



Apprentissage supervisé

Fabrice Rossi

TELECOM ParisTech

Mai/Juin 2009

Introduction et modélisation mathématique

Apprentissage supervisé

Qualité d'un modèle

Régression

Régression linéaire

Régularisation

Non linéaire

Discrimination

Moindres carrés

Analyse discriminante

Maximisation de la marge

Non linéaire

Sélection de modèle

Introduction et modélisation mathématique

Apprentissage supervisé

Qualité d'un modèle

Régression

Régression linéaire

Régularisation

Non linéaire

Discrimination

Moindres carrés

Analyse discriminante

Maximisation de la marge

Non linéaire

Sélection de modèle



Définition informelle

1. observations d'un phénomène
 2. construction d'un modèle de ce phénomène
 3. prévisions et analyse du phénomène grâce au modèle
- le tout automatiquement (sans intervention humaine)



Définition informelle

1. observations d'un phénomène
 2. construction d'un modèle de ce phénomène
 3. prévisions et analyse du phénomène grâce au modèle
- le tout automatiquement (sans intervention humaine)

Modélisation mathématique :

- observations d'un phénomène \Rightarrow des données $z_i \in \mathcal{Z}$



Définition informelle

1. observations d'un phénomène
 2. construction d'un modèle de ce phénomène
 3. prévisions et analyse du phénomène grâce au modèle
- le tout automatiquement (sans intervention humaine)

Modélisation mathématique :

- observations d'un phénomène \Rightarrow des données $z_i \in \mathcal{Z}$
- deux grandes catégories de données :
 1. cas non supervisé :
 - pas de structure interne à z
 - classification, règles d'association, etc.



Définition informelle

1. observations d'un phénomène
 2. construction d'un modèle de ce phénomène
 3. prévisions et analyse du phénomène grâce au modèle
- le tout automatiquement (sans intervention humaine)

Modélisation mathématique :

- observations d'un phénomène \Rightarrow des données $z_i \in \mathcal{Z}$
- deux grandes catégories de données :
 1. cas **non supervisé** :
 - pas de structure interne à z
 - classification, règles d'association, etc.
 2. cas **supervisé** :
 - $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$
 - modélisation du lien entre x et y
 - pour faire des prévisions : connaissant x , on prédit y



Apprentissage supervisé

■ discrimination/classement :

- $\mathcal{Y} = \{1, \dots, q\}$: q classes d'objets
- prévision : placer une nouvelle observation x dans une des q classes
- **applications** : diagnostic médical (malade/sain), reconnaissance de caractères, etc.



Apprentissage supervisé

■ discrimination/classement :

- $\mathcal{Y} = \{1, \dots, q\}$: q classes d'objets
- prévision : placer une nouvelle observation x dans une des q classes
- **applications** : diagnostic médical (malade/sain), reconnaissance de caractères, etc.

■ ranking/scoring :

- apprendre un ordre sur un ensemble d'objets
- prévision : donner des objets intéressants (grands au sens de l'ordre) ; dire si un objet est plus intéressant qu'un autre ; donne un score d'intérêt à un objet
- $\mathcal{Y} = \{0, 1\}$: 1 pour intéressant, 0 pour intéressant
- autres choix possibles pour \mathcal{Y} (par ex. \mathbb{R} ou tout ensemble ordonné)
- **applications** : recherche d'informations (*page rank* de Google), suggestions (amazon, netflix)



■ régression :

- $\mathcal{Y} = \mathbb{R}$ ou $\mathcal{Y} = \mathbb{R}^p$
- prévision : associer une valeur numérique à une nouvelle observation
- **applications** : certaines formes de *scoring* (note d'un objet, d'un consommateur), prévisions de la valeur future d'une action, etc.



■ régression :

- $\mathcal{Y} = \mathbb{R}$ ou $\mathcal{Y} = \mathbb{R}^p$
- prévision : associer une valeur numérique à une nouvelle observation
- **applications** : certaines formes de *scoring* (note d'un objet, d'un consommateur), prévisions de la valeur future d'une action, etc.

■ sortie structurée :

- \mathcal{Y} est un ensemble structuré complexe : ensemble de fonctions, chaînes de caractères, arbres, graphes, etc.
- prévision : associer un objet de l'ensemble complexe à une nouvelle observation
- **application** : inférence grammaticale (associer un arbre de syntaxe à un texte), traduction automatique, etc.



Vocabulaire

- x : variables **explicatives** (espace associé \mathcal{X})
- y : variable **à expliquer** (espace associé \mathcal{Y})
- un **modèle** g : une fonction de \mathcal{X} dans \mathcal{Y}
- $g(x)$ est la **prédition/prévision** du modèle pour l'entrée x
- l'ensemble des données à partir desquelles on construit le modèle est **l'ensemble d'apprentissage**
- collisions Français et Anglais :

Français	Anglais
Classification	<i>Clustering</i>
Classement	<i>Classification ou ranking</i>
Discrimination	<i>Classification</i>

■ buts principaux :

- obtenir un « **bon** » modèle : la prévision obtenue est proche de la vraie valeur
- obtenir **rapidement** un modèle **rapide** : temps de construction du modèle et temps nécessaire à l'obtention d'une prévision
- pouvoir **garantir** les performances : avec une probabilité de $1 - r$, la prévision sera bonne à ϵ près

■ buts annexes :

- obtenir un modèle **compréhensible** : comment le modèle prend il une décision ?
- obtenir un modèle **modifiable** : pouvoir prendre en compte de nouvelles données, s'adapter à un environnement changeant, etc.



Erreur de prédiction

Qu'est-ce qu'une bonne prédiction ?

- on considère une observation $z = (x, y)$ et une prédiction $g(x)$ faite par un modèle
- la qualité de $g(x)$ peut être mesurée par une dissimilarité / définie sur \mathcal{Y} : $l(g(x), y)$ doit être petit
- l est le **critère d'erreur** :
 - régression :
 - distances classiques sur \mathbb{R}^p
 - en général $\|g(x) - y\|^2$ et parfois $|g(x) - y|$ dans \mathbb{R} pour les méthodes de régression dites robustes
 - discrimination :
 - décompte des erreurs : $\delta_{g(x) \neq y}$
 - matrice des coûts de confusion : par ex. prédire $g(x) = 1$ alors que $y = 0$ peut être plus coûteux que prédire $g(x) = 0$ quand $y = 1$ (diagnostic médical)



Erreur d'un modèle

Qu'est-ce qu'un bon modèle ?

■ Vision « naïve » :

- données d'évaluation $\mathcal{T}_M = (x_i, y_i)_{i=1}^M$
- l est le critère d'erreur dans \mathcal{Y}
- l'erreur du modèle g est donnée par

$$\hat{L}(g; \mathcal{T}_M) = \frac{1}{M} \sum_{i=1}^M l(g(x_i), y_i)$$

- erreur du modèle : moyenne des erreurs de prédiction
- erreur empirique



Erreur d'un modèle

Qu'est-ce qu'un bon modèle ?

■ Vision « naïve » :

- données d'évaluation $\mathcal{I}_M = (x_i, y_i)_{i=1}^M$
- I est le critère d'erreur dans \mathcal{Y}
- l'erreur du modèle g est donnée par

$$\hat{L}(g; \mathcal{I}_M) = \frac{1}{M} \sum_{i=1}^M I(g(x_i), y_i)$$

- erreur du modèle : moyenne des erreurs de prédiction
- erreur empirique

■ interprétation intuitive :

- exigence raisonnable : ne pas se tromper en moyenne
- la moyenne résume bien la dispersion des erreurs



Erreur d'un modèle

Qu'est-ce qu'un bon modèle ?

■ modélisation statistique du processus :

- on suppose que le phénomène étudié est engendré par une loi de probabilité P **inconnue** sur $\mathcal{X} \times \mathcal{Y}$
- chaque couple observé (x, y) est tiré aléatoirement selon P



Erreur d'un modèle

Qu'est-ce qu'un bon modèle ?

- modélisation statistique du processus :
 - on suppose que le phénomène étudié est engendré par une loi de probabilité P inconnue sur $\mathcal{X} \times \mathcal{Y}$
 - chaque couple observé (x, y) est tiré aléatoirement selon P
- l'erreur du modèle g est donnée par

$$L(g) = E_P\{I(g(x), y)\}$$

c.-à-d. l'espérance de l'erreur de prédiction sous la distribution des données



Erreur d'un modèle

Qu'est-ce qu'un bon modèle ?

- modélisation statistique du processus :
 - on suppose que le phénomène étudié est engendré par une loi de probabilité P inconnue sur $\mathcal{X} \times \mathcal{Y}$
 - chaque couple observé (x, y) est tiré aléatoirement selon P
- l'erreur du modèle g est donnée par

$$L(g) = E_P\{I(g(x), y)\}$$

c.-à-d. l'espérance de l'erreur de prédiction sous la distribution des données

- remarque : le calcul exact de $L(g)$ est impossible car P est inconnue

■ pourquoi de l'aléatoire ?

- bruit dans les observations
- données incomplètes
- variabilité naturelle

■ pourquoi une distribution P fixée ?

- stationnarité
- condition nécessaire à l'inférence : si un phénomène change constamment, on ne peut pas le prédire
- extensions possibles aux variations lentes

■ pourquoi l'espérance ?

- naturelle dans un cadre statistique
- pour s'affranchir de la variabilité des nouvelles observations



Pratique vs statistique

- la loi des grands nombres dit que

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N U_i = E(U)$$

quand les U_i sont indépendantes et distribuées comme U

- si les données d'évaluation $\mathcal{T}_M = (x_i, y_i)_{i=1}^M$ sont distribuées selon P et indépendantes, alors

$$\lim_{M \rightarrow \infty} \hat{L}(g; \mathcal{T}_M) = L(g)$$



Pratique vs statistique

- la loi des grands nombres dit que

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N U_i = E(U)$$

quand les U_i sont indépendantes et distribuées comme U

- si les données d'évaluation $\mathcal{T}_M = (x_i, y_i)_{i=1}^M$ sont distribuées selon P et indépendantes, alors

$$\lim_{M \rightarrow \infty} \hat{L}(g; \mathcal{T}_M) = L(g)$$

- indépendance statistique ?

- (x_i, y_i) est choisie sans rien savoir des tirages précédents
- chaque observation (x_i, y_i) apporte de nouvelles informations



Interprétation

■ comment interpréter $L(g)$?

- on considère M observations $(x_i, y_i)_{i=1}^M$
- on calcule

$$\hat{L}(g; (x_i, y_i)_{i=1}^M) = \frac{1}{M} \sum_{i=1}^M l(g(x_i), y_i)$$

- alors pour M « grand », $\hat{L}(g; (x_i, y_i)_{i=1}^M) \simeq L(g)$



Interprétation

■ comment interpréter $L(g)$?

- on considère M observations $(x_i, y_i)_{i=1}^M$
- on calcule

$$\hat{L}(g; (x_i, y_i)_{i=1}^M) = \frac{1}{M} \sum_{i=1}^M I(g(x_i), y_i)$$

- alors pour M « grand », $\hat{L}(g; (x_i, y_i)_{i=1}^M) \simeq L(g)$

■ remarque :

- en discrimination, $\mathcal{Y} = \{1, \dots, q\}$
- si $I(g(x), y) = \delta_{g(x) \neq y}$, alors la qualité

$$L(g) = E_P\{I(g(x), y)\} = P(g(x) \neq y)$$

correspond à la probabilité d'erreur de classement



Interprétation

- par exemple en discrimination à deux classes, avec $L(g) = 0.1$:
 - la probabilité de se tromper de classe est de 10 %
 - en moyenne sur un grand nombre d'observations, on se trompera une fois sur dix
 - cela n'exclut pas de se tromper 5 fois de suite, la probabilité est simplement faible :
 - 1 chance sur cent mille
 - si on répète de très nombreuses fois le tirage de 5 observations, alors on se trompera sur les 5 observations seulement dans un cas sur cent mille en moyenne



Interprétation

- par exemple en discrimination à deux classes, avec $L(g) = 0.1$:
 - la probabilité de se tromper de classe est de 10 %
 - en moyenne sur un grand nombre d'observations, on se trompera une fois sur dix
 - cela n'exclut pas de se tromper 5 fois de suite, la probabilité est simplement faible :
 - 1 chance sur cent mille
 - si on répète de très nombreuses fois le tirage de 5 observations, alors on se trompera sur les 5 observations seulement dans un cas sur cent mille en moyenne
- on peut donner des intervalles de confiance sur $\hat{L}(g; (x_i, y_i)_{i=1}^M)$ autour de $L(g)$ en fonction de M de la forme

$$P \left\{ \left| \hat{L}(g; (x_i, y_i)_{i=1}^M) - L(g) \right| > \epsilon \right\} < 1 - \delta$$



Définition informelle

L'erreur en généralisation d'un modèle est celle des prédictions obtenues sur des nouvelles observations

- notion cruciale en apprentissage supervisé
- mathématiquement, il s'agit simplement de $L(g)$
- **problème fondamental** : comment estimer l'erreur en généralisation alors qu'on ne connaît pas P ?
- loi des grands nombres ?



Problème d'estimation

- processus d'apprentissage :
 - ensemble d'apprentissage : N observations
 $\mathcal{D}_N = (x_i, y_i)_{i=1}^N$, distribuées selon P et indépendantes
 - l'algorithme choisi construit un modèle g qui dépend de \mathcal{D}_N
- que dire de

$$\hat{L}(g; (x_i, y_i)_{i=1}^N) = \frac{1}{N} \sum_{i=1}^N I(g(x_i), y_i)$$



Problème d'estimation

- processus d'apprentissage :
 - ensemble d'apprentissage : N observations
 $\mathcal{D}_N = (x_i, y_i)_{i=1}^N$, distribuées selon P et indépendantes
 - l'algorithme choisi construit un modèle g qui dépend de \mathcal{D}_N
- que dire de

$$\hat{L}(g; (x_i, y_i)_{i=1}^N) = \frac{1}{N} \sum_{i=1}^N I(g(x_i), y_i)$$

- rien (simplement) car la loi des grands nombres ne s'applique pas ici :
 - les (x_i, y_i) sont indépendants
 - mais les $I(g(x_i), y_i)$ ne le sont pas à cause de g



K plus proches voisins

- algorithme classique de discrimination/régression
- N observations $\mathcal{D}_N = (x_i, y_i)_{i=1}^N$ et un paramètre K
- on suppose que \mathcal{X} est muni d'une dissimilarité d
- algorithme de calcul de $g_K(x)$:
 1. calcul des dissimilarités $d(x, x_i)$ pour $1 \leq i \leq N$
 2. tri des dissimilarités tels que $d(x, x_{j_i}) \leq d(x, x_{j_{i+1}})$
 3. $g_K(x)$ est
 - la classe majoritaire dans les K labels y_{j_1}, \dots, y_{j_k} en discrimination
 - le centre de gravité des K vecteurs y_{j_1}, \dots, y_{j_k} en régression



K plus proches voisins

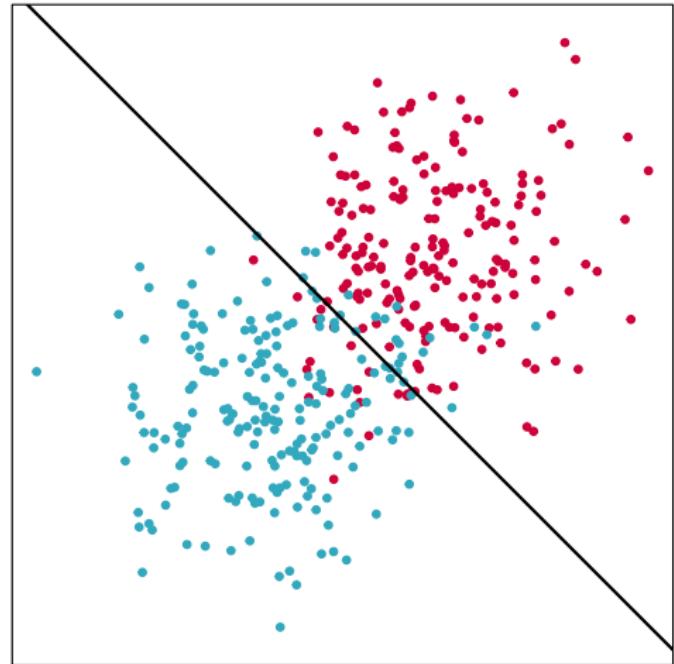
- algorithme classique de discrimination/régression
- N observations $\mathcal{D}_N = (x_i, y_i)_{i=1}^N$ et un paramètre K
- on suppose que \mathcal{X} est muni d'une dissimilarité d
- algorithme de calcul de $g_K(x)$:
 1. calcul des dissimilarités $d(x, x_i)$ pour $1 \leq i \leq N$
 2. tri des dissimilarités tels que $d(x, x_{j_i}) \leq d(x, x_{j_{i+1}})$
 3. $g_K(x)$ est
 - la classe majoritaire dans les K labels y_{j_1}, \dots, y_{j_K} en discrimination
 - le centre de gravité des K vecteurs y_{j_1}, \dots, y_{j_K} en régression
- on a $g_1(x_i) = y_i$ et donc pour tout critère / raisonnable,

$$\hat{L}(g_1; (x_i, y_i)_{i=1}^N) = \frac{1}{N} \sum_{i=1}^N I(g_1(x_i), y_i) = 0$$



Exemple

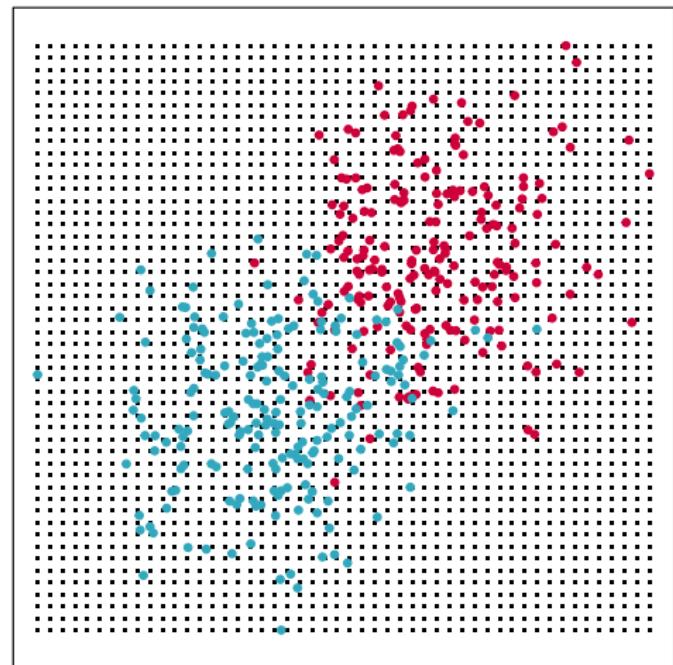
- classement
- frontière optimale linéaire





Exemple

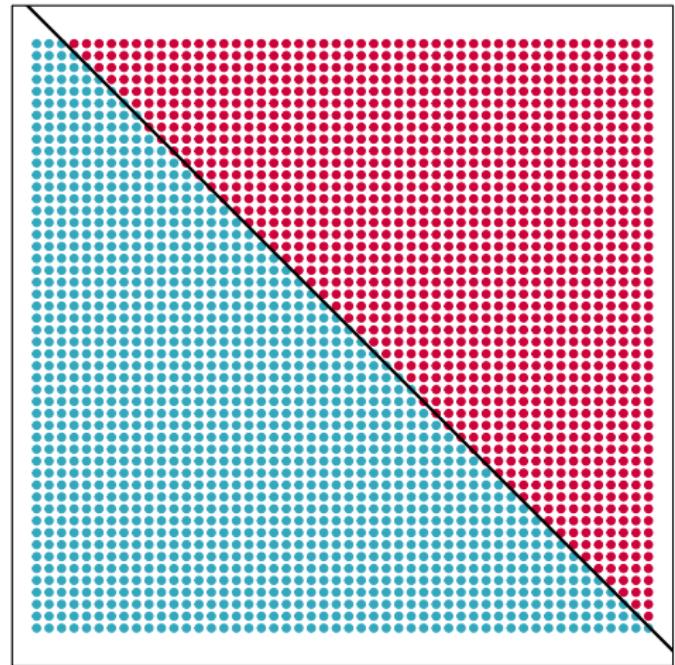
- classement
- frontière optimale linéaire
- grille d'évaluation





Exemple

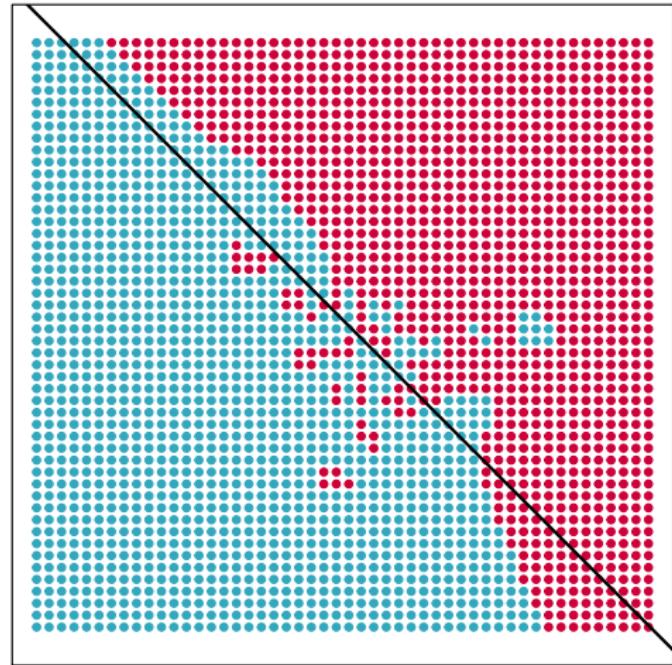
- classement
- frontière optimale linéaire
- classement optimal





Exemple

- classement
- frontière optimale linéaire
- $L(g) \simeq 0.0968$
- $\hat{L}(g) = 0$

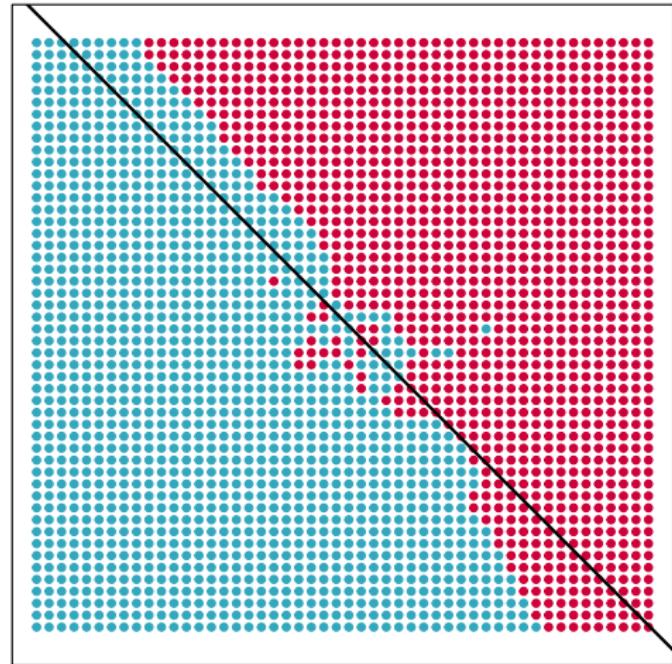


1 voisin



Exemple

- classement
- frontière optimale linéaire
- $L(g) \simeq 0.0892$
- $\hat{L}(g) = 0.065$

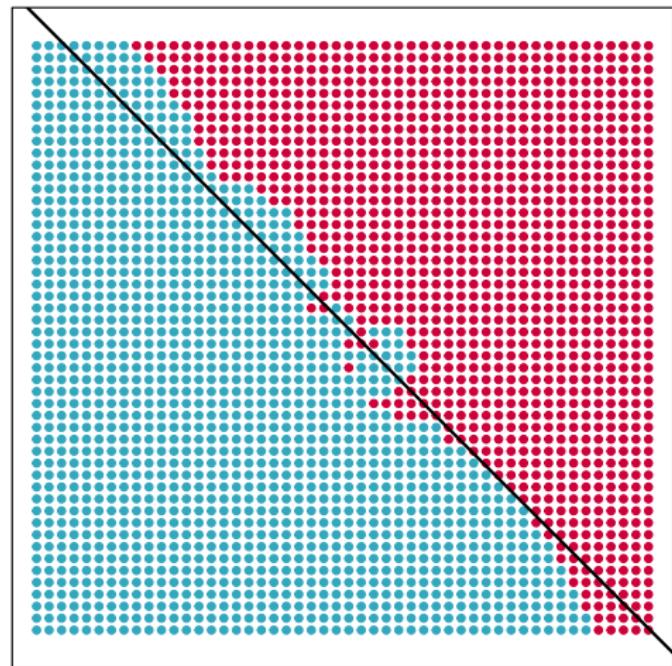


3 voisins



Exemple

- classement
- frontière optimale linéaire
- $L(g) \simeq 0.0524$
- $\hat{L}(g) = 0.085$

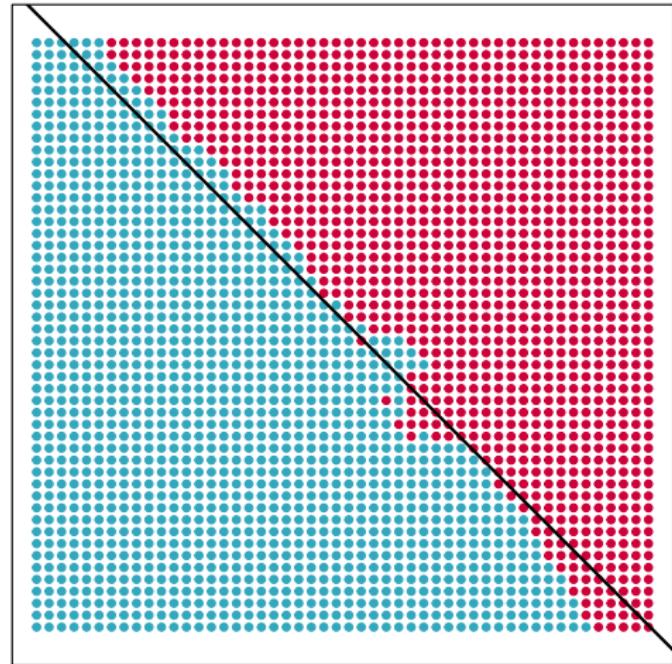


5 voisins



Exemple

- classement
- frontière optimale linéaire
- $L(g) \simeq 0.0416$
- $\hat{L}(g) = 0.0875$

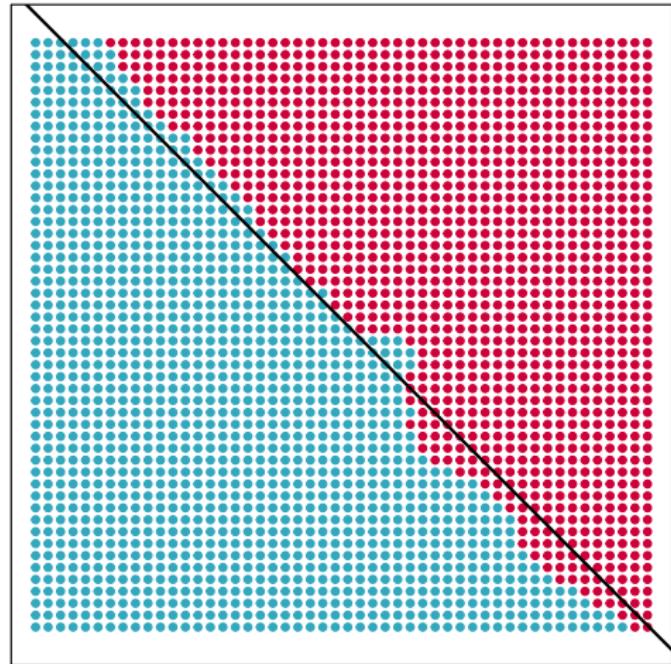


11 voisins



Exemple

- classement
- frontière optimale linéaire
- $L(g) \simeq 0.0404$
- $\hat{L}(g) = 0.085$

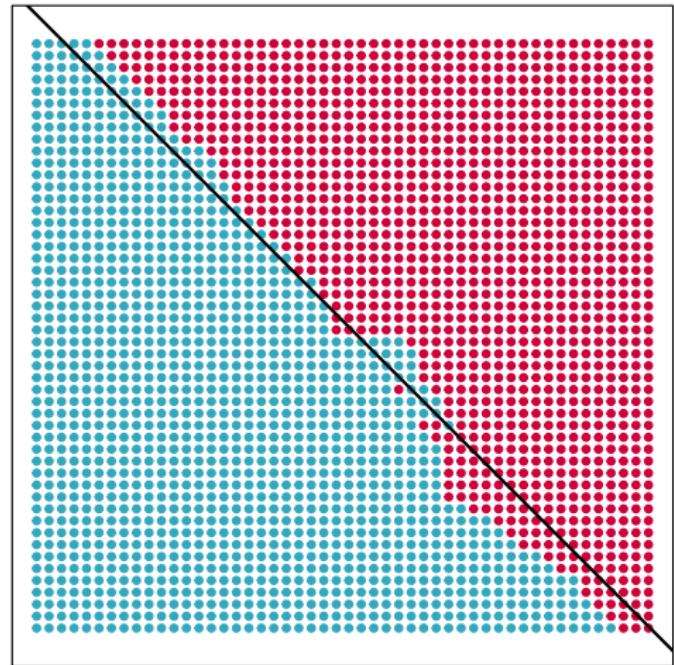


15 voisins



Exemple

- classement
- frontière optimale linéaire
- $L(g) \simeq 0.0456$
- $\hat{L}(g) = 0.095$



21 voisins

- l'exemple précédent est très représentatif :
 - on peut souvent construire g tel que $g(x_i) = y_i$ sur \mathcal{D}_N
 - pour un critère d'erreur raisonnable, on aura donc
 $\hat{L}(g; (x_i, y_i)_{i=1}^N) = 0$
 - mais en général, $L(g) > 0$
- l'erreur empirique sur l'ensemble d'apprentissage est généralement (très) optimiste
- c'est une mauvaise estimation de l'erreur en généralisation

Point à retenir

obtenir une bonne estimation des performances d'un modèle est la principale difficulté de l'apprentissage automatique



■ l'apprentissage supervisé

- construit un modèle pour prédire y à partir de x
- en s'appuyant sur un ensemble d'apprentissage constitué d'exemples d'associations (x, y)

■ suite du cours :

- quelques modèles et algorithmes associés
- méthodologie :
 - comment évaluer les performances d'un modèle ?
 - comment choisir un bon modèle ?

