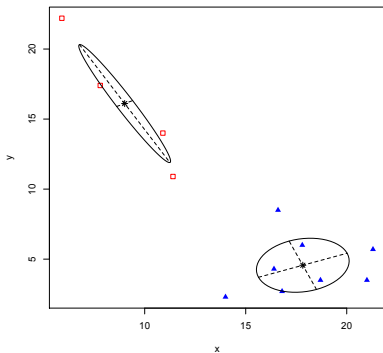
Ainsi

$$\hat{m}_i = \arg \max_m \hat{p}_{im}.$$

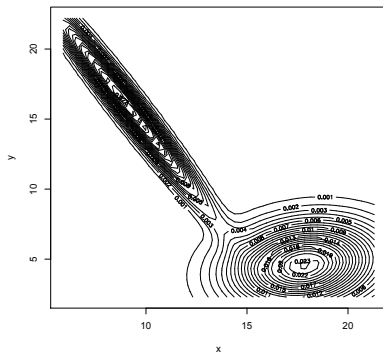
Example : EU in 1993

R package mclust.

Classification



Density Contour Plot



Lien avec les k -means

Méthode du k -means

- (i) Estimation de M centroïdes.
- (ii) Chaque donnée est affectée au centroïde le plus proche

Modèles de mélange :

- (i) Estimation M moyennes et matrices de covariance.
- (ii) Chaque donnée est affecté au groupe dont la composante du mélange est la plus probable

→ La partition obtenue dépend des centroïdes, mais également des matrices de covariances, qui déterminent la forme des groupes

Lien avec les estimateurs à noyau de la densité

Posons

$$\hat{f}_n(x) = \frac{1}{n} \sum_{i=1}^n \phi\left(\frac{x - x_i}{h}\right),$$

où ϕ est la densité de $\mathcal{N}(0, 1)$.

Dans ce mélange :

- ▶ autant de composantes que d'observations
- ▶ les coefficients α_i du mélange sont égaux à $1/n$
- ▶ les μ_m sont les observations : x_1, \dots, x_n
- ▶ h (qui joue le rôle des Σ_m) est une “taille de fenêtre” à choisir

Principe de l'algorithme EM

Maximisation directe de la vraisemblance difficile \implies approche alternée (comme pour k -means / algorithme de Lloyd)

Algorithme EM ( : *Expectation - Maximisation*) :

Initialisation : choix d'un mélange de départ.

Expectation Pour chaque donné x_i , calculer la probabilité que x_i soit dans le groupe m

Maximization Étant données les affectations des données en groupes, estimer les paramètres μ_m et Σ_m par maximum de vraisemblance

Itérer 2 et 3 jusqu'à convergence

Exemple en dimension 1 avec 2 composantes

$$f(x) = (1 - \pi)\phi(x; \mu_1, \sigma_1) + \pi\phi(x; \mu_2, \sigma_2)$$

Étape 1 : (Initialisation) $\pi^{(0)}$, $\mu_1^{(0)}$, $\sigma_1^{(0)}$, $\mu_2^{(0)}$, $\sigma_2^{(0)}$

Étape 2 : (Expectation) Connaissant $\pi^{(k)}$, $\mu_1^{(k)}$, $\sigma_1^{(k)}$, $\mu_2^{(k)}$, et $\sigma_2^{(k)}$, on estime la probabilité que x_i soit dans le groupe 2 par :

$$r_i^{(k)} = \frac{\pi^{(k)}\phi(x_i; \mu_2^{(k)}, \sigma_2^{(k)})}{(1 - \pi^{(k)})\phi(x_i; \mu_1^{(k)}, \sigma_1^{(k)}) + \pi^{(k)}\phi(x_i; \mu_2^{(k)}, \sigma_2^{(k)})}, \forall i$$

Exemple en dimension 1 avec 2 composantes

$$f(x) = (1 - \pi)\phi(x; \mu_1, \sigma_1) + \pi\phi(x; \mu_2, \sigma_2)$$

Étape 1 : (Initialisation) $\pi^{(0)}$, $\mu_1^{(0)}$, $\sigma_1^{(0)}$, $\mu_2^{(0)}$, $\sigma_2^{(0)}$

