Analyse des données

Apprentissage supervisé et non supervisé

[Apprentissage non supervisé : Clustering]

3ème année ENSEIRB-MATMECA - CISD

Méthodes de Machine Learning (Apprentissage Automatique)

Non supervisées

$X^1, \ldots, X^p$: variables (quantitatives ou qualitatives)

Clustering

Création d'une nouvelle variable qualitative

Exemple : k-means, CAH, GMM,…

Réduction de dimension

Création de nouvelles variables quantitatives qui résument $X^1, \ldots, X^p$.

Exemples de méthodes linéaires :

- *ACP si les données sont quantitatives*
- *ACM si les données sont qualitatives*
- *ACPmixte si les données sont mixtes*

Exemple de méthodes non linéaire : AutoEncoders,…

Supervisées

$X^1, \ldots, X^p$: variables d'entrées

Y : variable de sortie (quantitative ou qualitative

Régression : Y quantitatif

Exemples de méthodes linéaires :

- *Régression linéaire simple et multiple si entrées quantitatives*
- *ANOVA si entrées qualitatives*
- *ANCOVA si les entrées sont mixtes*

Exemples de méthodes non linéaires :

- *Arbre de décision et forêts aléatoires (entrées mixtes)*
- *SVM, réseaux de neurones*

Classification : Y qualitatif

Exemples de méthodes linéaires :

- *Régression logistique (Y binaire)*
- *LDA et QDA (entrées quantitatives)*

Exemples de méthodes non linéaires :

- *KNN, réseaux de neurone (entrées quantitatives)*
- *Bayésien naif, arbres et forêts aléatoires (entrées mixtes)*

Introduction

- Les méthodes de clustering prennent en entrée une matrice de données $\mathbf{X}$ de dimension $n \times p$.
- Les lignes $\mathbf{x}_1, ..., \mathbf{x}_n$ de cette matrice sont n observations d'un vecteur $(X^1, ..., X^p)$ de p variables aléatoires.
- L'objectif du clustering : trouver des sous-groupes homogènes au sein de ces observations c'est à dire une partition de l'ensemble $\Omega = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ des observations (des individus).
- Un sous-groupe s'appelle une classe.

Remarques :

- Clustering = Classification non supervisée = création de classes d'observations qui se ressemblent (création d'une nouvelle variable qualitative Y).
- Clustering $\neq$ Classification supervisée = prédiction pour une nouvelle observation de sa classe d'appartenance (variable qualitative Y connue).
- Clustering $\neq$ ACP = création de nouvelles variables quantitatives, combinaison des variables $(X^1, ..., X^p)$ résumant le mieux possibles les données.

Il existe de nombreux algorithmes de clustering qui se distinguent par :

- la nature des données : quantitatives, qualitatives ou mixtes,
- la nature de la structure de classification : partition ou hiérarchie,
- la nature de l'approche utilisée : approche géométrique (distance, dissimilarité, similarité) ou approche probabiliste (modèles de mélange).

Ici, on s'intéresse au clustering de données quantitatives, à l'aide d'approches géométriques utilisant les distances Euclidiennes.

Plan

Les données en entrée

Comme en ACP, les données en entrée sont quantitatives.

	$X^1 \ldots$	X^j	$\ldots X^p$
1			
$\vdots$		$\vdots$	
i	$\ldots$	x_{ij}	$\ldots$
$\vdots$		$\vdots$	
n			

On notera :

$\mathbf{X} = (x_{ij})_{n \times p}$ la matrice des données brutes où $x_{ij} \in \mathbb{R}$.

$\Omega = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ l'ensemble des observations $\mathbf{x}_i \in \mathbb{R}^p$ (les lignes de $\mathbf{X}$) dont on cherche une partition en K classes.

On notera w_i le poids de l'observation i avec en général :

- $w_i = \frac{1}{n}$ (ou $w_i = 1$) pour des observations aléatoires,
- $w_i \neq \frac{1}{n}$ pour des données agrégées ou redressées.

A SAVOIR :

Afin de donner la même importance à toutes les variables dans le calcul des distances Euclidiennes, les données sont souvent standardisées avant d'appliquer le clustering.

	$Z^1 \ldots$	Z^j	$\ldots Z^p$	
1				
$\vdots$		$\vdots$		
i	$\ldots$	$z_{ij} = \frac{x_{ij} - \bar{x}^j}{s_j}$	$\ldots$	
$\vdots$		$\vdots$		
n				
$\bar{\mathbf{z}}$	$\ldots$	0	$\ldots$	
s	$\ldots$	1	$\ldots$	

avec :

$\bar{x}^j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}$ la moyenne de la variable j

$s_j = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \bar{x}^j)^2}$ l'écart-type de la variable j

$\mathbf{Z} = (z_{ij})_{n \times p}$ la matrice des données standardisées.

$\Omega = \{\mathbf{z}_1, ..., \mathbf{z}_n\}$ est alors l'ensemble des observations (lignes de $\mathbf{Z}$) dont on cherche une partition en K classes.

Un exemple avec R

Le jeu de donnée indique la quantité de protéines consommée dans 9 types d'aliments dans 25 (anciens) pays européens : 25 observations et 9 variables quantitatives.

	Red.Meat	White.Meat	Eggs	Milk	Fish	Cereals	Starchy.Foods	Nuts	Fruite.veg.
Alban	10.1	1.4	0.5	8.9	0.2	42	0.6	5.5	1.7
Aust	8.9	14.0	4.3	19.9	2.1	28	3.6	1.3	4.3
Belg	13.5	9.3	4.1	17.5	4.5	27	5.7	2.1	4.0
Bulg	7.8	6.0	1.6	8.3	1.2	57	1.1	3.7	4.2
Czech	9.7	11.4	2.8	12.5	2.0	34	5.0	1.1	4.0
Den	10.6	10.8	3.7	25.0	9.9	22	4.8	0.7	2.4
E_Ger	8.4	11.6	3.7	11.1	5.4	25	6.5	0.8	3.6
Finl	9.5	4.9	2.7	33.7	5.8	26	5.1	1.0	1.4
Fr	18.0	9.9	3.3	19.5	5.7	28	4.8	2.4	6.5
Greece	10.2	3.0	2.8	17.6	5.9	42	2.2	7.8	6.5
Hung	5.3	12.4	2.9	9.7	0.3	40	4.0	5.4	4.2
Ireland	13.9	10.0	4.7	25.8	2.2	24	6.2	1.6	2.9
Italy	9.0	5.1	2.9	13.7	3.4	37	2.1	4.3	6.7
Nether	9.5	13.6	3.6	23.4	2.5	22	4.2	1.8	3.7
Nor	9.4	4.7	2.7	23.3	9.7	23	4.6	1.6	2.7
Pol	6.9	10.2	2.7	19.3	3.0	36	5.9	2.0	6.6
Port	6.2	3.7	1.1	4.9	14.2	27	5.9	4.7	7.9
Rom	6.2	6.3	1.5	11.1	1.0	50	3.1	5.3	2.8
Spain	7.1	3.4	3.1	8.6	7.0	29	5.7	5.9	7.2
Swed	9.9	7.8	3.5	24.7	7.5	20	3.7	1.4	2.0
Switz	13.1	10.1	3.1	23.8	2.3	26	2.8	2.4	4.9
UK	17.4	5.7	4.7	20.6	4.3	24	4.7	3.4	3.3
USSR	9.3	4.6	2.1	16.6	3.0	44	6.4	3.4	2.9
W_Ger	11.4	12.5	4.1	18.8	3.4	19	5.2	1.5	3.8
Yugo	4.4	5.0	1.2	9.5	0.6	56	3.0	5.7	3.2

$\Rightarrow \Omega = \{\mathbf{x}_1, \ldots, \mathbf{x}_{25}\}$: 25 points de $\mathbb{R}^9$ pondérés par $w_i = \frac{1}{25}$ (ou $w_i = 1$).

