

Fondamentaux du Data Mining

DU Analyste Data Science

Magali Champion



14/01/2021

Introduction

Contexte de grande dimension

On considère une matrice de données X et un vecteur d'observations Y à expliquer. Les observations portent sur p variables, mesurées sur n individus. Il existe plusieurs situations de **grande dimension** :

- n grand et p de taille raisonnable
 - ▶ situation favorable d'un point de vue théorique,
 - ▶ problème de stockage informatique,
 - ▶ solutions informatiques de BigData (Hadoop,...).

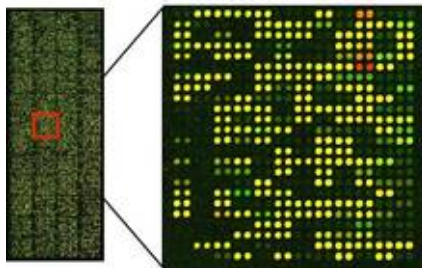


Introduction

Contexte de grande dimension

On considère une matrice de données X et un vecteur d'observations Y à expliquer. Les observations portent sur p variables, mesurées sur n individus. Il existe plusieurs situations de **grande dimension** :

- n de taille raisonnable et p grand ($p \gg n$)
 - ▶ problèmes théoriques sous-jacents,
 - ▶ réduction de dimension, sélection de variables.



Introduction

Contexte de grande dimension

On considère une matrice de données X et un vecteur d'observations Y à expliquer. Les observations portent sur p variables, mesurées sur n individus. Il existe plusieurs situations de **grande dimension** :

- n et p grands
 - ▶ situation la plus compliquée,
 - ▶ outils informatiques de BigData + outils stats de grande dimension.



Introduction

Qu'est-ce que le data mining?

Le data mining est aussi connu sous le nom de **fouille de données**. Il regroupe un ensemble de techniques et d'outils issus de la statistique, de l'informatique et de la science de l'information.

Motivations :

- découvrir des règles, relations, dépendances à travers une grande quantité de données, **Data mining non-supervisé**
- utiliser un ensemble de données pour prédire des informations, des comportements. **Data mining supervisé**

Avec le changement de paradigme (grande dimension), le data mining a considérablement évolué ces dernières années.

Comment trouver un diamant dans un tas de charbon sans se salir les mains?

CEO de SAS

Introduction

Apprentissage non-supervisé

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un n -échantillon $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$ de (X^1, \dots, X^p) .

Motivation : découvrir des règles, relations, dépendances dans les données $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$.



Introduction

Apprentissage supervisé

On considère :

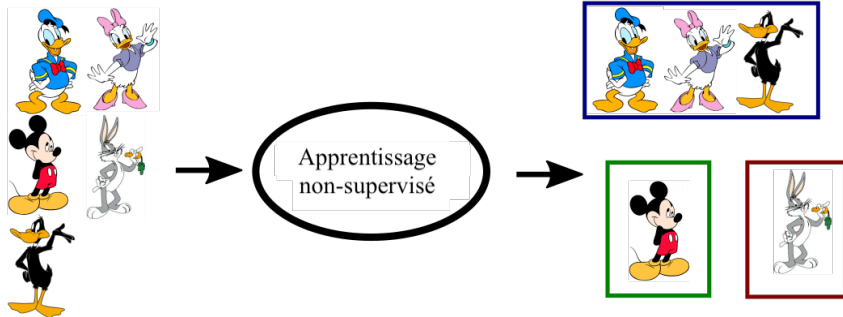
- p variables explicatives (X^1, \dots, X^p) ,
- un vecteur d'observations Y à expliquer,
- un n -échantillon $(x_i^1, \dots, x_i^p, y_i)_{1 \leq i \leq n}$ de (X^1, \dots, X^p, Y) .

Motivation : utiliser les observations $(x_i^1, \dots, x_i^p, y_i)_{1 \leq i \leq n}$ pour prédire des informations, des comportements.

On parle d'apprentissage supervisé car les $(y_i)_{1 \leq i \leq n}$ permettent de guider le processus d'estimation.





Introduction

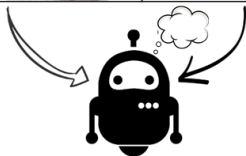
Apprentissage supervisé vs non-supervisé



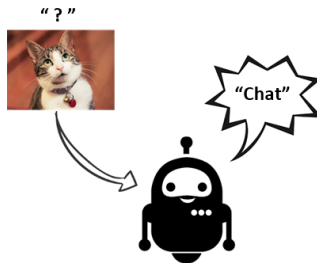
Introduction

Apprentissage supervisé vs non-supervisé

x	y
	"Chien"
	"Chien"
	"Chat"
	"Chien"



Apprentissage Supervisé



Utilisation finale

Introduction

Exemples d'application

- Entreprise et relation clients :
 - ▶ création de profils clients
 - ▶ ciblage de clients potentiels
- Finances et assurances :
 - ▶ minimisation de risques financiers
 - ▶ détection de fraudes
- Biomédical :
 - ▶ analyse du génome
 - ▶ identification de sous-groupes de patients
- Internet :
 - ▶ détection de spam
 - ▶ mise au point de systèmes de recommandation

Introduction

Jeu de données fil rouge

Les données *fil rouge* que vous aurez à traiter en autonomie seront des données issues du service de location de logements de particuliers **airbnb** :

- $X \in \mathcal{M}(n, p)$ désigne la matrice qui contient des informations concernant p variables pour n logements distincts,
- $y \in \mathbb{R}^n$ est la variable réponse, connue pour les n logements mais que l'on souhaite prédire, ici le quartier d'où provient le logement.



Introduction

Jeu de données non-supervisé

A titre d'exemples, nous utiliserons également le jeu de données **fromages**, qui contient des données relatives à la composition de 29 fromages.

```
fromages <- read.table(file="fromages.txt", header=T,  
                      row.names=1, sep="\t", dec=".")  
head(fromages,5)
```

##	calories	sodium	calcium	lipides	retinol	folates	prote
## Carre delEst	314	353.5	72.6	26.3	51.6	30.3	
## Babybel	314	238.0	209.8	25.1	63.7	6.4	
## Beaufort	401	112.0	259.4	33.3	54.9	1.2	
## Bleu	342	336.0	211.1	28.9	37.1	27.5	
## Camembert	264	314.0	215.9	19.5	103.0	36.4	
##	cholesterol	magnesium					
## Carre delEst	70	20					
## Babybel	70	27					
## Beaufort	120	41					
## Bleu	90	27					
## Camembert	60	20					

Introduction

Jeu de données supervisé

A titre d'exemples, nous utiliserons également le jeu de données **ptitanic**, qui contient des données relatives à 1309 passagers du titanic.

```
library(rpart.plot)
data("ptitanic")
head(ptitanic,5)
```

##	pclass	survived	sex	age	sibsp	parch
## 1	1st	survived	female	29.0000	0	0
## 2	1st	survived	male	0.9167	1	2
## 3	1st	died	female	2.0000	1	2
## 4	1st	died	male	30.0000	1	2
## 5	1st	died	female	25.0000	1	2

Section 1

Apprentissage non-supervisé

- méthodes descriptives -

Introduction

En quoi consiste l'apprentissage non-supervisé?