Étape 2 : (Expectation) Connaissant $\pi^{(k)}$, $\mu_1^{(k)}$, $\sigma_1^{(k)}$, $\mu_2^{(k)}$, et $\sigma_2^{(k)}$, on estime la probabilité que x_i soit dans le groupe 2 par :

$$r_i^{(k)} = \frac{\pi^{(k)}\phi(x_i; \mu_2^{(k)}, \sigma_2^{(k)})}{(1 - \pi^{(k)})\phi(x_i; \mu_1^{(k)}, \sigma_1^{(k)}) + \pi^{(k)}\phi(x_i; \mu_2^{(k)}, \sigma_2^{(k)})}, \forall i$$

Étape 2 : (Maximization step) nouvelles moyennes et variances :

$$\mu_1^{(k+1)} = \frac{\sum_{i=1}^n (1 - r_i^{(k)})x_i}{\sum_{i=1}^n (1 - r_i^{(k)})}, \quad \mu_2^{(k+1)} = \frac{\sum_{i=1}^n r_i^{(k)}x_i}{\sum_{i=1}^n r_i^{(k)}}$$
$$(\sigma_1^{(k+1)})^2 = \frac{\sum_{i=1}^n (1 - r_i^{(k)})(x_i - \mu_1^{(k)})^2}{\sum_{i=1}^n (1 - r_i^{(k)})}, \quad (\sigma_2^{(k+1)})^2 = \frac{\sum_{i=1}^n r_i^{(k)}(x_i - \mu_2^{(k)})^2}{\sum_{i=1}^n r_i^{(k)}}$$

$$\text{et enfin, probabilité du mélange : } \pi^{(k+1)} = \sum_{i=1}^n r_i^{(k)}.$$

Exemple en dimension 1 avec 2 composantes

$$f(x) = (1 - \pi)\phi(x; \mu_1, \sigma_1) + \pi\phi(x; \mu_2, \sigma_2)$$

Étape 1 : (Initialisation) $\pi^{(0)}$, $\mu_1^{(0)}$, $\sigma_1^{(0)}$, $\mu_2^{(0)}$, $\sigma_2^{(0)}$

Étape 2 : (Expectation) Connaissant $\pi^{(k)}$, $\mu_1^{(k)}$, $\sigma_1^{(k)}$, $\mu_2^{(k)}$, et $\sigma_2^{(k)}$, on estime la probabilité que x_i soit dans le groupe 2 par :

$$r_i^{(k)} = \frac{\pi^{(k)}\phi(x_i; \mu_2^{(k)}, \sigma_2^{(k)})}{(1 - \pi^{(k)})\phi(x_i; \mu_1^{(k)}, \sigma_1^{(k)}) + \pi^{(k)}\phi(x_i; \mu_2^{(k)}, \sigma_2^{(k)})}, \forall i$$

Étape 2 : (Maximization step) nouvelles moyennes et variances :

$$\mu_1^{(k+1)} = \frac{\sum_{i=1}^n (1 - r_i^{(k)})x_i}{\sum_{i=1}^n (1 - r_i^{(k)})}, \quad \mu_2^{(k+1)} = \frac{\sum_{i=1}^n r_i^{(k)}x_i}{\sum_{i=1}^n r_i^{(k)}}$$
$$(\sigma_1^{(k+1)})^2 = \frac{\sum_{i=1}^n (1 - r_i^{(k)})(x_i - \mu_1^{(k)})^2}{\sum_{i=1}^n (1 - r_i^{(k)})}, \quad (\sigma_2^{(k+1)})^2 = \frac{\sum_{i=1}^n r_i^{(k)}(x_i - \mu_2^{(k)})^2}{\sum_{i=1}^n r_i^{(k)}}$$

$$\text{et enfin, probabilité du mélange : } \pi^{(k+1)} = \sum_{i=1}^n r_i^{(k)}.$$

Complexité du modèle

Pour M composantes, avec $x \in \mathbb{R}^d$, les paramètres sont :

- ▶ M moyennes, soit $M \times d$ réels
- ▶ M matrice de covariances, soit $M \times d(d+1)/2$ réel
- ▶ $(M-1)$ coefficients α_m

Exemple :

- (i) Avec $M = 3$ composantes, en dimension $d = 4$, on a $3 \times 4 + 3 \times 20/2 + 2 = 44$ paramètres réels à estimer.
- (ii) Si l'on dispose de $n = 150$ observations, on a donc 600 “nombres” dans le jeu de données, pour estimer 44 paramètres, soit environ :

13.6 “nombres” par paramètre à estimer...