@

```r
load("protein.rda")
```

```r
# Ecart-types des 9 variables (colonnes de X)
apply(protein, 2, sd)
```

```
##      Red.Meat    White.Meat          Eggs          Milk          Fish       Cereals Starchy.Foods
##           3.3           3.7           1.1           7.1           3.4          11.0           1.6
##          Nuts   Fruite.veg.
##           2.0           1.8
```

```r
# Standardisation des données
Z <- scale(protein)
```

```r
# moyenne des 9 variables standardisées (colonnes de Z)
apply(Z, 2, mean)
```

```
##      Red.Meat    White.Meat          Eggs          Milk          Fish       Cereals Starchy.Foods
##      -3.4e-16       2.9e-16      -3.1e-16      -2.0e-16       8.9e-17      -3.7e-16      -3.8e-16
##          Nuts   Fruite.veg.
##      -2.2e-16      -6.1e-16
```

```r
# Ecart-types des 9 variables standardisées (colonnes de Z)
apply(Z, 2, sd)
```

```
##      Red.Meat    White.Meat          Eggs          Milk          Fish       Cereals Starchy.Foods
##             1             1             1             1             1             1             1
##          Nuts   Fruite.veg.
##             1             1
```

Données standardisées.

	Red.Meat	White.Meat	Eggs	Milk	Fish	Cereals	Starchy.Foods	Nuts	Fruite.veg.
Alban	0.08	-1.76	-2.18	-1.16	-1.20	0.92	-2.25	1.22	-1.35
Aust	-0.28	1.65	1.22	0.39	-0.64	-0.39	-0.41	-0.89	0.09
Belg	1.10	0.38	1.04	0.05	0.06	-0.51	0.87	-0.49	-0.08
Bulg	-0.61	-0.51	-1.20	-1.24	-0.91	2.23	-1.94	0.32	0.04
Czech	-0.04	0.95	-0.12	-0.65	-0.67	0.19	0.44	-0.99	-0.08
Den	0.23	0.79	0.68	1.11	1.65	-0.94	0.32	-1.19	-0.96
E_Ger	-0.43	1.00	0.68	-0.85	0.33	-0.70	1.36	-1.14	-0.30
Finl	-0.10	-0.81	-0.21	2.33	0.45	-0.54	0.50	-1.04	-1.52
Fr	2.44	0.54	0.33	0.34	0.42	-0.38	0.32	-0.34	1.31
Greece	0.11	-1.33	-0.12	0.07	0.47	0.86	-1.27	2.38	1.31
Hung	-1.35	1.22	-0.03	-1.04	-1.17	0.72	-0.17	1.17	0.04
Ireland	1.22	0.57	1.58	1.22	-0.61	-0.75	1.18	-0.74	-0.69
Italy	-0.25	-0.76	-0.03	-0.48	-0.26	0.41	-1.33	0.62	1.42
Nether	-0.10	1.54	0.59	0.88	-0.52	-0.90	-0.05	-0.64	-0.24
Nor	-0.13	-0.87	-0.21	0.87	1.59	-0.84	0.20	-0.74	-0.80
Pol	-0.87	0.62	-0.21	0.31	-0.38	0.35	0.99	-0.54	1.37
Port	-1.08	-1.14	-1.64	-1.72	2.91	-0.48	0.99	0.82	2.09
Rom	-1.08	-0.43	-1.28	-0.85	-0.97	1.58	-0.72	1.12	-0.74
Spain	-0.82	-1.22	0.15	-1.20	0.80	-0.28	0.87	1.42	1.70
Swed	0.02	-0.03	0.50	1.07	0.95	-1.16	-0.35	-0.84	-1.18
Switz	0.98	0.60	0.15	0.94	-0.58	-0.61	-0.90	-0.34	0.42
UK	2.26	-0.59	1.58	0.49	0.00	-0.72	0.26	0.17	-0.46
USSR	-0.16	-0.89	-0.75	-0.07	-0.38	1.03	1.30	0.17	-0.69
W_Ger	0.47	1.25	1.04	0.24	-0.26	-1.24	0.57	-0.79	-0.19
Yugo	-1.62	-0.78	-1.55	-1.07	-1.08	2.16	-0.78	1.32	-0.52

$\Rightarrow \Omega = \{\mathbf{z}_1, \ldots, \mathbf{z}_{25}\}$ est un ensemble de 25 points de $\mathbb{R}^9$ dont on cherche une partition.

Plan

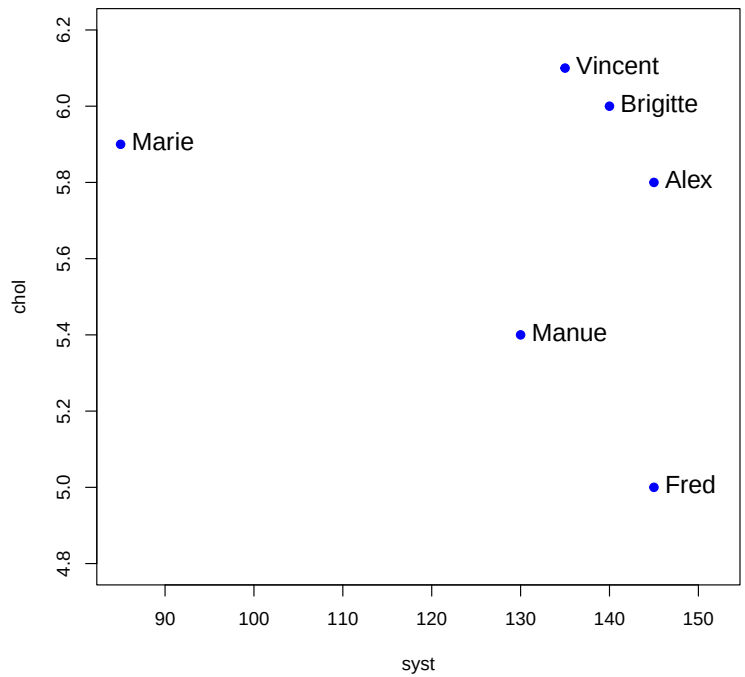
Les K-means

Une partition en K classes de Ω est un ensemble de classes non vides, deux à deux disjointes et dont la réunion forme Ω.

On notera $P_K = (C_1, \ldots, C_k, \ldots, C_K)$.