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un n -échantillon $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$ de (X^1, \dots, X^p) .

Motivation : découvrir des règles, relations, dépendances dans les données $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$.



Introduction

En quoi consiste l'apprentissage non-supervisé?

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un n -échantillon $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$ de (X^1, \dots, X^p) .

Motivation : découvrir des règles, relations, dépendances dans les données $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$.



Techniques de clustering et d'analyse factorielle

Techniques de clustering

Qu'est-ce que c'est?

Le **clustering**, aussi appelé classification automatique, partitionnement ou classification non-supervisée, a pour objectif de créer des groupes d'observations homogènes tels que :

- les individus au sein d'un groupe soient les plus semblables possibles,
- les groupes soient les plus différents possibles les uns des autres.

Les groupes ainsi construits sont appelés **clusters** (ou classes).

Objectif double : meilleure compréhension des données & réduction de dimension.

Techniques de clustering

Problématiques

Les **problématiques** associées au clustering sont de différents types :

- nature des observations : les données sont-elles binaires, textuelles, numériques... ?
- notion de similarité : comment définir une similarité ou dissimilarité entre observations?
- définition d'un cluster,
- interprétation d'un cluster : comment résumer un cluster? Par la moyenne de ces représentants?
- évaluation des performances d'un algorithme de clustering,
- définition du nombre de clusters.

Techniques de clustering

Comment définir le nombre de clusters?

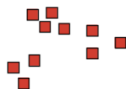


Techniques de clustering

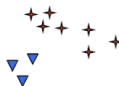
Comment définir le nombre de clusters?



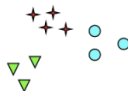
2 clusters?



4 clusters?



6 clusters?

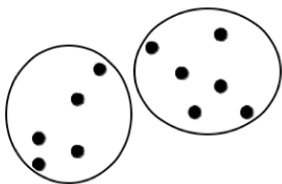


Techniques de clustering

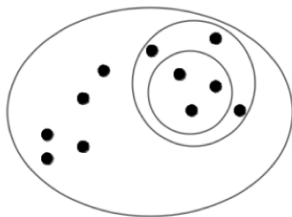
Différentes approches

Il existe 3 grandes familles de clustering :

- les approches **par partitionnement**¹, pour lesquelles les classes construites sont toujours disjointes,
- les approches **hiérarchiques**², pour lesquelles les classes sont disjointes ou incluses les unes dans les autres,
- les approches spectrales.



1



2

Techniques de clustering

Approches par partitionnement

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un n -échantillon $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$ de (X^1, \dots, X^p) .

Objectif : Construire une partition des données en $k < n$ clusters $(C_j)_{1 \leq j \leq k}$.

Techniques de clustering

Approches par partitionnement

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un n -échantillon $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$ de (X^1, \dots, X^p) .

Objectif : Construire une partition des données en $k < n$ clusters $(C_j)_{1 \leq j \leq k}$.

Approche naïve : construire toutes les partitions possibles et en retenir la meilleure.

⚠ Le nombre de partitions augmente de manière exponentielle : il s'agit d'un problème NP difficile.

Techniques de clustering

Approches par partitionnement : k -means

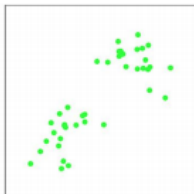
Le k -means (Forgy, 1965; MacQueen, 1967) est l'une des méthodes les plus anciennes et traditionnelles pour effectuer de la classification non-supervisée.

Principe :

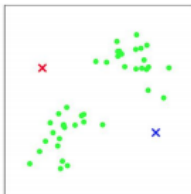
- 1 choisir k éléments initiaux (centres de gravité) $(\mu_j)_{1 \leq j \leq k}$ pour les k clusters (C_1, \dots, C_k) à construire,
- 2 affecter chaque observation x_i à la classe C_j dont le centre μ_j est le plus proche,
- 3 recalculer le centre de gravité de chaque cluster (C_1, \dots, C_k) ,
- 4 itérer jusqu'à stabilité des clusters.

Techniques de clustering

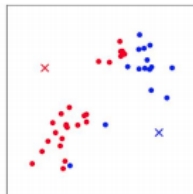
Approches par partitionnement : k -means



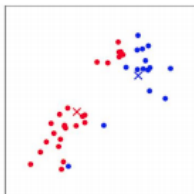
(a)



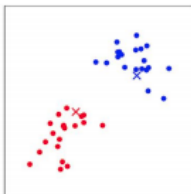
(b)



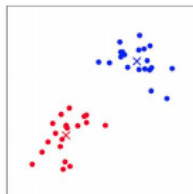
(c)



(d)



(e)



(f)

Techniques de clustering

Approches par partitionnement : *k*-means (R)

```
# centrage des données  
fromages <- scale(fromages, center=T, scale=T)
```

```
# k-means avec k=3 clusters  
kmean <- kmeans(fromages, centers=3)  
head(kmean$cluster)
```

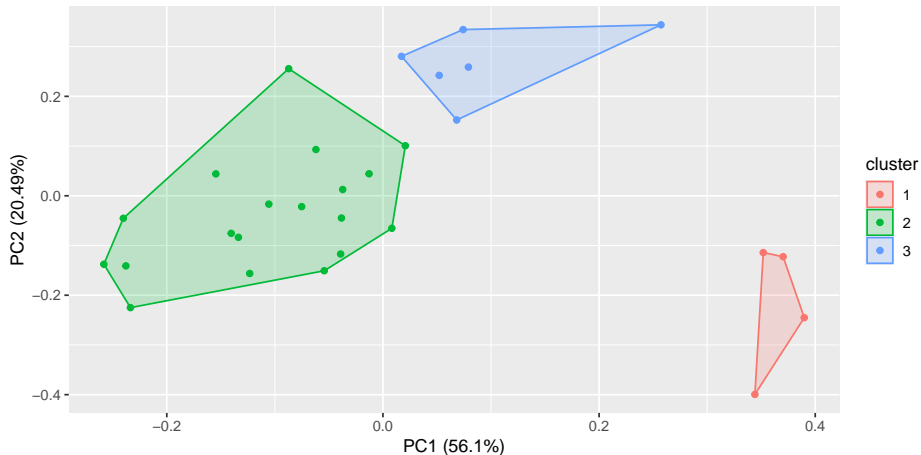
```
## CarreDelEst      Babybel      Beaufort      Bleu      Camembert  
##              3              2              2              2              3
```

```
# cluster 1  
names(which(kmean$cluster==1))  
  
## [1] "Fr.frais20nat."      "Fr.frais40nat."      "Petitsuisse40"  
## [4] "Yaourtlaitent.nat."
```