Hypothèses sur la variance

Pour simplifier, rajouter des hypothèses sur les Σ_m , e.g., :

Famille sphérique :

- ▶ $\Sigma_1 = \Sigma_2 = \dots \Sigma_M = \sigma^2 \text{Id}_d$
- ▶ Pour chaque m , $\Sigma_m = \sigma_m^2 \text{Id}_d$

Famille diagonale :

- ▶ $\Sigma_1 = \Sigma_2 = \dots \Sigma_M = \text{diag}(\sigma_1^2, \dots, \sigma_d^2)$
- ▶ $\forall m, \Sigma_m = \text{diag}(\sigma_{1,m}^2, \dots, \sigma_{d,m}^2)$

Matrice de covariance

Rappel ; une matrice symétrique est diagonalise en base orthonormale.

On peut décrire une matrice de covariance par :

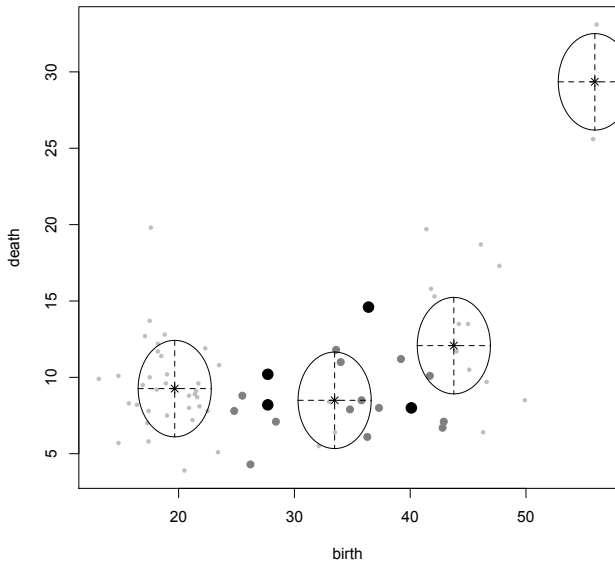
- ▶ son déterminant : \sim le volume
- ▶ ses valeurs propres : \sim la forme
- ▶ ses vecteurs propres normalisés : \sim l'orientation

$$\Sigma_m = v_m U_n D_m U_n^\top$$

- ▶ volume : $v_m = (\det(\Sigma_m))^{1/d}$
- ▶ forme : D_m matrice diagonale des valeurs propres, normalisée par $1/v_m$
- ▶ orientation : U_n matrice des vecteurs propres

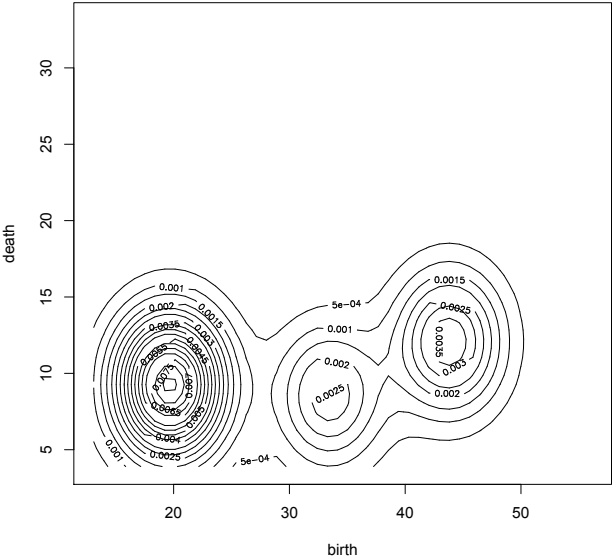
Clustering result II

Classification Uncertainty



Clustering result III

Density Contour Plot



Plan

Introduction

k -means

Modèles de mélanges gaussiens


Classification hiérarchique ascendante

Classification Hiérarchique






Principe :

- ▶ Former une **structure hiérarchique** des données allant de n groupes à 1 groupe. Les données ne sont donc pas partitionnées en une seule étape.
- ▶ On distingue les méthodes :
 - ▶ *ascendantes* : série de fusions de n à 1 groupes
 - ▶ *descendantes* : série de divisions de 1 à n groupes

Rem:

- ▶ Le résultat de la classification est représenté graphiquement sous la forme d'un **dendrogramme**.
- ▶ Les fusions (ou divisions) sont effectuées successivement selon une mesure de dissimilarité entre classes. On parle également de **lien** ( : (*linkage*).

