

# CES DATA scientist

## Apprentissage supervisé : théorie et algorithmes

**Joseph Salmon**

<http://josephsalmon.eu>

Télécom Paristech, Institut Mines-Télécom

# Motivation – enjeux

- ▶ Point de départ : diverses méthodes usuelles et retour sur quelques éléments de statistiques
- ▶ Compréhension de méthodes classiques de classification
- ▶ Implémenter celles-ci en Python

# Plan

## Prérequis

Optimisation / statistiques / Algèbre / Probabilités

## Cadre et notations

Contexte et pré-traitement

Modèle

Du cadre binaire au cadre multi-classe

## Quelques méthodes de classification

Prédiction linéaire et indicatrices

Analyse discriminante linéaire (LDA)

Analyse discriminante quadratique (QDA)

Bayésien Naïf

Régression logistique

K-nn

## Régularisation

Rappel : moindres carrés ordinaires

Régularisation avec  $\|\cdot\|_2^2$

Validation Croisée (CV)

Régularisation avec  $\|\cdot\|_1$

# Sommaire

## Prérequis

Optimisation / statistiques / Algèbre / Probabilités

## Cadre et notations

## Quelques méthodes de classification

## Régularisation

# Optimisation / statistiques

## Optimisation/Analyse

- ▶ Projection sur un sous-espace
- ▶ Régression linéaire, moindre carrés :

$$Y = X\theta + \varepsilon,$$

$$\hat{\theta} = (X^\top X)^{-1} X^\top Y \text{ (quand } X^\top X \text{ est inversible)}$$

$X = [x_1, \dots, x_n]^\top = [\mathbf{x}_1, \dots, \mathbf{x}_p]$  :  $n$  observations,  $p$  features

- ▶ Méthode de Newton

## Algèbre

- ▶ Décomposition spectrale des matrices symétriques (carrées) :

$$\Sigma = UDU^\top \in \mathbb{R}^{p \times p}$$

pour une matrice orthonormale  $U$  (i.e.,  $U^\top U = \text{Id}_p$ ) et une matrice diagonale  $D = \text{diag}(d_1, \dots, d_p)$

# Probabilités

- ▶ Notations :  $\mathbb{P}$  : probabilité,  $\mathbb{E}$  : espérance
- ▶ Les lois gaussiennes en dimension  $p$  quelconque ont comme densité :

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} \sqrt{|\Sigma|}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\} .$$

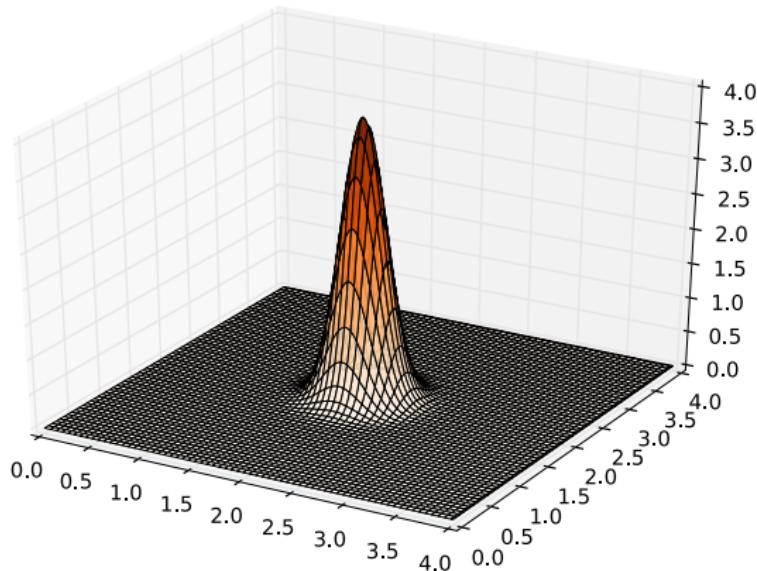
qui est gouvernée par deux paramètres :