Proposer une "bonne" et une "mauvaise" partition $P_3 = (C_1, C_2, C_3)$ en 3 classes des 6 observations de $\mathbb{R}^2$ ci-dessous.

	syst	chol
Brigitte	140	6.0
Marie	85	5.9
Vincent	135	6.1
Alex	145	5.8
Manue	130	5.4
Fred	145	5.0

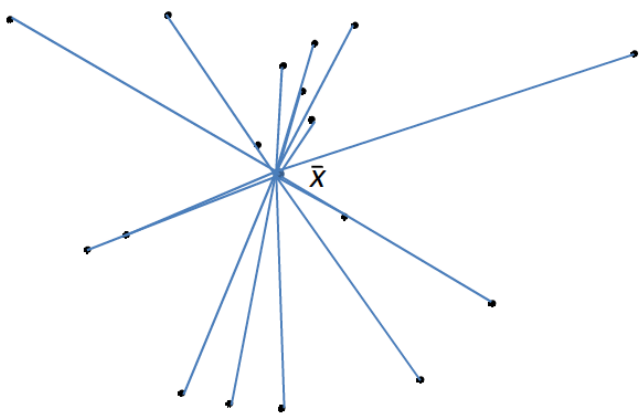


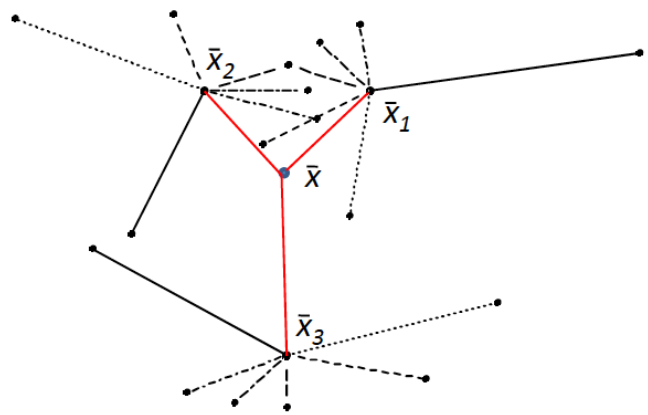
La méthode des K-means cherche une partition $P_K = (C_1, ..., C_K)$ de Ω qui minimise l'inertie intra-classe c'est à dire la variabilité des observations dans les classes :

$$\min_{(C_1,\ldots,C_K)} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} w_i d^2(\mathbf{x}_i, \bar{\mathbf{x}}_k)$$

où $d(\mathbf{x}_i, \bar{x}_k) = \|\mathbf{x}_i - \bar{\mathbf{x}}_k\|$ est la distance Euclidienne entre le point $\mathbf{x}_i = (x_{i1}, ..., x_{ip})^T$ et le centre de gravité $\bar{\mathbf{x}}_k = (\bar{x}_k^1, ..., \bar{x}_k^p)^T$ de la classe C_k :

$$\bar{\mathbf{x}}_k = \frac{\sum_{\mathbf{x}_i \in C_k} w_i \mathbf{x}_i}{\sum_{\mathbf{x}_i \in C_k} w_i}.$$





On peut montrer que

$$\underbrace{\sum_{i=1}^{n} w_i d^2(x_{i.}, \bar{x})}_{\text{Inertie totale}} = \underbrace{\sum_{k=1}^{K} \sum_{i \in C_k} w_i d^2(x_{i.}, \bar{x}_k)}_{\text{Inertie intra-classe}} + \underbrace{\sum_{k=1}^{K} \mu_k d^2(\bar{x}_k, \bar{x})}_{\text{Inertie inter-classe}}$$

où

- $\mu_k = \sum_{\mathbf{x}_i \in C_k} w_i$ est le poids de la classe C_k,
- $\bar{\mathbf{x}} = (\bar{x}^1, ..., \bar{x}^p)^T$ est le centre de gravité de Ω.

A SAVOIR :

L'inertie totale est indépendante de la partition. Donc la partition P_K qui minimise l'inertie intra-classe (l'hétérogénéité des classes) maximise l'inertie inter-classe (la séparation des classes).

L'algorithme des K-means

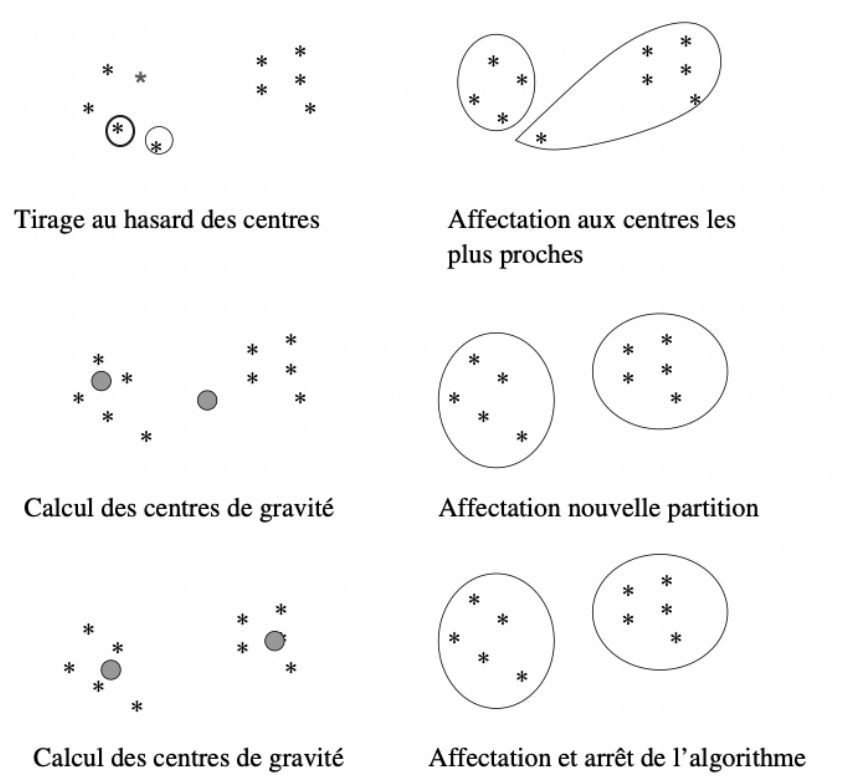
En entrée : l'ensemble $\Omega = (\mathbf{x}_1, ..., \mathbf{x}_n)$ de n points de $\mathbb{R}^p$ et le nombre K de classes.

1. On initialise les centres de gravité $\bar{\mathbf{x}}_k$ des classes en tirant aléatoirement K points dans Ω.
2. On répète jusqu'à la convergence :
   a) Chaque observation $\mathbf{x}_i$ de Ω est assignée à la classe C_k dont le centre $\bar{\mathbf{x}}_k$ est le plus proche (au sens de la distance Euclidienne).
   b) On met à jour les centre de gravités des classes.

En sortie : une partition $P_K = (C_1, ..., C_K)$ qui minimise localement l'inertie intra-classe.

L'algorithme a convergé lorsque aucune observation n'a changé de classe entre deux itérations successives.

Un petit exemple



Propriétés de l'algorithme des K-means

- L'algorithme converge vers une partition réalisant un minimum local de l'inertie intra-classe.
- La partition finale dépend des centres initiaux. Si on relance l'algorithme avec une autre initialisation, la partition finale peut être différente. En pratique :
  - On lance N fois l'algorithme avec des initialisations aléatoires différentes.
  - On retient parmi les N partitions finales la meilleure c'est à dire celle ayant l'inertie intra-classe la plus petite.
- La complexité de l'algorithme est $o(KpnT)$ où T est le nombre d'itérations. L'algorithme s'applique donc à des grands jeux de données (et peut être facilement parallélisé).
- La difficulté de cet méthode est qu'il faut connaitre le nombre de classes K.