Liens usuels

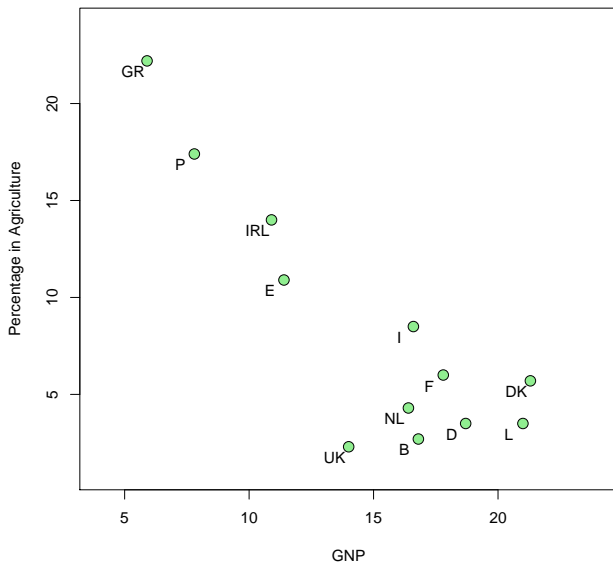
- (i) **Lien simple** ( : *single linkage*) : distance du plus proche voisin
- (ii) **Lien complet** ( : *complete linkage*) : distance du diamètre maximum
- (iii) **Lien moyen** ( : *average / group-average linkage*) : distance moyenne
- (iv) **Lien entre centroïdes** ( : *centroid linkage*) : distance entre barycentres
- (v) **Lien de Ward** ( : *Ward's linkage*) : distance de Ward.

Rem:

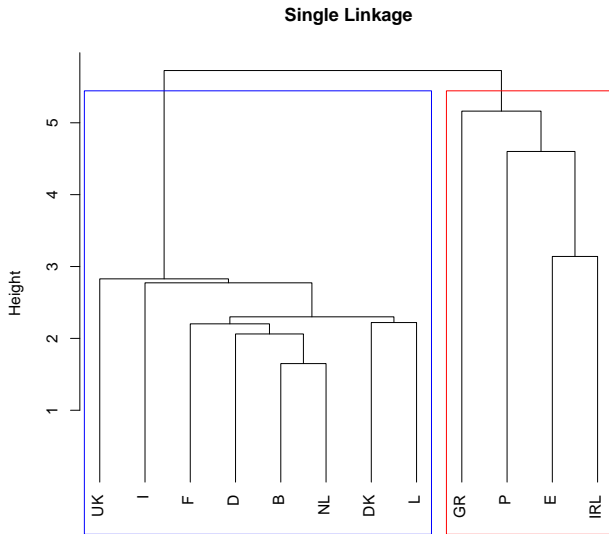
- ▶ Ces liens conduisent (typiquement) à des hiérarchies différentes
- ▶ Les classifications associées possèdent également des propriétés différentes