Introduction et modélisation mathématique

Apprentissage supervisé

Qualité d'un modèle

Régression

Régression linéaire

Régularisation

Non linéaire

Discrimination

Moindres carrés

Analyse discriminante

Maximisation de la marge

Non linéaire

Sélection de modèle



Régression linéaire

■ exemple le plus élémentaire d'apprentissage automatique :

- on dispose de N couples de réels (x_i, y_i) (l'ensemble d'apprentissage)
- on cherche deux réels a et b tels que $y_i \simeq ax_i + b$ pour tout $1 \leq i \leq N$
- le modèle est linéaire :
 - la fonction qui aux paramètres associe le modèle est linéaire $(a, b) \mapsto (x \mapsto ax + b)$
 - le modèle lui même est affine

■ stratégie de construction du modèle :

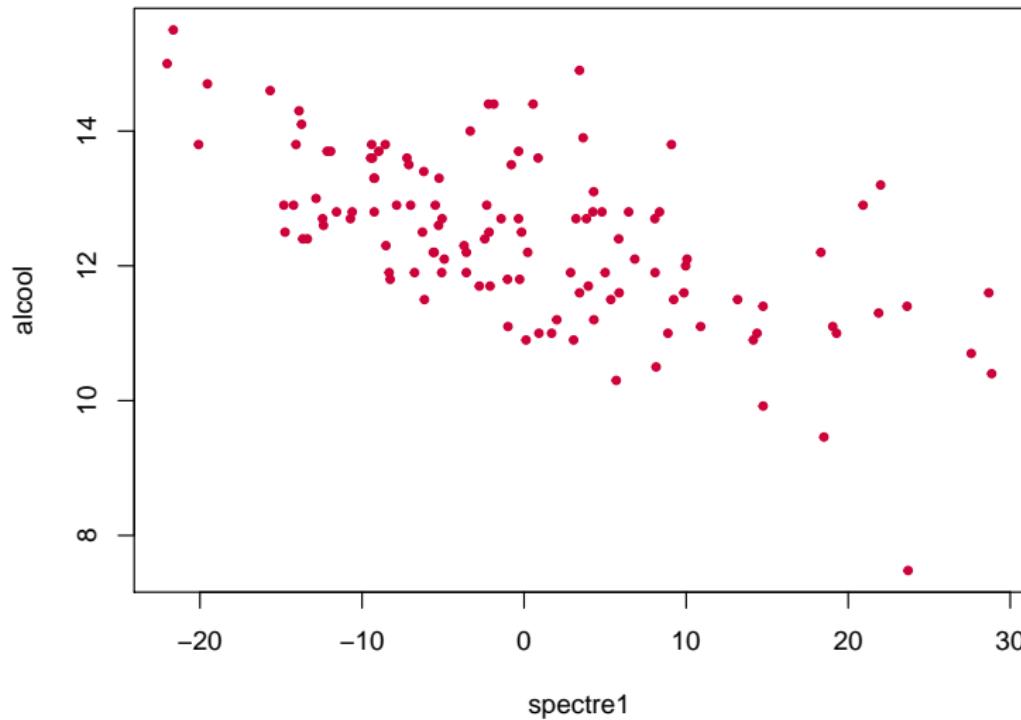
- minimisation de l'erreur des moindres carrés

$$(a^*, b^*) = \arg \min_{a,b} \sum_{i=1}^N (ax_i + b - y_i)^2$$



Exemple

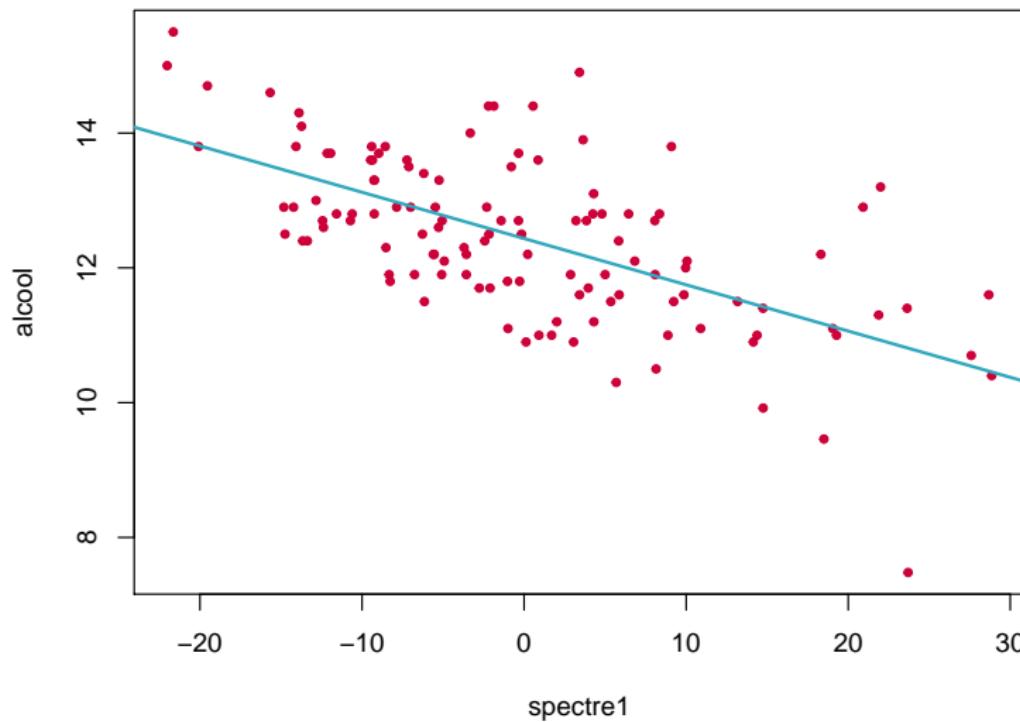
Analyse spectroscopique de vins





Exemple

Analyse spectroscopique de vins





Résolution

- si $E(a, b) = \sum_{i=1}^N (ax_i + b - y_i)^2$, on a

$$\nabla_a E(a, b) = 2 \left(a \sum_{i=1}^N x_i^2 + \sum_{i=1}^N x_i(b - y_i) \right)$$

et

$$\nabla_b E(a, b) = 2N \left(b + \frac{1}{N} \sum_{i=1}^N (ax_i - y_i) \right)$$

- $\nabla E = 0$ conduit à une unique solution (a^*, b^*)



Leçon générale

- méthode de construction de la régression linéaire
 - choix d'une classe de modèles

$$\mathcal{F} = \{f : \mathbb{R} \rightarrow \mathbb{R} \mid \exists (a, b) \in \mathbb{R}^2, f(x) = ax + b\}$$

- choix dans la classe du modèle d'erreur empirique minimale sur l'ensemble d'apprentissage

$$f^* = \arg \min_{f \in \mathcal{F}} \hat{L}(f; \{(x_1, y_1), \dots, (x_N, y_N)\})$$



Leçon générale

- méthode de construction de la régression linéaire
 - choix d'une classe de modèles

$$\mathcal{F} = \{f : \mathbb{R} \rightarrow \mathbb{R} \mid \exists (a, b) \in \mathbb{R}^2, f(x) = ax + b\}$$

- choix dans la classe du modèle d'erreur empirique minimale sur l'ensemble d'apprentissage

$$f^* = \arg \min_{f \in \mathcal{F}} \hat{L}(f; \{(x_1, y_1), \dots, (x_N, y_N)\})$$

- principe de la minimisation du risque empirique



Leçon générale

- méthode de construction de la régression linéaire
 - choix d'une classe de modèles

$$\mathcal{F} = \{f : \mathbb{R} \rightarrow \mathbb{R} \mid \exists (a, b) \in \mathbb{R}^2, f(x) = ax + b\}$$

- choix dans la classe du modèle d'erreur empirique minimale sur l'ensemble d'apprentissage

$$f^* = \arg \min_{f \in \mathcal{F}} \hat{L}(f; \{(x_1, y_1), \dots, (x_N, y_N)\})$$

- principe de la **minimisation du risque empirique** :

- méthode centrale de l'apprentissage automatique
- lien très fort avec l'optimisation continue
- problème associé : $\hat{L}(f; \mathcal{D})$ est optimiste



Régression linéaire multiple

- extension à plusieurs variables explicatives :

- $\mathcal{X} = \mathbb{R}^p$ et $\mathcal{Y} = \mathbb{R}$
- modèles considérés

$$\mathcal{F} = \left\{ f : \mathbb{R}^p \rightarrow \mathbb{R} \mid f(x) = \beta_0 + \sum_{i=1}^p \beta_i x_i \right\}$$

- vision apprentissage : minimisation du risque empirique



Régression linéaire multiple

- extension à plusieurs variables explicatives :

- $\mathcal{X} = \mathbb{R}^p$ et $\mathcal{Y} = \mathbb{R}$
- modèles considérés

$$\mathcal{F} = \left\{ f : \mathbb{R}^p \rightarrow \mathbb{R} \mid f(x) = \beta_0 + \sum_{i=1}^p \beta_i x_i \right\}$$

- vision apprentissage : minimisation du risque empirique

- vision statistique classique :

- les X_i sont des variables aléatoires à valeurs dans \mathbb{R}
- ε est un bruit (aléatoire)
- Y est distribuée selon

$$Y = \beta_0 + \sum_{i=1}^p \beta_i X_i + \varepsilon$$



■ notations simplificatrices :

- on ajoute une « variable » x_0 toujours égale à 1
- on note $Y = (y_1, \dots, y_N)^T$ et X la matrice dont les colonnes sont les variables :

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ \vdots & & \ddots & & \vdots \\ 1 & x_{N1} & x_{N2} & \dots & x_{Np} \end{pmatrix}$$

- on cherche alors $\beta = (\beta_0, \dots, \beta_p)^T$ tel que $Y \simeq X\beta$

■ minimisation de l'erreur quadratique

$$\beta^* = \arg \min_{\beta} \|Y - X\beta\|^2 = \arg \min_{\beta} \sum_{i=1}^N (Y_i - (X\beta)_i)^2$$



- $\nabla_{\beta} \|Y - X\beta\|^2 = 0$ conduit aux équations normales

$$(X^T X)\beta^* = X^T Y$$

- résolution (coût et stabilité croissants) :

- approche directe en $\mathcal{O}(p^3 + Np^2)$

$$\beta^* = (X^T X)^{-1} X^T Y$$

- décomposition QR en $\mathcal{O}(Np^2)$

$X = QR$ avec Q orthogonale et R triangulaire supérieure

- décomposition en valeurs singulières en $\mathcal{O}(Np^2)$

$X = UDV^T$ avec D diagonale, et U et V orthogonales)

- en général, on utilise la décomposition QR



Maximum de vraisemblance

- le modèle probabiliste s'écrit $Y = X\beta + \varepsilon$
- hypothèses supplémentaires :
 - observations statistiquement indépendantes
 - bruit ε gaussien $\mathcal{N}(0, \sigma^2)$
- vraisemblance de $(x_i, y_i)_{1 \leq i \leq N}$

$$\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N \prod_{i=1}^N \exp\left(-\frac{1}{2\sigma^2}(y_i - x_i\beta)^2\right)$$

- maximiser la log vraisemblance revient donc à minimiser

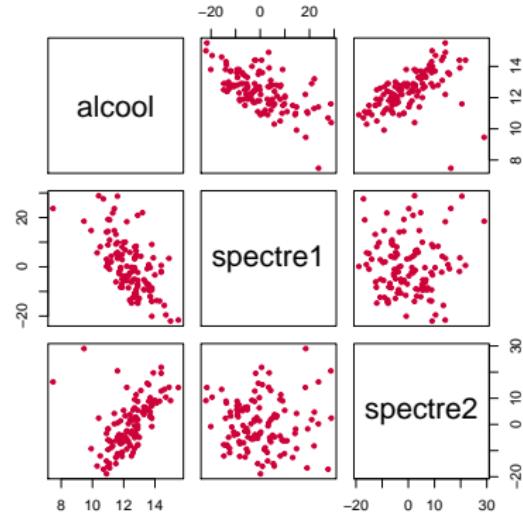
$$\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - x_i\beta)^2$$

- moindres carrés = maximum de vraisemblance
 - pour des observations indépendantes
 - et pour un bruit gaussien
- homoscédasticité :
 - la variance du bruit ne dépend ni de x , ni de y
 - hypothèse assez forte
- modèle probabiliste :
 - donne plus d'information : distribution des poids, significativité, etc.
 - plus souple que les moindres carrés (cf aussi le cas de la classification)
 - par exemple : bruit hétéroscédastique (variance non uniforme)
 - mais plus complexe à mettre en œuvre



Exemple

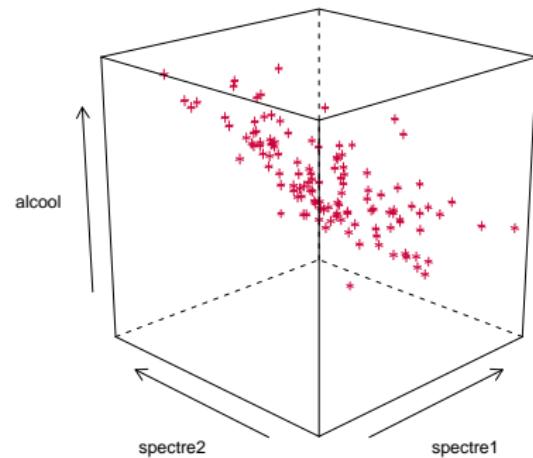
- taux d'alcool dans le vin en fonction de deux variables spectrales





Exemple

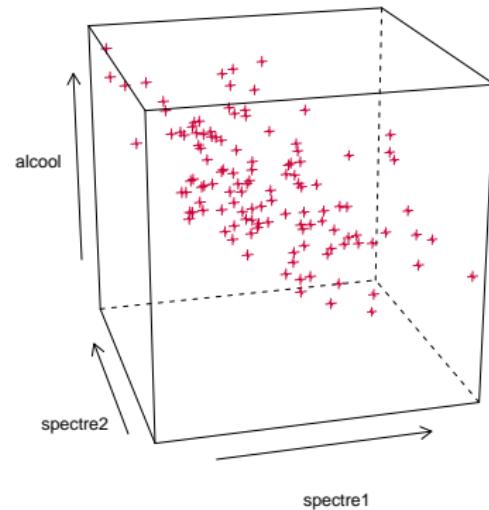
- taux d'alcool dans le vin en fonction de deux variables spectrales





Exemple

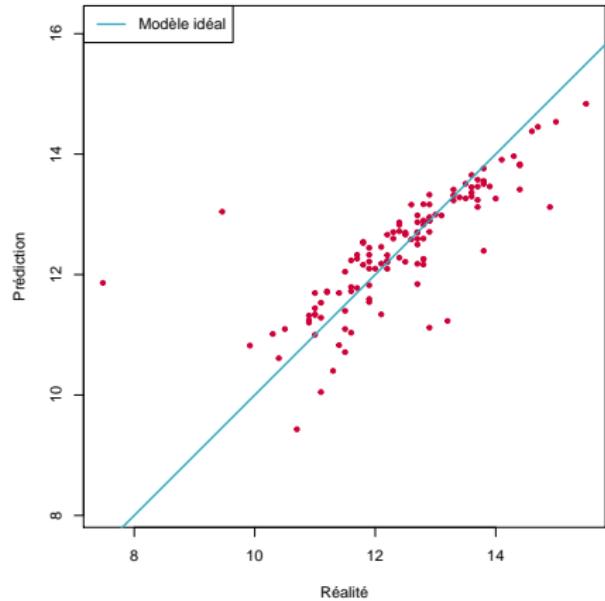
- taux d'alcool dans le vin en fonction de deux variables spectrales





Exemple

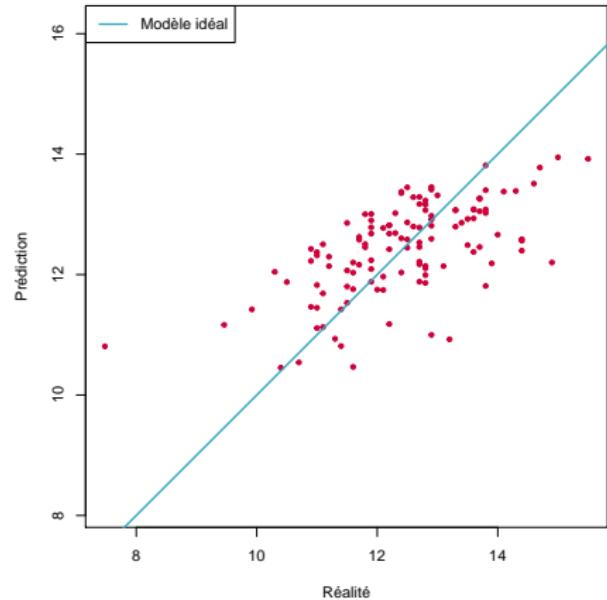
- taux d'alcool dans le vin en fonction de deux variables spectrales
- prédition vs réalité
- visualisation classique quand $p \geq 2$
- $\hat{L}(g) = 0.54$





Exemple

- taux d'alcool dans le vin en fonction de deux variables spectrales
- prédition vs réalité
- visualisation classique quand $p \geq 2$
- une variable : $\hat{L}(g) = 0.92$





Prédictions simultanées

- régression avec $y \in \mathbb{R}^q$:

- Y est la matrice des variables à prédire (une ligne par observation)

$$Y = \begin{pmatrix} y_{11} & \dots & y_{1q} \\ \vdots & & \vdots \\ y_{N1} & \dots & y_{Nq} \end{pmatrix}$$

- β est maintenant une matrice $(p+1) \times q$

- minimisation de l'erreur quadratique (erreur gaussienne hétéroscléastique)

$$\beta^* = \arg \min_{\beta} \|Y - X\beta\|^2 = \arg \min_{\beta} \sum_{j=1}^q \sum_{i=1}^N (Y_{ij} - (X\beta)_{ij})^2$$

- revient à réaliser q régressions linéaires multiples

■ fonction `lm` du package `stats` :

- modèle linéaire par moindres carrés (méthode QR)
- interprétation statistique classique (significativité, etc.)
- support des formules (au sens R) :
 - données sous forme d'une `data.frame`
 - formules du type `y~a+b-1` pour préciser les variables explicatives (ici `a` et `b`) et supprimer le terme constant `-1`
- fonction `predict` pour les prédictions

■ nombreuses extensions :

- modèles linéaires généralisés
- séries temporelles
- etc.

- deux régimes « extrêmes » :

- si N est grand devant p :
 - beaucoup plus d'observations que de variables
 - le modèle linéaire n'est généralement **pas assez complexe**
 - si N est petit devant p :
 - beaucoup plus de variables que d'observations
 - le modèle linéaire est généralement **trop complexe**

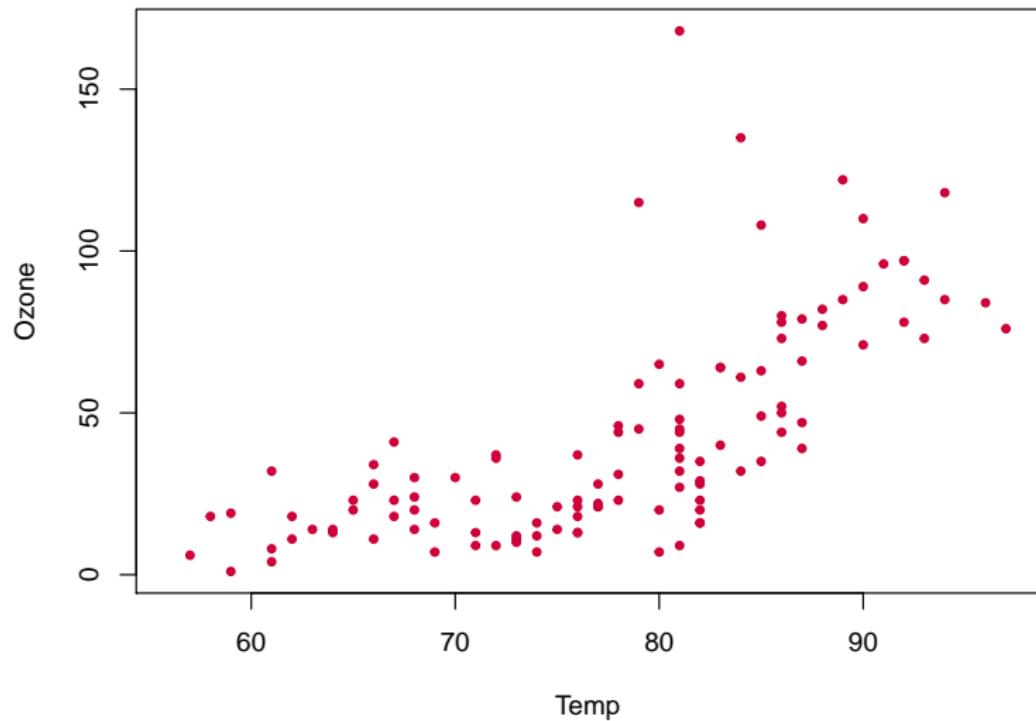
- plage d'utilisation directe : quand N est de l'ordre de αp

- trois grandes questions :

1. comment augmenter la complexité ?
2. comment réduire la complexité ?
3. comment choisir la complexité adaptée aux données ?

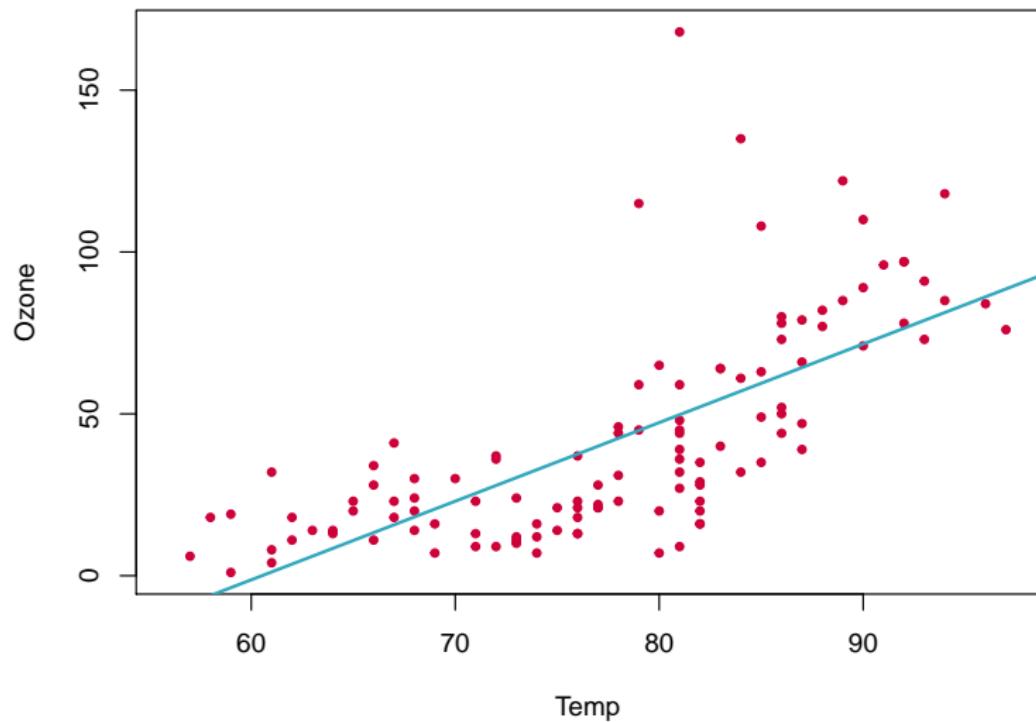


Trop simple





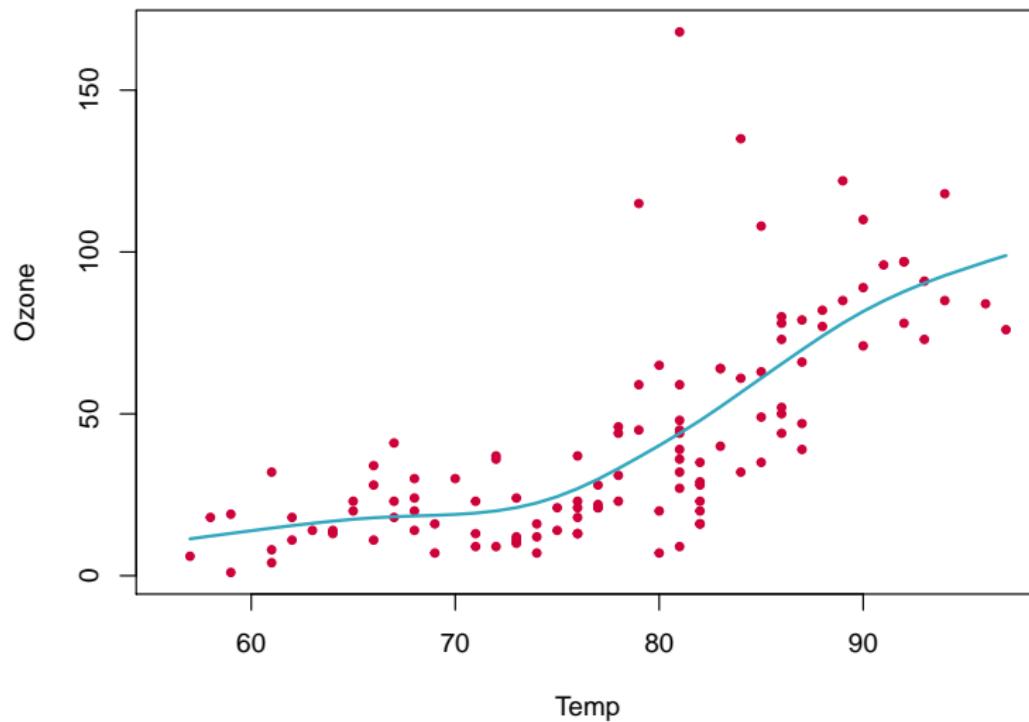
Trop simple



régression linéaire



Trop simple

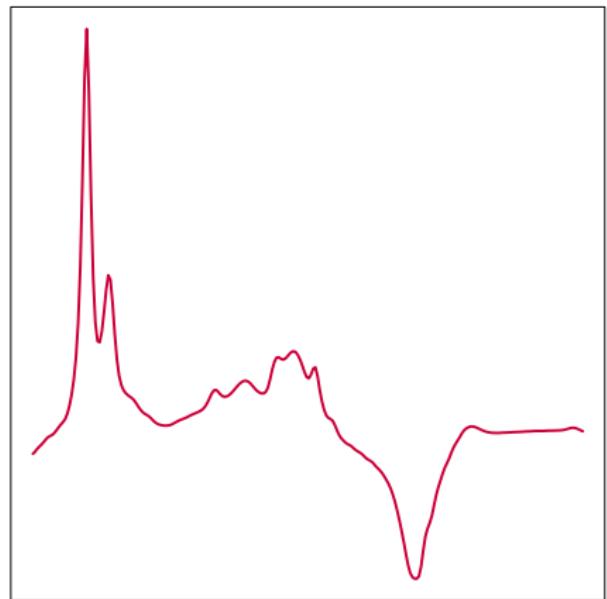


méthode non linéaire



Trop complexe

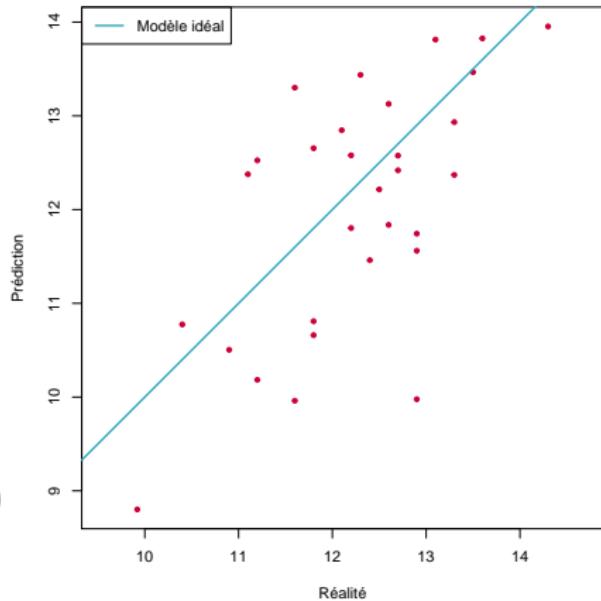
- taux d'alcool dans le vin en fonction du spectre complet
- 256 variables explicatives (!)
- 91 observations





Trop complexe

- taux d'alcool dans le vin en fonction du spectre complet
- 256 variables explicatives (!)
- 91 observations
- prédition vs réalité sur 30 nouvelles observations

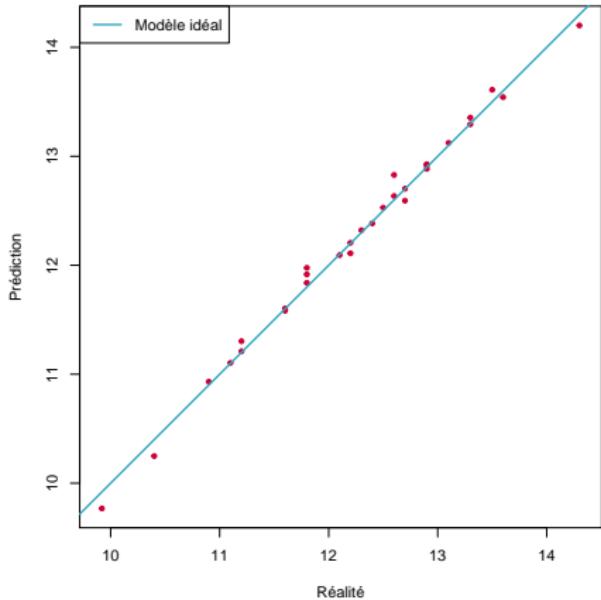


Modèle linéaire



Trop complexe

- taux d'alcool dans le vin en fonction du spectre complet
- 256 variables explicatives (!)
- 91 observations
- prédition vs réalité sur 30 nouvelles observations



Modèle linéaire « réduit »

Définition informelle

la complexité d'une classe de modèles se mesure par la qualité prédictive qu'elle peut atteindre sur un ensemble d'apprentissage

■ pas assez complexe :

- aucun modèle de la classe ne prédit bien y à partir de x
- par exemple $y = x^2$ et régression linéaire
- Ozone et température

■ trop complexe :

- certains modèles de la classe ne font aucune erreur sur l'ensemble d'apprentissage
- en général, $Y = X\beta$ a une infinité de solutions quand N est petit devant p
- Alcool et spectre



Sélection de modèle

- approche de base :
 - choix d'une classe de modèles
 - puis choix du modèle d'erreur empirique minimale
- ne fonctionne pas quand la classe est trop complexe :
 - exemple : pour la prévision du degré alcoolique en fonction du spectre, l'erreur empirique est nulle
 - phénomène de **sur-apprentissage** (*overfitting*)
- approche hiérarchique :
 - choix de plusieurs classes de modèles, de complexités différentes
 - minimisation de l'erreur empirique dans chaque classe
 - puis choix du modèle parmi les candidats



Sélection de modèle

- approche de base :
 - choix d'une classe de modèles
 - puis choix du modèle d'erreur empirique minimale
- ne fonctionne pas quand la classe est trop complexe :
 - exemple : pour la prévision du degré alcoolique en fonction du spectre, l'erreur empirique est nulle
 - phénomène de **sur-apprentissage** (*overfitting*)
- approche hiérarchique :
 - choix de plusieurs classes de modèles, de complexités différentes
 - minimisation de l'erreur empirique dans chaque classe
 - puis choix du modèle parmi les candidats
 - **comment ?**



Ensemble de validation

- l'erreur empirique sur l'ensemble d'apprentissage est un mauvais choix car :
 - la loi des grands nombres ne s'applique pas (dépendance)
 - l'estimation des performances est optimiste
- solution élémentaire (solutions plus sophistiquées dans la suite du cours) :
 - utiliser d'autres données distribuées aussi selon P
 - $\mathcal{V}_M = (x_i, y_i)_{i=1}^M$: ensemble de validation indépendant
 - la loi des grands nombres s'applique : $\hat{L}(g; \mathcal{V}_M) \simeq L(g)$
 - **point crucial : le modèle doit être construit sans utiliser \mathcal{V}_M**
- méthode :
 - choix de plusieurs classes de modèles
 - minimisation de l'erreur empirique dans chaque classe
 - choix du modèle parmi les candidats par minimisation de l'erreur de validation



Réduire la complexité

■ source du problème :

- le système $Y = X\beta$ a une infinité de solution quand Y est dans l'image de X
- quand p est grand devant N , c'est très probable :
 - moins d'équations (les N observations)
 - que d'inconnues (les $p + 1$ poids β_j)



Réduire la complexité

■ source du problème :

- le système $Y = X\beta$ a une infinité de solution quand Y est dans l'image de X
- quand p est grand devant N , c'est très probable :
 - moins d'équations (les N observations)
 - que d'inconnues (les $p + 1$ poids β_j)

■ attaquer la source du problème :

- réduire le nombre de variables
- classes de modèles, pour tout $S \subset \{1, \dots, p\}$:

$$\mathcal{F}_S = \left\{ f : \mathbb{R}^p \rightarrow \mathbb{R} \mid f(x) = \beta_0 + \sum_{i \in S} \beta_i x_i \right\}$$

- choisir un modèle revient à choisir les variables utilisées



Sélection de variables

■ recherche exhaustive :

- faisable quand p est petit : $2^p - 1$ configurations
- accélération par *branch and bound* : faisable jusqu'à $p \simeq 30$

■ heuristiques d'exploration :

- croissante (*forward*) :

- on ajoute des variables progressivement
- $S_1 = \{j_1\}$ est la variable qui donne le meilleur modèle linéaire à une variable
- $S_2 = \{j_1, j_2\}$ est obtenu en trouvant la variable j_2 qui donne avec j_1 (fixée) le meilleur modèle linéaire à deux variables
- etc.

- décroissante (*backward*) :

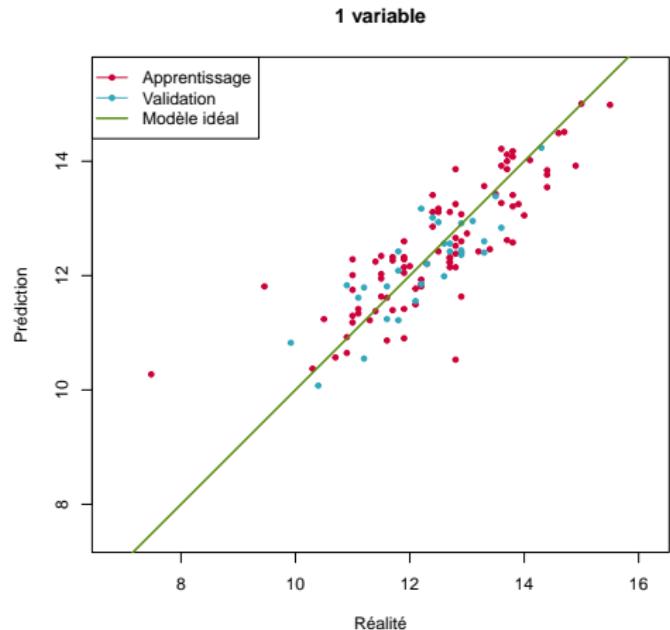
- même principe mais en enlevant des variables
- on commence donc par considérer le modèle complet

- mélange des deux...