Un exemple avec R

K-means des données `protein` non standardisées avec $K = 4$ classes

```
load("protein.rda")
```

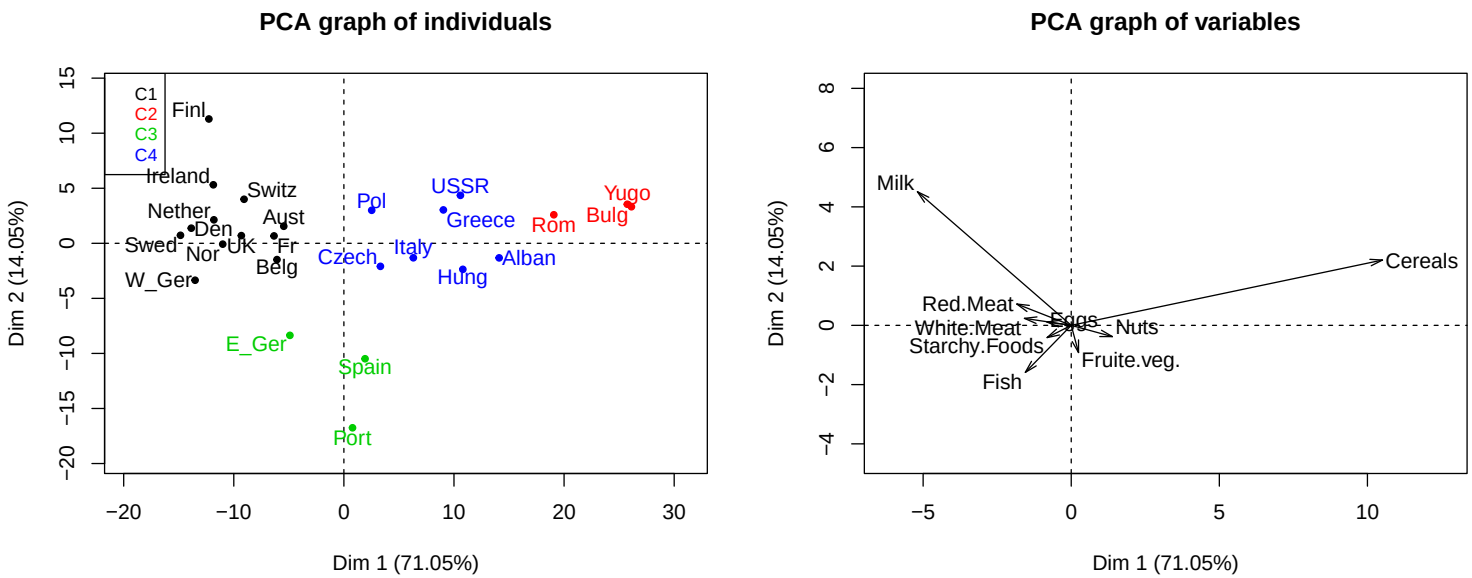
```
# K-means avec K=4 classes et N=5 initialisations

res <- kmeans(protein, centers = 4, nstart = 5)
P4 <- res$cluster
```

```
P4 <- as.factor(P4)
levels(P4) <- paste("C", 1:4, sep="")
P4
```

```
##   Alban    Aust    Belg    Bulg   Czech     Den   E_Ger    Finl      Fr  Greece    Hung Ireland
##      C4      C1      C1      C2      C4      C1      C3      C1      C1      C4      C4      C1
##   Italy  Nether     Nor     Pol    Port     Rom   Spain    Swed   Switz      UK    USSR   W_Ger
##      C4      C1      C1      C4      C3      C2      C3      C1      C1      C1      C4      C1
##    Yugo
##      C2
## Levels: C1 C2 C3 C4
```

```r
# ACP non normée (projection des données centrées)

library(FactoMineR)
res <- PCA(data.frame(P4, protein), scale.unit = FALSE, quali.sup = 1, graph = FALSE)
```

```r
# Plans factoriel de l'ACP non normée avec les pays colorés en fonction de leur classe

par(mfrow=c(1,2))
plot(res, habillage = 1, invisible = "quali", graph.type = "classic")
plot(res, choix = "var", graph.type = "classic")
```



Interprétation ? Conclusion ?

K-means des données `protein` standardisées avec $K = 4$ classes

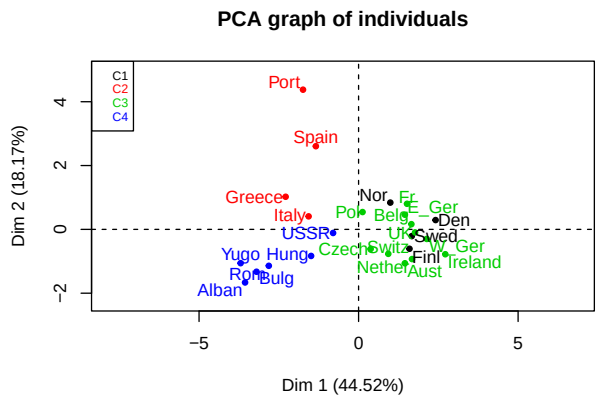
```
# Standardisation des données

Z <- scale(protein)
```

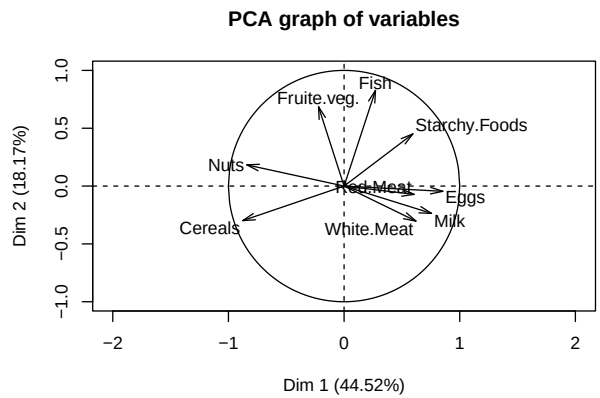
```
# K-means avec K=4 classes et N=5 initialisations

res <- kmeans(Z, centers = 4, nstart = 5)
P4 <- res$cluster
```

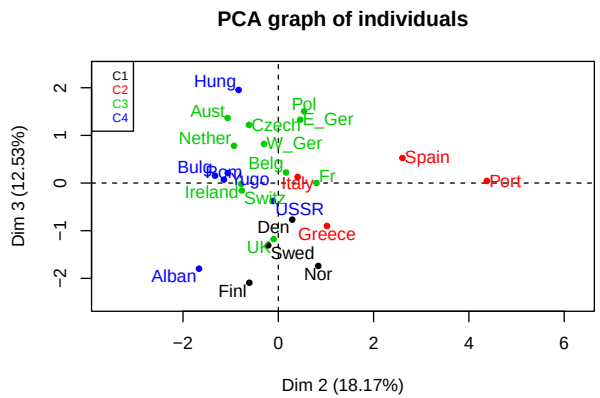
```
P4 <- as.factor(P4)
levels(P4) <- paste("C", 1:4, sep="")
P4
```