Techniques de clustering

Approches par partitionnement : *k*-means (R)

```
autoplot(kmean, fromages, frame=TRUE)
```



Techniques de clustering

Approches par partitionnement : k -means

Remarques :

- les centres initiaux sont souvent choisis aléatoirement, des initialisations différentes peuvent donc mener à des clusters différents,
 - ▶ faire plusieurs essais,
 - ▶ utiliser du clustering hiérarchique pour déterminer les centres initiaux,
- l'utilisation du k -means nécessite de connaître le nombre de clusters,
 - ▶ fixer k à priori,
 - ▶ chercher la meilleure partition pour différentes valeurs de k et chercher un coude pour la décroissance de l'inertie globale,
- le k -means est sensible à la présence d'outliers et est en difficulté lorsque les clusters sont de différentes tailles ou densités.

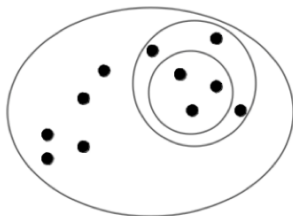
Techniques de clustering

Approches hiérarchiques

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un n -échantillon $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$ de (X^1, \dots, X^p) .

Objectif : Construire une partition **hiérarchique** des données en $k < n$ clusters $(C_j)_{1 \leq j \leq k}$.



Techniques de clustering

Approches hiérarchiques

Il existe **deux** types d'approches :

- clustering hiérarchique ascendant ou agglomératif (**CAH**) :
 - ▶ commencer en considérant que chaque individu est un cluster à lui seul,
 - ▶ à chaque étape, regrouper les clusters les plus proches jusqu'à obtenir 1 ou k clusters,
- clustering hiérarchique descendant ou divisif :
 - ▶ commencer en considérant un seul cluster contenant l'ensemble des individus,
 - ▶ à chaque étape, diviser un cluster jusqu'à obtenir des clusters ne contenant qu'un point ou k clusters.

Techniques de clustering

Approches hiérarchiques : CAH

Principe :

- ➊ placer les n individus dans leur propre cluster (n clusters au total),
- ➋ calculer la similarité entre chaque couple de clusters,
- ➌ chercher les deux clusters les plus proches basé sur cette mesure de similarité,
- ➍ fusionner ces deux clusters,
- ➎ recalculer la similarité entre chaque couple de clusters,
- ➏ itérer jusqu'à l'obtention de 1 ou k clusters.

Techniques de clustering

Approches hiérarchiques : CAH

Principe :

- ➊ placer les n individus dans leur propre cluster (n clusters au total),
- ➋ calculer la similarité entre chaque couple de clusters,
- ➌ chercher les deux clusters les plus proches basé sur cette mesure de similarité,
- ➍ fusionner ces deux clusters,
- ➎ recalculer la similarité entre chaque couple de clusters,
- ➏ itérer jusqu'à l'obtention de 1 ou k clusters.

Point-clé : définir le critère de choix de clusters à fusionner (critère d'agrégation)

Techniques de clustering

Approches hiérarchiques : CAH

Quelques critères d'agrégation :

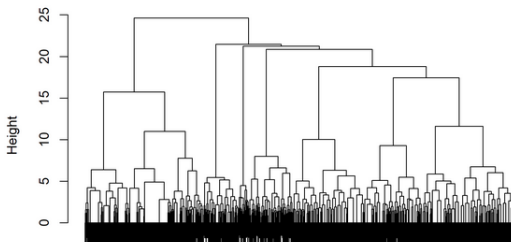
- saut minimal (single linkage) : basée sur la distance entre les deux points les plus proches de chaque cluster,
 - ▶ tendance à construire des clusters très généraux
- saut maximal (complete linkage) : basée sur la distance entre les deux points les plus éloignés de chaque cluster,
 - ▶ tendance à construire des clusters très spécifiques
- saut moyen : basée sur la distance moyenne entre les points des clusters,
 - ▶ tendance à construire des clusters de variance proche
- méthode de Ward
 - ▶ chaque cluster est représenté par son centre de gravité,
 - ▶ agglomération des clusters basée sur l'inertie intra-classe.

Techniques de clustering

Approches hiérarchiques : représentations graphiques

Les résultats des approches hiérarchiques sont représentés sous la forme d'un **dendrogramme**, qui fournit une visualisation de la hiérarchie des partitions obtenues sous la forme d'un arbre dont :

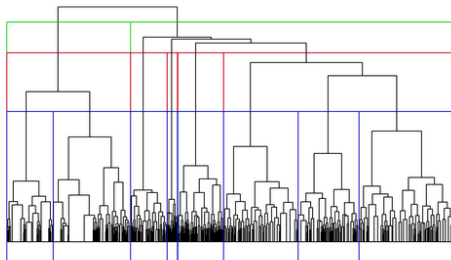
- les nœuds sont les différents clusters construits,
- les feuilles sont les individus,
- la racine est la partition finale en un unique cluster.



Techniques de clustering

Approches hiérarchiques : représentations graphiques

Choix du nombre de clusters : l'axe des ordonnées du dendrogramme indique la valeur du critère d'agrégation (avant fusion), on cherche donc des coupures nettes dans le dendrogramme.



Techniques de clustering

Approches hiérarchiques : CAH (R)

```
# matrice des distances entre individus  
d.fromage <- dist(fromages)  
  
# CAH - critère de Ward  
cah.ward <- hclust(d.fromage,method="ward.D2")
```

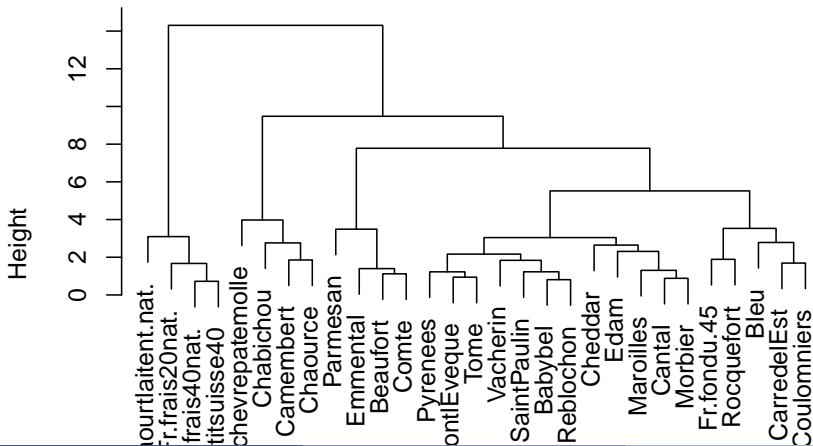
Techniques de clustering

Approches hiérarchiques : CAH (R)

```
# dendrogramme
```

```
plot(cah.ward)
```