Exemple

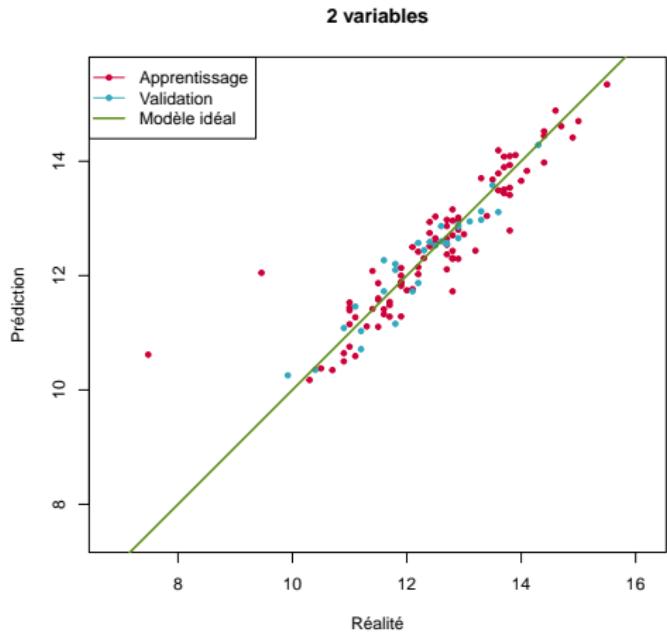
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

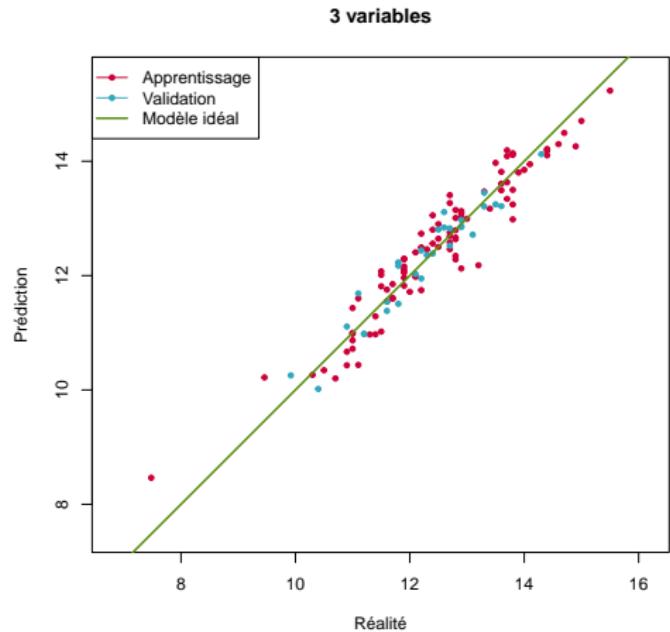
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

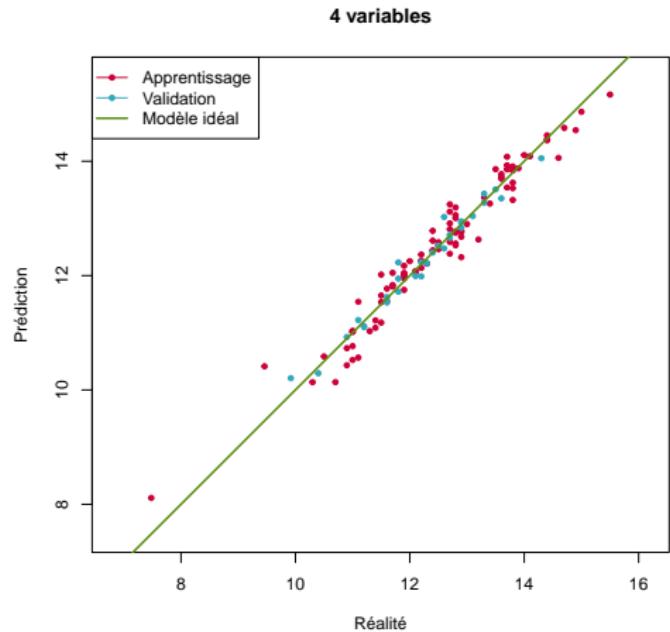
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

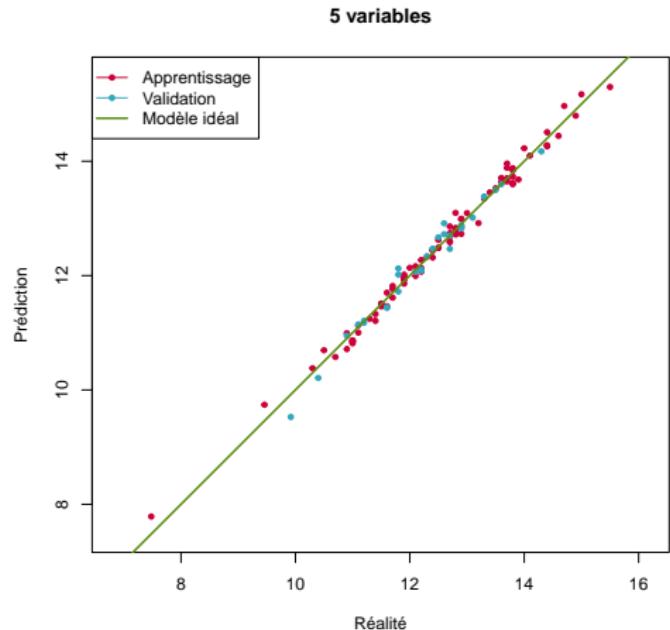
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

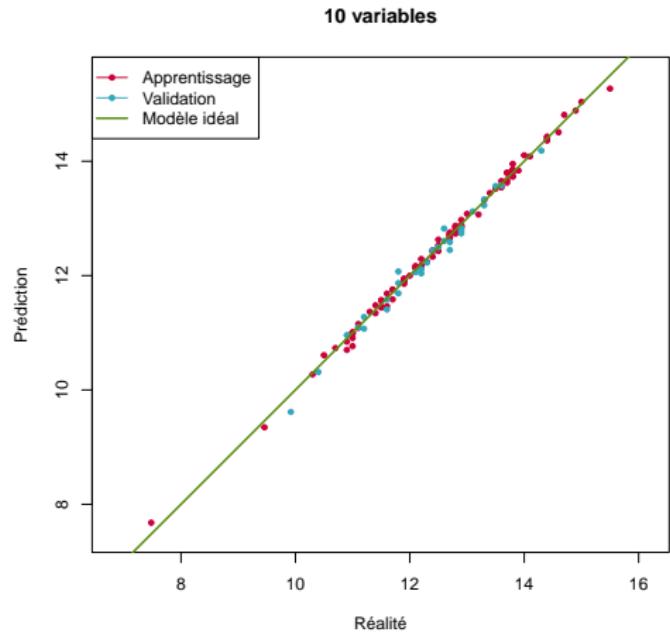
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

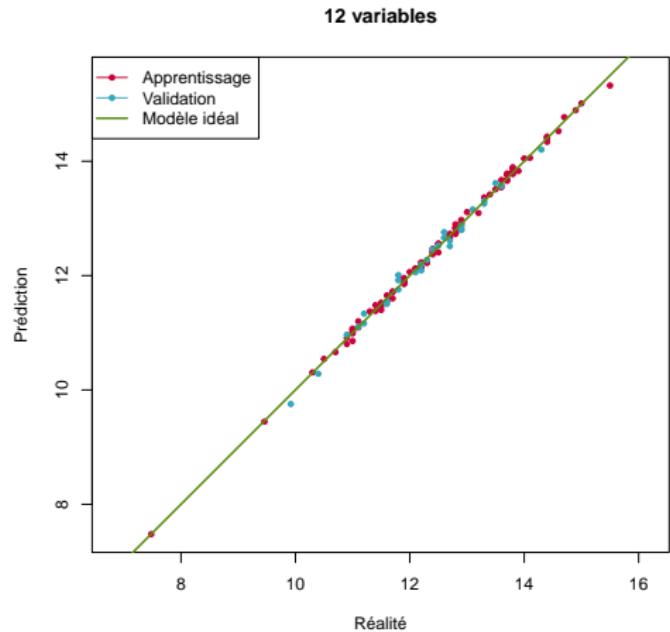
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

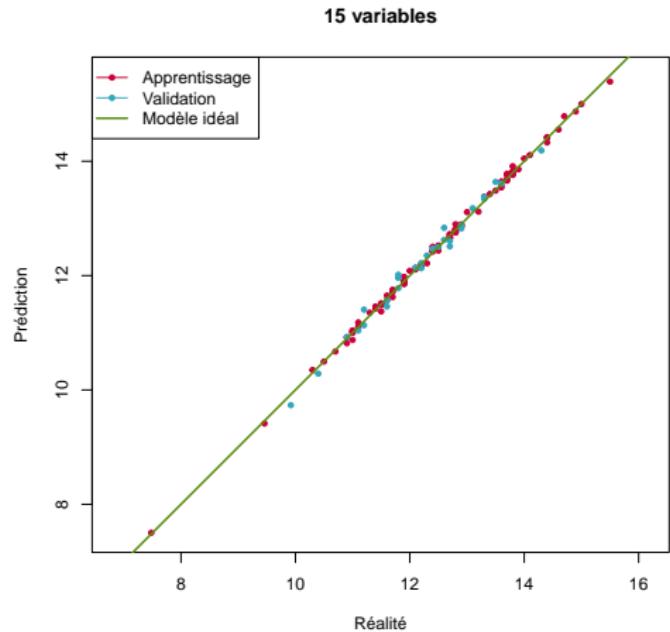
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

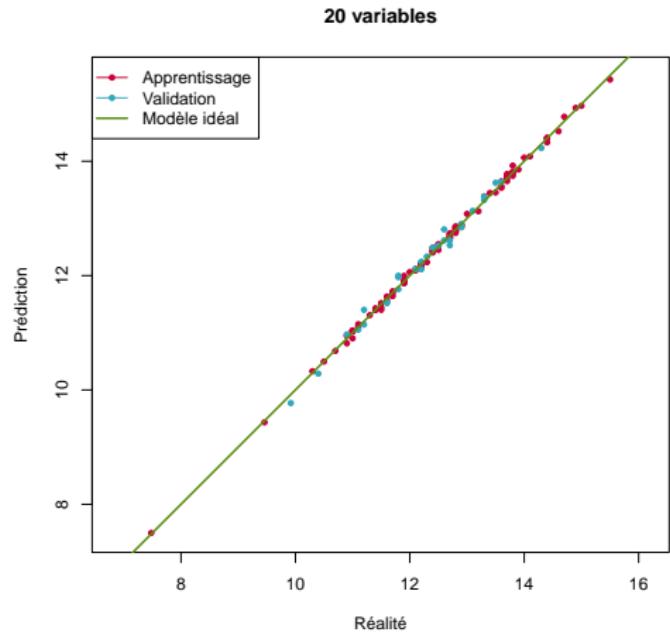
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

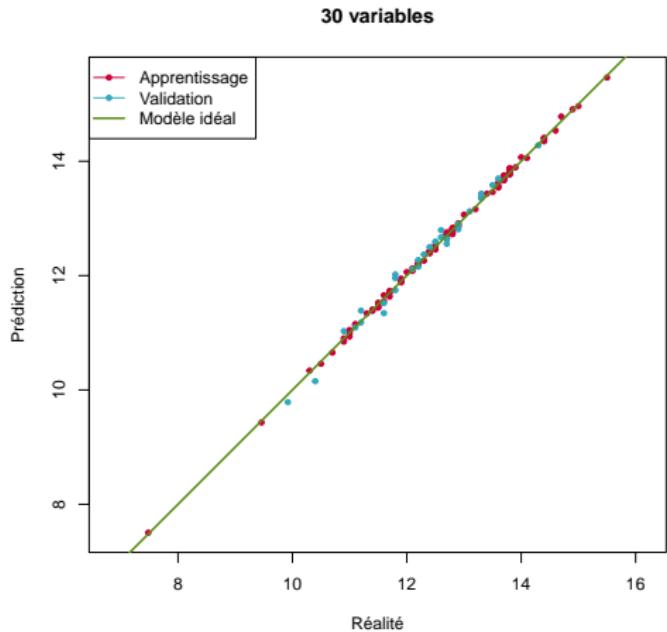
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

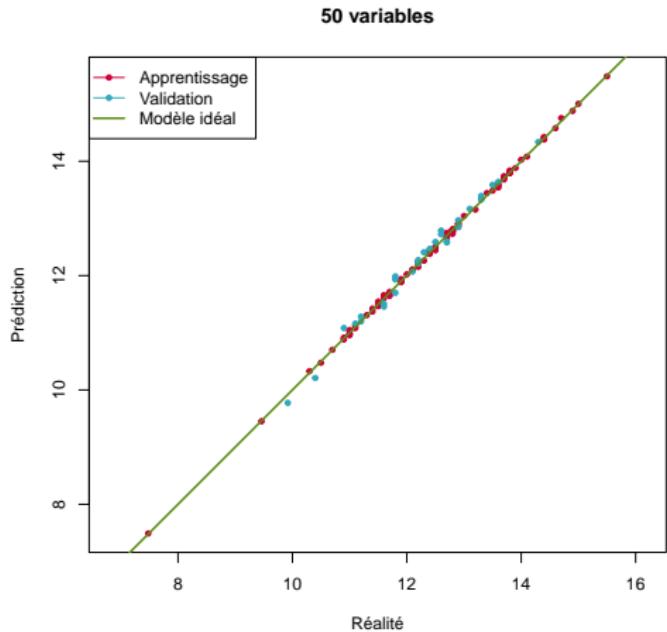
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

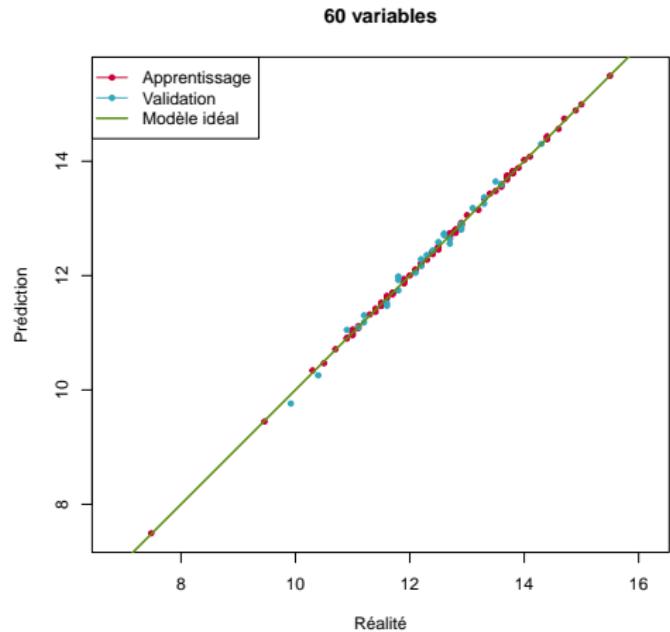
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

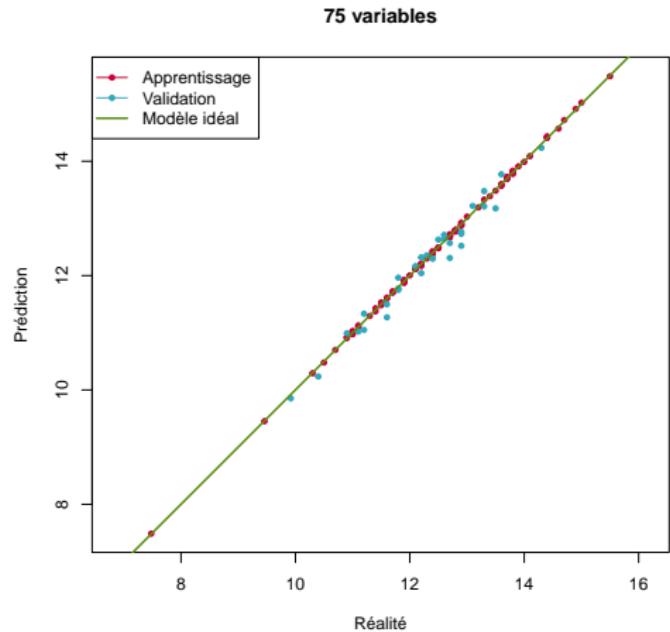
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Exemple

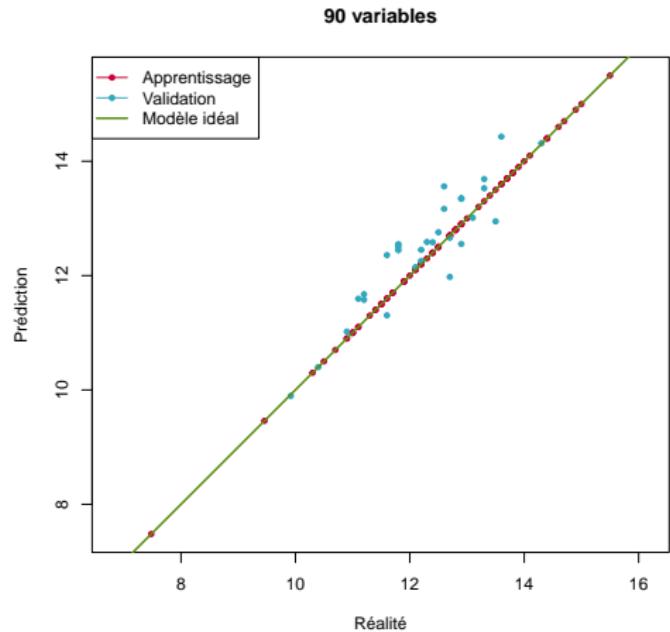
- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





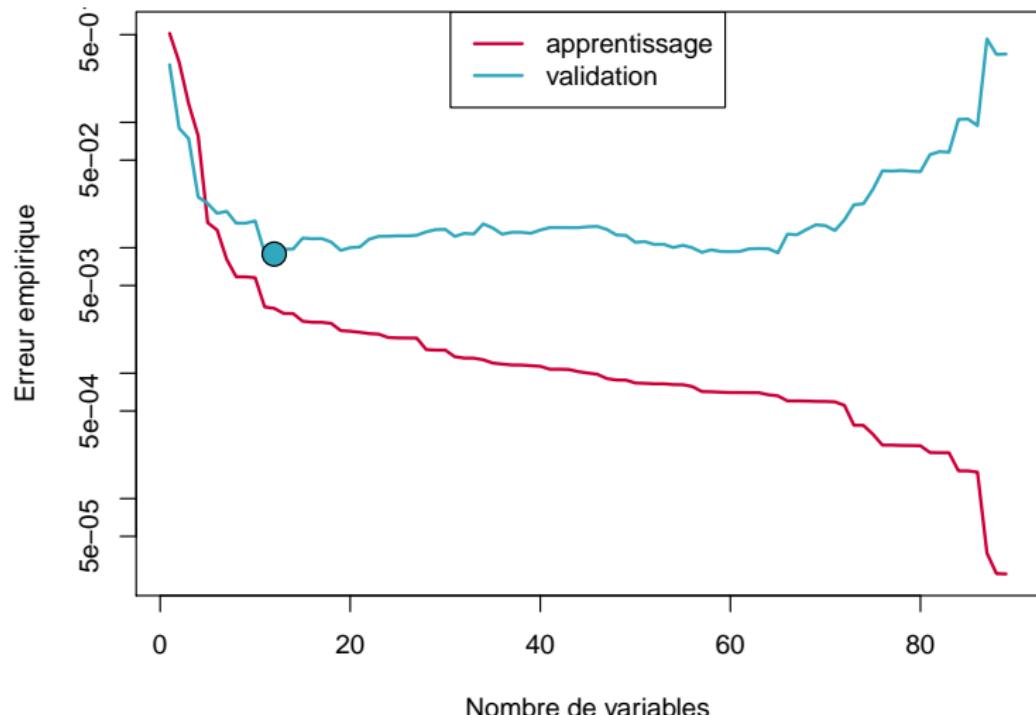
Exemple

- taux d'alcool dans le vin en fonction du spectre
- 91 observations \Rightarrow aucune erreur quand $p = 90$





Choix du modèle



12 variables

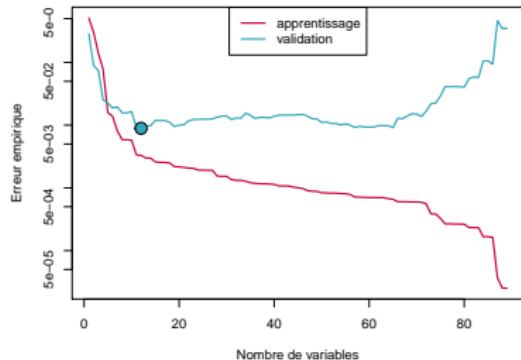


Choix du modèle

courbe classique du sur-apprentissage :

- décroissance constante de l'erreur empirique sur l'**ensemble d'apprentissage**
- décroissance puis croissance sur l'**ensemble de validation**

la bonne évaluation des performances est celle fournie par l'**ensemble de validation**

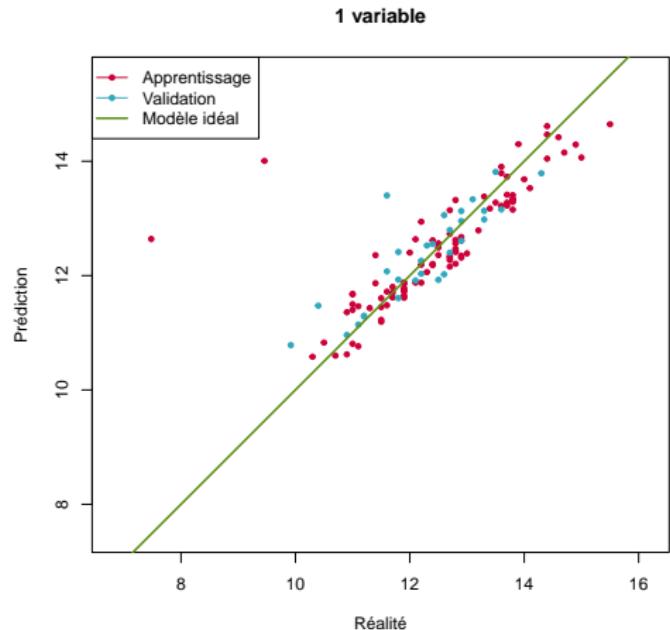


- les variables d'origine ne sont pas nécessairement les plus adaptées
- technique de réduction de la complexité (régression sur composantes principales) :
 - réaliser une ACP des données
 - construire des modèles linéaires sur 1, 2, ..., p composantes principales
 - choisir le meilleur modèle, c'est-à-dire le bon nombre de composantes
- extension :
 - choisir des composantes orthogonales et corrélées avec la variable à prédire Y
 - c'est la régression PLS (*Partial Least Squares*)
- les composantes sont ordonnées : sélection *forward* par nature



Exemple

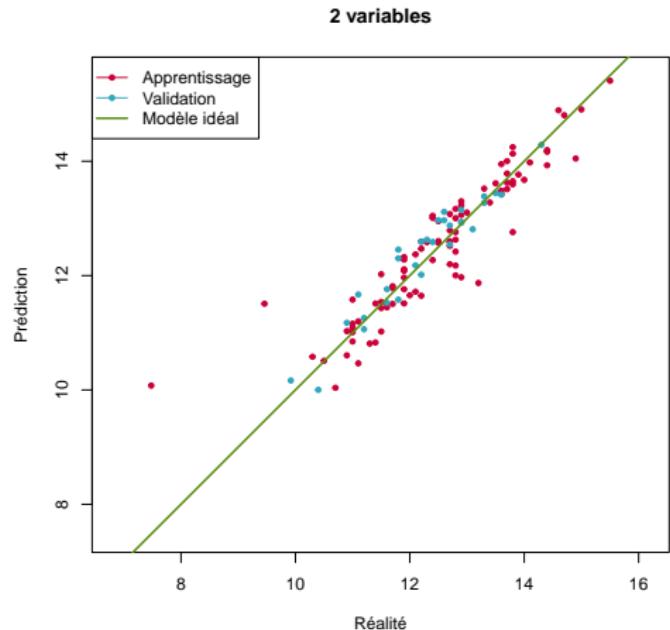
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

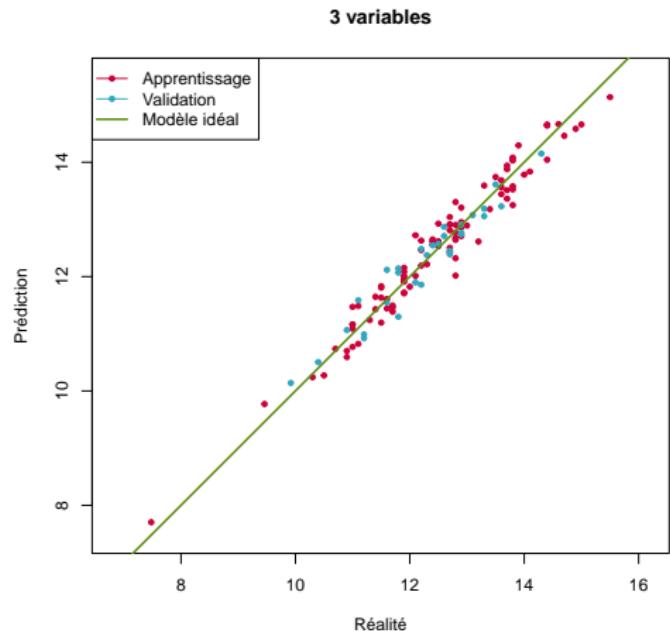
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

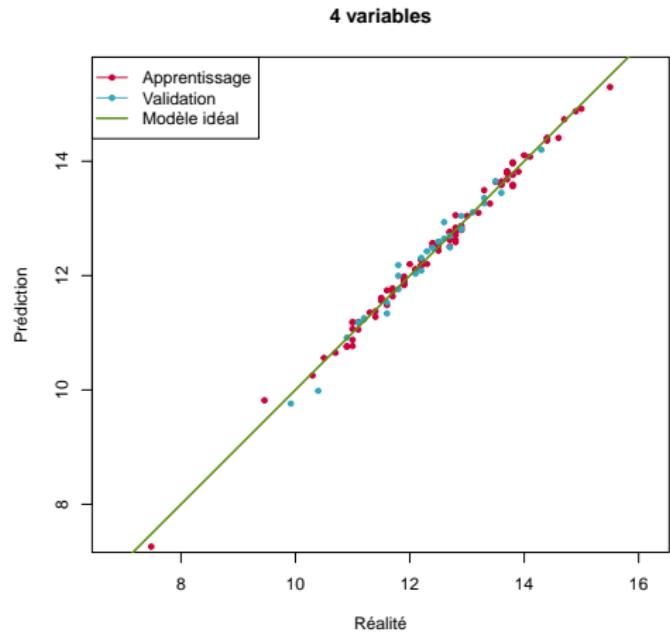
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

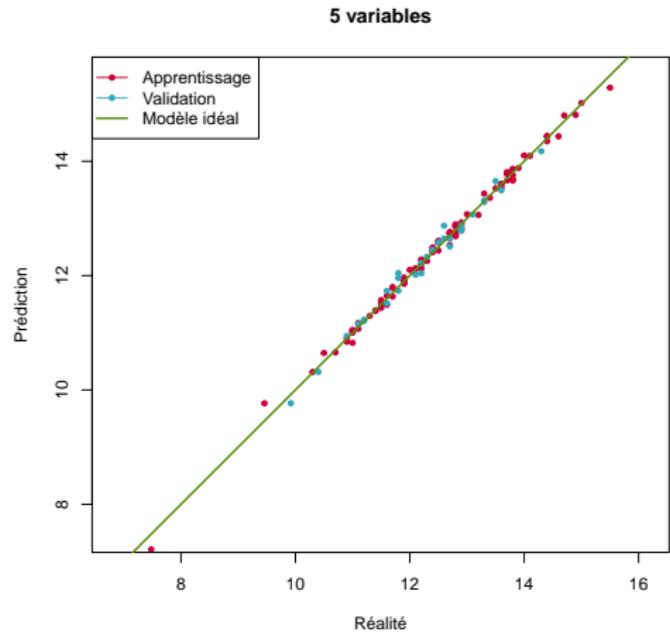
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

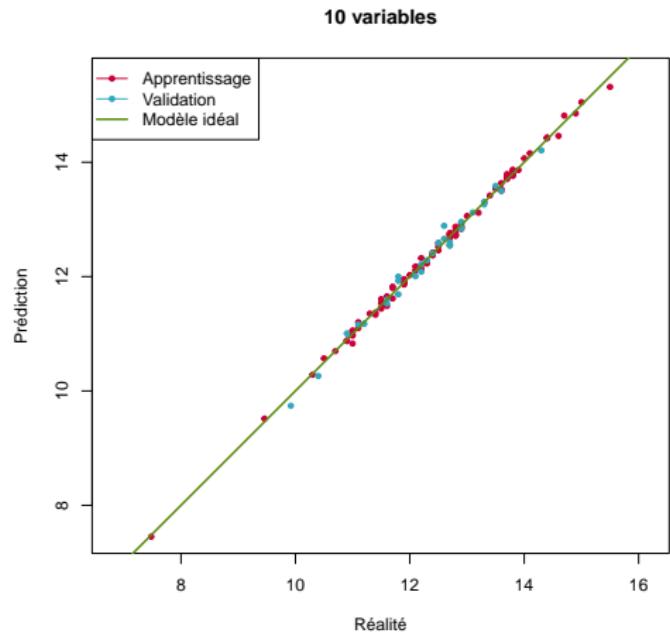
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

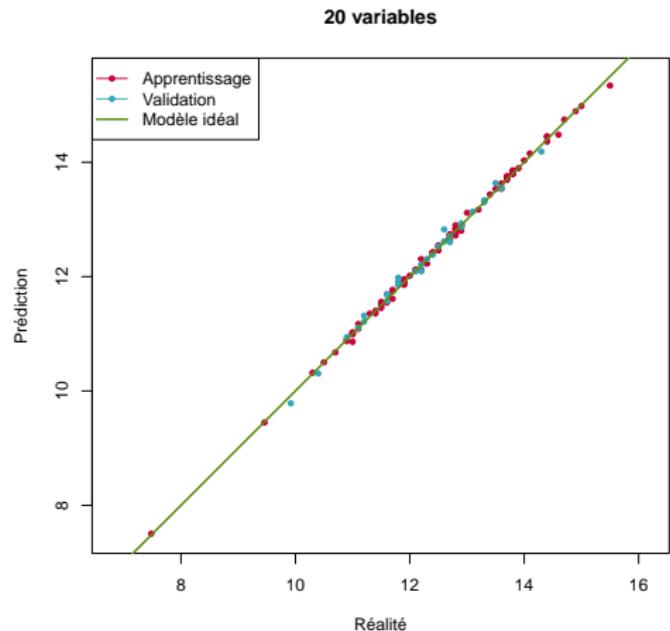
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

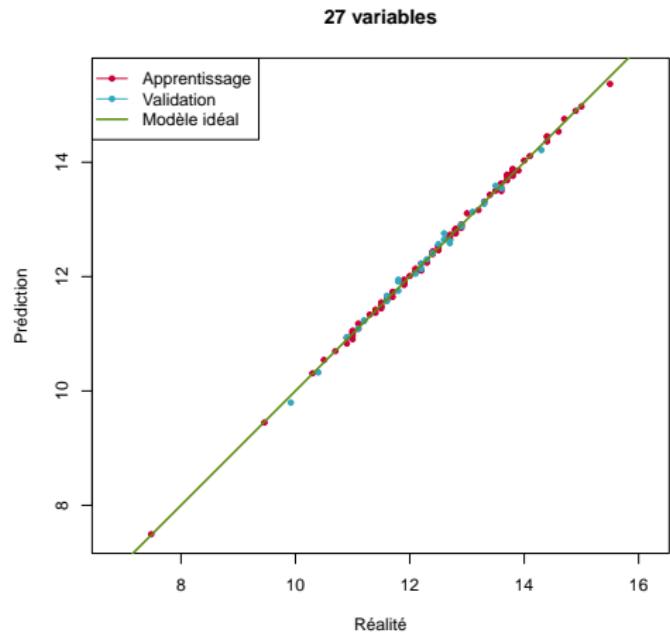
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

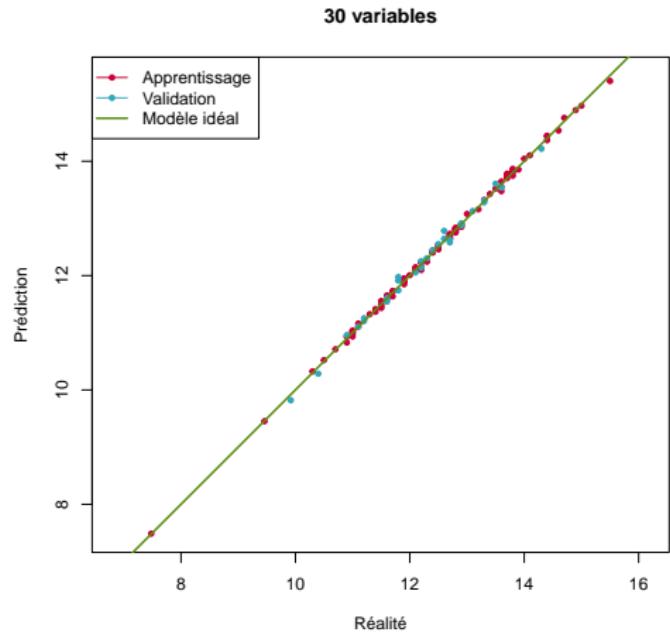
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

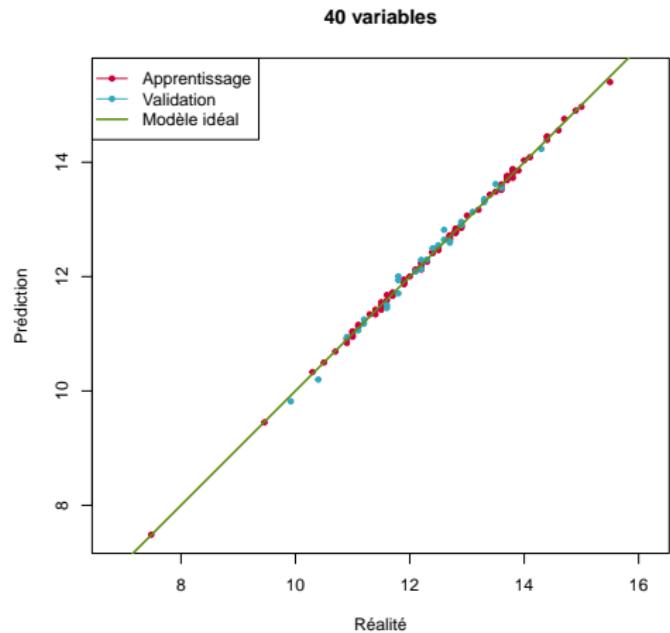
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