Exemple

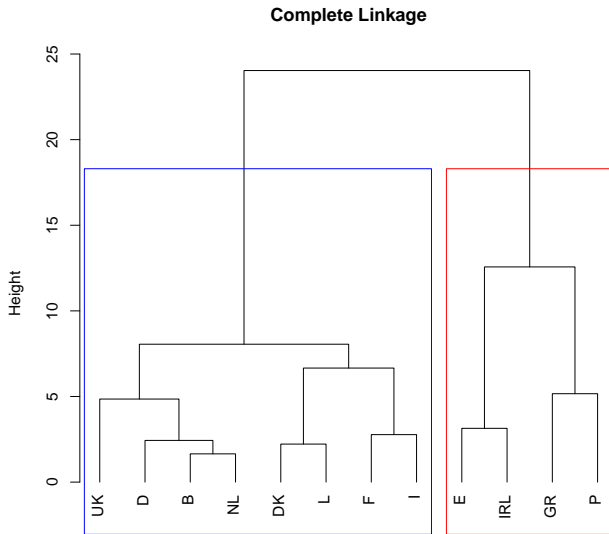
EU in 1993



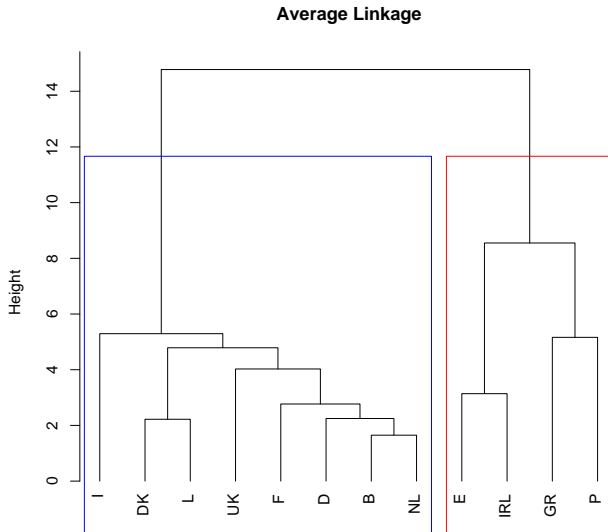
Lien simple



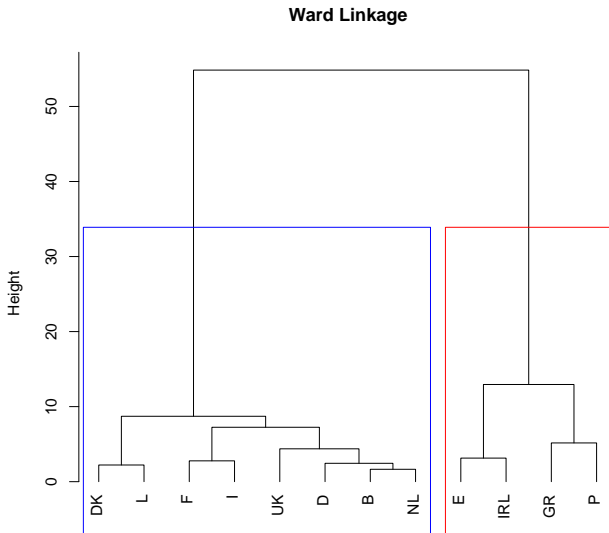
Lien complet



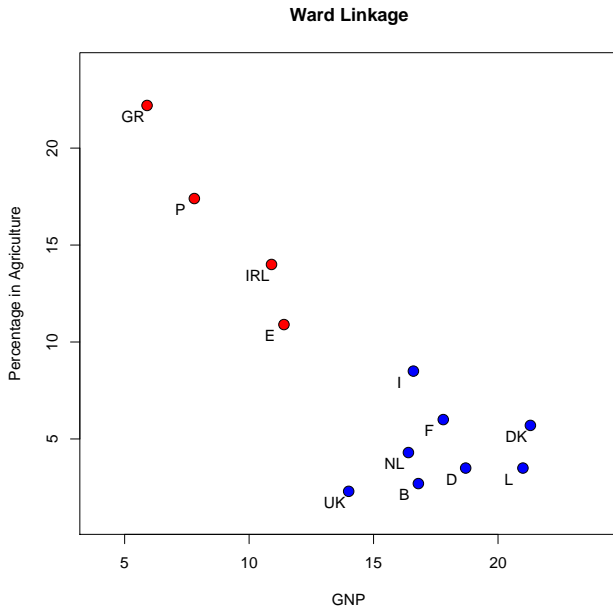
Lien moyen



Lien de Ward



Résultat pour deux groupes



Algorithme basique de la CAH

Principe :

1. **Initialisation** : former $K = n$ groupes $\mathcal{C}_1, \dots, \mathcal{C}_n$ contenant chacun un point
2. **Sélection** : trouver les deux clusters (e.g., \mathcal{C}_i et \mathcal{C}_j) les plus proches
3. **Fusion** : fusionner \mathcal{C}_i et \mathcal{C}_j , puis diminuer K de 1
4. **Itération** : itérer sur les points 2 et 3 jusqu'à $K = 1$