Cluster Dendrogram



Techniques de clustering

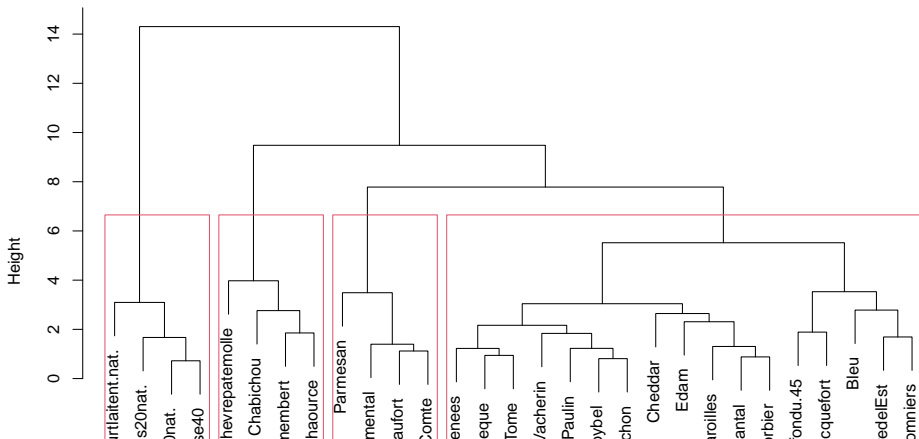
Approches hiérarchiques : CAH (R)

dendrogramme avec matérialisation des groupes

```
plot(cah.ward)
```

```
rect.hclust(cah.ward, k=4)
```

Cluster Dendrogram



Techniques de clustering

Approches hiérarchiques : CAH (R)

```
# découpage en 4 groupes  
groupes.cah <- cutree(cah.ward,k=4)
```

```
# cluster 1  
names(which(groupes.cah==4))
```

```
## [1] "Fr.frais20nat."      "Fr.frais40nat."      "Petitsuisse40"  
## [4] "Yaourtlaitent.nat."
```

Techniques de clustering

Classification mixte

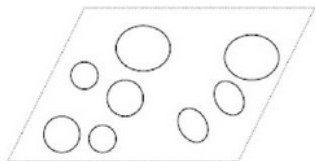
La **classification mixte** (Lebart, 1984) - combinaison du k -means et de la CAH - a été mise en place dans le but de pallier les difficultés de la CAH lors du passage à la grande dimension (calcul de la similarité de tous les individus 2 à 2).

Principe :

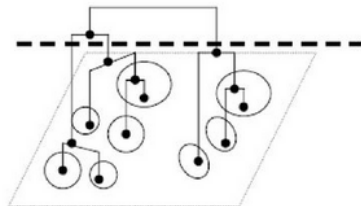
- Partitionnement préliminaire des individus en $K > k$ clusters par k -means
 - ▶ permet de diminuer la dimension du problème
- Classification hiérarchique CAH sur les K clusters initiaux
 - ▶ on utilise les barycentres des clusters obtenus lors de l'étape 1
 - ▶ le nombre de clusters final est déterminé en coupant l'arbre hiérarchique
- Consolidation de la partition par réaffectation des individus dans les clusters
 - ▶ permet d'augmenter l'inertie inter-classe et l'homogénéité des clusters

Techniques de clustering

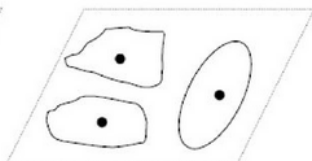
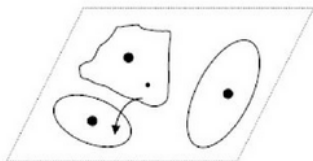
Classification mixte



Etape 1: Partitionnement préliminaire



Etape 2: CAH



Etape 3: Partition finale et consolidation de la partition par réallocations

Analyse factorielle

Qu'est-ce que c'est?

L'**analyse factorielle** a pour objectif de rechercher des relations complexes entre des variables en résumant l'information en un petit nombre seulement de facteurs. Elle est particulièrement adaptée au cadre de la grande dimension.

Objectif double : meilleure compréhension des données & réduction de dimension.

Il existe différentes méthodes d'analyse factorielle : l'**analyse en composantes principales** (ACP), l'analyse factorielle des correspondances (AFC), l'analyse des correspondances multiples (ACM), l'analyse factorielle de données mixtes (AFDM), l'analyse factorielle multiple (AFM) et l'analyse factorielle multiple hiérarchique (AFMH).

Analyse factorielle

Analyse en Composantes Principales (ACP)

L'**Analyse en Composantes Principales** construit des facteurs (composantes principales) qui résument l'information contenue dans un jeu de données sous la forme de combinaisons linéaires de variables.

Principe : construction de :

- une matrice A de taille $p \times r$ ($r \ll p$) contenant en colonne les coefficients des combinaisons linéaires des anciennes variables (les vecteurs engendrant le nouvel espace),
- une matrice Z de taille $n \times r$ contenant les r nouvelles variables,

telles que :

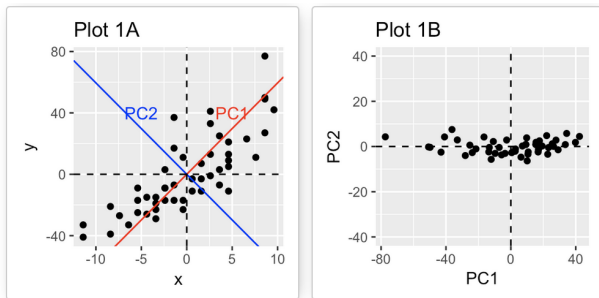
$$Z = XA.$$

Analyse factorielle

Analyse en Composantes Principales (ACP)

Les composantes principales Z^1, \dots, Z^r sont construites de telle sorte à garder le plus d'information possible contenue dans X^1, \dots, X^p : la variance des coordonnées des n individus sur chaque nouvel axe doit être maximale.

Ainsi, le nuage de points se répartit bien sur l'axe, gardant la diversité initiale du nuage, ce qui ne serait pas le cas si tous les points étaient projetés au même endroit (variance nulle).