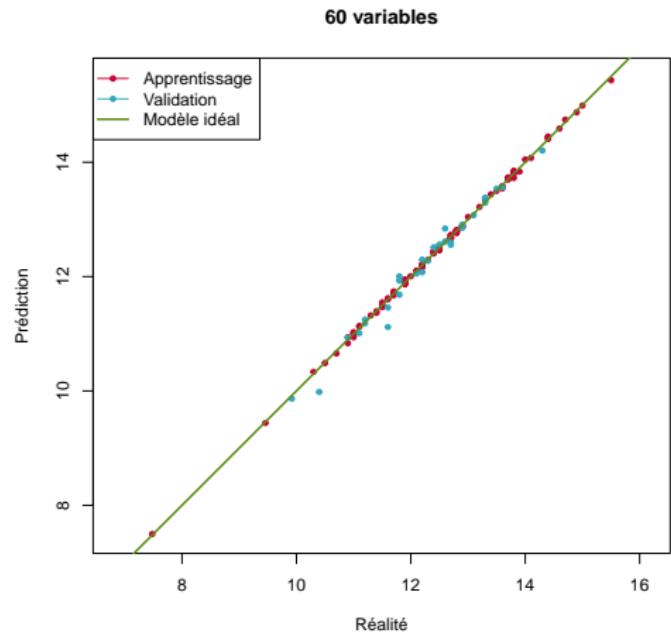
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Exemple

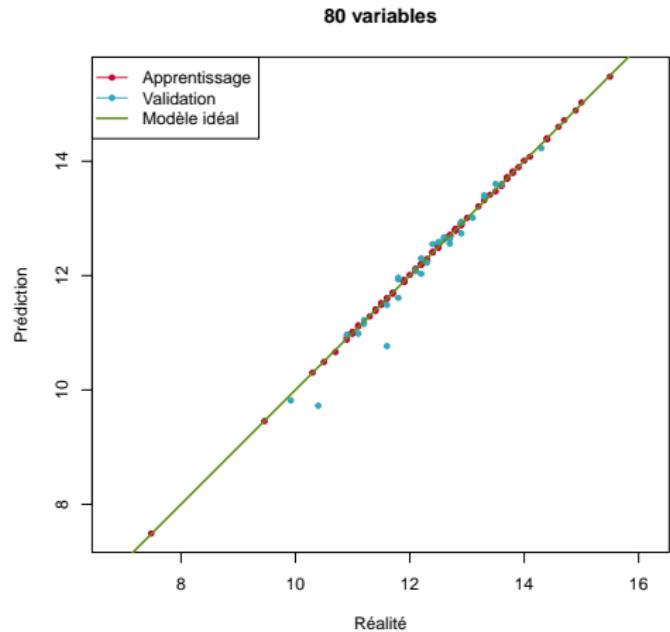
- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





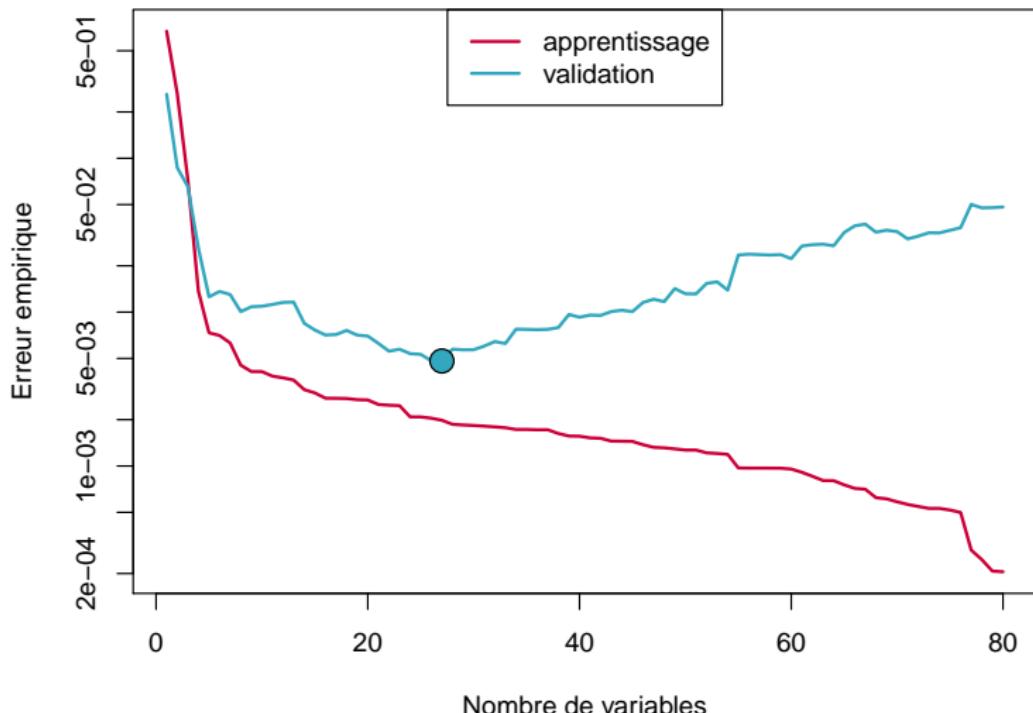
Exemple

- taux d'alcool dans le vin en fonction du spectre
- variables induites par l'ACP





Choix du modèle



27 variables ACP



Régularisation

- limitation de la sélection : tout ou rien
- approche concurrente par contrainte de régularité :
 - idée sous-jacente :
 - un bon modèle s'appuie sur la « continuité » de la nature
 - si $u \simeq v$ alors $g(u) \simeq g(v)$
 - dans le cas linéaire :
 - $|\langle u, v \rangle| \leq \|u\| \|v\|$ (Cauchy-Schwarz)
 - donc $|\langle x_1, \beta \rangle - \langle x_2, \beta \rangle| = |\langle x_1 - x_2, \beta \rangle| \leq \|x_1 - x_2\| \|\beta\|$
 - donc $\|\beta\|$ donne une mesure de la régularité d'un modèle linéaire
- classe de modèles

$$\mathcal{F}_C = \left\{ f : \mathbb{R}^p \rightarrow \mathbb{R} \mid f(\mathbf{x}) = \beta_0 + \sum_{i=1}^N \beta_i x_i, \|\beta\| \leq C \right\}$$

■ résoudre

$$\beta^* = \arg \min_{\beta, \|\beta\| \leq C} \|Y - X\beta\|^2$$

peut sembler plus complexe qu'en l'absence de la contrainte

■ mais par dualité convexe, il existe un λ tel que β^* soit aussi solution de

$$\beta^* = \arg \min_{\beta} \left(\|Y - X\beta\|^2 + \lambda \|\beta\|^2 \right)$$

■ on parle de **régression ridge**



- la résolution est simple car le problème est toujours quadratique en β
- $\nabla_{\beta} (\|Y - X\beta\|^2 + \lambda\|\beta\|^2) = 0$ conduit aux équations normales modifiées

$$(X^T X + \lambda I)\beta^* = X^T Y$$

où I est la matrice identité (de taille $p + 1$)

- le conditionnement de $X^T X + \lambda I$ s'améliore avec λ



Mise en œuvre

- algorithme :

1. calculer la SVD de X , $X = UDV^T$
2. calculer $Z = U^T Y$
3. pour quelques valeurs de λ (par exemple des puissances de 10) :
 - 3.1 calculer la matrice diagonale $K(\lambda)$ définie par
$$K(\lambda)_{ii} = D_{ii}/(D_{ii}^2 + \lambda)$$
 - 3.2 calculer

$$\beta^* = VK(\lambda)Z$$

4. choisir le modèle optimal (sur un ensemble de validation)



■ algorithme :

1. calculer la SVD de X , $X = UDV^T$
2. calculer $Z = U^T Y$
3. pour quelques valeurs de λ (par exemple des puissances de 10) :
 - 3.1 calculer la matrice diagonale $K(\lambda)$ définie par
$$K(\lambda)_{ii} = D_{ii}/(D_{ii}^2 + \lambda)$$
 - 3.2 calculer

$$\beta^* = VK(\lambda)Z$$

4. choisir le modèle optimal (sur un ensemble de validation)

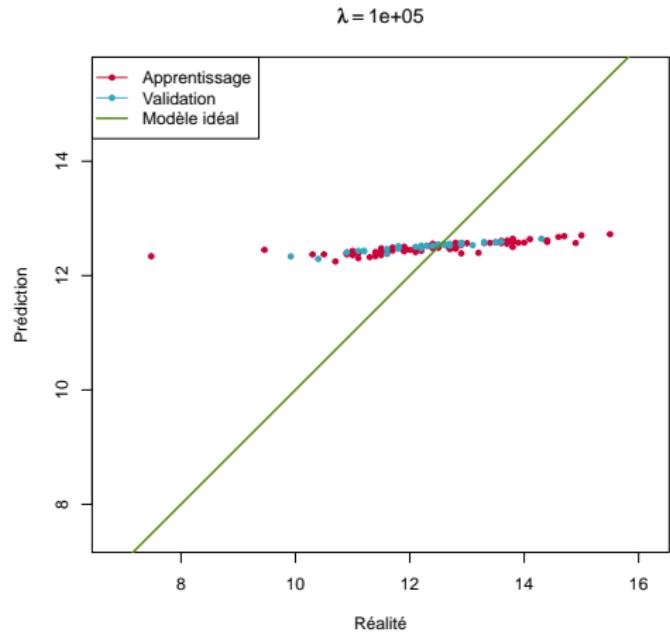
■ détails pratiques :

- régulariser β_0 n'est pas une bonne idée : une bonne valeur pour β_0 est la moyenne des y_i
- un changement d'échelle des x_i change la solution de la régression ridge pas celle de la régression classique : on centre et on réduit les données avant traitement



Exemple

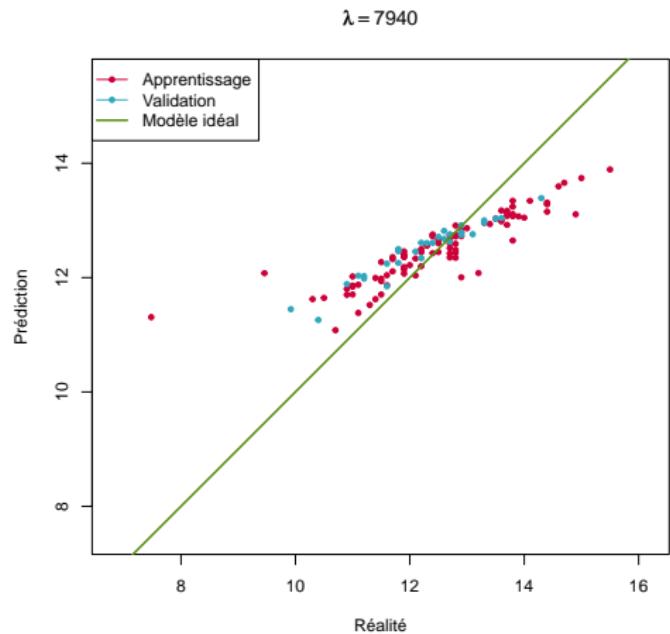
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

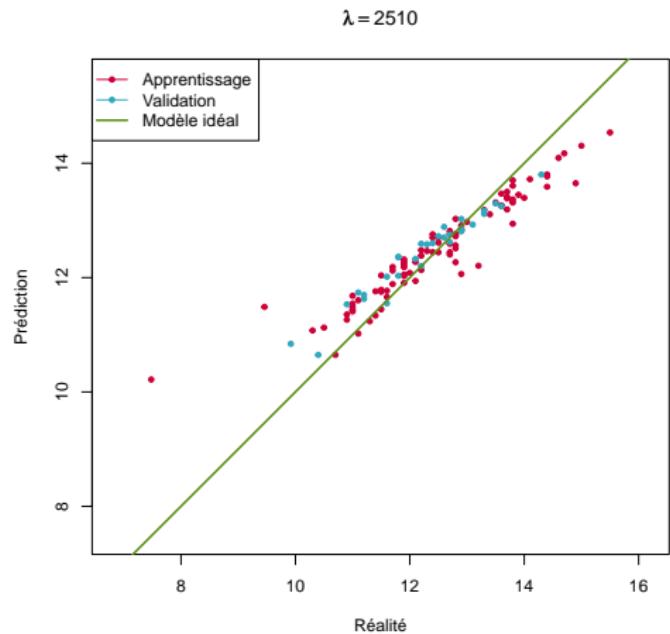
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

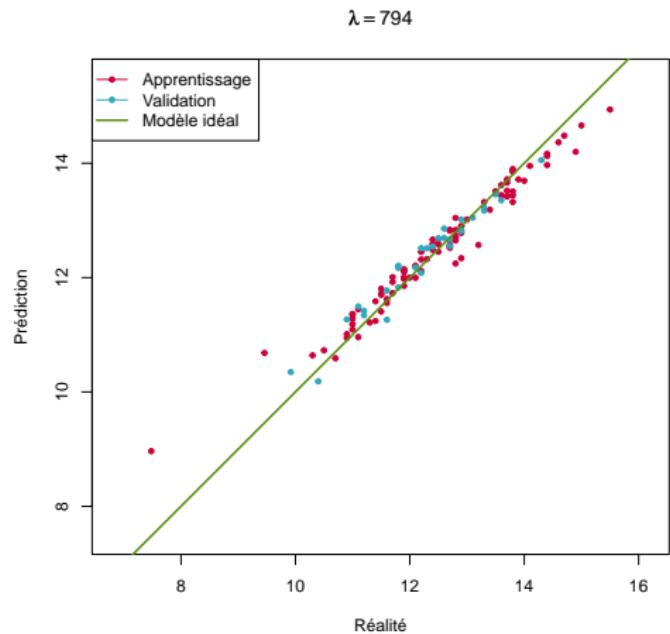
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

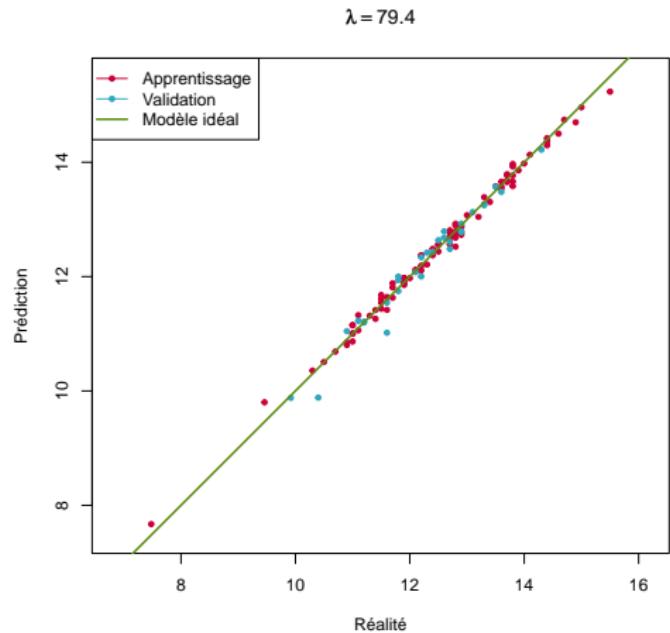
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

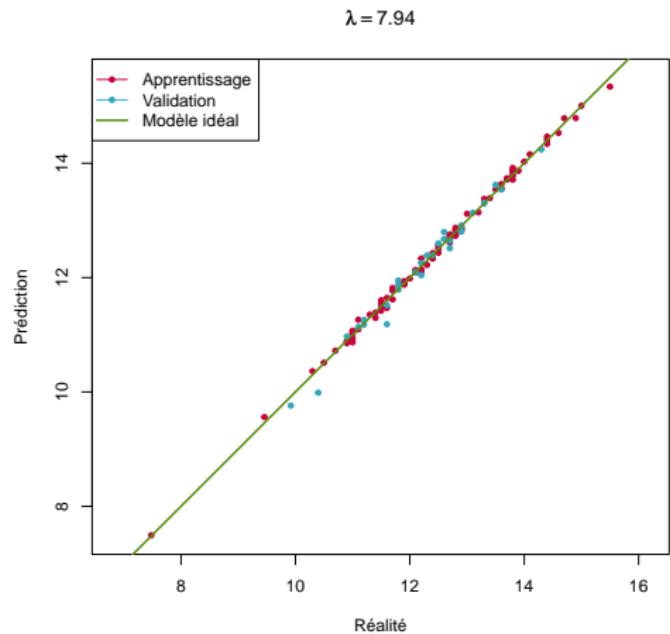
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

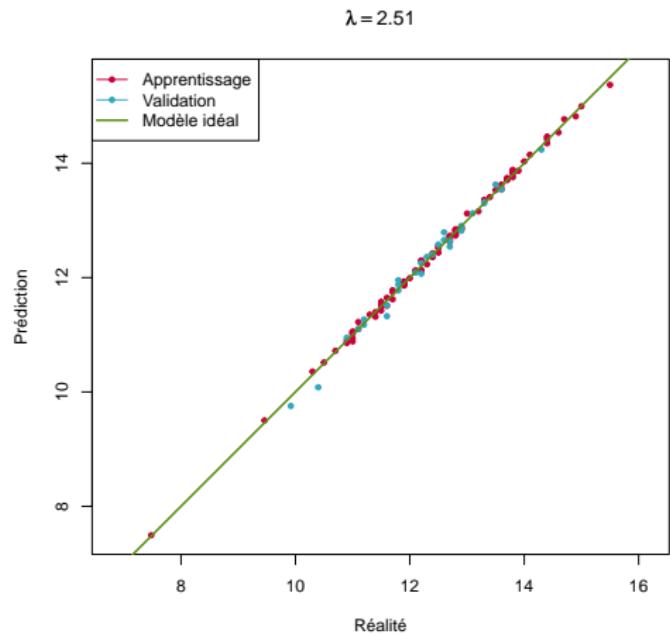
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

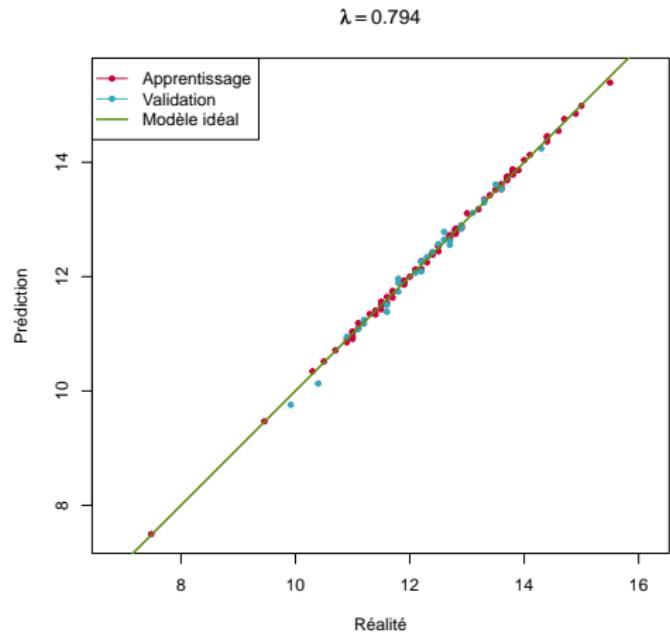
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

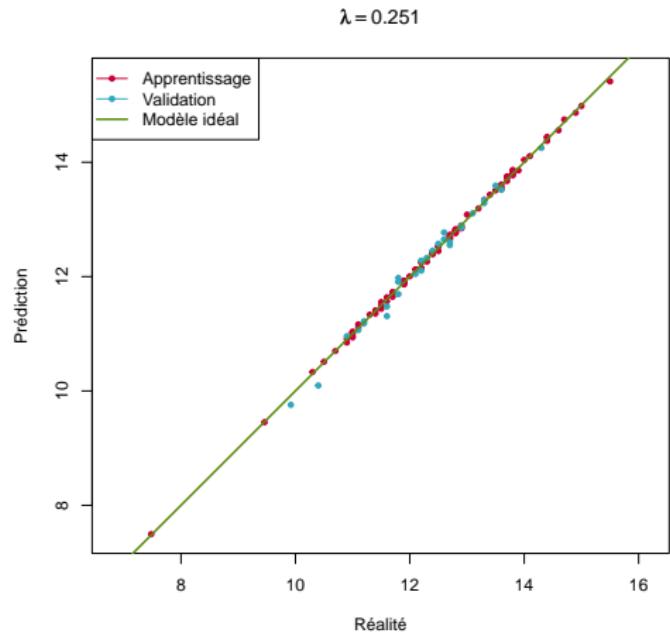
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

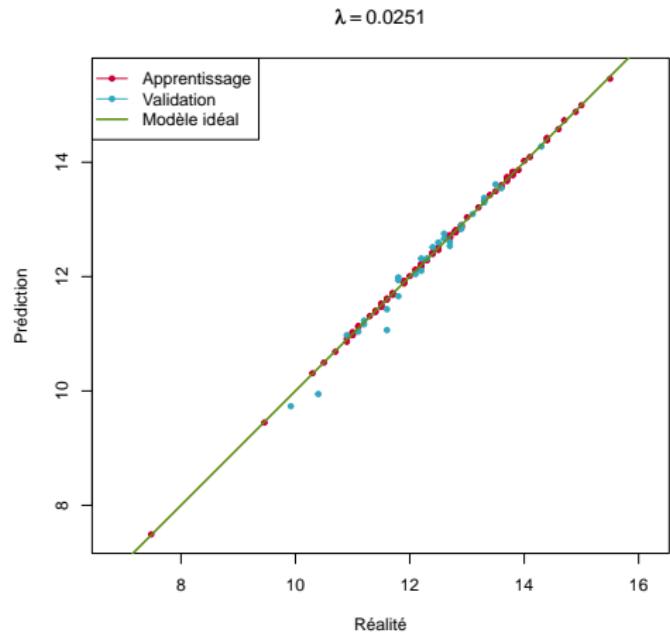
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

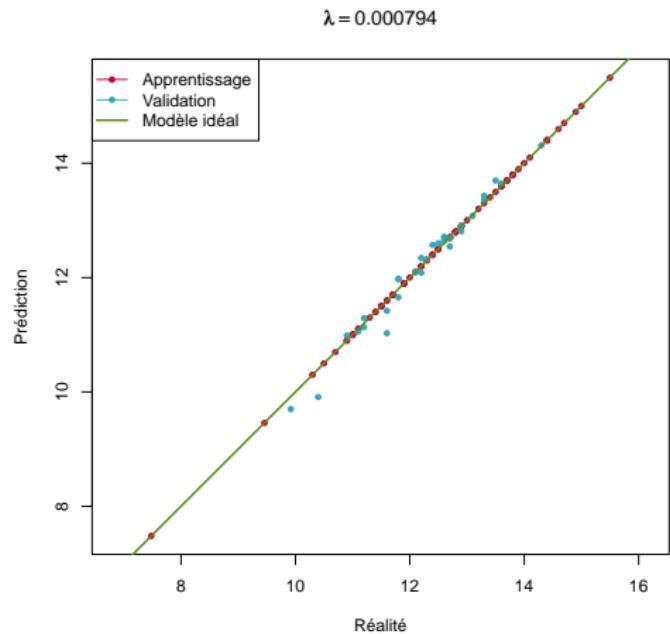
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





Exemple

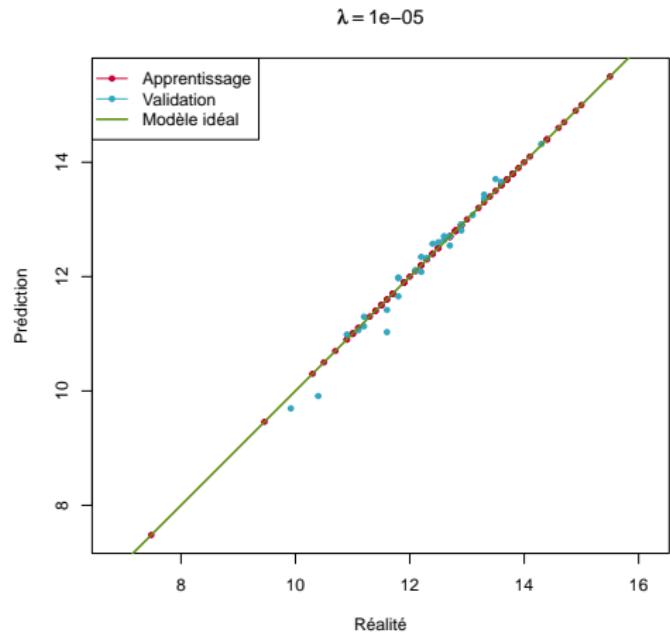
- taux d'alcool dans le vin en fonction du spectre
- régression ridge





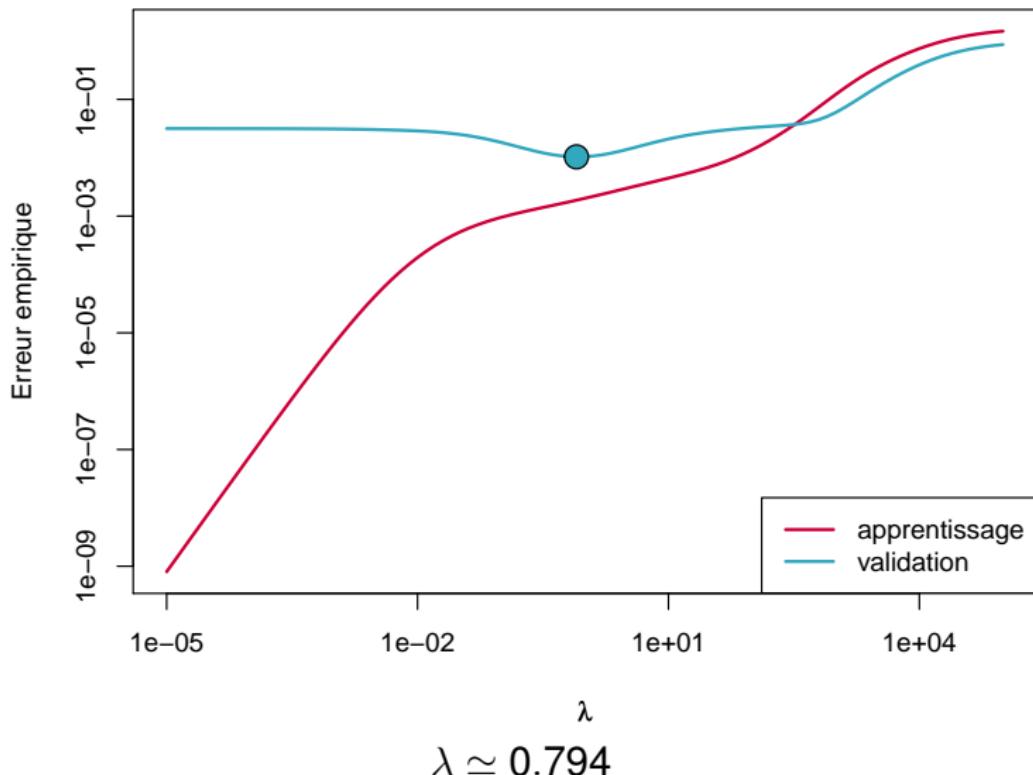
Exemple

- taux d'alcool dans le vin en fonction du spectre
- régression ridge



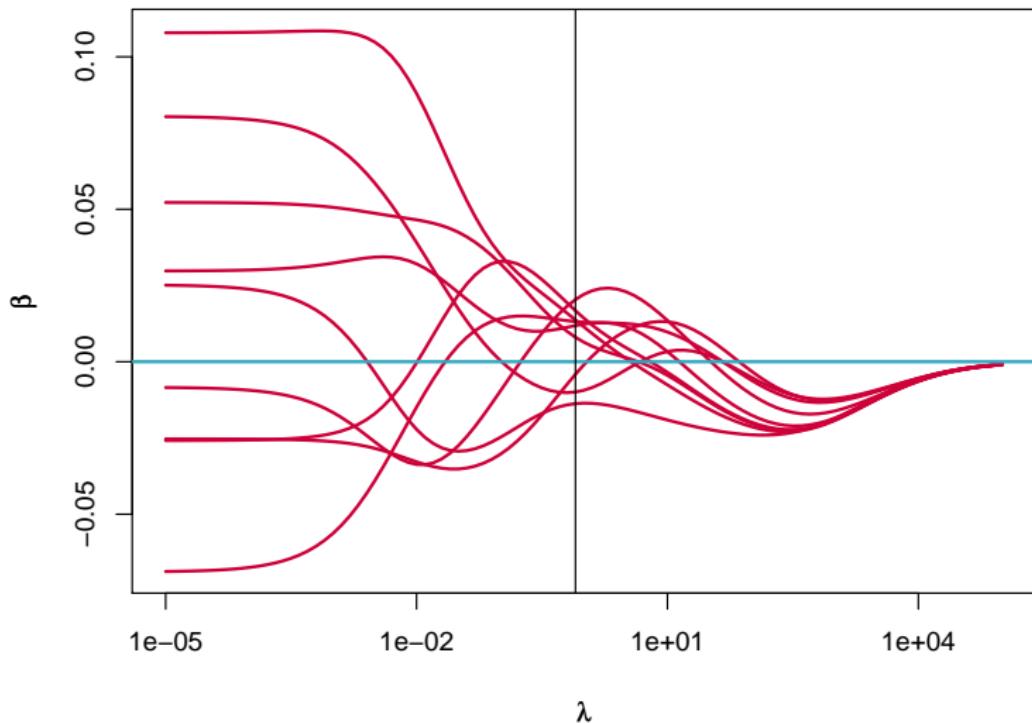


Choix du modèle





Évolution des coefficients





■ sélection *forward/backward* :

- + sélection de variables
- + très efficace avec une implémentation adaptée : $\mathcal{O}(Nk^2)$ pour une procédure forward naïve jusqu'à k variables
- décisions binaires

■ projections :

- + sélection de variables
- + efficace : $\mathcal{O}(Np^2)$ (avec une implémentation à la forward)
- variables transformées

■ régression ridge :

- + souple
- + efficace : SVD en $\mathcal{O}(Np^2)$ puis $\mathcal{O}(p^2 + Np)$ par valeur de λ
- pas de sélection de variables



Régularisation L_1

- régression ridge : mesure de régularité $\|\beta\|_2$
- méthode « lasso » :
 - mesure de régularité $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$
 - point de vue modèle

$$\mathcal{F}_C = \left\{ f : \mathbb{R}^p \rightarrow \mathbb{R} \mid f(x) = \beta_0 + \sum_{i=1}^N \beta_i x_i, \sum_{i=1}^p |\beta_i| \leq C \right\}$$

- point de vue optimisation

$$\beta^* = \arg \min_{\beta} \left(\|Y - X\beta\|^2 + \lambda \sum_{i=1}^p |\beta_i| \right)$$

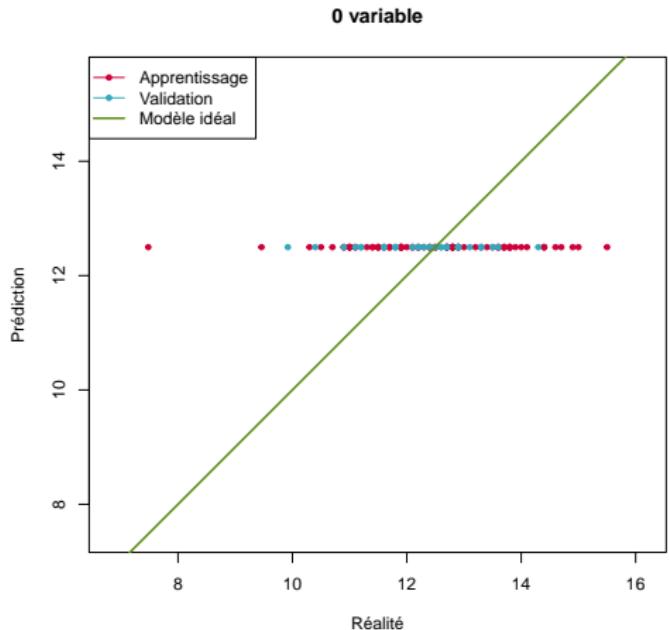
- intérêt : produit naturellement des coefficients nuls
- résolution par programmation quadratique

- algorithme LARS : Least Angle Regression
- idée :
 - ajout progressif des variables (avec sorties possibles)
 - mais sans prendre le coefficient optimal associé à la nouvelle variable
 - sans sortie : lars ; avec sortie : lasso
- même type de coût algorithmique qu'une procédure forward, mais avec plus d'itérations
- calcule un chemin :
 - on montre que l'évolution des paramètres en fonction de λ est affine par morceaux
 - l'algorithme trouve tous les points de jonction



Exemple

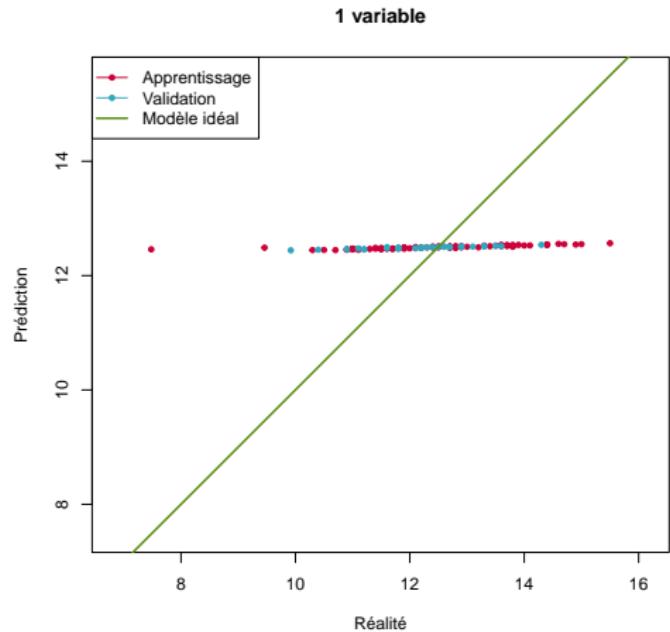
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