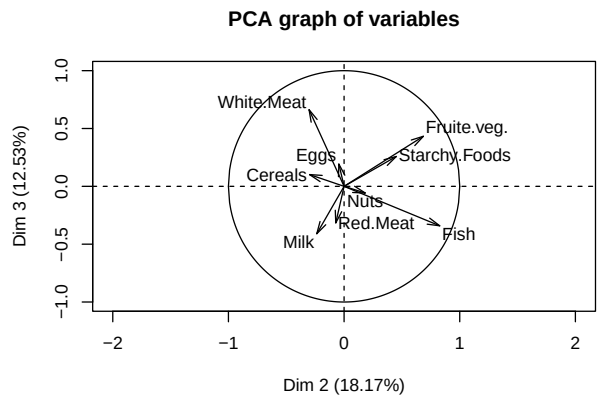
```
##   Alban    Aust    Belg    Bulg   Czech     Den   E_Ger    Finl      Fr  Greece    Hung Ireland
##      C4      C3      C3      C4      C3      C1      C3      C1      C3      C2      C4      C3
##   Italy  Nether     Nor     Pol    Port     Rom   Spain    Swed   Switz      UK    USSR   W_Ger
##      C2      C3      C1      C3      C2      C4      C2      C1      C3      C3      C4      C3
##    Yugo
##      C4
## Levels: C1 C2 C3 C4
```

```r
# ACP normée (projection des données standardisées)
res <- PCA(data.frame(P4, protein), scale.unit = TRUE, quali.sup = 1, graph = FALSE)

# Plans factoriel de l'ACP normée avec les pays colorés en fonction de leur classe
par(mfrow=c(2,2))
plot(res, axes = c(1,2), habillage = 1, invisible = "quali", graph.type = "classic")
plot(res, axes = c(1,2), choix = "var",  graph.type = "classic")
plot(res, axes = c(2,3), habillage = 1, invisible = "quali", graph.type = "classic")
plot(res, axes = c(2,3), choix = "var", graph.type = "classic")
```









Interprétation ?

Plan

La classification ascendante hiérarchique (CAH) de Ward

La structure classificatoire est maintenant la hiérarchie.

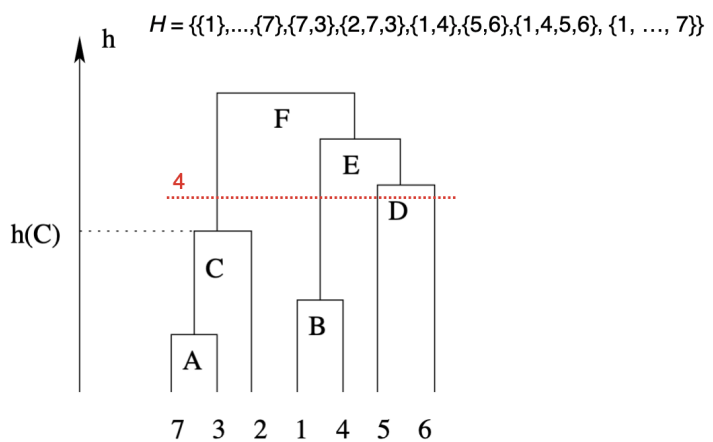
Une hiérarchie (binaire) est un ensemble H de classes non vides de Ω, dont chaque classe est la réunion de deux classes, et qui doit contenir Ω (la classe avec toutes les observations) et les singletons (les n classes avec une seule observation). Le nombre de classes (mis à part les singletons) d'une hiérarchie binaire vaut $n-1$.

Une hiérarchie indicée est une coupe (H, h) où H est une hiérarchie et h est une fonction de H dans $\mathbb{R}^+$ telle que :

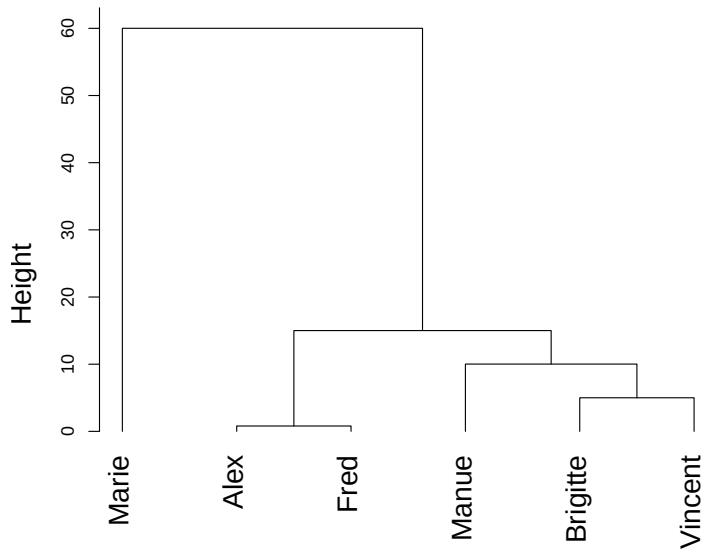
$$\forall A \in H, h(A) = 0 \Leftrightarrow A \text{ est un singleton}$$
$$\forall A, B \in H, A \neq B, A \subset B \Rightarrow h(A) \leq h(B) \text{ (h croissante)}$$

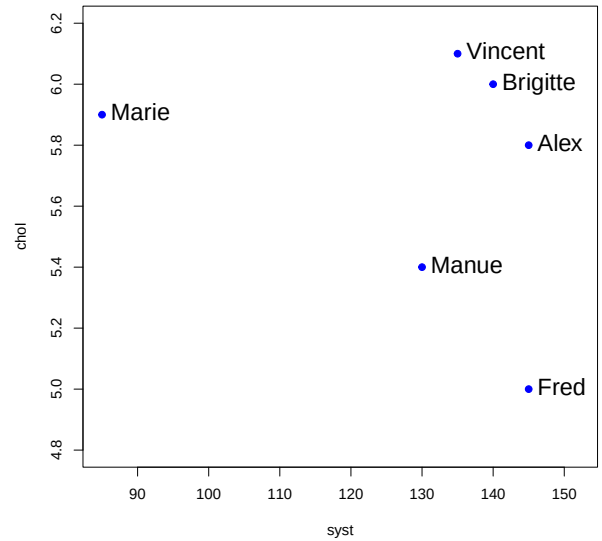
Un dendrogramme (ou arbre hiérarchique) est la représentation graphique d'une hiérarchie indicée et la fonction h mesure la hauteur des classes dans ce dendrogramme.

Les partitions sont obtenues en coupant le dendrogramme par une séquence de lignes horizontales.



Un petit exemple





- Quelle est la hiérarchie H des 6 points de $\mathbb{R}^2$? Quelles sont les hauteurs de ces classes ?
- Représenter les classes de H sur le graphique de droite.
- Quelle sont les partitions en 2 et 3 classes associées à ce dendrogramme ?

L'algorithme de CAH

(a) Initialisation
La partition initiale est la partition des singletons $P_n = (C_1, \ldots, C_n)$ avec $C_k = \{k\}$.

(b) Etape agrégative
On agrège les deux classes C_k et $C_{k'}$ de la partition P_K en K classes obtenue à l'étape précédente qui se ressemble le plus c'est à dire qui minimisent une mesure d'agrégation $D(C_k, C_{k'})$ (mesure de dissimilarité entre classes). Une nouvelle partition P_{K-1} en $K-1$ classes est ainsi obtenue.