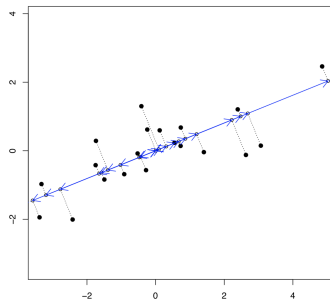
Analyse factorielle

Analyse en Composantes Principales (ACP)

Choix du nombre d'axes : déterminer le nombre r d'axes à retenir est une problématique centrale pour faire de la réduction de dimension.

Il existe de nombreux critères basés par exemple sur :

- la règle de Kaiser,
- l'éboulis des valeurs propres,
- la part d'inertie.



Analyse factorielle

Analyse en Composantes Principales sur R

```
library(FactoMineR)
```

```
# ACP sur les données
```

```
pca <- PCA(fromages,graph=FALSE)
```

```
pca$var$contrib # contribution des variables
```

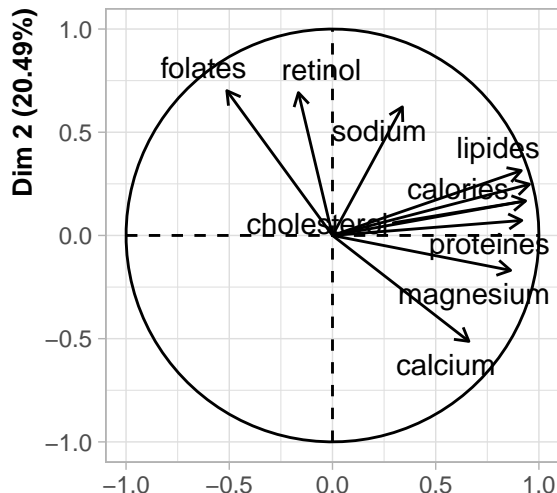
##	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
## calories	18.0381519	3.3377051	0.4098099	2.833464456	0.1375
## sodium	2.2695983	21.0354417	33.2181578	34.687197617	0.0290
## calcium	8.6489254	14.2744855	3.8710220	37.966028140	5.4498
## lipides	16.5715684	5.3536248	1.4469703	5.327536187	1.4223
## retinol	0.5395505	25.9659859	41.9841176	8.156922776	23.1419
## folates	5.1862456	26.6746427	2.9732139	1.538462405	57.9355
## proteines	16.6594517	0.2861261	2.7055987	0.180977683	4.6397
## cholesterol	17.3372783	1.5187216	0.1298788	9.306440451	1.4161
## magnesium	14.7492299	1.5532666	13.2612311	0.002970284	5.8277

Analyse factorielle

Analyse en Composantes Principales sur R

```
plot(pca, choix="var")
```

PCA graph of variables

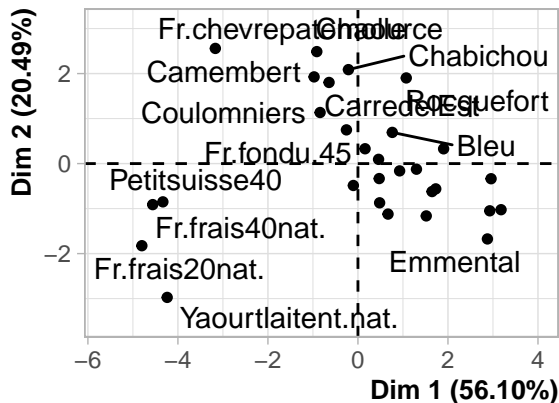


Analyse factorielle

Analyse en Composantes Principales sur R

```
plot(pca, choix="ind")
```

PCA graph of individuals

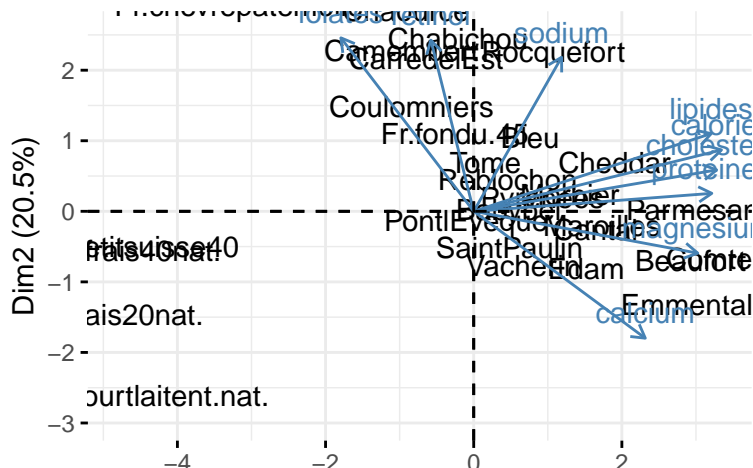


Analyse factorielle

Analyse en Composantes Principales sur R

```
library(factoextra)
fviz_pca_biplot(pca, geom = "text")
```

PCA – Biplot



Section 2

Apprentissage supervisé

- méthodes prédictives -

Introduction

En quoi consiste l'apprentissage supervisé?

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un vecteur d'observations Y à expliquer,
- un n -échantillon $(x_i^1, \dots, x_i^p, y_i)_{1 \leq i \leq n}$ de (X^1, \dots, X^p, Y) .

Motivation : utiliser les observations $(x_i^1, \dots, x_i^p, y_i)_{1 \leq i \leq n}$ pour prédire des informations, des comportements.

On parle d'apprentissage supervisé car les $(y_i)_{1 \leq i \leq n}$ permettent de guider le processus d'estimation.

Introduction

En quoi consiste l'apprentissage supervisé?

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un vecteur d'observations Y à expliquer,
- un n -échantillon $(x_i^1, \dots, x_i^p, y_i)_{1 \leq i \leq n}$ de (X^1, \dots, X^p, Y) .

Motivation : utiliser les observations $(x_i^1, \dots, x_i^p, y_i)_{1 \leq i \leq n}$ pour prédire des informations, des comportements.

Problème de régression si les y_i sont continus

classification si les y_i sont discrets

On parle d'apprentissage supervisé car les $(y_i)_{1 \leq i \leq n}$ permettent de guider le processus d'estimation.