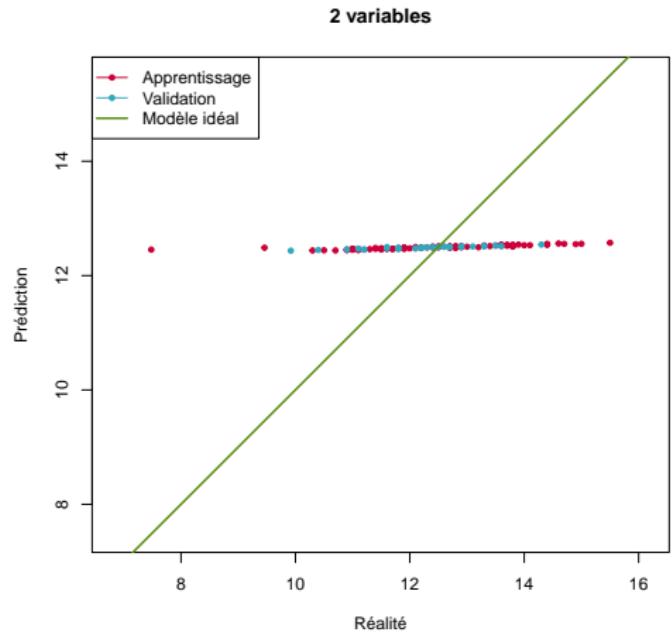
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

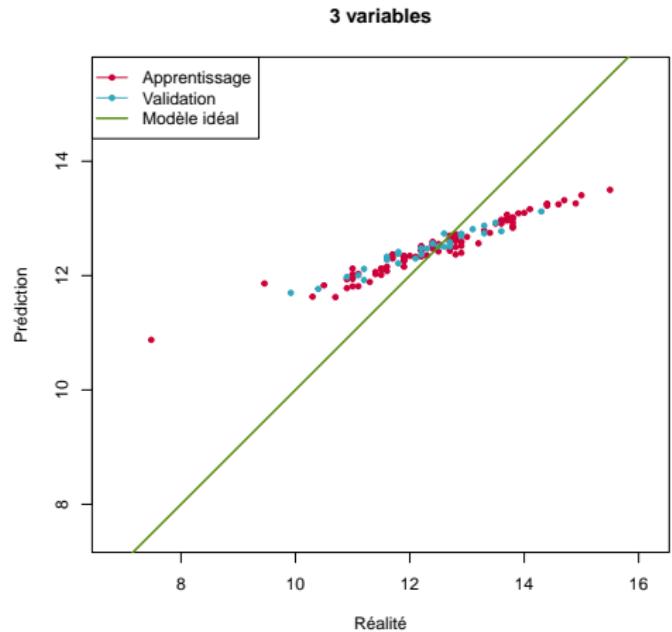
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

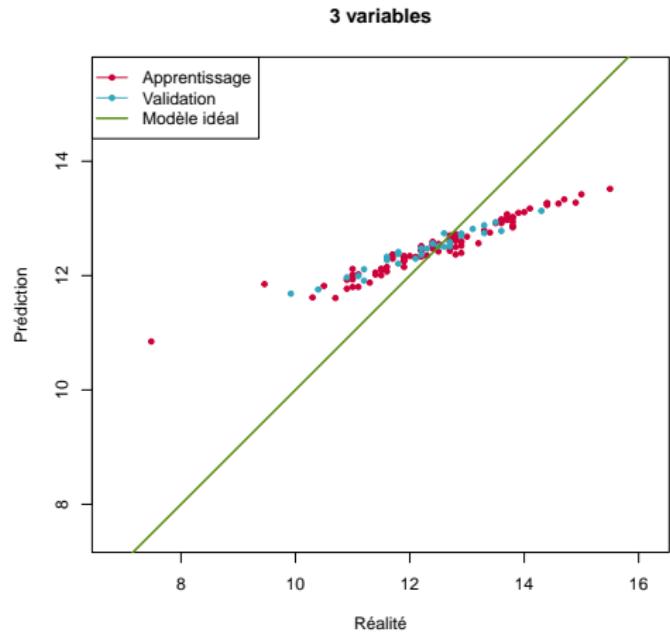
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

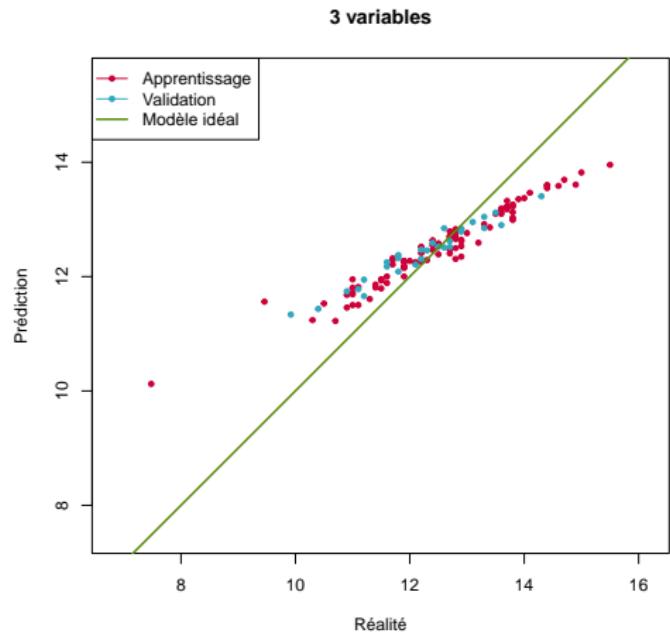
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

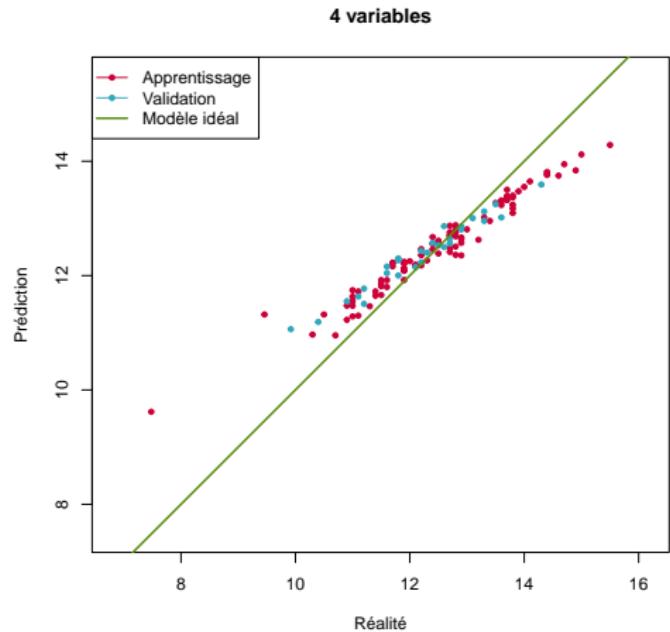
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

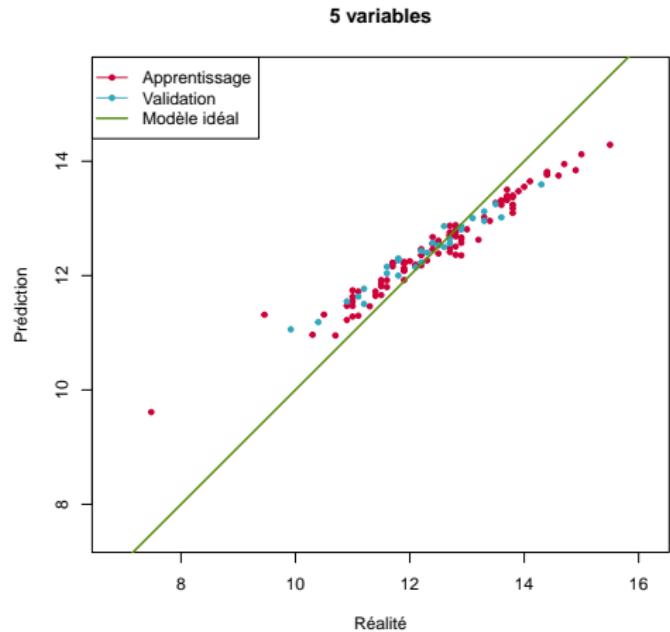
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

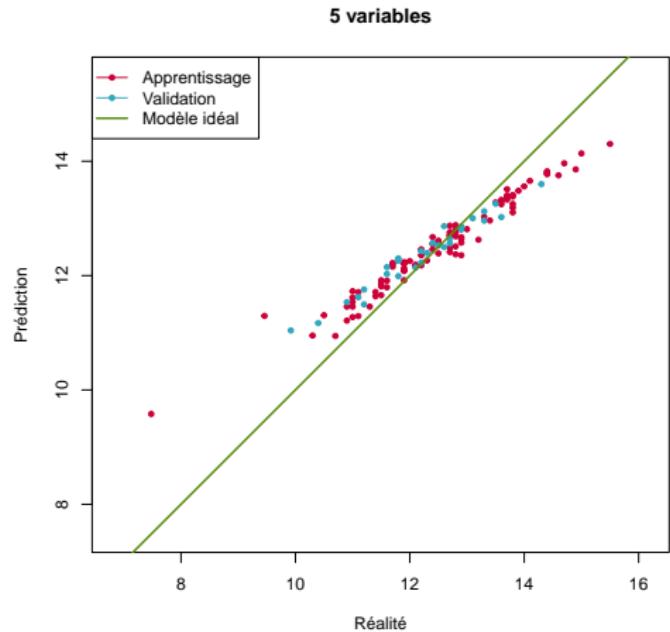
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

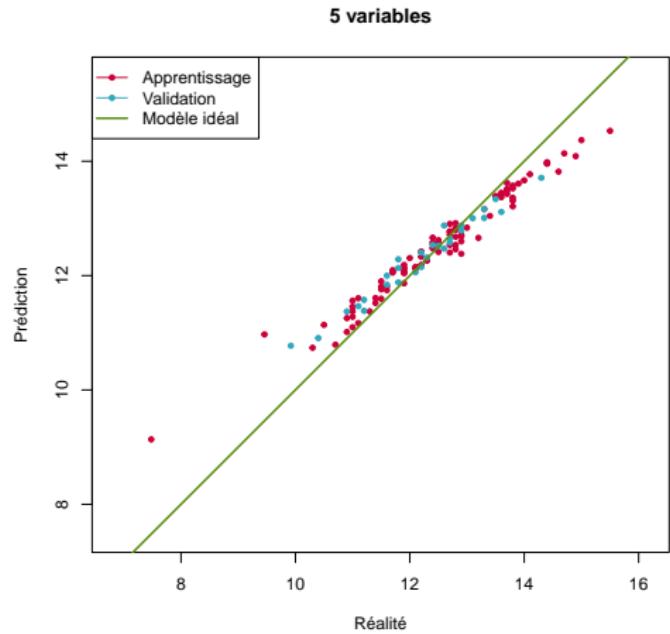
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

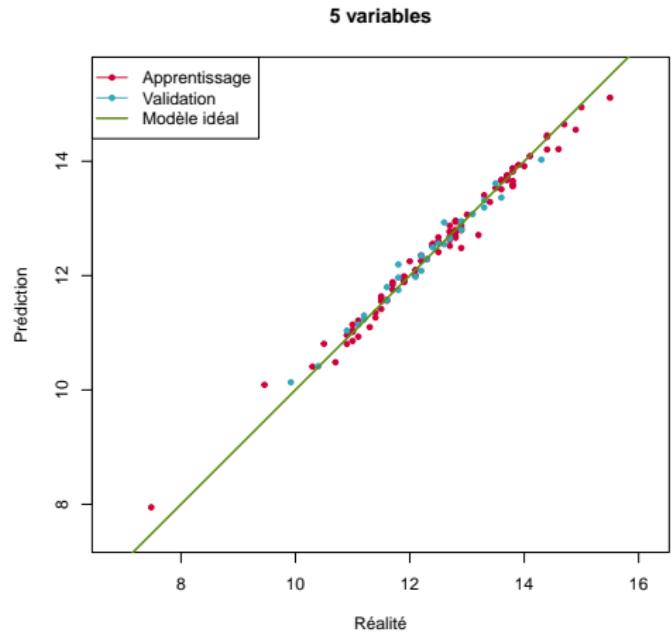
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

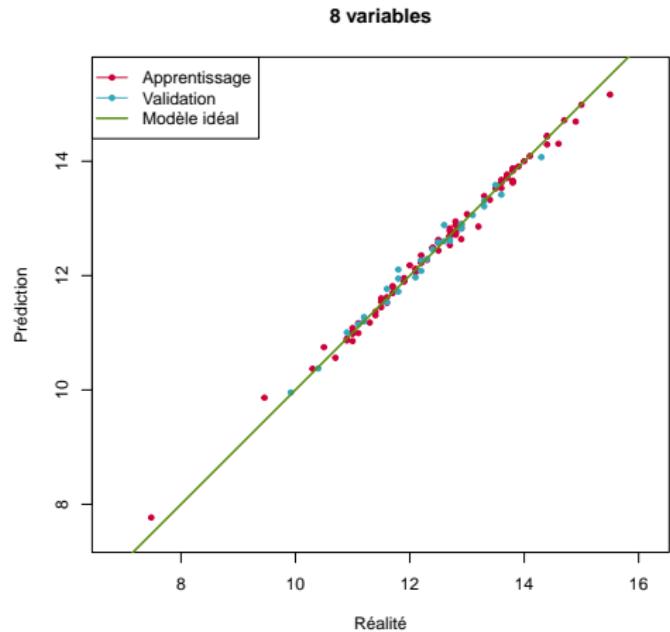
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

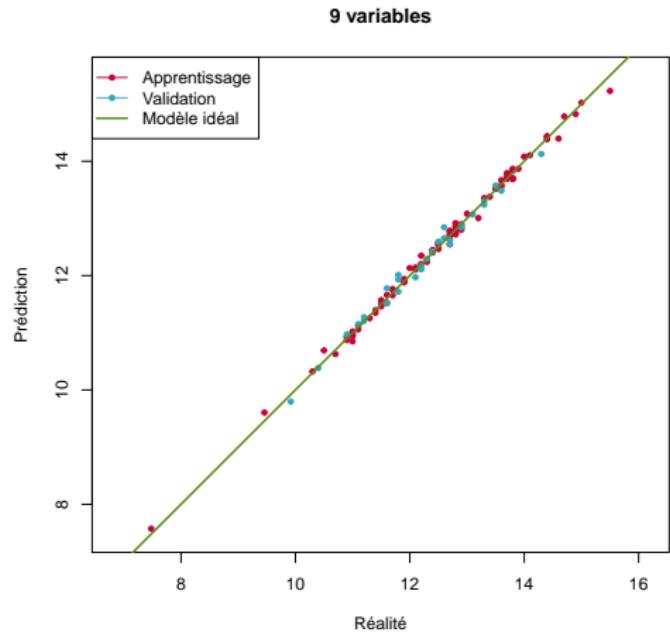
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

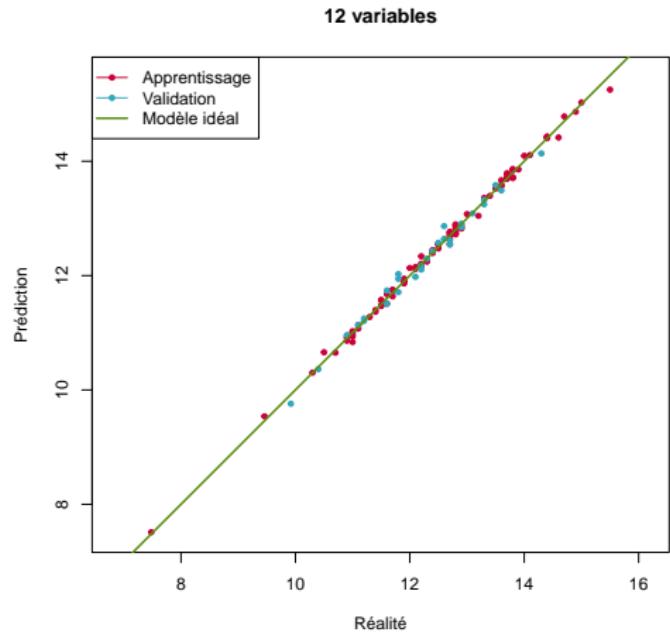
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

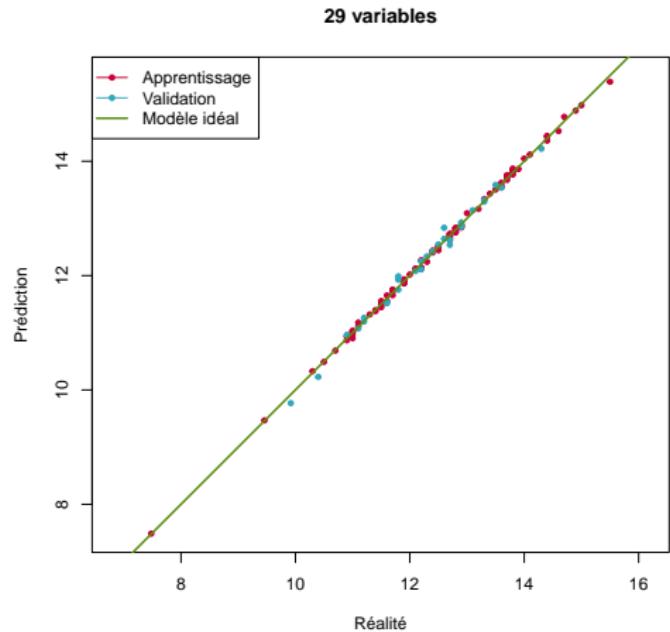
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

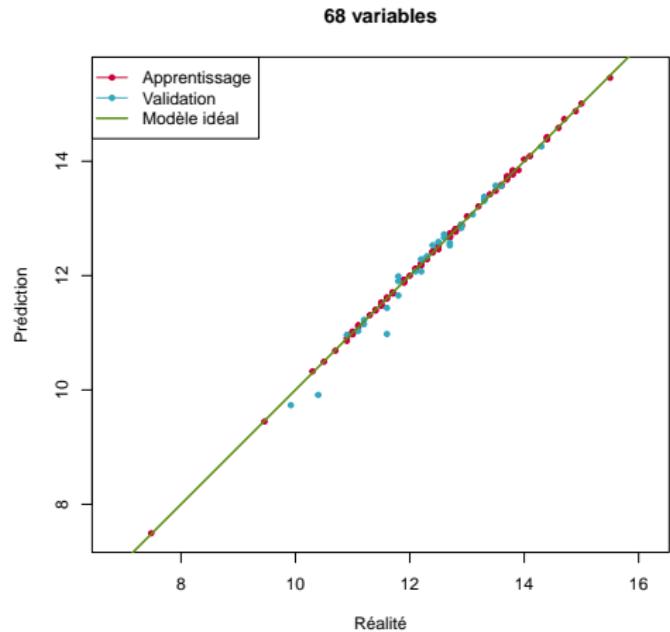
- taux d'alcool dans le vin en fonction du spectre
- lasso





Exemple

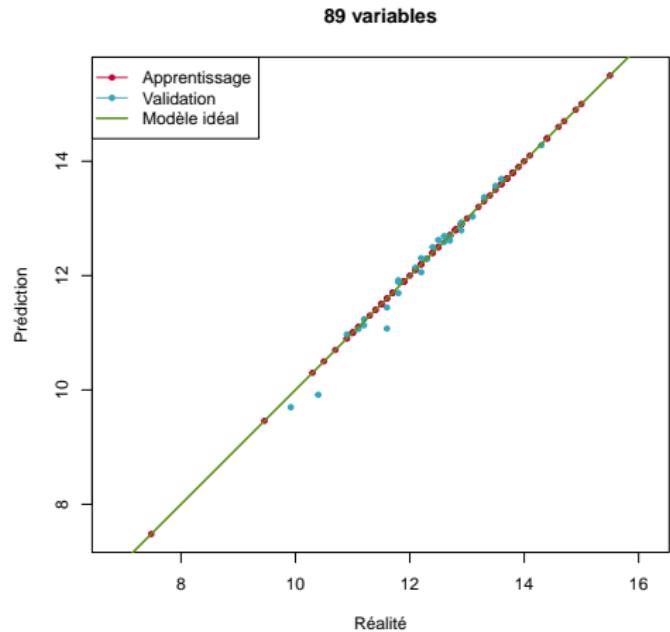
- taux d'alcool dans le vin en fonction du spectre
- lasso





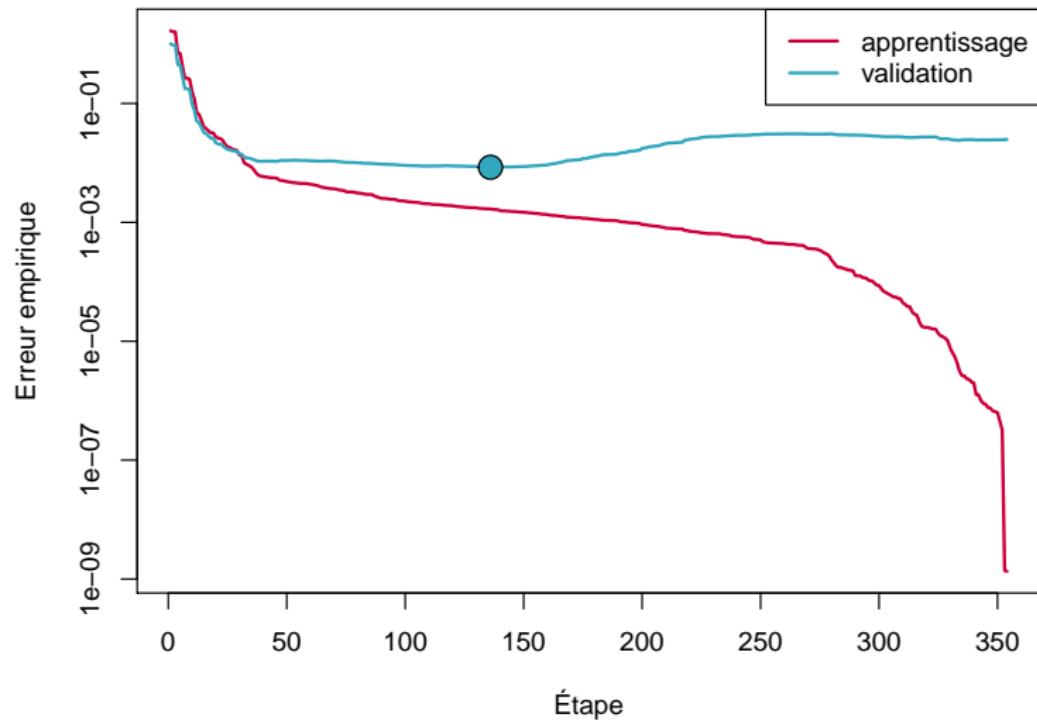
Exemple

- taux d'alcool dans le vin en fonction du spectre
- lasso





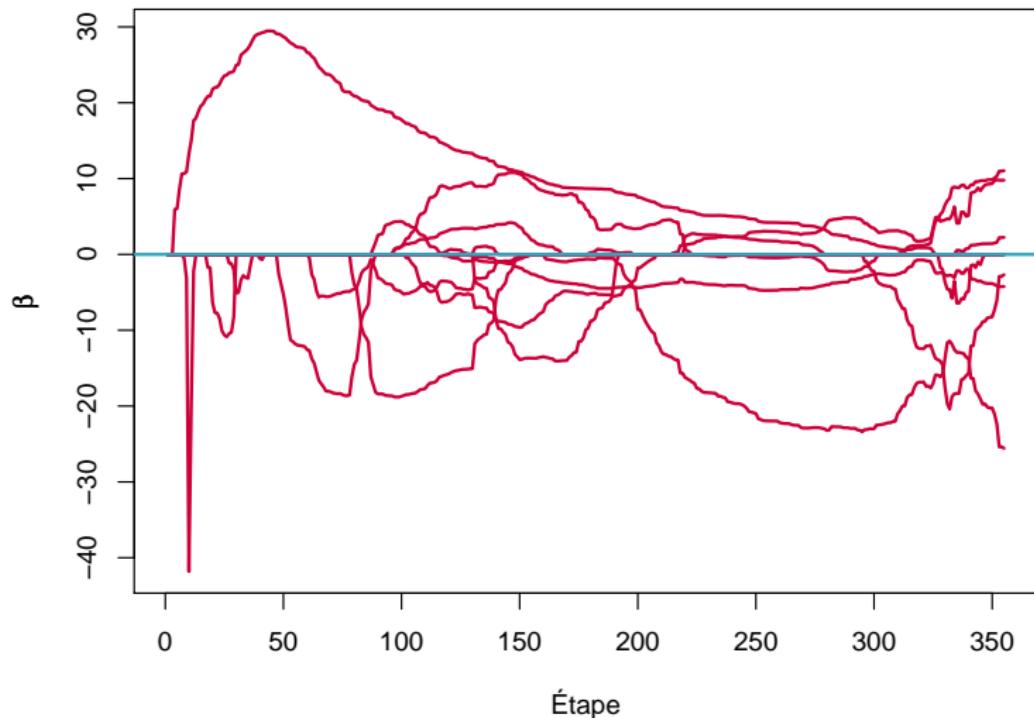
Choix du modèle



étape 136, 29 variables actives

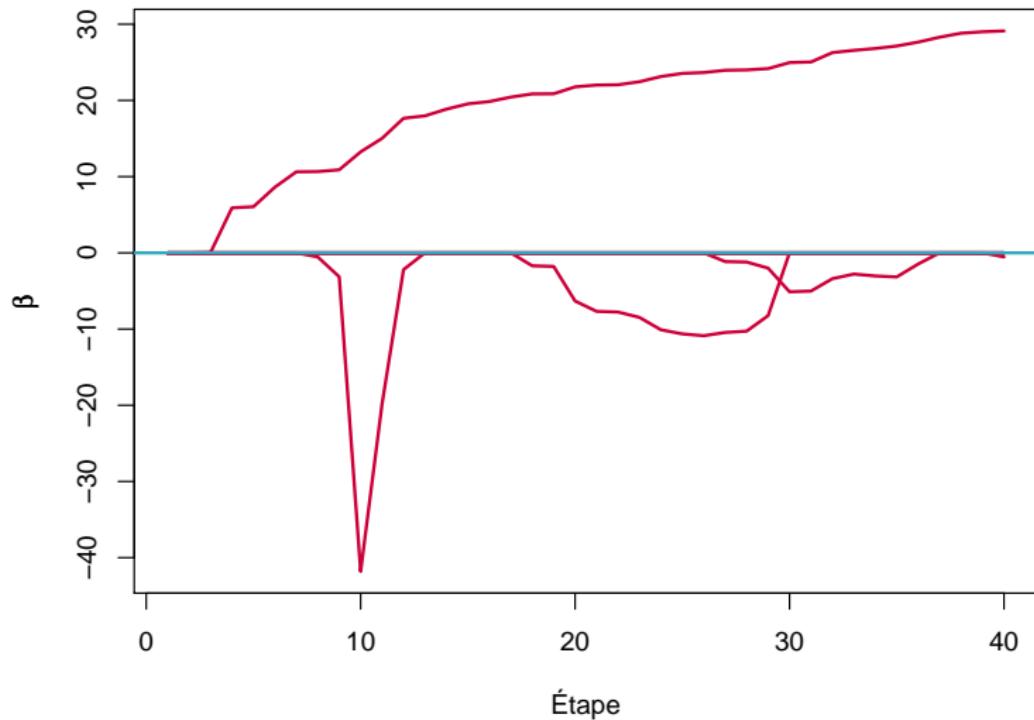


Évolution des coefficients





Évolution des coefficients



- sélection de variables dans le modèle linéaire :
 - package `leaps`
 - fonction `regsubsets`
 - propose la recherche exhaustive avec *branch and bound* et les recherches heuristiques classiques
- projection :
 - package `pls`
 - fonction `pcr` pour la régression sur composantes principales
 - fonction `plsr` pour la régression PLS
- régression ridge : fonction `lm.ridge` du package MASS
- lasso : fonction `lars` du package `lars`



- régression linéaire :
 - méthode simple et efficace pour la régression
 - à toujours tester en premier !
- limitations :
 - données dans \mathbb{R}^p seulement
 - $x \mapsto g(x)$ est affine : insuffisant dans certains cas
 - quand p est grand par rapport à N , le modèle linéaire peut être trop puissant
- limiter la puissance :
 - sélection de variables
 - régularisation
- leçons générales :
 - contrôler la régularité d'un modèle par une pénalité
 - sélectionner un modèle grâce à un ensemble de validation



Non linéarité

- en régression linéaire, $x \mapsto g(x)$ est affine et
 $\beta \mapsto (x \mapsto g(x))$ est linéaire
- certains problèmes ne sont pas linéaires/affines :
 - non linéarité intrinsèque (emballage d'une réaction chimique, par ex.)
 - variables manquantes (inconnues)
- corriger le modèle en gardant la linéarité $\beta \mapsto g$:
 - lever la limitation sur le modèle
 - conserver la simplicité du choix de β (optimisation quadratique)
- idée simple : transformer les données



Transformer les données

■ principe :

- fonction de transformation $\phi : \mathbb{R}^p \rightarrow \mathbb{R}^q$
- régression linéaire sur $(\phi(x_i), y_i)_{1 \leq i \leq N}$
- si ϕ est bien choisie, on obtient des variables linéairement indépendantes dans \mathbb{R}^q :
 - N équations à q inconnues pour $\phi(x_i) \simeq y_i$
 - si q est de l'ordre de N , on trouve toujours une solution exacte
- $x \mapsto \langle \phi(x), \beta \rangle$ n'est plus affine !