(c) Répeter l'étape (b) jusqu'à obtenir la partition en une classe $P_1 = (\Omega)$

(d) Calculer la hauteur h des classes de la hiérarchie H avec :

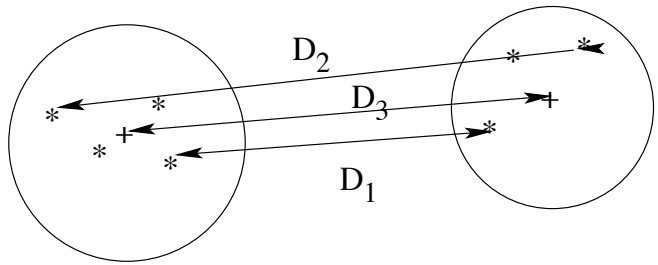
$$h(C_k \cup C_{k'}) = D(C_k \cup C_{k'})$$

Attention : cet algorithme dépend du choix de la mesure d'agrégation :

$$D : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) \to \mathbb{R}^+$$

qui mesure la dissimilarité entre deux classes C_k et $C_{k'}$ de Ω.

Quelles sont les mesures d'agrégation ?



La mesure d'agrégation du lien minimum (single link) :

$$D(C_k, C_{k'}) = \min_{\mathbf{x}_i \in C_k,\ \mathbf{x}_{i'} \in C_{k'}} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

La mesure d'agrégation du lien maximum (complete link) :

$$D(C_k, C_{k'}) = \max_{\mathbf{x}_i \in C_k,\ \mathbf{x}_{i'} \in C_{k'}} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

Ces deux premières mesures d'agrégation utilisent uniquement les dissimilarités (non obligatoirement des distances Euclidiennes) entre les observations.

La mesure d'agrégation de Ward est définie par :

$$D(C_k, C_{k'}) = \frac{\mu_k \mu_{k'}}{\mu_k + \mu_{k'}} d^2(\bar{\mathbf{x}}_k, \bar{\mathbf{x}}_{k'})$$

où

- $\mu_k = \sum_{\mathbf{x}_i \in C_k} w_i$ est le poids de la classe C_k,
- $\bar{\mathbf{x}}_k = \frac{1}{\mu_k} \sum_{\mathbf{x}_i \in C_k} w_i \mathbf{x}_i$ est le centre de gravité de la classe C_k.

La mesure d'agrégation de Ward utilise la distance Euclidiennes entre les centres de gravités des classes. Elle ne peut donc (normalement) être utilisée que pour des données quantitatives.

Remarque : les hauteurs trouvées avec les mesures d'agrégation du lien minimum, lien maximum et de Ward vérifient la propriété $A \subset B \Rightarrow h(A) \leq h(B)$.

Mais cette propriété n'est pas vérifiée pour les mesures d'agrégation suivantes :

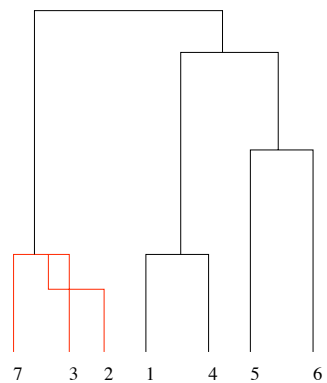
- Mesure d'agrégation du lien moyen :

$$D(C_k, C_{k'}) = \frac{1}{n_k n_{k'}} \sum_{\mathbf{x}_i \in C_k} \sum_{\mathbf{x}_{i'} \in C_{k'}} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

- la mesure d'agrégation des centroides :

$$D(C_k, C_{k'}) = d^2(\bar{\mathbf{x}}_k, \bar{\mathbf{x}}_{k'})$$

Des inversions peuvent donc être observées avec les mesures d'agrégation du lien moyen et des centroides.



En général, on utilise toujours la mesure d'agrégation de Ward dont l'algorithme se réecrit pour ne prendre en entrée que la matrice des distances (normalement Euclidiennes) entre les observations.

Algorithme de la CAH de Ward.

(a) Initialisation : construire la matrice $\Delta = (\delta_{ij})_{n\times n}$ des mesures de Ward entre les singletons :

$$\delta_{ij} = D(\{i\},\{j\}) = \frac{w_i w_j}{w_i + w_j} d^2(\mathbf{x}_i, \mathbf{x}_j).$$

(b) Etape agrégative

1. Agréger les deux classes A et B de P_K qui minimisent $D(A, B)$.
2. Mettre à jour la matrice Δ des mesures de Ward entre les classes de P_{K-1} en calculant la mesure de Ward entre $A \cup B$ et les autres classes de P_{K-1} avec la formule de Lance et Williams :

$$D(A\cup B, C) = \frac{\mu_A + \mu_C}{\mu_A + \mu_B + \mu_C} D(A, C) + \frac{\mu_B + \mu_C}{\mu_A + \mu_B + \mu_C} D(B, C) - \frac{\mu_C}{\mu_A + \mu_B + \mu_C} D(A, B)$$

(c) Recommencer l'étape (b) jusqu'à obtenir la partition en une classe.

L'algorithme de CAH de Ward prend donc en entrée :

- le vecteur $\mathbf{w} = (w_i)_{i=1,\dots,n}$ des poids des observations,
- la matrice $\mathbf{D} = (d_{ij})_{n\times n}$ des distances Euclidiennes entre les observations.

Avec la fonction `hclust` de R :

- `hclust(d=`Δ`, method="ward.D")` lorsque les poids sont uniformes,
- `hclust(d=`Δ`, method="ward.D", members=`$\mathbf{w}$`)` sinon.

et Δ la matrice des mesures de Ward entre les singletons :

- $\Delta = \frac{\mathbf{D}^2}{2n}$ lorsque les poids sont $w_i = \frac{1}{n}$,
- $\Delta = \frac{\mathbf{D}^2}{2}$ lorsque les poids sont $w_i = 1$

avec $\mathbf{D}^2 = (d_{ij}^2)_{n\times n}$.

Propriétés de le CAH de Ward

- La CAH de Ward optimise le même critère que les K-means. En effet, on peut montrer qu'à chaque étape, la CAH de Ward agrège les deux classes qui donnent la nouvelle partition de plus petite inertie intra-classe (parmi toutes les partitions obtenues en agrégant deux classes).
- La somme des hauteurs du dendrogramme de Ward est égale à l'inertie totale.
- La somme des $K-1$ plus grandes hauteurs est l'inertie inter-classe de la partition en K classes du dendrogramme.
- La complexité de l'algorithme est quadratique en fcontion du nombre d'observations. L'algorithme ne s'applique pas à des grand jeux de données.
- Cet algorithme est parfois utilisé avec des matrices de dissimilarités ou de distances non Euclidiennes. On peut montrer que dans ce cas, le critère optimisé est la pseudo-inertie intra-classe où l'inertie d'une classe est définie par :

$$I(C_k) = \sum_{\mathbf{x}_i \in C_k} \sum_{\mathbf{x}_j \in C_k} \frac{w_i w_j}{2\mu_k} d^2(\mathbf{x}_i, \mathbf{x}_j).$$