Introduction

Vocabulaire

Ensemble d'apprentissage : les $(x_i^1, \dots, x_i^p, y_i)_{1 \leq i \leq n}$ utilisés pour mettre en place la règle de décision,

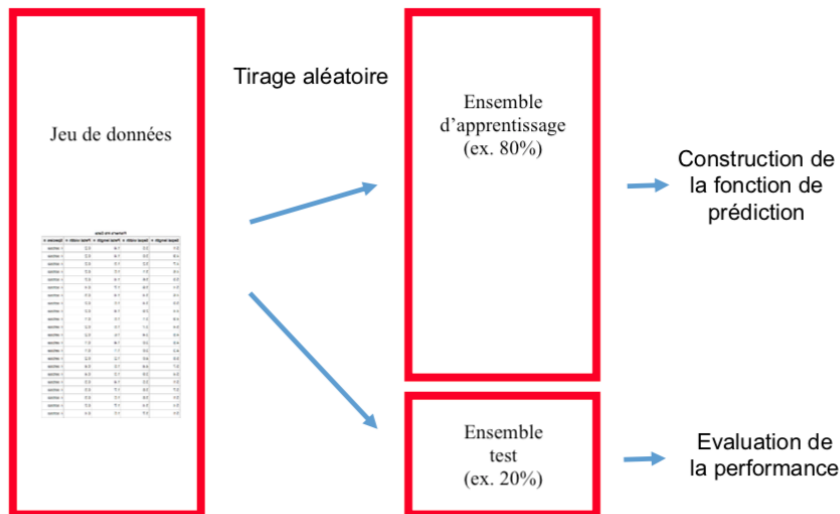
Fonction de prédiction : règle de décision permettant de prédire la valeur/classe de tout nouvel individu connaissant la valeur de ses p variables,

Ensemble test : toute nouvelle observation $(x_{n+1}^1, \dots, x_{n+1}^p)$ n'appartenant pas à l'échantillon d'apprentissage qui va nous permettre d'évaluer la performance de la fonction de prédiction.

Sur-apprentissage : lorsque le modèle mis en place sur des données est trop ajusté à ces données et ne se généralisent pas bien à d'autres données.

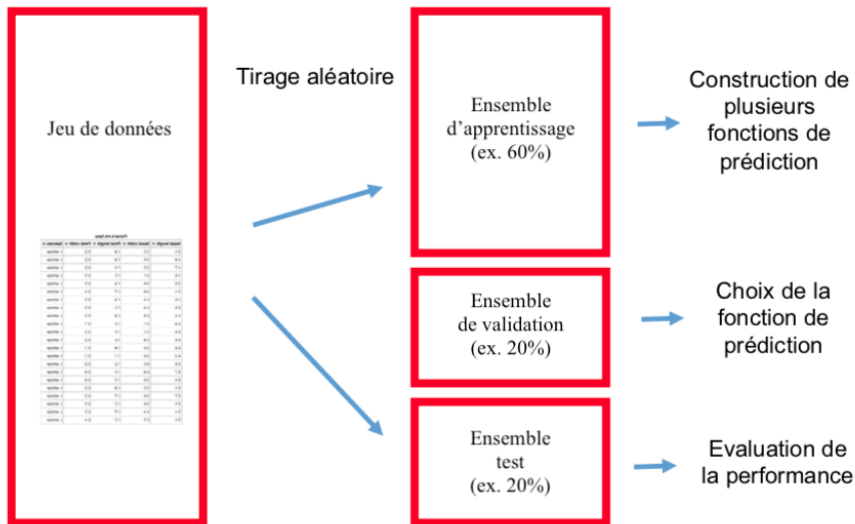
Introduction

Performances prédictives



Introduction

Performances prédictives



Introduction

Performances prédictives

La **validation croisée** est une technique de rééchantillonnage qui permet d'obtenir une mesure de la variabilité de l'erreur et de faire de la sélection de modèles.

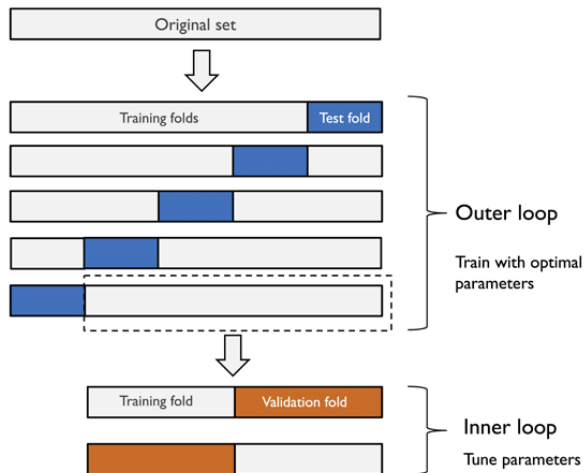
Principe :

- découpage de l'ensemble d'apprentissage en k groupes,
- construction de la fonction de prédiction sur l'ensemble des $k - 1$ groupes,
- évaluation de l'erreur sur le k -ième groupe,
- calculer la moyenne des k erreurs obtenues.

Si $k = n$, on parle de validation croisée leave-one-out, sinon, de k -folds validation croisée.

Introduction

Performances prédictives



Techniques de régression

Régression linéaire

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un vecteur d'observations Y à expliquer,
- un n -échantillon $(x_i^1, \dots, x_i^p, y_i)_{1 \leq i \leq n}$ de (X^1, \dots, X^p, Y) .

On appelle **modèle linéaire gaussien** un modèle statistique qui peut s'écrire sous la forme :

$$\forall i \in \llbracket 1, n \rrbracket, \quad Y_i = \beta_0 + \beta_1 X_i^1 + \dots + \beta_p X_i^p + \varepsilon_i,$$

où les $(\varepsilon_i)_{1 \leq i \leq n}$ sont des termes d'erreur non observés, i.i.d $\varepsilon_i \stackrel{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$.

Les paramètres $(\beta_1, \dots, \beta_p)$ expriment le lien linéaire qui existe entre Y et les variables (X^1, \dots, X^p) .

Techniques de régression

Moindres carrés et prédiction

Les paramètres du modèle peuvent être estimés par la **méthode des moindres carrés**, qui consiste à minimiser la somme des carrés des erreurs :

$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i^1 - \dots - \beta_p x_i^p)^2,$$

Les estimateurs $(\hat{\beta}_0, \dots, \hat{\beta}_p)$ ainsi définis sont alors utilisés dans un cadre de **prédiction** : étant donnée une nouvelle observation $(x_{n+1}^1, \dots, x_{n+1}^p)$

$$\hat{y}_{n+1} = \hat{\beta}_0 + \hat{\beta}_1 x_{n+1}^1 + \dots + \hat{\beta}_p x_{n+1}^p.$$

Techniques de régression

Moindres carrés et prédiction

Les paramètres du modèle peuvent être estimés par la **méthode des moindres carrés**, qui consiste à minimiser la somme des carrés des erreurs :

$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{n} \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i^1 - \dots - \beta_p x_i^p)^2,$$

Les estimateurs $(\hat{\beta}_0, \dots, \hat{\beta}_p)$ ainsi définis sont alors utilisés dans un cadre de **prédiction** : étant donnée une nouvelle observation $(x_{n+1}^1, \dots, x_{n+1}^p)$

$$\hat{y}_{n+1} = \hat{\beta}_0 + \hat{\beta}_1 x_{n+1}^1 + \dots + \hat{\beta}_{n+1} x_{n+1}^p.$$

Remarque : le modèle de régression linéaire ne fonctionne pas si Y est catégorielle et si l'on se trouve dans un cadre de grande dimension.