- ▶ le vecteur d'**espérance**  $\boldsymbol{\mu} \in \mathbb{R}^p$
- ▶ la matrice de **covariance**  $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$  (symétrique)

Rem:  $|\Sigma| = \det(\Sigma)$  est le produit des valeurs propres de  $\Sigma$

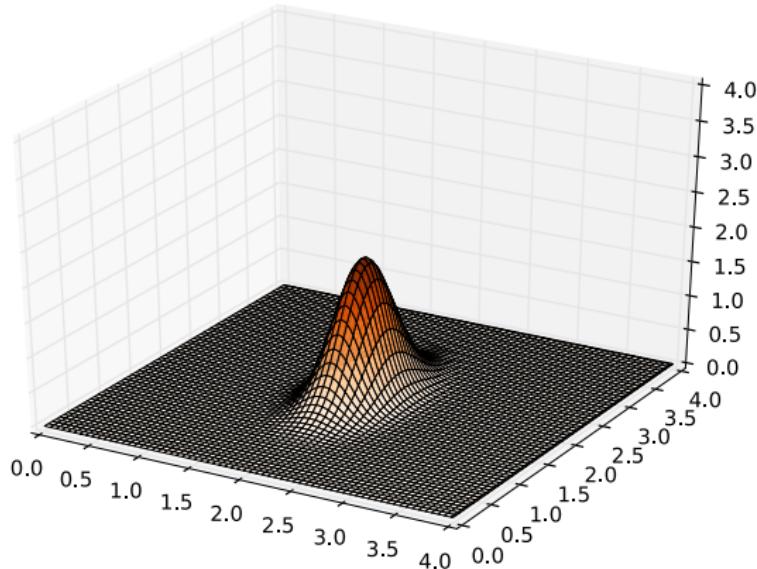
# Lois gaussiennes

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} : \text{cas isotrope}$$



# Lois gaussiennes

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} : \text{cas anisotrope}$$



# Sommaire

Prérequis

Cadre et notations

Contexte et pré-traitement

Modèle

Du cadre binaire au cadre multi-classe

Quelques méthodes de classification

Régularisation

# La classification : cadre binaire

Diagnostiquer des patients :



# La classification : cadre binaire

Diagnostiquer des patients : malades



# La classification : cadre binaire

Diagnostiquer des patients : malades sains



# La classification : cadre binaire

Classer des emails :



# La classification : cadre binaire

Classer des emails : pourriels (spams)



# La classification : cadre binaire

Classer des emails : pourriels (spams) normaux



# La classification : cadre binaire

Classer des clients :



# La classification : cadre binaire

Classer des clients : mauvais payeurs/fraudeurs



## La classification : cadre binaire

Classer des clients : ~~mauvais payeurs/fraudeurs~~ bon payeurs



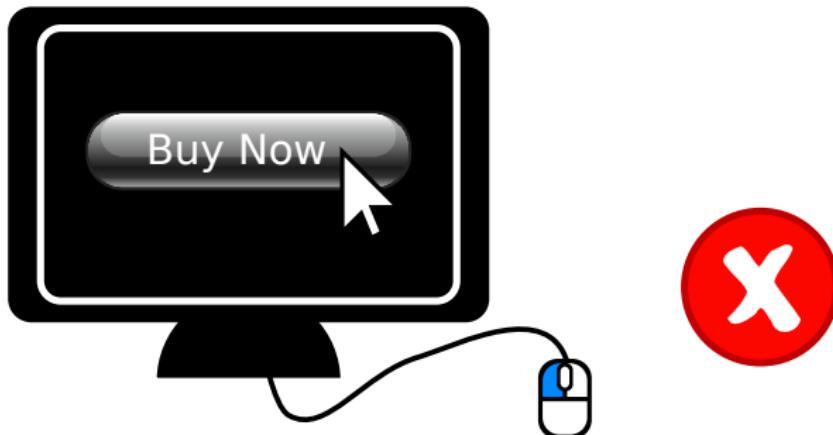
# La classification : cadre binaire

Classer les surfeurs :