■ exemple :

- $\phi(x) = (1, x, x^2)^T$
- classe de modèles

$$\mathcal{F} = \{f : \mathbb{R} \rightarrow \mathbb{R} \mid f(x) = \beta_0 + \beta_1 x + \beta_2 x^2\}$$

- modèles quadratiques



Mise en œuvre

- q fonctions de base $\phi_j : \mathbb{R}^p \rightarrow \mathbb{R}$
- matrice des prédicteurs

$$\Phi(X) = \begin{pmatrix} 1 & \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_q(x_1) \\ \vdots & & \ddots & & \vdots \\ 1 & \phi_1(x_N) & \phi_2(x_N) & \dots & \phi_q(x_N) \end{pmatrix}$$

- problème d'optimisation

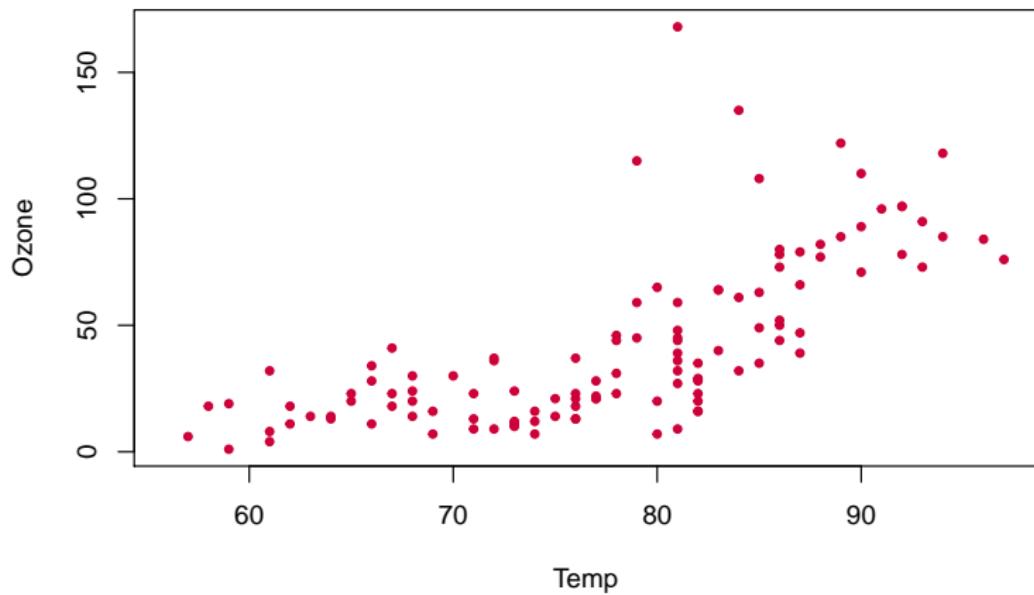
$$\beta^* = \arg \min_{\beta \in \mathbb{R}^{q+1}} \|Y - \Phi(X)\beta\|^2$$

- équations normales associées

$$(\Phi(X)^T \Phi(X))\beta^* = \Phi(X)^T Y$$



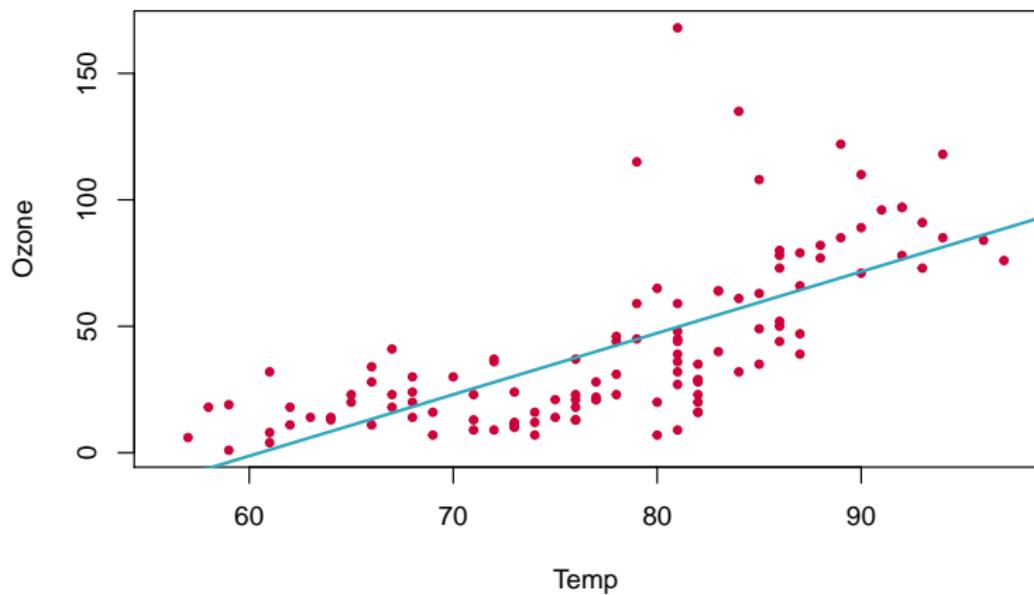
Exemple





Exemple

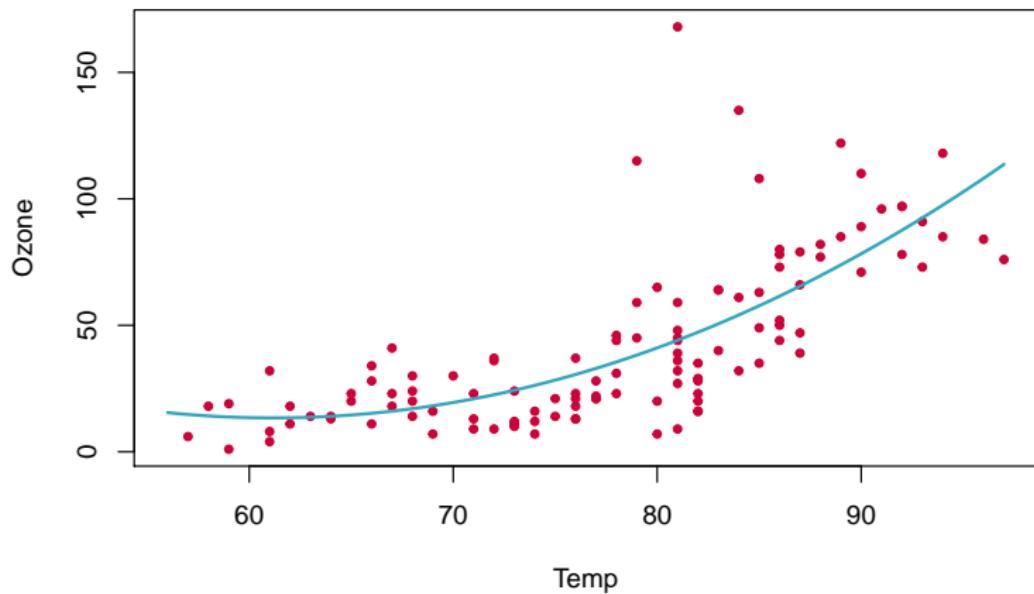
degré : 1





Exemple

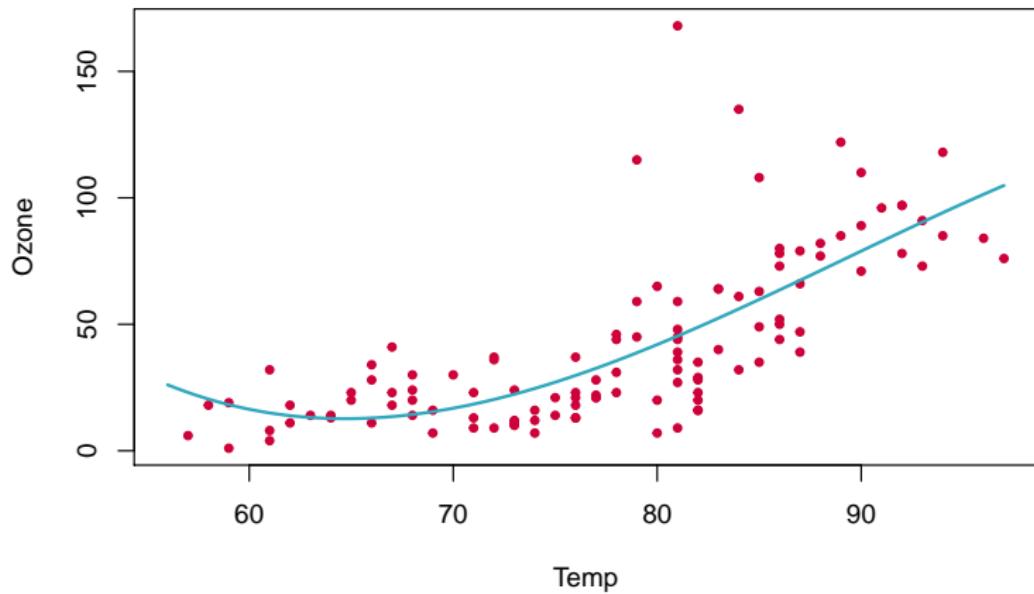
degré : 2





Exemple

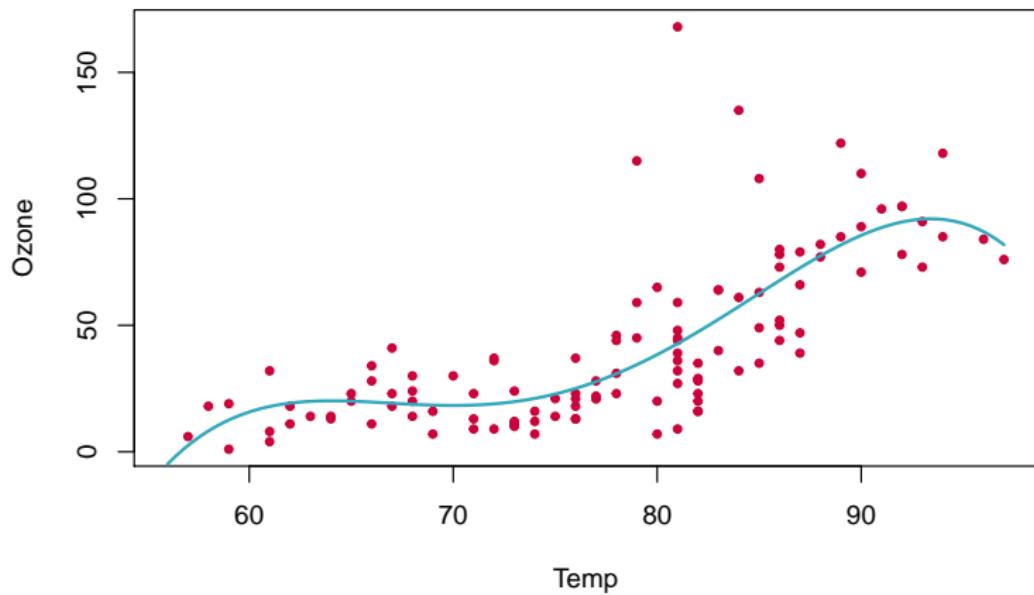
degré : 3





Exemple

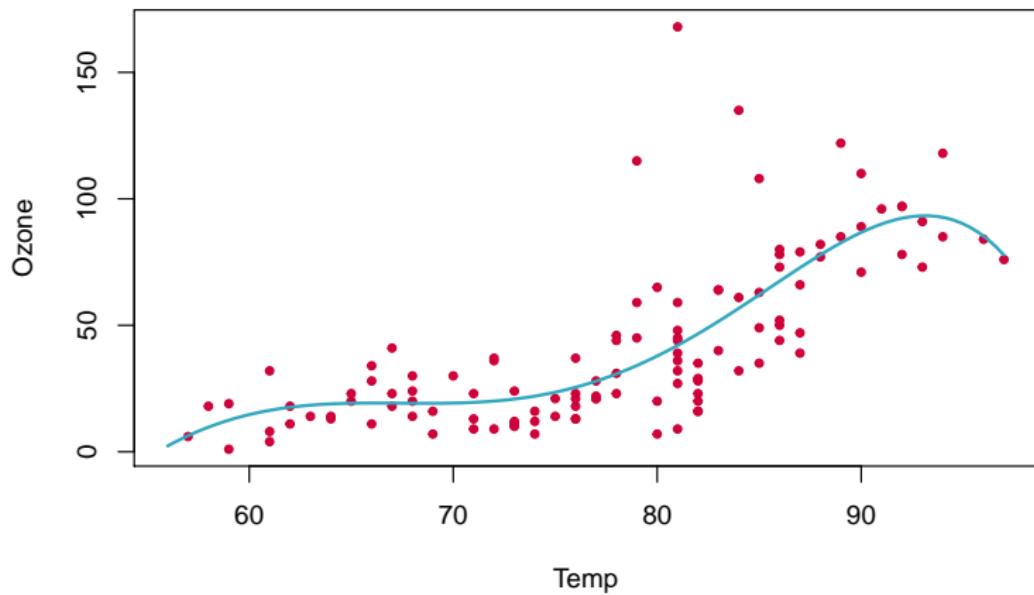
degré : 4





Exemple

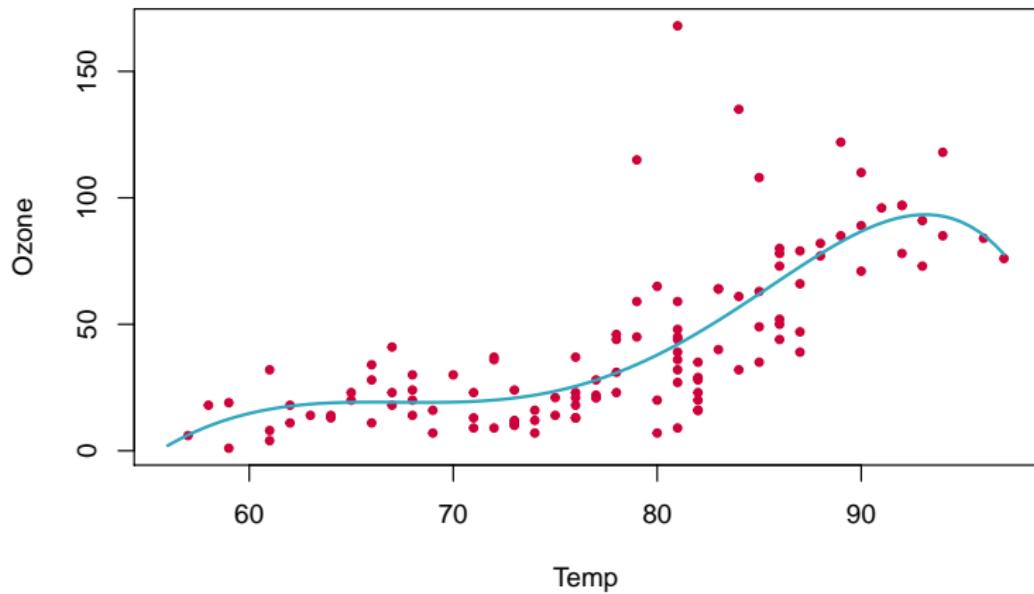
degré : 5





Exemple

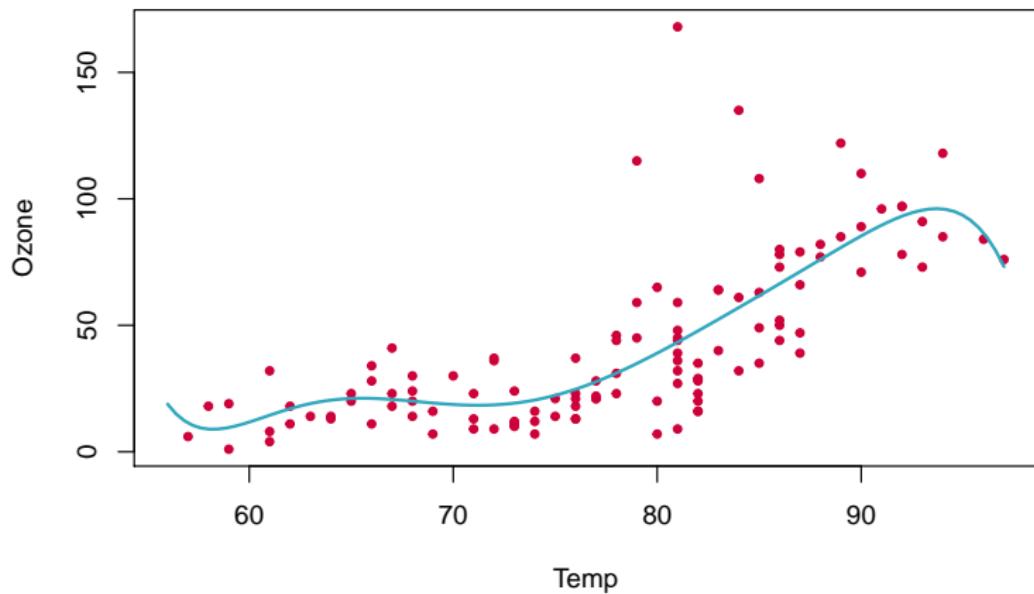
degré : 6





Exemple

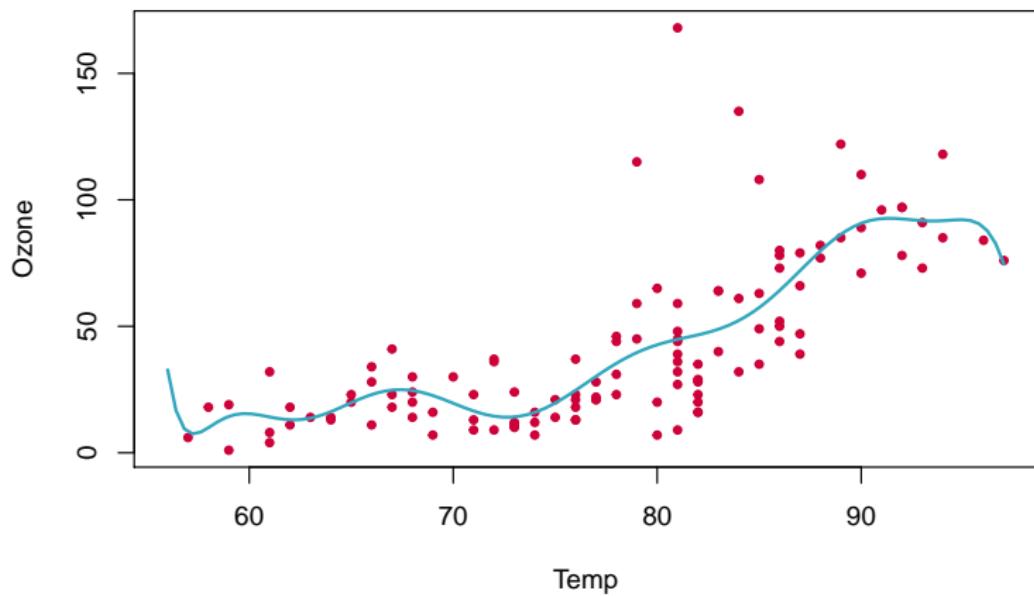
degré : 7





Exemple

degré : 25





■ choix de ϕ :

- quand $x \in \mathbb{R}$ pas de problème particulier : polynômes, splines, ondelettes, séries de Fourier, etc.
- quand $x \in \mathbb{R}^p$, explosion combinatoire :
 - $\frac{(p+d)!}{p!d!}$ monômes de degré inférieur ou égal à d sur p variables
 - même type de problème pour les autres solutions
 - solutions par approches gloutonnes : on ajoute progressivement des ϕ_j

■ coût algorithmique :

- la régression linéaire est en $\mathcal{O}(Np^2)$
- si $p \simeq N \Rightarrow \mathcal{O}(N^3)$: réduction du champ d'application

■ contrôle de la puissance :

- régularisation
- sélection de modèle



Équations normales

- on remarque que si $(\Phi(X)^T \Phi(X))$ est inversible

$$\beta^* = \Phi(X)^T \alpha^* = \sum_{i=1}^N \alpha_i^* \phi(x_i)$$

et donc

$$g(x) = \langle \phi(x), \beta^* \rangle = \sum_{i=1}^N \alpha_i^* \langle \phi(x), \phi(x_i) \rangle$$

- de plus

$$\alpha^* = (\Phi(X) \Phi(X)^T)^{-1} Y$$



Transformation implicite

- or $(\Phi(X)\Phi(X)^T)_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$
- pour construire et utiliser le modèle linéaire sur les $\phi(x_i)$ il suffit de connaître les produits scalaires $\langle \phi(x_i), \phi(x_j) \rangle$
- impact algorithmique :
 - $u = (u_1, u_2)$ and $v = (v_1, v_2)$
 - $\phi(u) = (1, \sqrt{2}u_1, \sqrt{2}u_2, \sqrt{2}u_1u_2, u_1^2, u_2^2)$: 3 opérations
 - $\langle \phi(u), \phi(v) \rangle$: 11 opérations
 - total : 17 opérations
- mais on montre que $\langle \phi(u), \phi(v) \rangle = \left(1 + \sum_{i=1}^2 u_i v_i\right)^2$: 5 opérations
- plus généralement $\langle \phi(u), \phi(v) \rangle = (1 + \langle u, v \rangle)^d$ pour une transformation ϕ utilisant tous les monômes de degré inférieur à d : temps de calcul en $O(p + d)$



■ nouvelle version de l'approche :

- choisir ϕ telle que $\langle \phi(u), \phi(v) \rangle$ se calcule efficacement
- calculer $\alpha^* = (\Phi(X)\Phi(X)^T)^{-1} Y$
- utiliser le modèle

$$x \mapsto \sum_{i=1}^N \alpha_i^* \langle \phi(x), \phi(x_i) \rangle$$

■ une fois $\Phi(X)\Phi(X)^T$ calculée, l'algorithme est en $\mathcal{O}(N^3)$:

- intéressant si ϕ envoie dans \mathbb{R}^q avec $q > N$
- mais dans ce cas le modèle est potentiellement trop puissant
- **régularisation ridge**



Régularisation ridge

- on cherche à contrôler $\|\beta^*\|^2$
- on a

$$\|\beta^*\|^2 = \sum_{i=1}^N \sum_{j=1}^N \alpha_i^* \alpha_j^* \langle \phi(x_i), \phi(x_j) \rangle$$

- on montre que

$$\beta^* = \arg \min_{\beta \in \mathbb{R}^{q+1}} \left(\|Y - \Phi(X)\beta\|^2 + \lambda \|\beta\|^2 \right)$$

correspond à

$$\alpha^* = \left(\Phi(X)\Phi(X)^T + \lambda I \right)^{-1} Y$$



Principe du noyau

- en fait ϕ est inutile, seuls les $\langle \phi(u), \phi(v) \rangle$ entre en jeu
- **noyau (kernel)** :
 - fonction K de $\mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$
 - symétrique : $K(u, v) = K(v, u)$
 - positive : $\sum_{i,j} \lambda_i \lambda_j K(u_i, v_j) \geq 0$
- on montre que pour tout noyau K , il existe une fonction ϕ telle que $K(u, v) = \langle \phi(u), \phi(v) \rangle$ dans un certain espace \mathcal{H} :
 - \mathcal{H} peut être très grand (de dimension infinie)
 - on n'a jamais besoin de calculer explicitement ϕ



- un noyau correspond à un produit scalaire :
 - peut être vue comme une similarité
 - peut être défini sur un espace quelconque :
 - chaînes de caractères (dénombrement de co-occurrence)
 - graphes (chemins communs)
 - etc.
 - \Rightarrow régression régularisée non linéaire sur des données arbitraires
- un noyau important dans \mathbb{R}^p , le **noyau Gaussien** :
 - $K(u, v) = \exp\left(-\frac{\|u-v\|^2}{2\sigma^2}\right)$
 - σ est un paramètre de sensibilité :
 - grand σ : peu sensible, comportement proche du linéaire
 - petit σ : très sensible, comportement proche des k plus proches voisins
- la matrice noyau $K_{ij} = K(x_i, x_j)$ remplace $\Phi(X)\Phi(X)^T$ dans les formules



Kernel Ridge Regression

- choisir un noyau et calculer la matrice $K_{ij} = K(x_i, x_j)$
- algorithme :
 1. diagonaliser K , $K = U^T D U$
 2. calculer $Z = U Y$
 3. pour quelques valeurs de λ (par exemple des puissances de 10) :
 - 3.1 calculer la matrice diagonale $V(\lambda)$ définie par
$$V(\lambda)_{ii} = 1/(D_{ii} + \lambda)$$
 - 3.2 calculer
$$\alpha^* = U^T V(\lambda) Z$$
 4. choisir le modèle optimal (sur un ensemble de validation)
- attention, il faut aussi choisir le noyau (ou ses paramètres) sur un ensemble de validation



- la régression linéaire s'étend facilement au non linéaire :
 - soit par transformation directe (peu de variables explicatives)
 - soit par le biais d'un noyau
- l'accroissement de la puissance rend cruciales :
 - l'utilisation d'une forme de régularisation
 - une sélection de modèle
- outil générique :
 - la régression ridge à noyau (*Kernel Ridge Regression*)
 - coût algorithmique acceptable $\mathcal{O}(N^3)$
 - champ d'application énorme grâce aux noyaux : données non numériques et modèles non linéaires
 - implémentation indépendante du noyau
- il existe de nombreuses autres méthodes non linéaires (par exemple les k plus proches voisins)

Introduction et modélisation mathématique

Apprentissage supervisé

Qualité d'un modèle

Régression

Régression linéaire

Régularisation

Non linéaire

Discrimination

Moindres carrés

Analyse discriminante

Maximisation de la marge

Non linéaire

Sélection de modèle



Discrimination

■ rappels :

- discrimination à deux classes : $\mathcal{Y} = \{A, B\}$
- critère d'erreur : $I(g(x), y) = \delta_{g(x) \neq y}$

■ modèle linéaire en discrimination :

- $g(x) = \langle x, \beta \rangle + \beta_0$ n'est pas directement utilisable
- solution simple :
 - $g(x) = \text{signe}(\langle x, \beta \rangle + \beta_0)$
 - $-1 \Rightarrow \text{classe } A$
 - $1 \Rightarrow \text{classe } B$

■ minimisation du risque empirique

$$\beta^* = \arg \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^N \delta_{\text{signe}(\langle x_i, \beta \rangle + \beta_0) \neq y_i}$$

optimisation combinatoire : impossible en pratique



■ solution simple :

- faire de la régression
- en cherchant à prédire $y_i = -1$ pour la classe A et $y_i = 1$ pour la classe B
- attention : l'opération signe n'est pas prise en compte pour le choix de β

■ on a donc

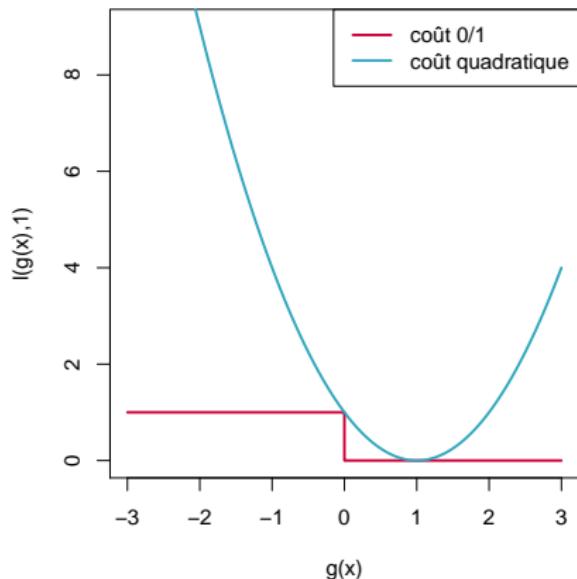
$$\beta^* = \arg \min_{\beta \in \mathbb{R}^p} \left(\sum_{x_i \in A} (\langle x_i, \beta \rangle + 1)^2 + \sum_{x_i \in B} (\langle x_i, \beta \rangle - 1)^2 \right)$$

avec l'augmentation habituelle des x par une variable constante



Interprétation

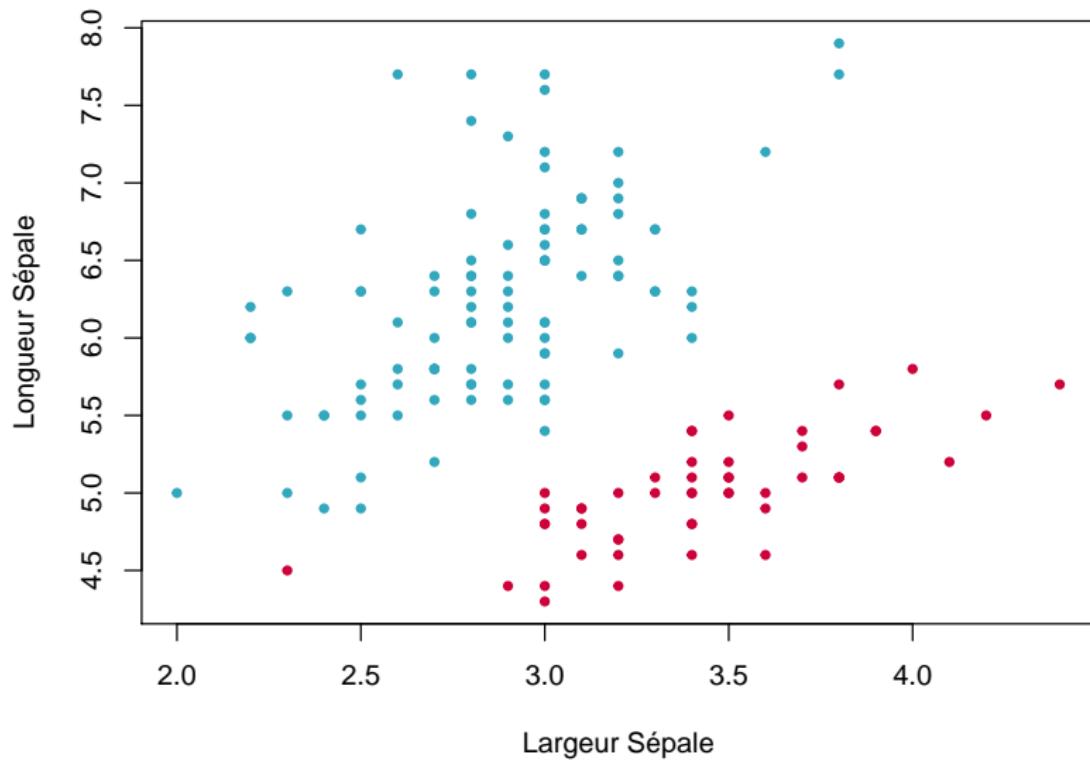
- approximation convexe du coût
- + facile à optimiser
- pénalise un trop bon classement



Solution acceptable mais limitée

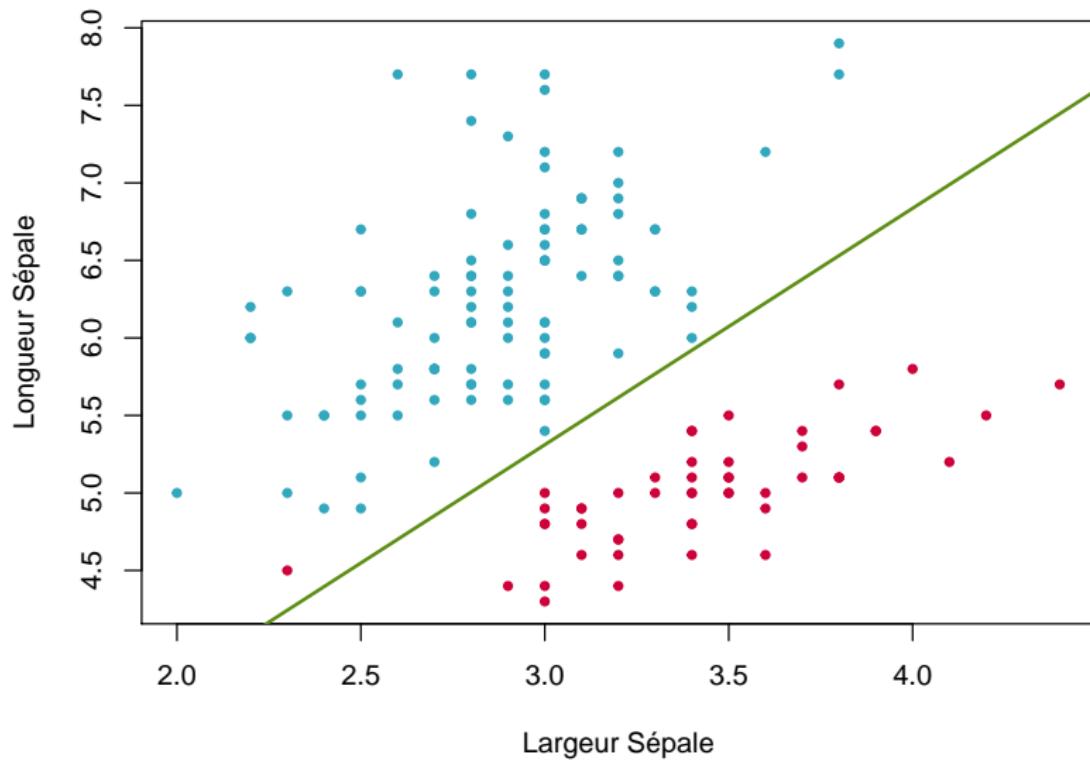


Exemple





Exemple





Plus de deux classes

■ très mauvaise solution :

- numérotter les classes de 1 à K
- faire une régression linéaire avec comme cible le numéro de la classe
- ne jamais faire ça :
 - induit une fausse structure d'ordre sur les classes
 - rend la régression plus difficile qu'elle ne devrait l'être
 - etc.