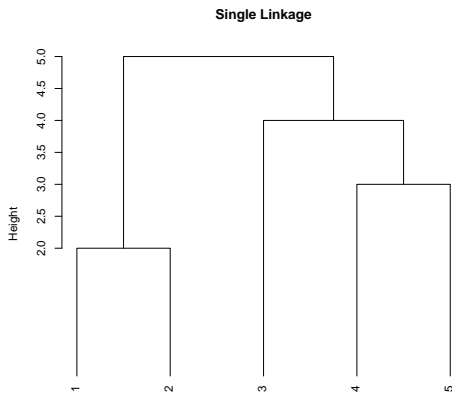
Construction du dendrogramme :

- Lors de la fusion de deux groupes, ces derniers sont reliés à une **hauteur** h correspondant à leur **dissimilarité**

Construction du dendrogramme : lien simple

Matrice de dissimilarités :

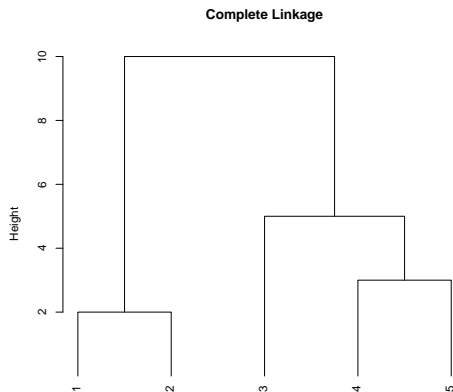
$$D = \begin{pmatrix} 0 & 2 & 6 & 10 & 9 \\ 2 & 0 & 5 & 9 & 8 \\ 6 & 5 & 0 & 4 & 5 \\ 10 & 9 & 4 & 0 & 3 \\ 9 & 8 & 5 & 3 & 0 \end{pmatrix}$$



Construction du dendrogramme : lien complet

Matrice de dissimilarités :

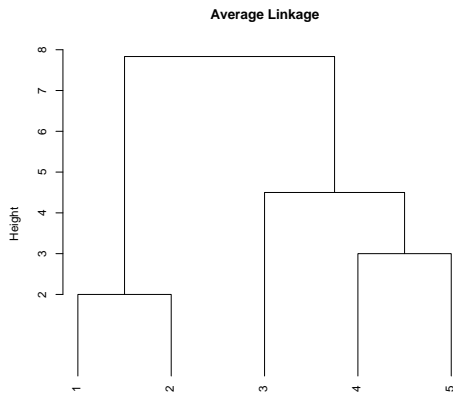
$$D = \begin{pmatrix} 0 & 2 & 6 & 10 & 9 \\ 2 & 0 & 5 & 9 & 8 \\ 6 & 5 & 0 & 4 & 5 \\ 10 & 9 & 4 & 0 & 3 \\ 9 & 8 & 5 & 3 & 0 \end{pmatrix}$$



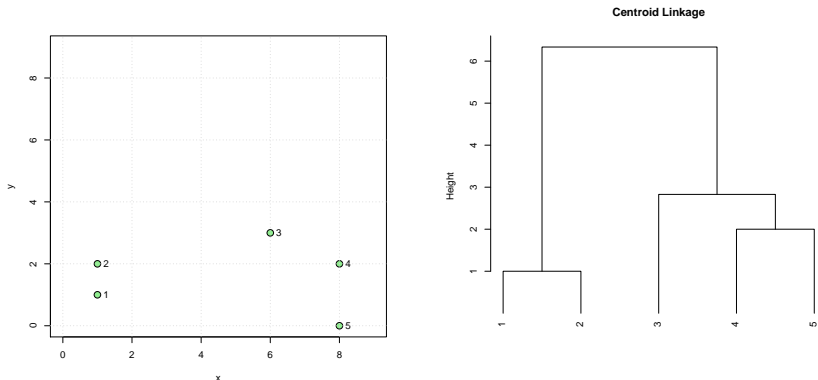
Construction du dendrogramme : lien moyen

Matrice de dissimilarités :

$$D = \begin{pmatrix} 0 & 2 & 6 & 10 & 9 \\ 2 & 0 & 5 & 9 & 8 \\ 6 & 5 & 0 & 4 & 5 \\ 10 & 9 & 4 & 0 & 3 \\ 9 & 8 & 5 & 3 & 0 \end{pmatrix}$$



Construction du dendrogramme : lien centroïde



1(1,1) 2(1,2) 3(6,3) 4(8,2) 5(8,0).