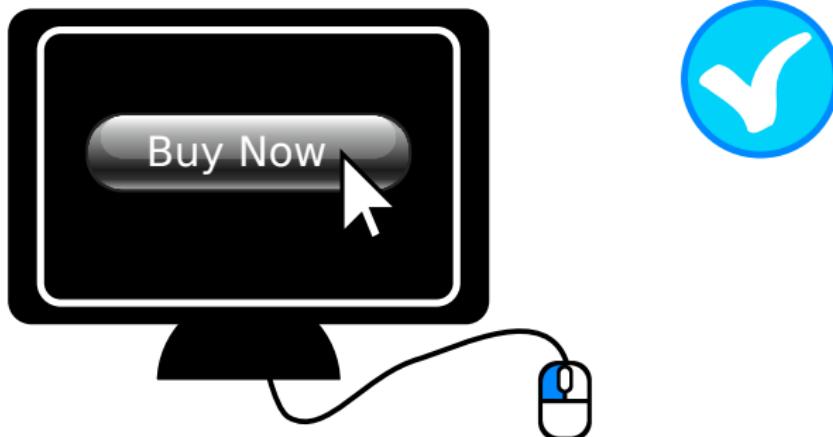
# La classification : cadre binaire

Classer les surfeurs : futurs acheteurs



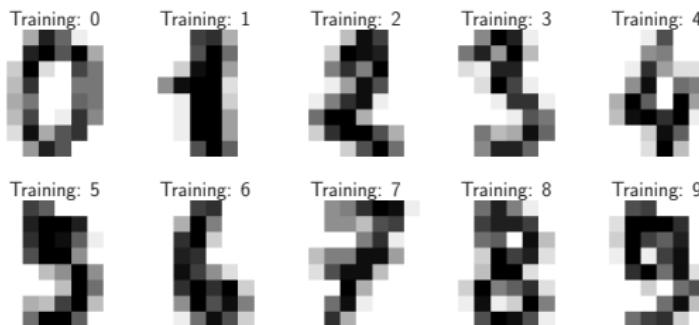
# La classification : cadre binaire

Classer les surfeurs : futurs acheteurs ou pas...



# La classification : cadre multi-classe

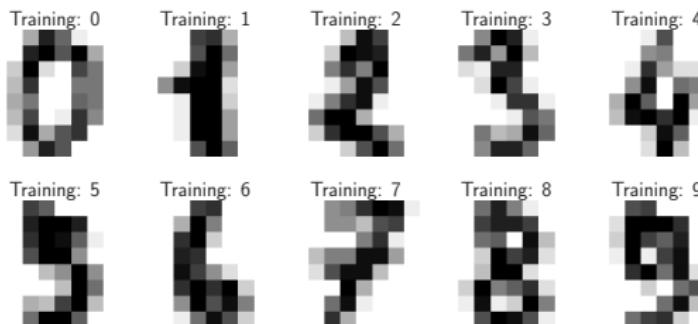
- ▶ Classer des chiffres numérisés (e.g., codes postaux de courriers 80's/90's)



- ▶ Classer des objets dans des images  
(<http://image-net.org/>, 2010's)

# La classification : cadre multi-classe

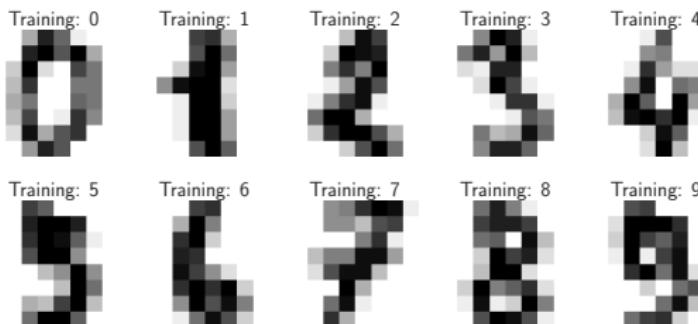
- ▶ Classer des chiffres numérisés (e.g., codes postaux de courriers 80's/90's)