Techniques de régression

Régressions pénalisées

Dans le cadre de la grande dimension, des estimateurs pénalisés peuvent être mis en place. L'estimateur **Lasso** (Tibshirani, 1996) est ainsi défini par :

$$\hat{\beta}_{lasso} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \quad \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1.$$

- La pénalité lasso est utilisée pour obtenir des solutions parcimonieuses, c'est-à-dire telles que beaucoup de coefficients soient nuls (plus λ est grand, plus les solutions sont parcimonieuses).
- L'estimateur lasso est en général un estimateur de grande variance avec des problèmes de stabilité, notamment en présence de variables corrélées.
- L'estimateur Lasso n'a pas d'écriture propre, il faut le déterminer par un algorithme d'optimisation (*Lars*).

Techniques de régression

Régressions pénalisées sous R

```
# mise en place du modèle linéaire
```

```
model <- lm(calories~.,data=fromages)  
summary(model)
```

```
##  
## Call:  
## lm(formula = calories ~ ., data = fromages)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -10.9261  -5.8959  -0.3326   3.8848  17.9170   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)  27.21563    10.42523   2.611   0.0167 *      
## sodium       0.03141     0.02470   1.272   0.2180        
## calcium     -0.01740     0.05054  -0.344   0.7343        
## lipides      6.78355     1.02424   6.623 1.89e-06 ***  
## retinol     -0.04326     0.08679  -0.499   0.6236      
```

Techniques de régression

Régressions pénalisées sous R

```
# fromage de chèvre
newdata <- data.frame(sodium=297, calcium=91.6, lipides=16.9,
                      retinol=113, folates=43.1, proteines=13.1,
                      cholesterol=69.6, magnesium=12.1)

# prédiction
predict(model, newdata)

##          1
## 223.2935
```

Techniques de régression

Régressions pénalisées sous R

```
library(glmnet)

# méthode lasso
lasso <- glmnet(y = fromages$calories, x = fromages[,2:9],
                lambda=0.2)

lasso$beta

## 8 x 1 sparse Matrix of class "dgCMatrix"
##              s0
## sodium      0.02454328
## calcium      .
## lipides      6.99493416
## retinol     -0.03314832
## folates     -0.01839563
## proteines    3.19426401
## cholesterol 0.50332418
## magnesium    0.05419546
```


Techniques de régression

Régression logistique

Lorsque Y est qualitative (0/1 par exemple), on utilisera le **modèle de régression logistique**, qui s'écrit sous la forme :

$$\begin{cases} Y \sim \mathcal{B}(p), \\ \log \left(\frac{\mathbb{P}(Y=1)}{\mathbb{P}(Y=0)} \right) = X\beta. \end{cases}$$

Remarque : le modèle de régression logistique ne s'écrit pas $Y = X\beta + \varepsilon$.

```
# préparation des données
```

```
TrainSet <- sample(nrow(ptitanic))[1:floor(80/100*nrow(ptitanic))]
```

```
TestSet <- c(1:nrow(ptitanic))[-TrainSet]
```

```
ptitanic_Train <- ptitanic[TrainSet,]
```

```
ptitanic_Test <- ptitanic[TestSet,]
```

Techniques de régression

Régression logistique (R)

modèle linéaire logistique sur l'échantillon d'apprentissage

```
model <- glm(survived~.,data=ptitanic_Train,family="binomial")
summary(model)
```

```
##
## Call:
## glm(formula = survived ~ ., family = "binomial", data = ptitanic,
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3181  -0.6522  -0.4199   0.6516   2.4846
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.759889   0.405240   9.278  < 2e-16 ***
## pclass2nd    -1.239986   0.264328  -4.691 2.72e-06 ***
## pclass3rd    -2.221022   0.256853  -8.647  < 2e-16 ***
## sexmale      -2.642046   0.193486 -13.655  < 2e-16 ***
## age          -0.036596   0.007484  -4.890 1.01e-06 ***
```

Techniques de régression

Régression logistique (R)

```
# prédiction sur l'échantillon test
```

```
tit_prob <- predict(model, ptitanic_Test, type = "response")  
head(tit_prob)
```

```
##           3           4           11           12           25           30  
## 0.9694669 0.4479897 0.2876986 0.9424994 0.9369445 0.5232726
```

```
# performance de la méthode
```

```
tit_pred <- rep("died", length(TestSet))  
tit_pred[tit_prob > .5] = "survived"  
table(tit_pred, ptitanic_Test$survived)
```

```
##  
## tit_pred   died survived  
##   died     107       31  
##   survived  17       55
```

```
mean(tit_pred == ptitanic_Test$survived)
```

```
## [1] 0.7714286
```

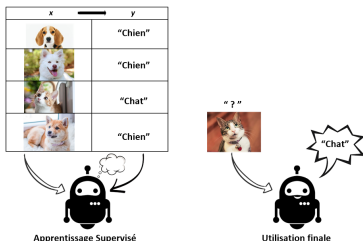
Techniques de classification

Objectifs

On considère :

- p variables explicatives (X^1, \dots, X^p) ,
- un n -échantillon $(x_i^1, \dots, x_i^p)_{1 \leq i \leq n}$ de (X^1, \dots, X^p) ,
- une partition des n individus en k classes (donnée par le vecteur $(y_i)_{1 \leq i \leq n}$).

Objectif : mettre en place une règle de décision qui attribue l'une des k classes à tout nouvel individu.



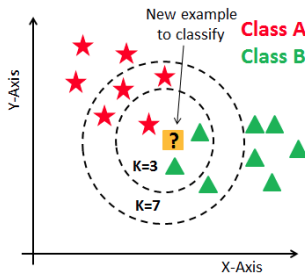
Techniques de classification

Méthode des k plus proches voisins

Principe :

- on compte parmi les k plus proches voisins d'un point x_{n+1} à classer le nombre de points $(n_\ell)_{1 \leq \ell \leq k}$ appartenant à chacune des classes.
- on estime la probabilité que x_{n+1} appartienne à la classe ℓ par :

$$\mathbb{P}(x_{n+1} \in C_\ell) = \frac{n_\ell}{k}.$$



Techniques de classification

Méthode des k plus proches voisins (R)

```
library(class)

# préparation des données (pas de factor)
ptitanic$pclass <- is.numeric(ptitanic$pclass)
ptitanic$sex <- is.numeric(ptitanic$sex)

# knn
results <- knn(ptitanic[TrainSet,-2],ptitanic[TestSet,-2],
               cl=ptitanic[TrainSet,"survived"],k=2)
head(results)

## [1] survived died      survived died      died      died
## Levels: died survived
```