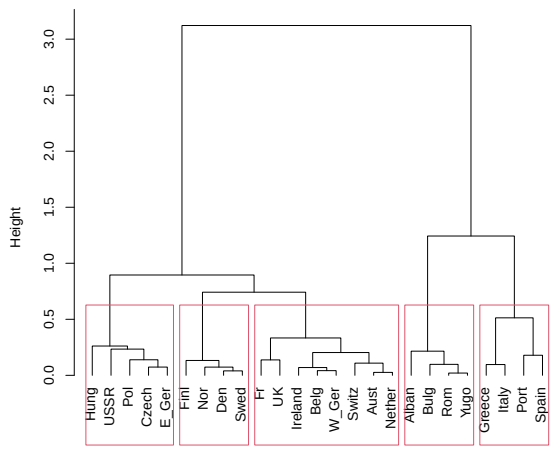
Un exemple avec R

CAH de Ward des données protein standardisées

```
load("protein.rda")
n <- nrow(protein)
# Standardisation avec l'écart-type non corrigé
Z <- scale(protein)*sqrt(n/(n-1))
```

```
#CAH de Ward avec pondérations de 1/n des observations
D <- dist(Z)
tree <- hclust(D^2/(2*n), method = "ward.D")
sum(tree$height) # la somme des hauteurs est égale à l'inertie totale (p=9)

## [1] 9
```

```
# Le dendrogramme suggère de couper en 3 ou 5 classes.
plot(tree, hang=-1, main="", sub="", xlab="")
rect.hclust(tree, k=5)
```



```r
P5 <- cutree(tree, k=5)
K <- 5
W <- sum(tree$height[1:(n-K)]) # inertie intra-classe

# Pourcentage d'inertie expliquée de la partition en 5 classes de Ward
(1-W/9)*100

## [1] 67
```

```r
# kmeans sur les mêmes données et le même nombre de classes
res2 <- kmeans(Z, centers = 5, nstart = 20)

# Pourcentage d'inertie expliquée par la partition en 5 classes des kmeans
res2$betweenss/res2$totss*100

## [1] 67

# On retrouve la même partition qu'avec la CAH de Ward
sum(res2$cluster == P5)

## [1] 0
```

Plan

Interprétation des résultats

On peut interpréter les classes d'une partition à partir :

- des variables actives : variables utilisées dans le processus de clustering,
- de variables illustratives : utilisées uniquement pour la description des classes.

Ces variables peuvent être quantitatives ou qualitatives.

En pratique, on interprétera souvent les classes par :

- les variables quantitatives : la moyenne dans la classe est-elle différente de la moyenne sur toutes les observations ?
- les modalités des variables qualitatives : une modalité est-elle plus fréquente dans la classe, la classe contient-elle toutes les observations possédant cette modalité ?

La valeur-test $t_k(X^j)$ d'une variable quantitative X^j dans une classe C_k est une statistique de test définie par :

$$t_k(X^j) = \frac{\bar{x}_k^j - \bar{x}^j}{\sqrt{\frac{s_j^2}{n_k}\frac{(n-n_k)}{n-1}}}$$

On voit que la valeur-test de X^j est d'autant plus grande (en valeurs absolue) que la moyenne de X^j dans C_k est différente de la moyenne de X^j sur toutes les observations. On dira alors que la variable X^j caractérise la classe C_k.

De plus, sous l'hypothèse nulle que les n_k observations de C_k sont tirées au hasard sans remise dans Ω, la statistique $t_k(X^j)$ suit approximativement une loi N(0,1). On va donc rejeter l'hypothèse nulle (et conclure que la variable X^j caractéristique la classe) si la `p.value` est petite qu'un certain seuil (0.05 par exemple).

La valeur-test $t_k(M^j)$ d'une modalité M^j (d'une variable qualitative) dans une classe C_k est une statistique de test définie par :

$$t_k(M^j) = \frac{\frac{n_{jk}}{n_{.k}} - \frac{n_{j.}}{n}}{\sqrt{\frac{(n-n_{.k})}{n-1}\frac{s_j^2}{n_{.k}}}}$$

avec $s_j^2 = \frac{n_{j.}}{n}(1 - \frac{n_{j.}}{n})$.

On voit que la valeur-test de M^j est d'autant plus grande (en valeurs absolue) que la fréquence de M^j dans C_k est différente de la fréquence de M^j sur toutes les observations. On dira alors que la modalité M^j caractérise la classe C_k.

De plus, sous l'hypothèse nulle que les n_k observations de C_k sont tirées au hasard sans remise dans Ω, la statistique $t_k(M^j)$ suit approximativement une loi N(0,1). On va donc rejeter l'hypothèse nulle (et conclure que la modalité M^j caractéristique la classe) si la `p.value` est petite qu'un certain seuil (0.05 par exemple).

Un exemple avec R

```r
# Partition en 5 classes de la CAH de Ward
load("protein.rda")
Z <- scale(protein)*sqrt(25/(24))
P5 <- cutree(hclust(dist(Z)^2/25), k = 5)
P5 <- as.factor(P5)
levels(P5) <- paste("C", 1:5, sep="")
```

```r
# Classe 1 = (Alban, Bulg, Greece, Italy, Rom, Yugo)
library(FactoMineR)
res <- catdes(data.frame(P5, Z), num.var = 1, proba = 0.05)
res$quanti$C1
```

```
##                v.test Mean in category Overall mean sd in category Overall sd p.value
## Cereals           3.8              1.39     -4.1e-16           0.69          1 0.00013
## Nuts              3.3              1.19     -2.7e-16           0.66          1 0.00107
## Milk             -2.2             -0.80     -2.2e-16           0.47          1 0.02692
## White.Meat       -2.6             -0.95      2.6e-16           0.48          1 0.00910
## Eggs             -3.0             -1.08     -3.6e-16           0.78          1 0.00287
## Starchy.Foods    -3.9             -1.41     -3.8e-16           0.57          1 0.00010

# Seules les variables avec une p.val < 0.05 sont affichées
```

Interprétation de la classe 1 à partir des variables quantitatives ?

```r
# Variable qualitative illustrative
zone <- c("east","west","west","east","east","north","east","north","west","south",
          "east","west","south","west","north","east","south","east","south","north",
          "west","west","east","west","east")

res <- catdes(data.frame(P5, zone), num.var=1)
res$category[-1]
```

```
## $C2
##           Cla/Mod Mod/Cla Global p.value v.test
## zone=west     100     100     32 9.2e-07    4.9
## zone=east       0       0     36 1.2e-02   -2.5
##
## $C3
##           Cla/Mod Mod/Cla Global p.value v.test
## zone=east      56     100     36  0.0024      3
##
## $C4
##            Cla/Mod Mod/Cla Global p.value v.test
## zone=north     100     100     16 7.9e-05    3.9
##
## $C5
##            Cla/Mod Mod/Cla Global p.value v.test
## zone=south      50     100     16    0.02    2.3
```

- Cla/Mod = proportion des observations ayant la modalité M^j qui sont dans la classe C_k,
- Mod/Cla = proportion des observations de la classe C_k qui possèdent la modalité M^j,
- Global = proportion d'observations qui ont la modalité M^j.

Interprétation de la classe 3 à partir des modalités de la variable qualitative zone ?

Plan

CAH de Ward + K-means

1. On effectue une CAH de Ward et on choisit un nombre de classes K.
2. On coupe l'arbre pour obtenir une partition en K classes.
3. On applique les K-means avec les centres de gravité de cette partition comme centres initiaux.

Remarques : La CAH de Ward permet d'avoir une idée du nombre de classes et d'avoir une bonne partition pour initialiser les K-means. Les K-means vont nécessairement améliorer la partition de Ward (en terme d'inertie intra-classe).