- ▶ Classer des objets dans des images  
(<http://image-net.org/>, 2010's)
- ▶ Classer des textes par thème (e.g., RCV20)

# La classification : cadre multi-classe

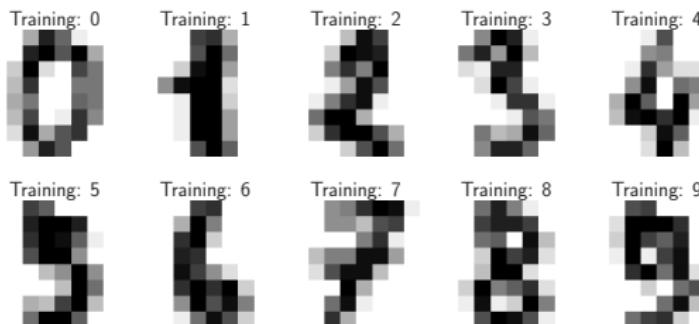
- ▶ Classer des chiffres numérisés (e.g., codes postaux de courriers 80's/90's)



- ▶ Classer des objets dans des images  
(<http://image-net.org/>, 2010's)
- ▶ Classer des textes par thème (e.g., RCV20)
- ▶ Classer des espèces animales/végétales (e.g., iris)

# La classification : cadre multi-classe

- ▶ Classer des chiffres numérisés (e.g., codes postaux de courriers 80's/90's)



- ▶ Classer des objets dans des images  
(<http://image-net.org/>, 2010's)
- ▶ Classer des textes par thème (e.g., RCV20)
- ▶ Classer des espèces animales/végétales (e.g., iris)

## Exemples de variables explicatives

- ▶ cadre médical : âge, CPS, patrimoine génétique, examens, patrimoine génétique, antécédent, etc.
- ▶ cadre de la détection de pourriels : langue, niveau de langages, mots-clefs (e.g., discount, Nigeria, etc.)
- ▶ cadre de la publicité en ligne : historique de navigation, cookie, site web, âge, etc.

# Numérisation des variables explicatives

Toutes les variables ne peuvent pas être utilisées telles quelles, il faut souvent un **pré-traitement**.

On parle de variable qualitative, quand une variable ne prend que des modalités discrètes et/ou non-numériques.

- ▶ Pour des variables continues pas de soucis (e.g., âge, températures, distances, etc.)
- ▶ on peut coder chaque modalité par un nombre, mais il faut se méfier car la proximité ne veut alors rien dire...

**Exemple:** (“parfums”) : en codant le parfum vanille par 0, le parfum fraise par 1 et chocolat par 2, on décrète implicitement que fraise est plus proche de vanille que de chocolat...

Plus d'informations :

<http://fastml.com/converting-categorical-data-into-numbers-with-pandas-and-scikit-learn/>

# Variables qualitatives

**Exemple:** couleurs, genres, villes, parfums, etc.

Encodage classique : variables fictives/indicatrices *dummy variables* aussi appelé “encodage à chaud” *one-hot encoder*.

Si la variable  $x$  peut prendre  $K$  modalités  $a_1, \dots, a_K$  on créer les  $K$  variables explicatives suivantes :  $\forall k \in \llbracket 1, K \rrbracket, \mathbf{1}_{a_k} \in \mathbb{R}^n$  définies par

$$\forall i \in \llbracket 1, n \rrbracket, \quad (\mathbf{1}_{a_k})_i = \begin{cases} 1, & \text{if } x_i = a_k \\ 0, & \text{sinon} \end{cases}$$

# Exemple d'encodage

Cas binaire : M/F, oui/non, j'aime/j'aime pas.