■ solutions par combinaisons :

- construire $K - 1$ modèles : 1 contre les autres classes
- construire $K(K - 1)/2$ modèles : 1 contre 1

■ solution par codage :

- représenter l'appartenance à la classe k par un vecteur de \mathbb{R}^K contenant $K - 1$ zéros et un 1 dans la variable k
- puis régression classique

- $x \mapsto \langle x, \beta \rangle$ est une projection de \mathbb{R}^p dans \mathbb{R}
- comment optimiser la projection pour bien répartir les exemples en deux classes ?
 - bien regrouper les projetés d'une même classe (variance intra petite)
 - bien éloigner les projetés de classes différences (variance inter grande)
- analyse discriminante de Fisher : maximisation du ratio entre les variances
- s'applique à plusieurs classes, C_1, \dots, C_K

■ décomposition de la covariance :

- covariance totale $T = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T$ (μ moyenne des x)
- covariances intraclasse $W_k = \frac{1}{N_k} \sum_{i \in C_k} (x_i - \mu_k)(x_i - \mu_k)^T$ (μ_k moyenne des x de la classe C_k)
- covariance interclasse $B = \frac{1}{N} \sum_{k=1}^K N_k (\mu_k - \mu)(\mu_k - \mu)^T$
- $T = B + W$, avec $W = \frac{1}{N} \sum_{k=1}^K N_k W_k$

■ projection = « multiplication » par β

- intraclasse : $\beta^T W \beta$
- interclasse : $\beta^T B \beta$

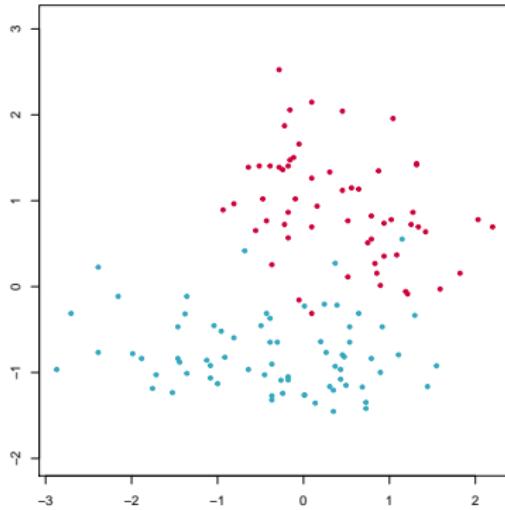


Critère de Fisher

- Critère de Fisher : maximiser $\frac{\beta^T B \beta}{\beta^T W \beta}$
- si β maximise le critère, on montre qu'il existe λ tel que $B\beta = \lambda W\beta$ (problème de valeur propre généralisé)
- en général W est inversible et β est donc vecteur propre de $W^{-1}B$ (associé à la plus grande valeur propre)
- algorithme basique (méthode de la puissance itérée) :
 - $\beta^{(0)}$ aléatoire
 - $\beta^{(t+1)} = \frac{1}{\|W^{-1}B\beta^{(t)}\|} W^{-1}B\beta^{(t)}$
 - converge vers un vecteur propre associé à la plus grande valeur propre
- puis on ajoute un seuil β_0 optimal (sous une hypothèse de distribution gaussienne)

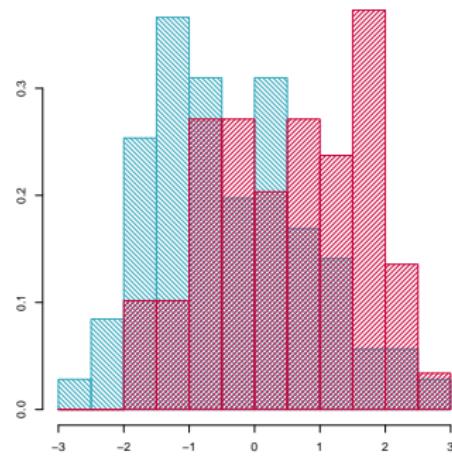
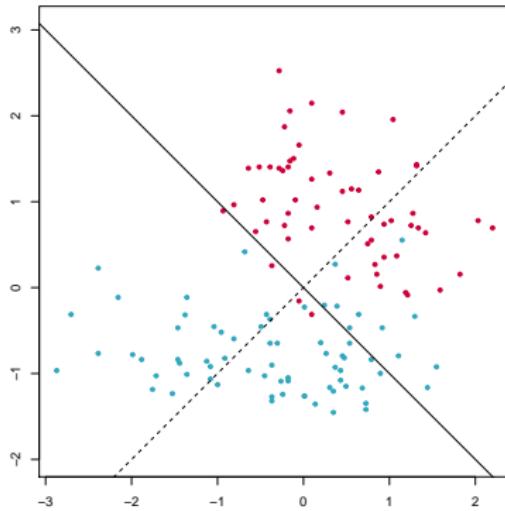


Exemple



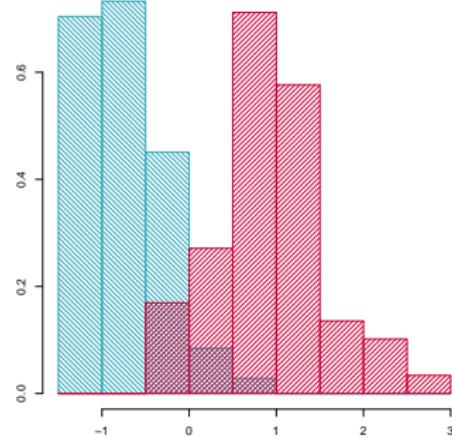
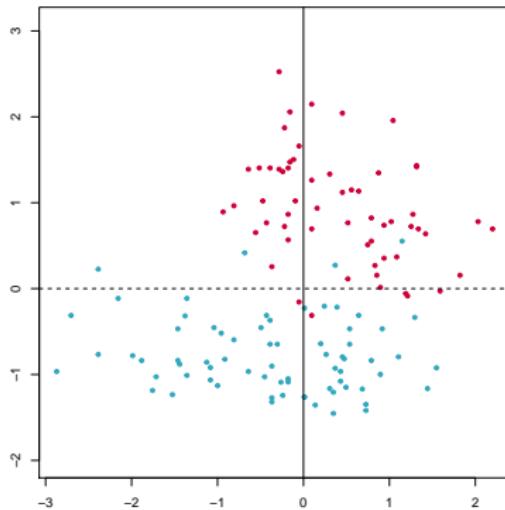


Exemple



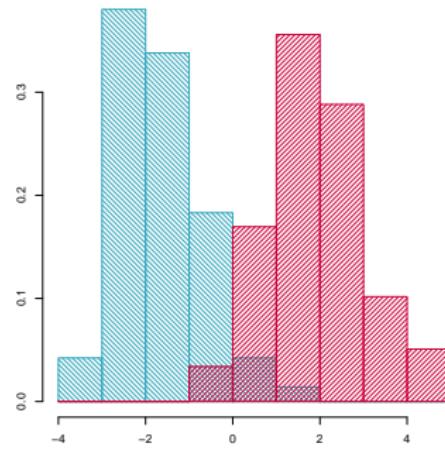
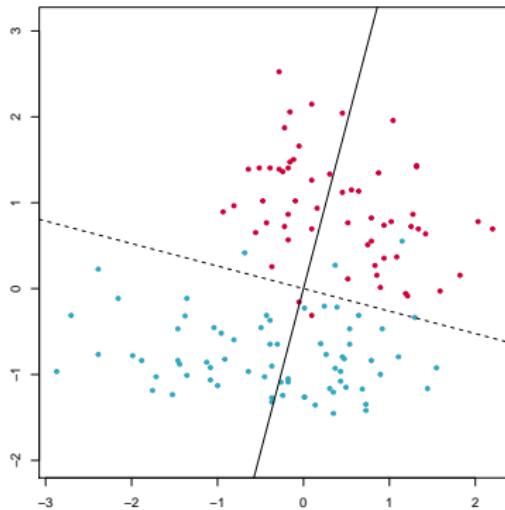


Exemple



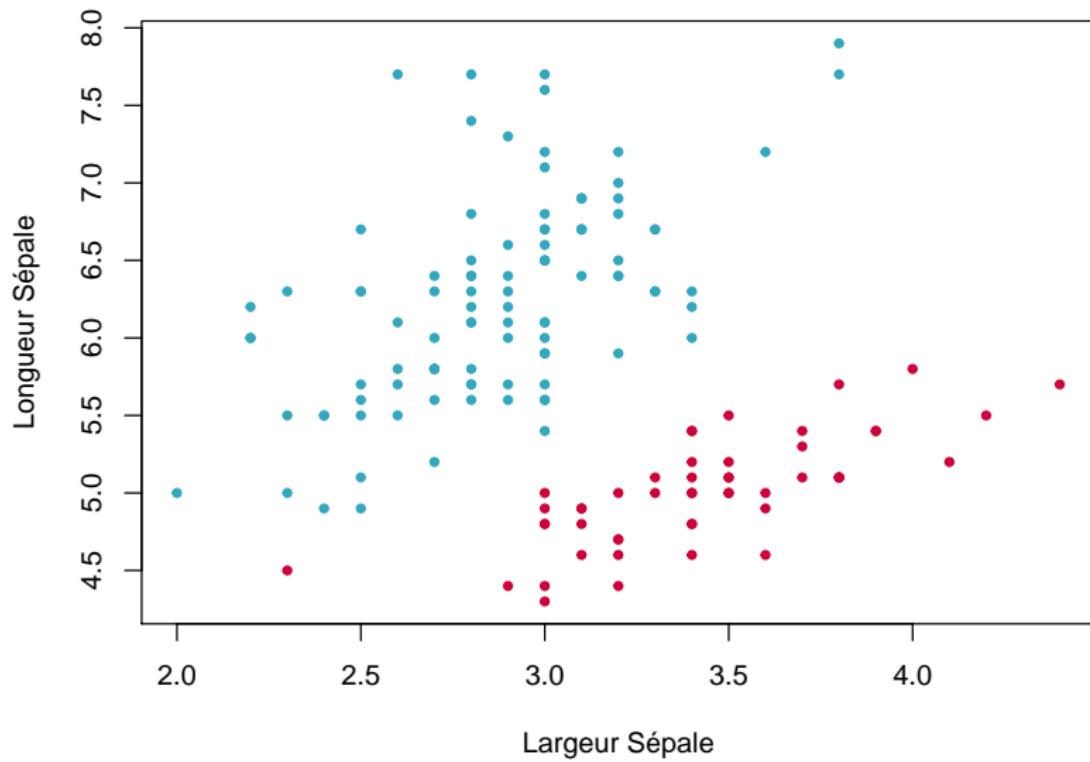


Exemple



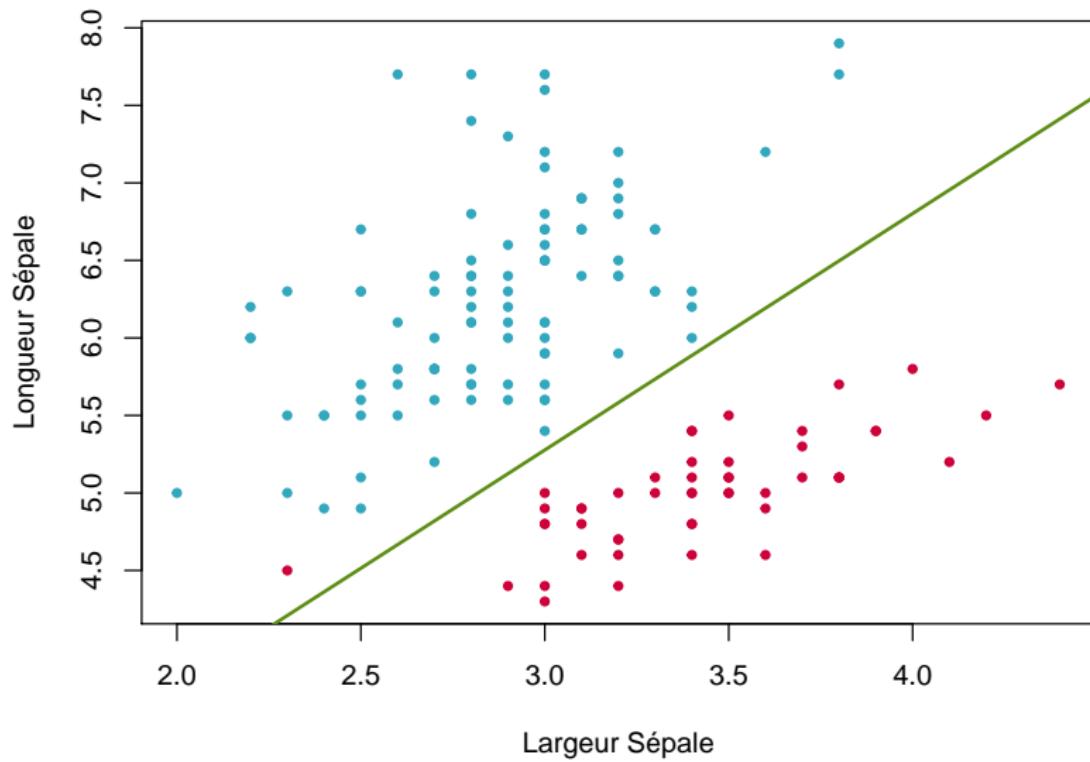


Exemple





Exemple





■ cas à deux classes :

- on montre facilement que l'analyse discriminante conduit au même hyperplan séparateur que la régression mais à des seuils (β_0) différents
- les méthodes donnent des résultats strictement identiques si les deux classes sont de mêmes tailles

■ cas à trois classes ou plus :

- les résultats sont très différents
- l'analyse discriminante fonctionne généralement mieux que la régression



Choix d'un séparateur linéaire

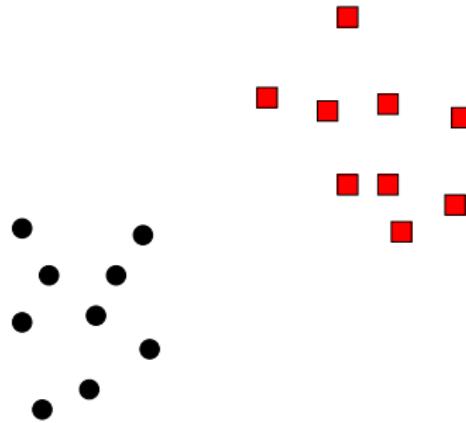
Deux problèmes principaux :

1. cas séparable : en général une infinité de solutions, comment choisir ?
 - critère additionnel : choix parmi les solutions exactes (avec aucune erreur)
 - critère alternatif : optimisation d'une autre grandeur (pas le nombre d'erreurs)
2. cas non séparable : comment minimiser le nombre d'erreurs ?
 - algorithme de coût acceptable
 - critère alternatif (bis)

Question subsidiaire : le cas non linéaire est-il fréquent ?

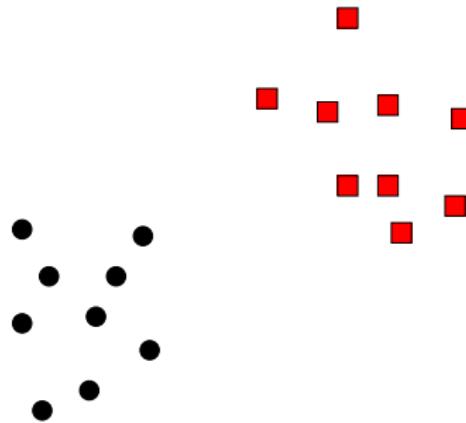


Maximisation de la marge





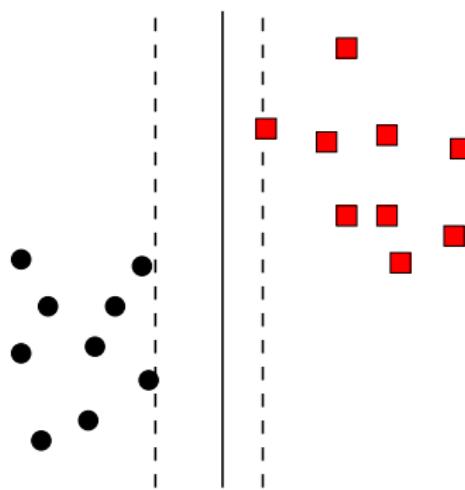
Maximisation de la marge



- Données linéairement séparables : une infinité de choix possibles



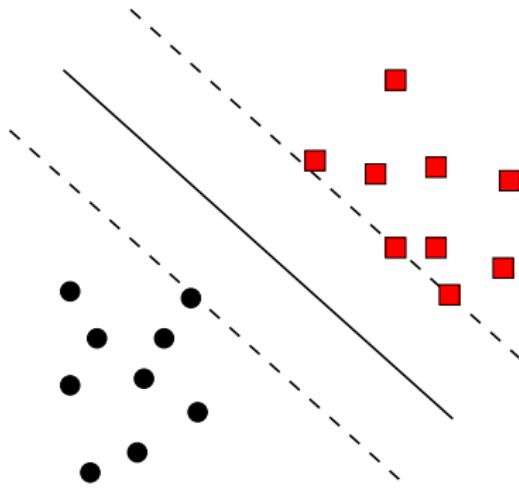
Maximisation de la marge



- Données linéairement séparables : une infinité de choix possibles
- Données proches du séparateur : petite « marge » \Rightarrow faible robustesse



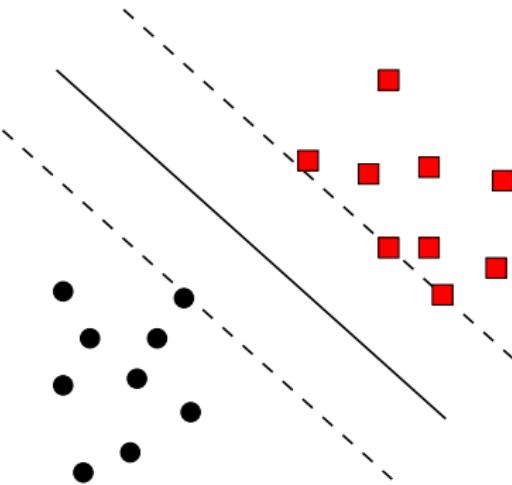
Maximisation de la marge



- Données linéairement séparables : une infinité de choix possibles
- Données proches du séparateur : petite « marge » \Rightarrow faible robustesse
- Un critère de choix possible : maximiser la marge



Maximisation de la marge



- Données linéairement séparables : une infinité de choix possibles
- Données proches du séparateur : petite « marge » \Rightarrow faible robustesse
- Un critère de choix possible : maximiser la marge
- Machine à vecteurs de support



Formulation du problème

- marge : distance entre le séparateur et l'observation la plus proche
- marge ($y_i \in \{-1, 1\}$) :

$$\min_i \frac{|\langle \beta, x_i \rangle + \beta_0|}{\langle \beta, \beta \rangle} = \min_i \frac{y_i(\langle \beta, x_i \rangle + \beta_0)}{\langle \beta, \beta \rangle},$$

en l'absence d'erreur, c.-à-d., avec $y_i(\langle \beta, x_i \rangle + \beta_0) > 0$

- normalisation par $\min_i y_i(\langle \beta, x_i \rangle + \beta_0)$:

$$(P_0) \quad \min_{\beta, \beta_0} \frac{1}{2} \langle \beta, \beta \rangle, \\ \text{sous les contraintes } y_i(\langle \beta, x_i \rangle + \beta_0) \geq 1, \quad 1 \leq i \leq N.$$

- problème d'optimisation quadratique sous contraintes linéaires
- formulation duale plus simple



Formulation duale

- (P_0) est équivalent à

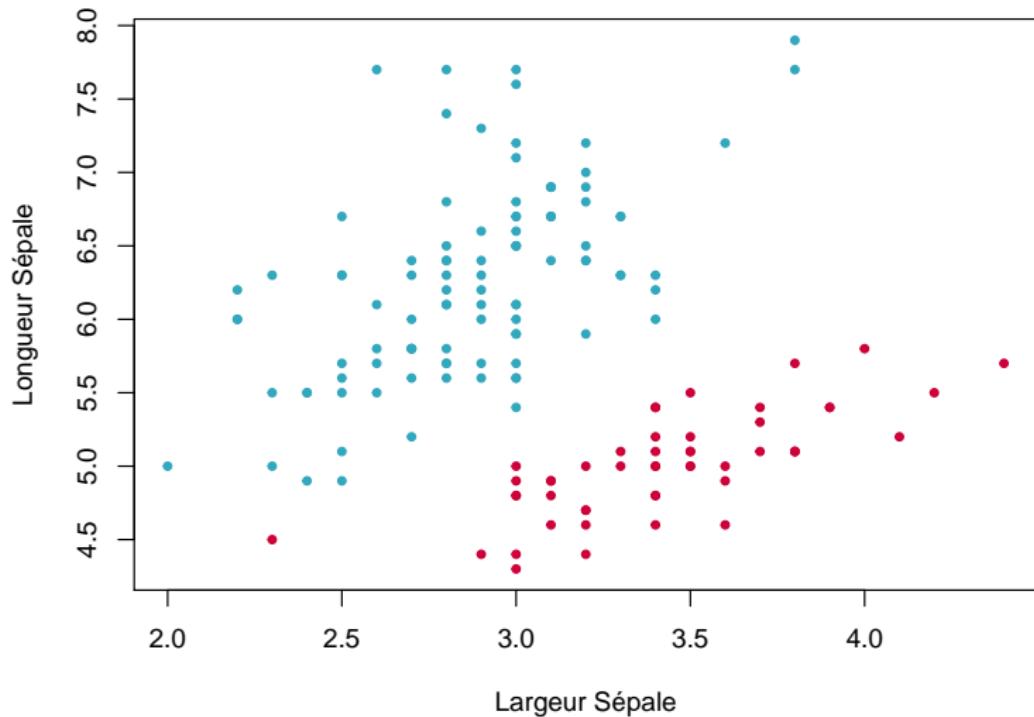
$$(D_0) \max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

sous les contraintes $\sum_{i=1}^N \alpha_i y_i = 0$ et $\alpha_i \geq 0$

- problème plus facile à résoudre :
 - toujours quadratique
 - contraintes plus simples
- on montre que $(y_i(\langle \beta, x_i \rangle + \beta_0) - 1) > 0 \Rightarrow \alpha_i = 0$:
 - les observations éloignées du séparateur n'interviennent pas dans la solution
 - la solution dépend uniquement des observations « sur la marge » : les **vecteurs de support** (contraintes saturées)
 - on a aussi $\langle \beta, x \rangle = \sum_{\alpha_i \neq 0} \alpha_i y_i \langle x_i, x \rangle$

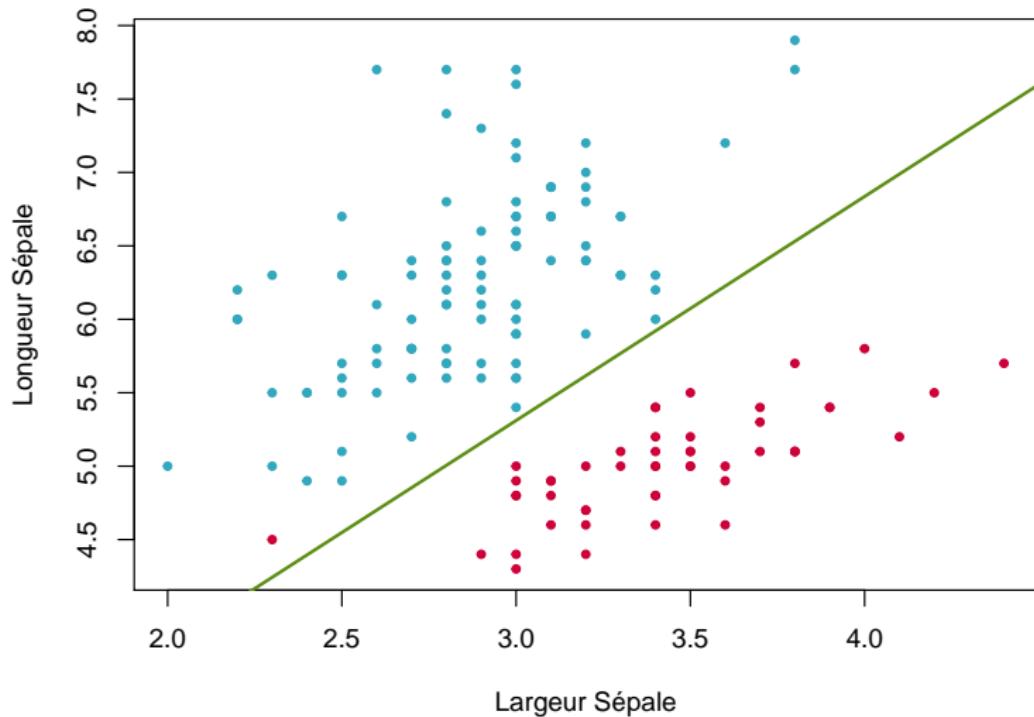


Exemple



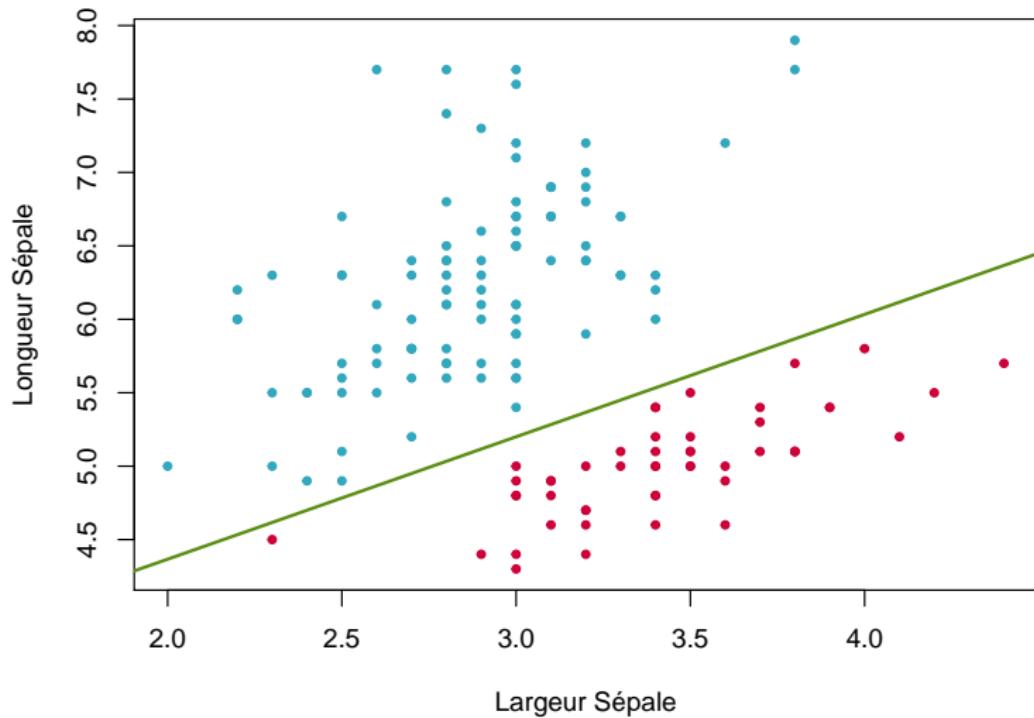


Exemple



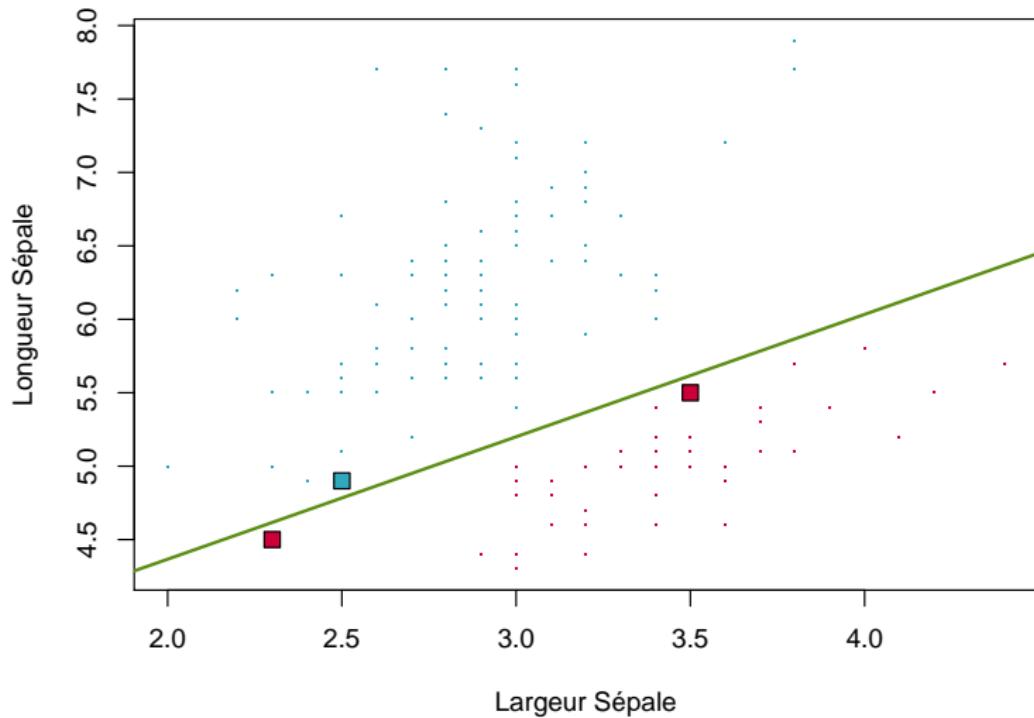


Exemple



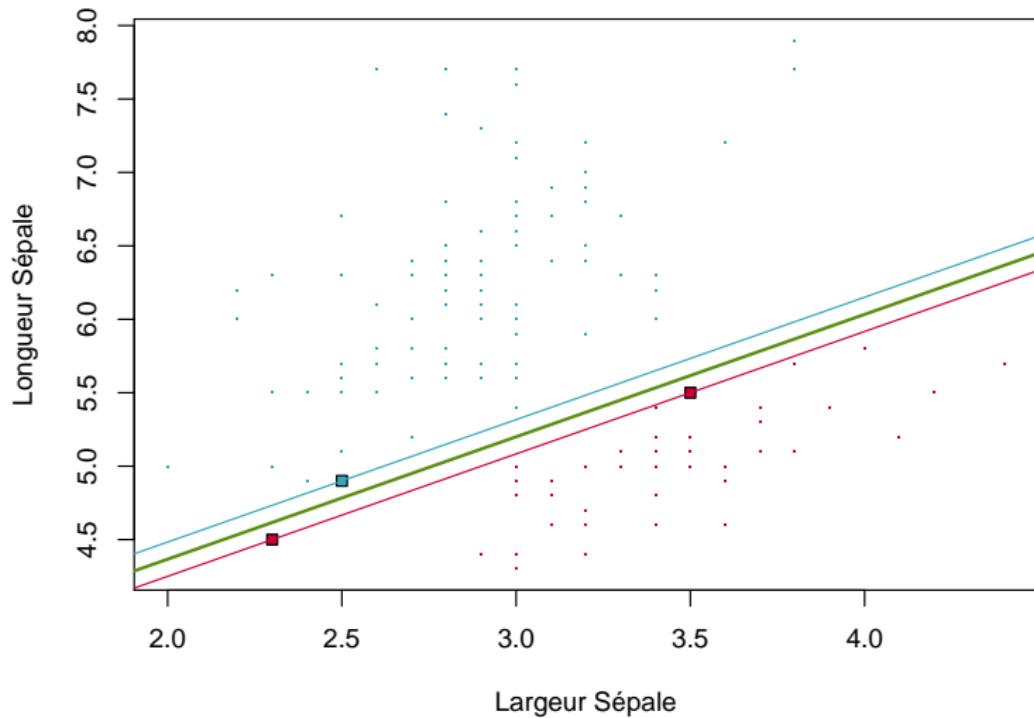


Exemple





Exemple





Cas non linéairement séparable

- le problème P_0 n'a pas de solution : pas de point admissible
- assouplir les contraintes :
 - autoriser des erreurs de classement
 - conserver la notion de marge pour les points bien classés
 - $y_i(\langle \beta, x_i \rangle + \beta_0) \geq 1 - \xi_i$ avec $\xi_i \geq 0$
 - les ξ_i sont les « variables ressort »
- nouveau problème :

$$(P_C) \quad \min_{\beta, \beta_0, \xi} \frac{1}{2} \langle \beta, \beta \rangle + C \sum_{i=1}^N \xi_i,$$

avec $y_i(\langle \beta, x_i \rangle + \beta_0) \geq 1 - \xi_i, \quad 1 \leq i \leq N,$
 $\xi_i \geq 0, \quad 1 \leq i \leq N.$

- variantes possibles (par exemple $C \sum_{i=1}^N \xi_i^2$)



Interprétation

- (P_C) s'écrit aussi

$$(P_C) \min_{\beta, \beta_0, \xi} \frac{1}{2} \langle \beta, \beta \rangle + C \sum_{i=1}^N \xi_i,$$

avec $\xi_i \geq 1 - y_i(\langle \beta, x_i \rangle + \beta_0)$, $1 \leq i \leq N$,

$$\xi_i \geq 0, \quad 1 \leq i \leq N.$$

- de façon équivalente :

$$(P_C) \min_{\beta, \beta_0} \frac{1}{2} \langle \beta, \beta \rangle + C \sum_{i=1}^N \max(1 - y_i(\langle \beta, x_i \rangle + \beta_0), 0)$$

- interprétation de C :

- compromis entre erreurs et marge, régularisation
- C grand : erreurs interdites, au détriment de la marge (le modèle « colle » aux données)
- C petit : marge maximisée, au détriment des erreurs
- choix de C : choix de modèle



Autre interprétation

- le coût

$$l(g(x), y) = \max(1 - yg(x), 0)$$

est appelé le *hinge loss*

- on remarque que

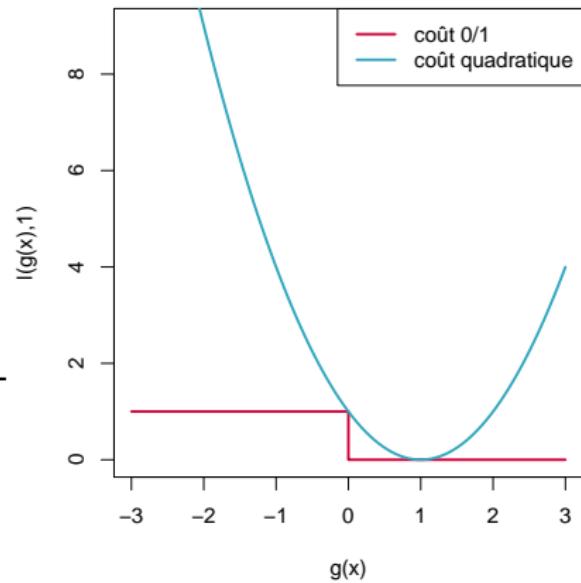
$$l(g(x), y) \geq \delta_{\text{signe}(g(x)) \neq y}$$

- le *hinge loss* est une majoration convexe du coût 0/1



Coût quadratique ou *hinge*

- approximation convexe du coût
- + facile à optimiser
- pénalise un trop bon classement

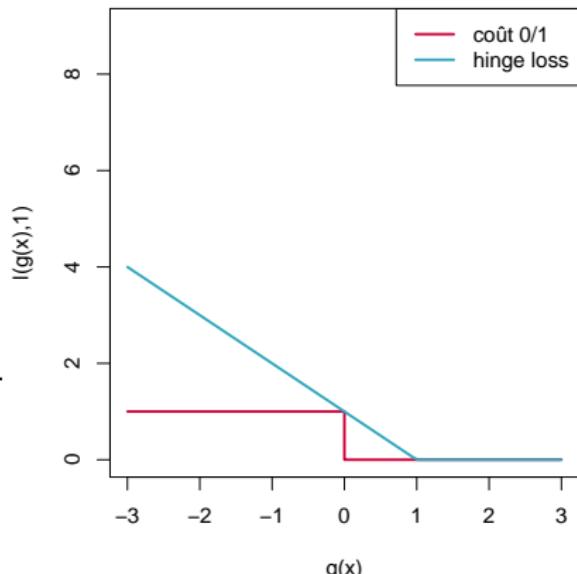


coût quadratique



Coût quadratique ou *hinge*

- approximation convexe du coût
- + facile à optimiser
- + ne pénalise pas les bons classements
- + n'explose pas



hinge loss



Problème dual

- (P_C) est équivalent à

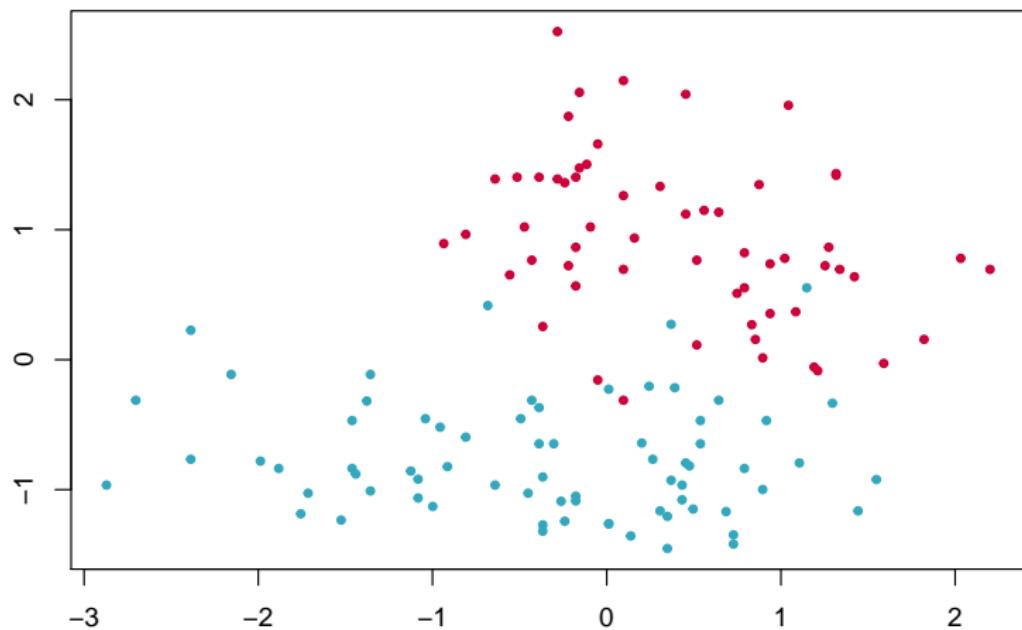
$$(D_C) \max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$$

sous les contraintes $\sum_{i=1}^N \alpha_i y_i = 0$ et $0 \leq \alpha_i \leq C$

- seul changement : valeur maximale sur les multiplicateurs
- coût algorithmique :
 - algorithme « exact » en $\mathcal{O}(N^3)$
 - algorithme plus heuristique en $\mathcal{O}(N^2)$ en pratique