TO DO: Corriger cet exemple avec
`sklearn.cluster.AgglomerativeClustering` et

<https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/>

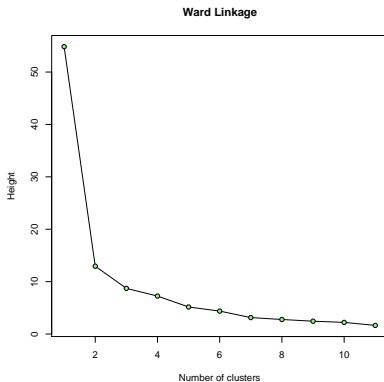
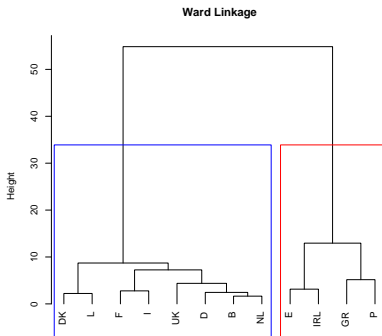
Construction des classes

Stratégies

- ▶ Couper le dendrogramme à un niveau de similarité fixé.
- ▶ Choisir à l'avance le nombre de groupes, et couper à un niveau convenable.
- ▶ Utiliser une heuristique pour sélectionner le nombre de groupes / niveau, par exemple, en comparant les différences de similarités (dissimilarités) entre deux fusions successives.

Mais : Monotonicité ?

Exemple : EU in 1993



Le choix de $K = 2$ classes paraît convenable.

Monotonicit  et inversion

Propri t  de monotonicit 

Un algorithme de CAH est dit monotone si les dissimilarit s h_1, \dots, h_{n-1} des fusions successives sont telles que

$$h_1 \leq h_2 \leq \dots \leq h_{n-1}.$$

Dans le cas contraire, on parle d'**inversion**

CAH monotones

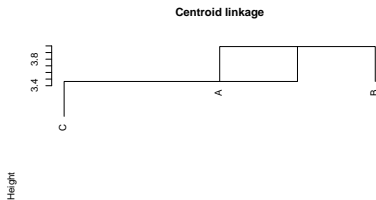
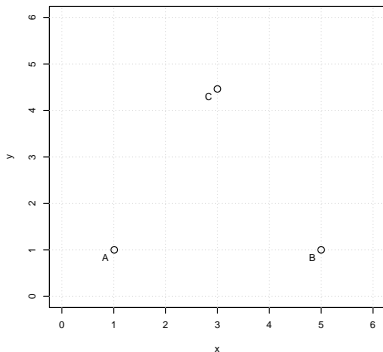
- lien simple
- lien complet
- lien moyen

CAH **non-monotone**

- lien entre centroides

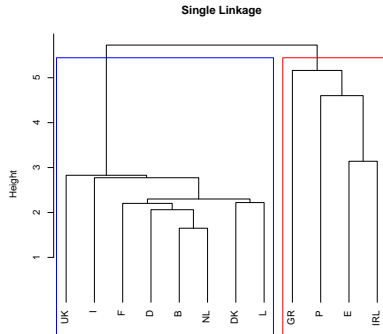
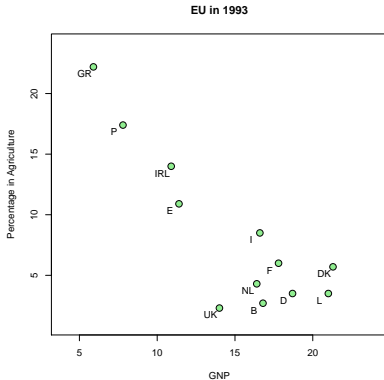
Exemple d'inversion