K-means + CAH de Ward

- On effectue les K-means pour résumer les n observations par les centres de gravité des K classes.
- On applique la CAH de Ward à ces K centres de gravités pondérés par les fréquences des classes.

Remarques : Les K-means permettent des réduire le nombre d'observations sans perdre trop d'information, pour pouvoir appliquer ensuite la CAH de Ward (dont la complexité ne permet pas de traiter de très grands jeux de données).

Et si les données sont qualitatives ou mixtes ?

Première possibilité : recoder les données en numérique et appliquer une méthode de clustering sur données quantitatives. Deux approches (au moins) pour recoder les données en numérique :

1. Faire une ACM (si les données sont qualitatives) ou une ACPmixte(si les données sont quantitatives et qualitatives) et récuperer les composantes principales qui sont numériques (toutes ou les q premières) pour recoder les n observations.

2. Faire un clustering des p variables en $q \leq p$ classes avec la méthode ClustOfVar qui gère les variables quantitatives et qualitatives. Puis récupérer les variables synthétiques des q classes qui sont numériques pour recoder les n observations.

Seconde possibilité : calculer une matrice de dissimilarités entre les n observations avec une mesure de dissimilarité adaptée aux données qualitatives ou mixtes (par exemple la "distance" de Gower implémentée dans la fonction `daisy` du package `cluster`. Puis appliquer une méthode de clustering sur matrice de dissimilarité (par exemple K-médoides, CAH de type lien minimum, maximum ou moyen).

Comment choisir le nombre de classes ?

Elbow method :

1. Calculer l'inertie intra-classe des partitions pour plusieurs valeurs de K (nombre de classes) dans une grille.
2. Choisir le nombre de classes dans cette grille avec la règle du coude déjà utilisée pour choisir le nombre de composantes principales en ACP.

Remarques :

- Inertie intra-classe = Somme des carrés intra-classes (poids w_i tous égaux à 1) = Within Sum of Squares (WSS).
- L'inertie intra-classe décroit avec le nombre de classes : elle vaut 0 pour une partition en $K = n$ classes et l'inertie totale des observations pour une partition en $K = 1$ classe.
- On peut visualiser cette décroissante sur un graphique de l'inertie intra-classe (en ordonnée) en fonction du nombre de classes (en abscisse). Par exemple avec la fonction `fviz_nbclust` du package `factoextra`.

Average silhouette :

1. Calculer le critère de silhouette moyenne des partitions pour une grille de valeurs de K (nombre de classes).
2. Choisir le nombre de classes dans cette grille qui maximise ce critère.

Calcul de la silhouette moyenne d'une partition $P_K = (C_1, ..., C_K)$:

- Silhouette d'une observation $\mathbf{x}_i$:

$$s(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{\max(b(\mathbf{x}_i), a(\mathbf{x}_i))},$$

  où $a(\mathbf{x}_i)$ est la distance moyenne entre l'observation $\mathbf{x}_i$ et les autres observations de sa classe, et $b(\mathbf{x}_i)$ est la plus petite distance entre l'observation $\mathbf{x}_i$ et les observations des autres classes (version simplifiée).

- Silhouette moyenne de P_K :

$$S(P_K) = \frac{1}{n} \sum_{i=1}^{n} s(\mathbf{x}_i).$$

Remarques :

- Le critère de la silhouette moyenne mesure combien les classes d'une partition sont homogènes et bien séparées les unes des autres.
- Il varie dans $[-1, 1]$:

$$s(\mathbf{x}_i) = \begin{cases} 1 - \frac{a(\mathbf{x}_i)}{b(\mathbf{x}_i)} & \text{si } a(\mathbf{x}_i) < b(\mathbf{x}_i) \\ 0 & \text{si } a(\mathbf{x}_i) = b(\mathbf{x}_i) \\ \frac{b(\mathbf{x}_i)}{a(\mathbf{x}_i)} - 1 & \text{si } a(\mathbf{x}_i) > b(\mathbf{x}_i) \end{cases}$$

- Il permet de comparer des partitions ayant des nombres de classes différentes, à la différence du critère d'inertie intra-classe qui décroit avec le nombre de classes.
- On peut visualiser sur un graphique l'évolution de ce critère en fonction du nombre de classes, par exemple avec la fonction `fviz_nbclust` du package `factoextra`.

Gap statistic :

1. Calculer la valeur de la gap statistic pour une grille de valeurs de K (nombre de classes).
2. Choisir le nombre de classes dans cette grille qui maximise ce critère.

Calcul de la gap statistic pour une partition $P_K = (C_1, ..., C_K)$ d'une matrice des données $\mathbf{X}$.

1. Calculer $W(P_K)$ où W est un critère de variation intra-classe (par exemple l'inertie intra-classe).
2. Construire B matrices $\mathbf{X}^{(1)}, ..., \mathbf{X}^{(B)}$ en simulant les valeurs de chaque colonne (version simplifiée), selon une loi uniforme sur l'étendu de la variable observée dans $\mathbf{X}$.
3. Pour $b = 1, ..., B$, construire une partition $P_K^{(b)}$ de $\mathbf{X}^{(b)}$ avec l'algorithme de clustering (e.g. les K-means), et calculer $W(P_K^{(b)})$.
4. Calculer :

$$Gap(P_K) = \log\left(W(P_K)\right) - \frac{1}{B}\sum_{b=1}^{B} \log\left(W(P_K^{(b)})\right)$$

Remarques :

- La gap statistic compare (ici) la variation intra-classe de la partition P_K avec la variation intra-classe moyenne de données non structurées en K classes.
- Plus il est grand, plus on s'éloigne de l'hypothèse d'absence de structure en K classes. Et donc plus le nombre de classes est "pertinent".
- Il permet de comparer des partitions ayant des nombres de classes différentes, à la différence du critère d'inertie intra-classe qui décroit avec le nombre de classes.
- On peut visualiser sur un graphique l'évolution de ce critère en fonction du nombre de classes, par exemple avec la fonction `fviz_nbclust` du package `factoextra`, qui utilise la fonction `clusGap` du package `cluster`.

Enfin, il existe de nombreux autres critères ou méthodes pour choisir le nombre de classes...

A SAVOIR : il n'y a pas de bon nombre de classes. Il est possible d'avoir plusieurs plusieurs nombre de classes qui donnent des valeurs très proches des critères optimisés. Et deux critères peuvent donner des nombres de classes optimaux différents ! Une piste : une bonne partition est aussi une partition qui s'interprète bien (pour laquelle il existe des variables ou des modalités qui expliquent ses classes).

D'autres méthodes de clustering...

Méthode	Particularité	Structure de classification	Package R
PAM	K-médoides	Partition	Cluster
SOM	Cartes Autoroganisatrices de Kohonen	Partition	som
fanny	Fuzzy K-means	Partition	Cluster
Model-based Clustering	Modèles de mélanges Gaussiens	Partition	mclust
Kmeans sparse	kmeans avec sélection de variables	Partition	spca
diana	Divisive clustering	Hiérarchie	Cluster
hclustvar	Clustering de variables	Hiérarchie	ClustOfVar
hclustgeo	Clustering avec contraintes spatiales	Hiérarchie	ClustGeo

Et bien d'autres :

CRAN Task View : Cluster Analysis & Finite Mixture Models

`https://cran.r-project.org/web/views/Cluster.html`