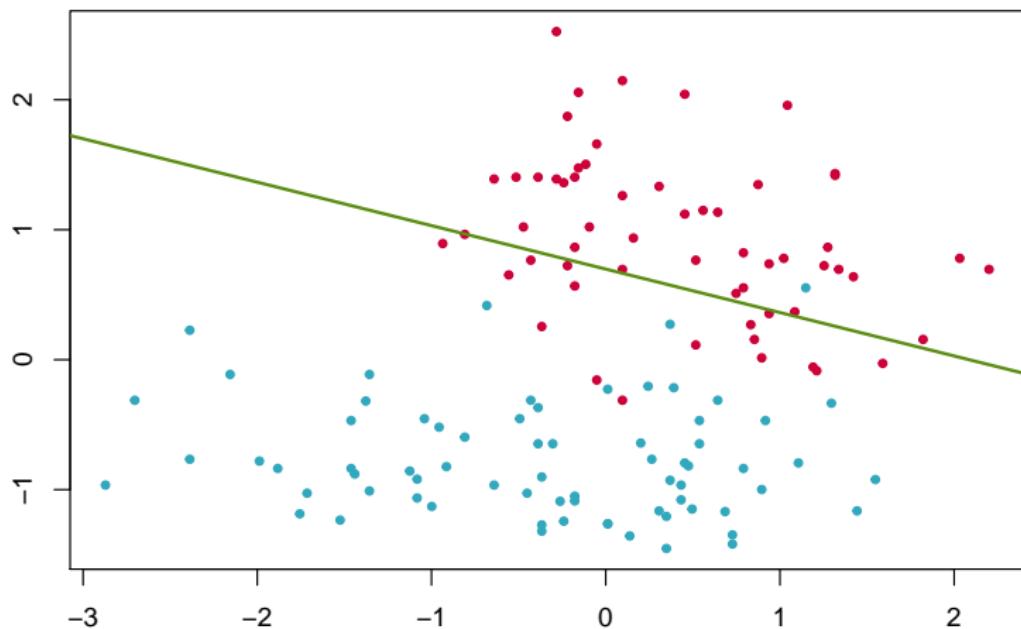
Exemple





Exemple

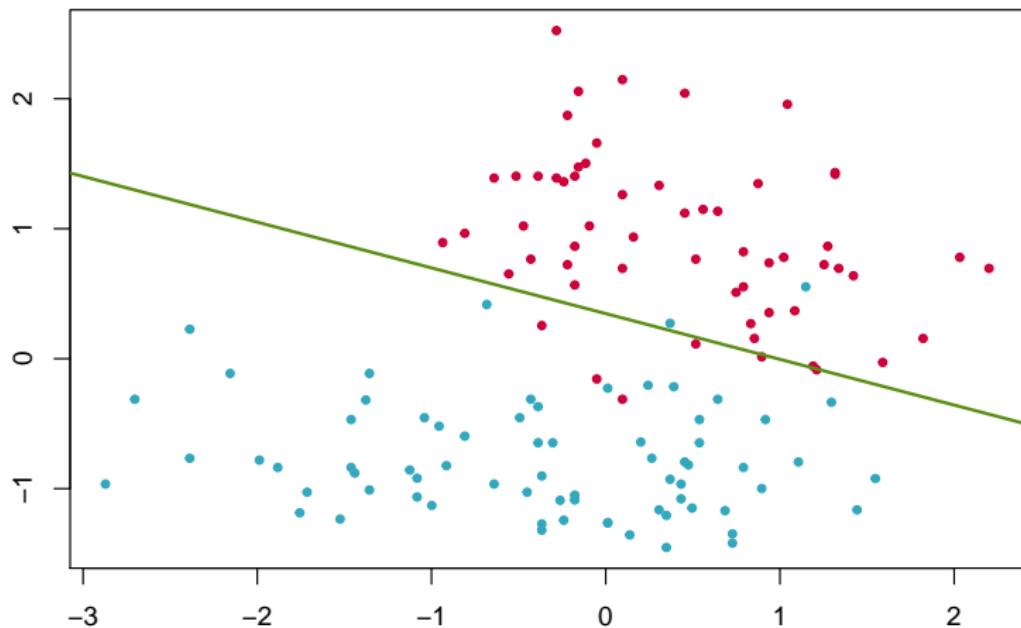
C=0.005, 18 erreurs





Exemple

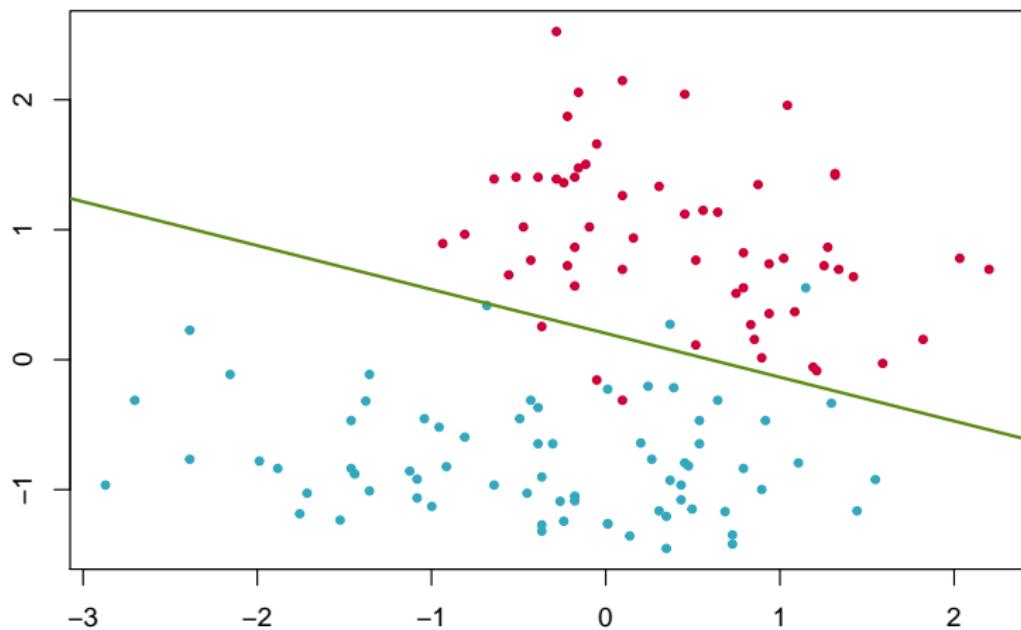
$C=0.0075, 8$ erreurs





Exemple

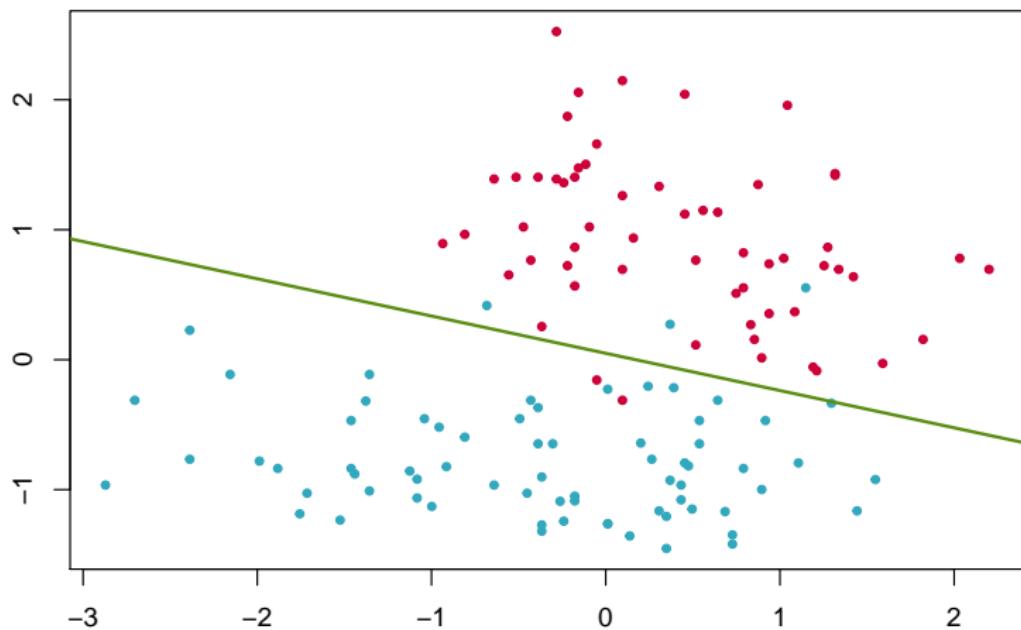
$C=0.01, 5$ erreurs





Exemple

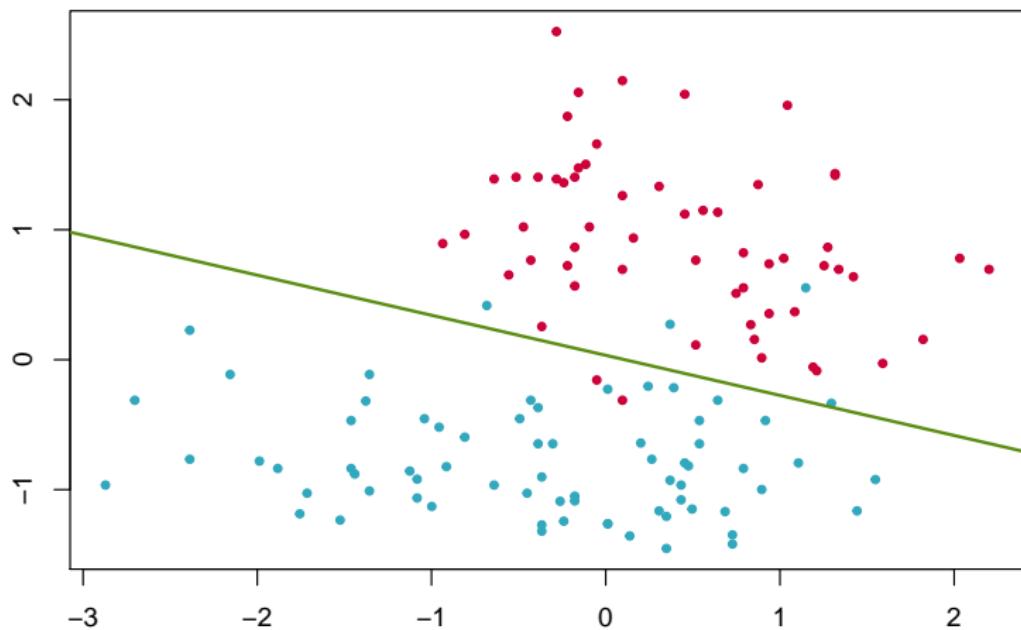
$C=0.1, 5$ erreurs





Exemple

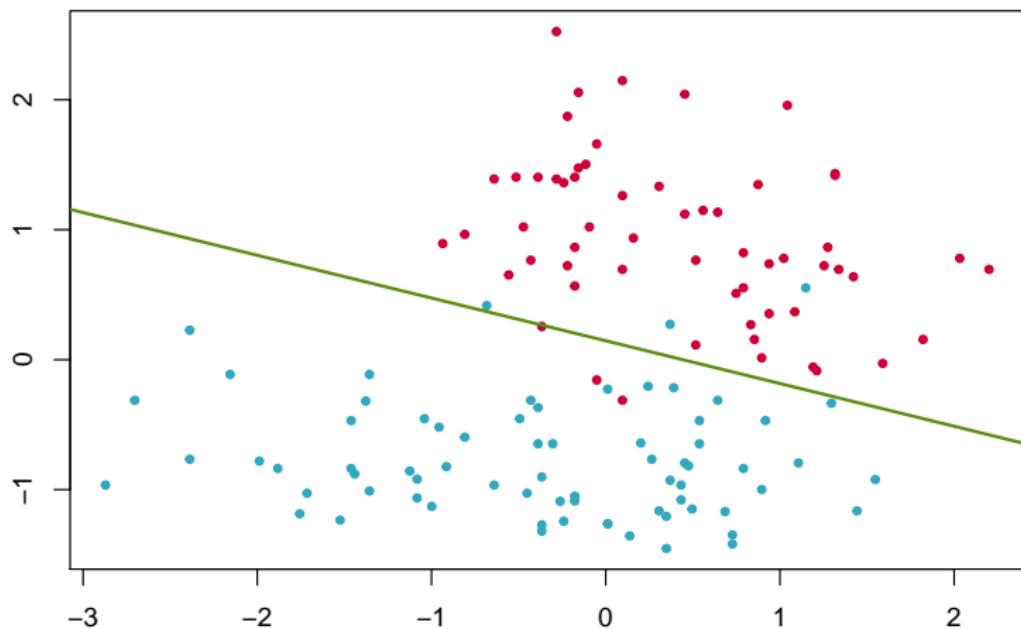
$C=1$, 6 erreurs





Exemple

$C=1e+06$, 6 erreurs





Résumé

- plusieurs choix pour la discrimination
- coût quadratique :
 - simple à mettre en œuvre
 - relativement rapide
 - assez peu adapté au cas multi-classes
 - résultats mitigés
- analyse discriminante :
 - meilleure justification que le coût quadratique
 - bien adapté au multi-classes
 - relativement rapide et simple
- machines à vecteurs de support :
 - solution robuste
 - extensions complexes au multi-classes
 - algorithme efficace mais sophistiqué
 - excellent résultats en pratique

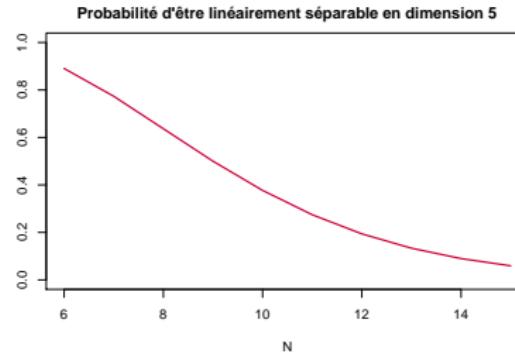
- régression :
 - fonction `lm` du package `stats`
 - nombreuses extensions associées
- analyse discriminante : fonction `lda` du package `MASS`
- machines à vecteurs de support :
 - fonction `svm` du package `e1071`
 - fonction `ksvm` du package `kernlab`



Linéarité et dimension

Résultats de Thomas Cover (1965)

- la « linéarité » d'un problème dépend de la dimension
- l'espérance du nombre maximum de points linéairement séparable en dimension p est $2p$
- l'espérance du nombre minimal de variables nécessaires pour séparer linéairement N point est $\frac{N+1}{2}$
- distribution de plus en plus « piquée » :

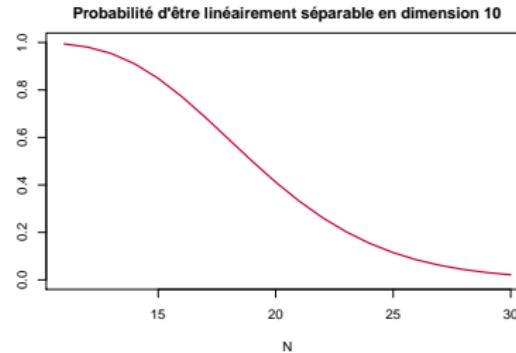




Linéarité et dimension

Résultats de Thomas Cover (1965)

- la « linéarité » d'un problème dépend de la dimension
- l'espérance du nombre maximum de points linéairement séparable en dimension p est $2p$
- l'espérance du nombre minimal de variables nécessaires pour séparer linéairement N point est $\frac{N+1}{2}$
- distribution de plus en plus « piquée » :

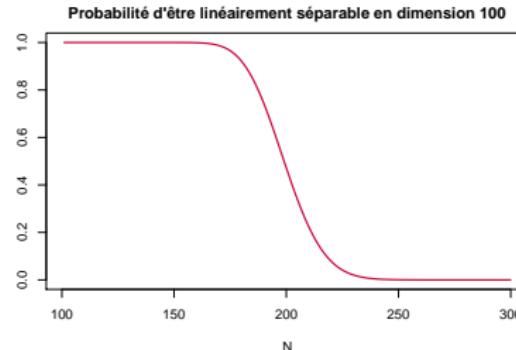




Linéarité et dimension

Résultats de Thomas Cover (1965)

- la « linéarité » d'un problème dépend de la dimension
- l'espérance du nombre maximum de points linéairement séparable en dimension p est $2p$
- l'espérance du nombre minimal de variables nécessaires pour séparer linéairement N point est $\frac{N+1}{2}$
- distribution de plus en plus « piquée » :

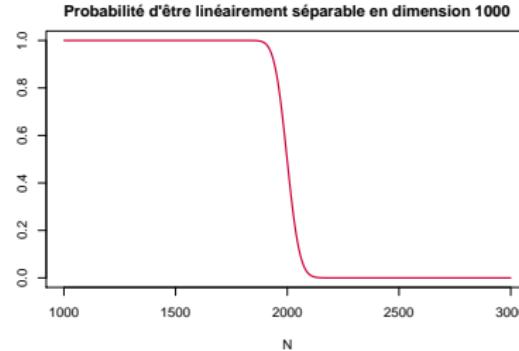




Linéarité et dimension

Résultats de Thomas Cover (1965)

- la « linéarité » d'un problème dépend de la dimension
- l'espérance du nombre maximum de points linéairement séparable en dimension p est $2p$
- l'espérance du nombre minimal de variables nécessaires pour séparer linéairement N point est $\frac{N+1}{2}$
- distribution de plus en plus « piquée » :





■ problèmes « simples » :

- $\frac{p}{N} \gg 2$: beaucoup de variables pour peu d'observations
- classifieur linéaire :
 - généralement une infinité de choix possibles
 - critère de choix très important
 - régularisation cruciale

■ problèmes « difficiles » :

- $\frac{N}{p} \ll 2$
- pas de séparateur linéaire
- peu de variables et/ou beaucoup d'observations
- données « contradictoires » (classes partiellement superposées)

■ la situation est la même qu'en régression



- même principe d'extension que pour la régression
- transformation explicite des variables
- transformation implicite :
 - passage par un noyau
 - régression ridge à noyau et erreur quadratique
 - analyse discriminante de Fisher à noyau
 - machines à vecteurs de support (MVS) à noyau :
 - la formulation duale fait apparaître les $\langle x_i, x_j \rangle$
 - il suffit de remplacer par un noyau pour obtenir une MVS non linéaire
- comme en régression, la difficulté est le choix du modèle

Introduction et modélisation mathématique

Apprentissage supervisé

Qualité d'un modèle

Régression

Régression linéaire

Régularisation

Non linéaire

Discrimination

Moindres carrés

Analyse discriminante

Maximisation de la marge

Non linéaire

Sélection de modèle



Le modèle parfait

- si les données d'apprentissage ne sont pas contradictoires, il existe un modèle parfait
- données contradictoires : $x_i = x_j$ et $y_i \neq y_j$
- modèle parfait :
 - algorithme des plus proches voisins
 - régression utilisant le noyau gaussien avec σ petit
 - etc.
- le modèle parfait n'a aucun intérêt car il colle au bruit :
 - apprentissage **par cœur**
 - sur-apprentissage
- principe du rasoir d'Occam : de deux modèles qui expliquent aussi bien un phénomène, on choisit le plus simple

Les multiples ne doivent pas être utilisés sans nécessité



Estimation des performances

- le problème fondamental est l'estimation de

$$L(g) = E_P\{I(g(x), y)\}$$

alors qu'on ne connaît pas P , la distribution des données

- la loi des grands nombres

$$\lim_{M \rightarrow \infty} \frac{1}{M} \sum_{i=1}^M I(g(x_i), y_i) = L(g)$$

nécessite des données indépendantes du modèle

- d'où la méthode de l'ensemble de validation :
 - on découpe les données observées en un ensemble d'apprentissage et un ensemble de validation
 - on construit de modèle sur l'apprentissage, on l'évalue sur la validation



■ Avantages :

- facile à mettre en œuvre
- temps de calcul raisonnable

■ Inconvénients :

- nécessite beaucoup de données :
 - au moins deux ensembles
 - si on veut évaluer un modèle sélectionné sur l'ensemble de validation, on doit utiliser un troisième ensemble : l'ensemble de **test**
- sensible au découpage
- réduit les données utilisées pour construire le modèle : résultats moins robustes



■ idée principale

- échanger les ensembles d'apprentissage et de validation
- apprendre un modèle sur $\mathcal{D} = (x_i, y_i)_{1 \leq i \leq N}$ et l'évaluer sur $\mathcal{D}' = (x_i, y_i)_{N+1 \leq i \leq N+M} \dots$
- puis apprendre un modèle sur \mathcal{D}' et l'évaluer sur $\mathcal{D} \dots$
- et enfin combiner les évaluations

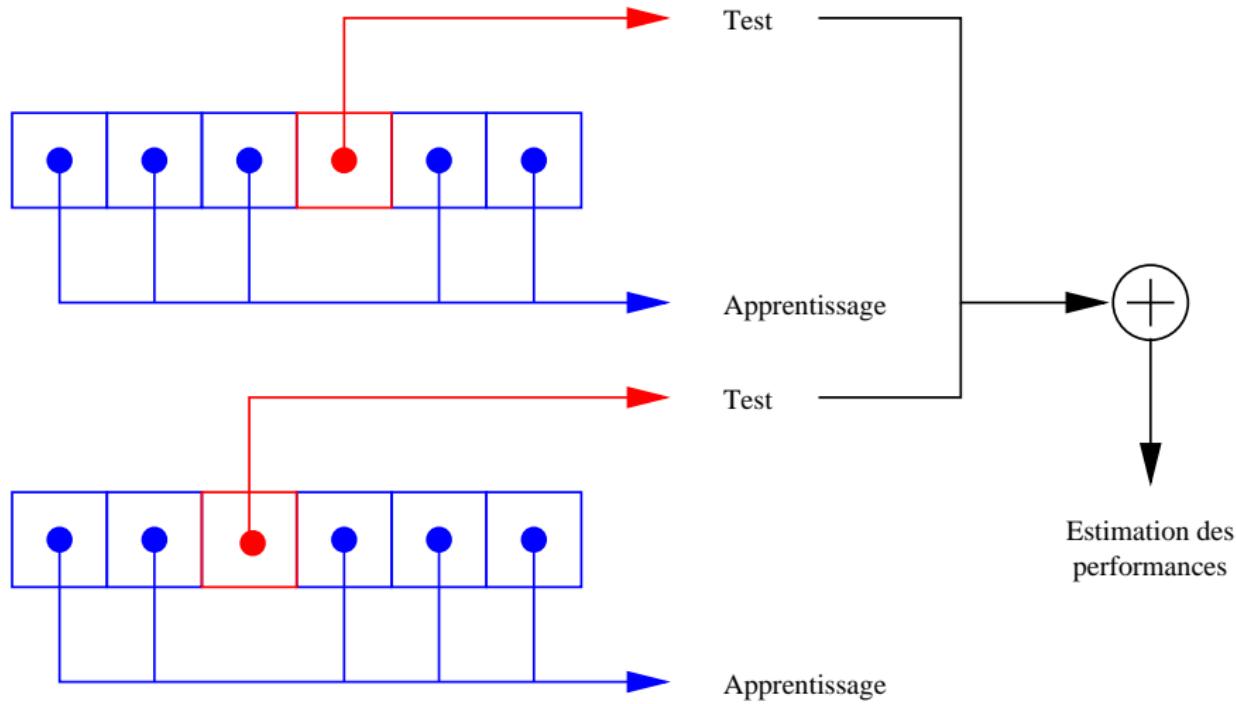
■ solution générale :

1. découpage des données en k sous-ensembles $\mathcal{D}_1, \dots, \mathcal{D}_n$
2. pour tout i :
 - 2.1 apprentissage sur l'union des \mathcal{D}_j avec $j \neq i$
 - 2.2 évaluation sur \mathcal{D}_i
3. combinaison des évaluations

■ si $k = N$ on parle de *leave-one-out*.



Validation croisée





- procédure détaillée :

- apprentissage sur $\bigcup_{j \neq i} \mathcal{D}_k \Rightarrow g_i$
- prédictions sur \mathcal{D}_i , $y_i^{(i)} = g_i(x_i)$ pour $x_i \in \mathcal{D}_i$
- donc pour tout $x_i \in \mathcal{D}$, on a une prédition $y_i^{(i)}$ (pour un certain i)
- évaluation : $\frac{1}{N} \sum_{i=1}^N \mathbb{I}_{\{y_i^{(i)} \neq y_i\}}$

- pas de classifieur unique !

- applications :

- évaluation de performances
- sélection de modèle :
 - évaluation des performances pour chaque configuration choisie (degré du polynôme, etc.)
 - choix de la meilleure configuration
 - construction d'un classifieur sur l'ensemble des données



Validation croisée

- avantages :

- facile à mettre en œuvre
- utilise toutes les données pour évaluer le modèle

- inconvénients :

- sensible au découpage et au nombre de blocs
- temps de calcul élevé
- ne donne pas directement un modèle

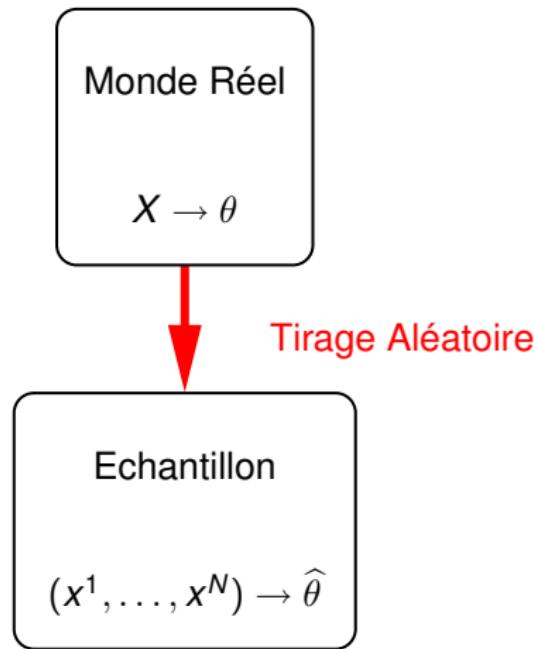
- solution la plus utilisée aujourd'hui

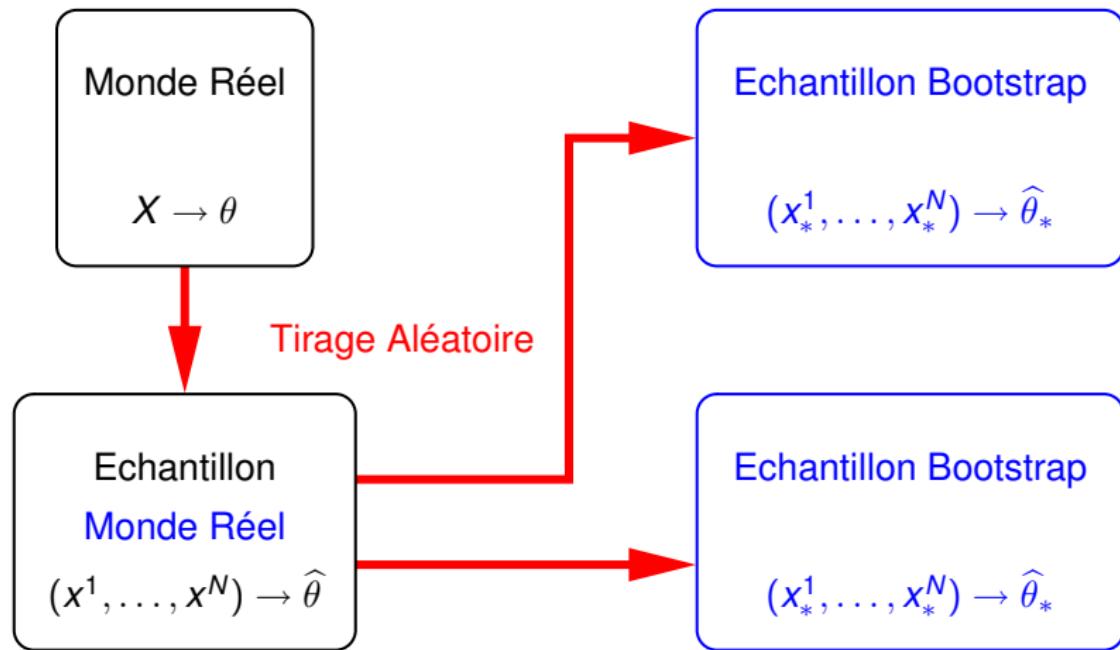
- ne dispense pas du découpage apprentissage/test



Méthode générale d'estimation de la qualité d'un estimateur, basée sur un ré-échantillonnage :

- on cherche à estimer θ , une statistique sur les observations (les x^i)
- on se donne $\hat{\theta}(x^1, \dots, x^N)$ un estimateur de θ
- on cherche à déterminer :
 - le biais de $\hat{\theta}$
 - la variance de $\hat{\theta}$
- solution :
 - fabriquer des échantillons *bootstrap*
 - un échantillon *bootstrap* : (x_*^1, \dots, x_*^N) obtenu par **tirage aléatoire uniforme avec remise** dans l'échantillon d'origine (x^1, \dots, x^N)
 - simule des nouveaux tirages pour les (x^1, \dots, x^N)







Estimation du biais

Algorithme :

1. pour b allant de 1 à n
 - 1.1 engendrer un échantillon bootstrap $(x_{*b}^1, \dots, x_{*b}^N)$
 - 1.2 calculer $\widehat{\theta}_{*b} = \widehat{\theta}(x_{*b}^1, \dots, x_{*b}^N)$
2. l'estimation du biais est

$$\frac{1}{n} \sum_{b=1}^n \widehat{\theta}_{*b} - \widehat{\theta}(x^1, \dots, x^N)$$

Idée, remplacer le monde réel par l'échantillon :

- le premier terme estime l'espérance de l'estimateur
- le second terme estime l'estimateur



Estimation de la variance

Algorithme :

1. pour b allant de 1 à n

 1.1 engendrer un échantillon bootstrap $(x_{*b}^1, \dots, x_{*b}^N)$

 1.2 calculer $\hat{\theta}_{*b} = \hat{\theta}(x_{*b}^1, \dots, x_{*b}^N)$

2. calculer

$$\hat{\theta}_* = \frac{1}{b} \sum_{b=1}^n \hat{\theta}_{*b}$$

3. l'estimation de la variance est

$$\frac{1}{n-1} \sum_{b=1}^n \left(\hat{\theta}_{*b} - \hat{\theta}_* \right)^2$$



Application à l'évaluation d'un modèle

Raisonnement :

- l'évaluation d'un modèle consiste à estimer ses performances
- l'erreur résiduelle sur l'ensemble d'apprentissage sous-estime l'erreur réelle
- idée, estimer l'ampleur de la sous-estimation par *bootstrap* :
 - calculer la sous-estimation pour un échantillon *bootstrap*
 - moyenner les sous-estimations pour beaucoup d'échantillons *bootstrap*
 - corriger l'erreur résiduelle en ajoutant la moyenne



Évaluation d'un modèle

Algorithme :

1. pour b allant de 1 à n
 - 1.1 engendrer un échantillon bootstrap $(x_{*b}^1, \dots, x_{*b}^N)$ (à partir de l'ensemble d'apprentissage)
 - 1.2 estimer le modèle optimal pour l'échantillon **bootstrap**
 - 1.3 calculer $\hat{\mathcal{B}}_{*b}$ comme la différence entre l'erreur résiduelle du modèle sur l'échantillon d'**apprentissage** et l'erreur résiduelle du modèle sur l'échantillon **bootstrap**
2. estimer l'erreur résiduelle $\hat{\mathcal{E}}$ du modèle optimal sur l'ensemble d'apprentissage
3. corriger cette erreur en lui ajoutant $\frac{1}{n} \sum_{b=1}^n \hat{\mathcal{B}}_{*b}$



Estimation **directe** de l'erreur du modèle optimal

- moyenne empirique de l'erreur commise sur l'ensemble d'apprentissage par le modèle construit sur l'échantillon *bootstrap* ($\hat{\mathcal{E}}_B$)
- moyenne empirique de l'erreur commise sur le complémentaire de l'échantillon *bootstrap* par le modèle construit sur l'échantillon (*bootstrap out-of-bag*, $\hat{\mathcal{E}}_{oob}$)
- *bootstrap 632* : combinaison de l'estimation *out-of-bag* et de l'estimation naïve (sur l'ensemble d'apprentissage)

$$\hat{\mathcal{E}}_{632} = 0.632 \hat{\mathcal{E}}_{oob} + 0.368 \hat{\mathcal{E}}$$

Probabilité qu'une observation de l'ensemble d'apprentissage soit dans un échantillon *bootstrap* : 0.632

■ Points positifs :

- facile à mettre en œuvre
- utilise toutes les données
- donne des intervalles de confiance

■ Points négatifs :

- temps de calcul très élevé
- nombreuses variantes
- ne donne pas directement un modèle

■ ne dispense pas du découpage apprentissage/test



- l'erreur empirique ne donne pas une bonne idée des performances en généralisation
- il faut **toujours** utiliser une méthode valide pour estimer les performances
- découpage et rééchantillonnage :
 - méthodes classiques et éprouvées
 - rééchantillonnage (validation croisée et *bootstrap*) : lent mais utilise toutes les données
 - validation (découpage) : rapide mais nécessite beaucoup de données