Client	Genre
1	H
2	F
3	H
4	F
5	F



$$\begin{pmatrix} F & H \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$$

Cas général : couleurs, villes, etc.

Client	Couleurs
1	Bleu
2	Blanc
3	Rouge
4	Rouge
5	Bleu



$$\begin{pmatrix} \text{Bleu} & \text{Blanc} & \text{Rouge} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

# “Feature engineering”

Création artificielle de variables explicatives :

- ▶ transformation simple de variables : terme de puissances de degré 2, 3, 4, . . . , log, exp etc.
- ▶ variables de types temporelles : cosinus/sinus avec des périodes adaptées
- ▶ interactions d'ordre supérieurs : par exemple les gens blonds avec des yeux bleus (catégories venant de deux variables explicatives couleur yeux / couleur cheveux)

Plus d'informations sur cette partie pré-traitement :

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

# Données manquantes

Dans certains cas il se peut que certaines variables explicatives ne soient pas renseignées. On parle alors de données manquantes.

La première chose à faire est de les annoter, en renseignant le fait qu'une donnée est manquante (e.g., en pandas: nan / na pour "not a number/not available").

**Exemple:**

```
>pandas.read_csv(file_name.csv, na_values='`-''')
```

Client	Age
1	19
2	-
3	26
4	18
5	43



Client	Age
1	19
2	NaN
3	26
4	18
5	43

# Traitement des données manquantes

- ▶ les enlever
- ▶ remplacer par la moyenne / la médiane
- ▶ remplacer par le mode (valeur la plus fréquente) pour le cas qualitatif
- ▶ se tourner vers méthodes robustes qui peuvent traiter un pourcentage d'erreur faible (moyennes tronquées)
- ▶ etc.

**Exemple:** données qualitatives

Client	Couleurs
1	Bleu
2	-
3	Rouge
4	Rouge
5	-



Client	Couleurs
1	Bleu
2	???
3	Rouge
4	Rouge
5	???

# Sommaire

Prérequis

**Cadre et notations**

Contexte et pré-traitement

**Modèle**

Du cadre binaire au cadre multi-classe

Quelques méthodes de classification

Régularisation

# Notations pour la classification multi-classes

Classes (en : *label*) :  $\mathcal{Y} = \{0, 1, \dots, K - 1\}$  ( $K$  classes)

Espace des caractéristiques (en : *features*) :  $\mathcal{X} \subset \mathbb{R}^p$

Observations : l'utilisateur reçoit un ensemble d'apprentissage composé de  $n$  couples  $(X_i, Y_i) \sim (X, Y)$  supposés *i.i.d.*, où

- ▶ les  $X_i \in \mathcal{X}$  correspondent aux **variables explicatives**
- ▶  $Y_i \in \mathcal{Y}$  correspondent aux **étiquettes**.

Formalisme vectoriel :

- ▶ Vecteur des étiquettes :  $\mathbf{Y} = (Y_1, \dots, Y_n)^\top \in \mathbb{R}^n$
- ▶ Matrice des caractéristiques/attributs :  
 $\mathbf{X} = (X_1, \dots, X_n)^\top \in \mathbb{R}^{n \times p}$

**Exemple**: pour le modèle binaire ( $K = 2$ ), parfois utile de prendre  $\mathcal{Y} = \{-1, 1\}$  (notamment quand on travaille avec des hyperplans)

# Classification supervisée

## Objectif

Être capable pour un nouvel élément  $X_{n+1} \in \mathcal{X}$  d'estimer sa classe  $Y_{n+1}$  par une quantité  $\hat{Y}_{n+1} \in \mathcal{Y}$

# Sommaire

Prérequis

Cadre et notations

Contexte et pré-traitement

Modèle

Du cadre binaire au cadre multi-classe

Quelques méthodes de classification

Régularisation

## De deux à plusieurs classes

On peut passer du cadre binaire au multi-classe pour toute méthode, e.g., il suffit de tester :

- ▶ “un contre tous” (en : **One-vs.-all**) : créer un classifieur par classe, et produire un score (par exemple une probabilité). Choisir alors la classe avec le score maximum.
- ▶ “un contre un” (en : **One-vs.-one**) : on calcule un classifieur pour toutes les  $K(K - 1)/2$  paires. Pour la prédiction on calcule tous les choix possibles et l'on prend la classe qui a reçu le plus de votes.