$$A(1.01, 1) \quad B(5, 1) \quad C(3, 1 + 2\sqrt{3})$$

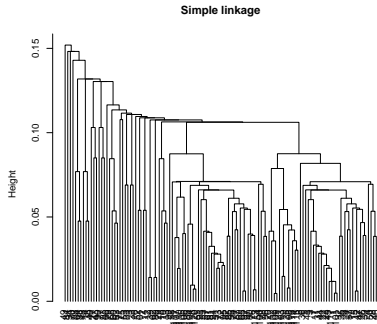
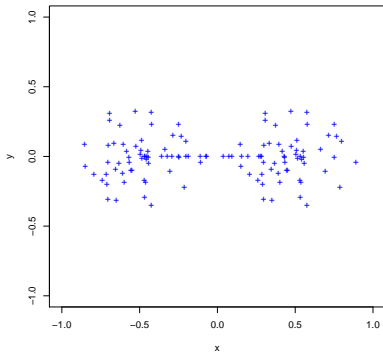


Comment interpreter le dendrogramme en présence d'une inversion ?

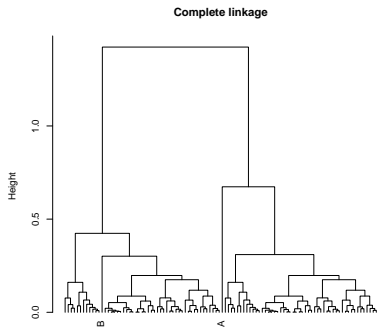
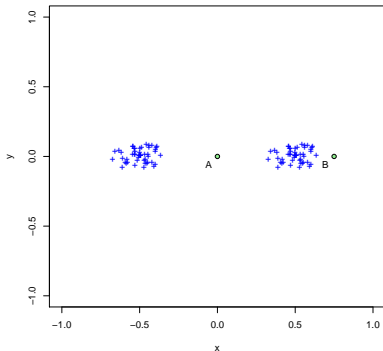
CAH lien simple : effet de chaînage



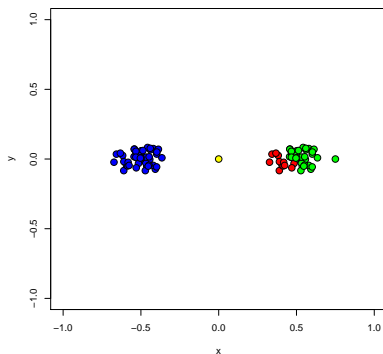
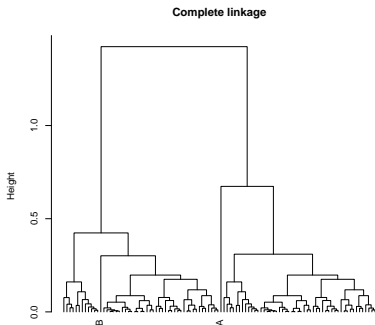
CAH lien simple : effet de chaînage



CAH lien complet : influence des outliers



CAH lien complet : influence des outliers



Coupe avec $K = 4$ groupes : la structure est perdue !

Complexité algorithmique

- ▶ L'algorithme “naïf” de CAH a une complexité algorithmique en $O(n^3)$:
 - ▶ $n \times n$ matrice de dissimilarité ;
 - ▶ $n - 1$ itérations.
- ▶ Il est possible toutefois d'améliorer les algorithmes pour obtenir une complexité en $O(n^2 \log n)$, et $O(n^2)$ pour la CAH par lien simple.
- ▶ Les quatres liens usuels :
 - ▶ *lien simple* ;
 - ▶ *lien complet* ;
 - ▶ *lien moyen* ;
 - ▶ *lien par centroides* ;ont donc une complexité calculatoire équivalente.

Conclusion

Lien simple

- ▶ **pros** : interprétation en terme de graphe ; CAH monotone.
- ▶ **cons** : effet de chaînage, ; fusions locales uniquement.

Lien complet

- ▶ **pros** : CAH monotone ; fusions globales.
- ▶ **cons** : sensibilité aux outliers.

Lien moyen et centroïde

- ▶ **pros** : lien moyen monotone ; fusions globales ; compromis entre lien simple et lien complet.
- ▶ **cons** : tendance à former des groupes compacts (sens usuel) ; lien centroïde non-monotone.