Techniques de classification

Méthode des k plus proches voisins (R)

```
# performances  
table(results, ptitanic_Test$survived)
```

```
##  
## results      died survived  
##    died        94         53  
##    survived    30         33
```

```
mean(results == ptitanic_Test$survived)
```

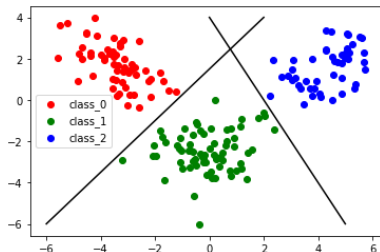
```
## [1] 0.6047619
```

Techniques de classification

Analyse Discriminante Linéaire (LDA)

L'**analyse discriminante linéaire** est un algorithme utilisé dans le cadre d'apprentissage supervisé. Il y a deux points de vues différents permettant de l'appréhender :

- géométriquement, il s'agit de chercher des hyperplans qui séparent au mieux les groupes,
- mathématiquement, cela revient à supposer que les lois des covariables sont des vecteurs gaussiens avec des valeurs de paramètres qui dépendent des classes.



Techniques de classification

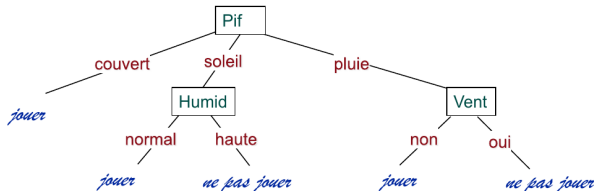
Arbres de décision

Les arbres de décision sont définis de la manière suivante :

- **racine** : ensemble de l'échantillon,
- **nœud** : une variable (au choix) et une division (partition de l'échantillon en 2 classes).

La **règle de division** doit être une règle facilement interprétable :

- pour une variable explicative quantitative : choix d'un seuil,
- pour une variable explicative qualitative : choix d'un groupe de modalités.



Techniques de classification

Arbres de décision

Principe :

- on cherche la variable permettant de découper le mieux possible l'échantillon en deux sous-ensembles,
- on utilise cette variable pour définir une partition de l'espace en deux sous-ensembles,
- on recommence récursivement jusqu'à ce que
 - ▶ il n'y ait plus de découpage possible,
 - ▶ si les sous-ensembles ont atteint une taille minimale fixée en amont.

Techniques de classification

Elegage des arbres de décision

Les arbres de décision sont souvent trop fins et donc instables (sur-ajustement des données). La dernière étape de l'algorithme consiste donc à élaguer l'arbre afin d'en réduire la complexité. Une solution pour éviter l'exploration complète de tous les sous-arbres possibles consiste à utiliser la méthode **CART** (Breiman et al., 1984).

Principe :

- construction de l'arbre maximal,
- construction d'une séquence d'arbres emboîtés
 - ▶ les arbres obtenus en supprimant l'une des dernières divisions sont comparés suivant un critère d'erreur de classement
 - ▶ l'arbre retenu est le meilleur
 - ▶ on remonte ainsi jusqu'à la racine,
- comparaison des arbres obtenus à l'aide d'un taux de mauvais classement estimé sur l'échantillon test.

Techniques de classification

Arbres de décision (R)

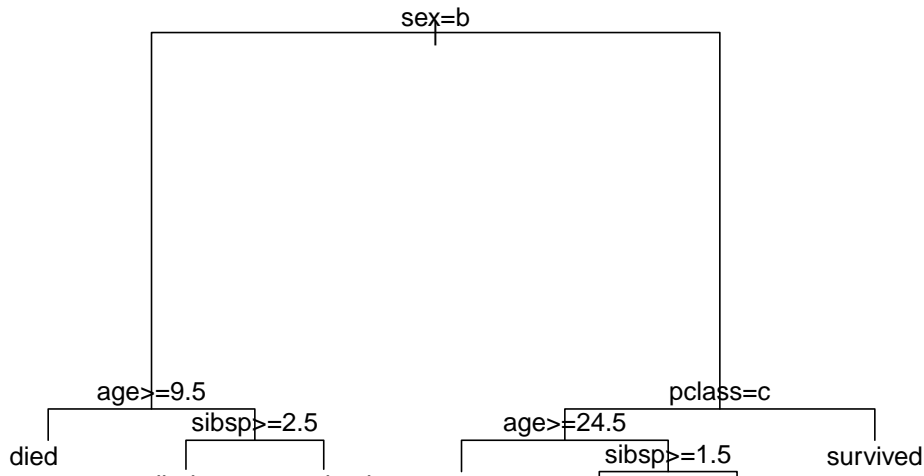
```
library(rpart)
model <- rpart(survived ~., data = ptitanic_Train,method="class")
print(model)
```

```
## n= 836
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
## 1) root 836 334 died (0.60047847 0.39952153)
##    2) sex=male 536 106 died (0.80223881 0.19776119)
##      4) age>=9.5 501  86 died (0.82834331 0.17165669) *
##      5) age< 9.5 35  15 survived (0.42857143 0.57142857)
##        10) sibsp>=2.5 14  1 died (0.92857143 0.07142857) *
##        11) sibsp< 2.5 21  2 survived (0.09523810 0.90476190) *
##    3) sex=female 300  72 survived (0.24000000 0.76000000)
##      6) pclass=3rd 117  58 died (0.50427350 0.49572650)
##        12) age>=24.5 44  17 died (0.61363636 0.38636364) *
##        13) age< 24.5 73  32 survived (0.43835616 0.56164384)
```

Techniques de classification

Arbres de décision (R)

```
par(xpd = NA) # otherwise on some devices the text is clipped  
plot(model)  
text(model, digits = 3)
```



Techniques de classification

Arbres de décision (R)

```
# prédiction  
head(predict(model, ptitanic_Test))
```

```
##           died  survived  
## 7  0.07103825 0.9289617  
## 12 0.07103825 0.9289617  
## 18 0.07103825 0.9289617  
## 23 0.82834331 0.1716567  
## 26 0.82834331 0.1716567  
## 30 0.82834331 0.1716567
```

```
predictions <- predict(model, ptitanic_Test, type="class")  
head(predictions)
```

```
##           7           12           18           23           26           30  
## survived survived survived      died      died      died  
## Levels: died survived
```

Techniques de classification

Arbres de décision (R)

```
# performances de la méthode  
table(predictions, ptitanic_Test$survived)
```

```
##  
## predictions died survived  
##      died      105      30  
##      survived   12      63
```

```
mean(predictions==ptitanic_Test$survived)
```

```
## [1] 0.8
```

Section 3

A vous de jouer!