# Outils de diagnostique

## Autre type d'information : "Matrice de confusion"

Estimer les quantités  $\mathbb{P}(\hat{Y}_{n+1} = k | Y = k')$  pour tout  $k, k'$   
c'est-à-dire que l'on veut estimer pour toutes les classes la  
probabilités d'estimer une autre classe.

Cela permet de dépister les erreurs courantes (e.g., les chiffres 7 et  
1 peuvent être très souvent confondus, 1 et 8 rarement)

Voir aussi : Courbe ROC, AUC, F1, etc.

Rem: Il y a de nombreuses méthodes de mesure d'erreurs qui  
peuvent être utiles dans divers contextes.

Leur énumération est donnée par exemple ici :

[http://scikit-learn.org/stable/modules/model\\_evaluation.html](http://scikit-learn.org/stable/modules/model_evaluation.html)

# Sommaire

Prérequis

Cadre et notations

Quelques méthodes de classification

Prédiction linéaire et indicatrices

Analyse discriminante linéaire (LDA)

Analyse discriminante quadratique (QDA)

Bayésien Naïf

Régression logistique

K-nn

Régularisation

# Prédiction linéaire et indicatrices

Idée naïve : utiliser un outils de prédiction linéaire pour faire de la classification (ne jamais faire ça après aujourd'hui)

Simple, mais ne marche pas, on va le voir quand même pour s'en convaincre, faire des premières manipulations

# Prédiction linéaire et indicatrices

Idée simple : utiliser une méthode de régression pour estimer  $\mathbb{P}(Y = 0|X = x), \mathbb{P}(Y = 1|X = x), \dots, \mathbb{P}(Y = K - 1|X = x)$  et choisir la classe qui donne le plus grande probabilité.

Rem:  $\mathbb{P}(Y_{n+1} = k|X = X_{n+1}) = \mathbb{E}(Z^{(k)}|X = X_{n+1})$

Solution possible : résoudre  $K$  problèmes de régression,  $k = 0, \dots, K - 1$  on définit  $Z^{(k)} \in \mathbb{R}^n$  le vecteur de coordonnées

$$Z_i^{(k)} = \mathbb{1}_{Y_i=k} = \begin{cases} 1 & \text{si } Y_i = k, \\ 0 & \text{sinon.} \end{cases}$$

Estimateur des moindres carrés

$$\boxed{\theta^{(k)} = \arg \min_{\theta \in \mathbb{R}^p} \|\mathbf{X}\theta - Z^{(k)}\|^2}$$

Prédiction (conditionnelle)  $Y_{n+1}^{(k)} = X_{n+1}\theta^{(k)}$

Le classifieur final

$$\hat{Y}_{n+1} = \arg \max_{k \in \{0, \dots, K-1\}} Y_{n+1}^{(k)} \quad (\text{ex-aequo départagés au hasard})$$

---

**Exo:** Retrouver la formule de l'estimateur des moindres carrés

# Prédiction linéaire et indicatrice

Seconde interprétation :

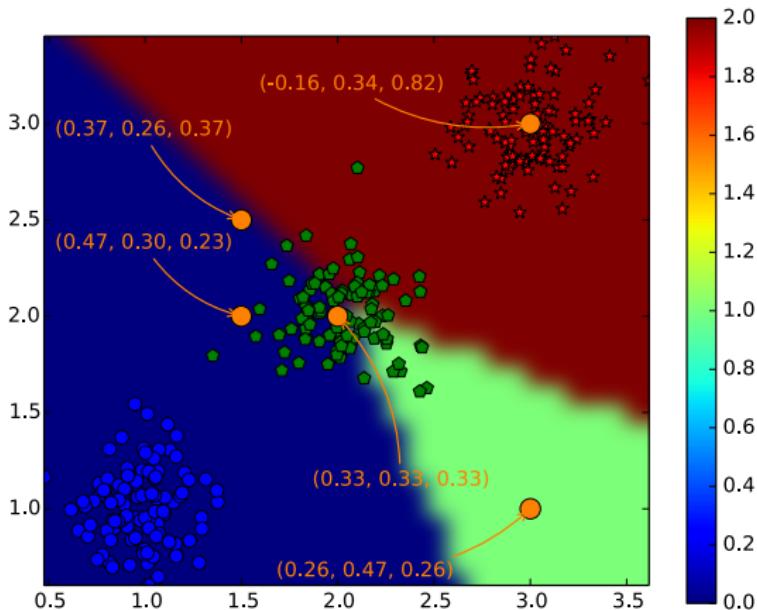
$$e_k = (0, \dots, 0, 1, 0, \dots, 0)^\top \in \mathbb{R}^K \text{ (1 en } k^{\text{e}} \text{ place)}$$

$$\arg \min_{M \in \mathbb{R}^{K \times p}} \left( \sum_{i=1}^n \|e_{Y_i} - X_i^\top M\|^2 \right)$$

puis

$$\hat{Y}_{n+1} = \arg \min_{k \in \{0, \dots, K-1\}} \|e_k - X_{n+1}^\top M\|^2$$

# Un exemple : prédition linéaire et indicatrices



Données avec trois classes : les triplets indiquent les probabilités estimées des classes 0, 1 et 2 aux points indiqués

# Avantages / Inconvénients : prédition linéaire et indicatrice

## Avantages

- ▶ Simple : sans hypothèse de modèle (encore que)
- ▶ Implémentable facilement avec un solveur de moindres-carrés
- ▶  $\sum_{k=0}^{K-1} Y_{n+1}^{(k)} = 1$  si la matrice des caractéristiques contient la variable constante (*i.e.*, une colonne de 1)

**Exo:** Prouver ce point (utiliser la projection sur les colonnes)

## Inconvénients

- ▶ les estimations  $Y_{n+1}^{(k)}$  de  $\mathbb{E}(Z^{(k)}|X = X_{n+1})$  n'ont pas de raison d'être positives, et peuvent donc ne pas l'être !
- ▶ Solution :  $\theta^{(k)} = \arg \min_{\theta \in \mathbb{R}^p} \| \mathbf{X}\theta - Z^{(k)} \|^2$  ? ne résout pas le pb, pour un nouveau point la prédition peut être négative
- ▶ effet masque

**TP:** Faire la partie TP associée

# Sommaire

Prérequis

Cadre et notations

Quelques méthodes de classification

Prédiction linéaire et indicatrices

Analyse discriminante linéaire (LDA)

Analyse discriminante quadratique (QDA)

Bayésien Naïf

Régression logistique

K-nn

Régularisation

# Analyse discriminante linéaire (I)

## Hypothèse de mélange gaussien

Pour tout  $k$ , la loi conditionnelle de  $X$  sachant  $Y = k$  est gaussienne  $\mathcal{N}_p(\mu_k, \Sigma_k)$  (on note  $f_k$  leur densités respectives)

- ▶ Vecteurs des centres de classes :  $\mu_k \in \mathbb{R}^p$
- ▶ Matrices de covariance :  $\Sigma_k$  sont symétriques de taille  $p \times p$
- ▶ Probabilités de la classe  $k$  :  $\pi_k = \mathbb{P}\{Y = k\}$

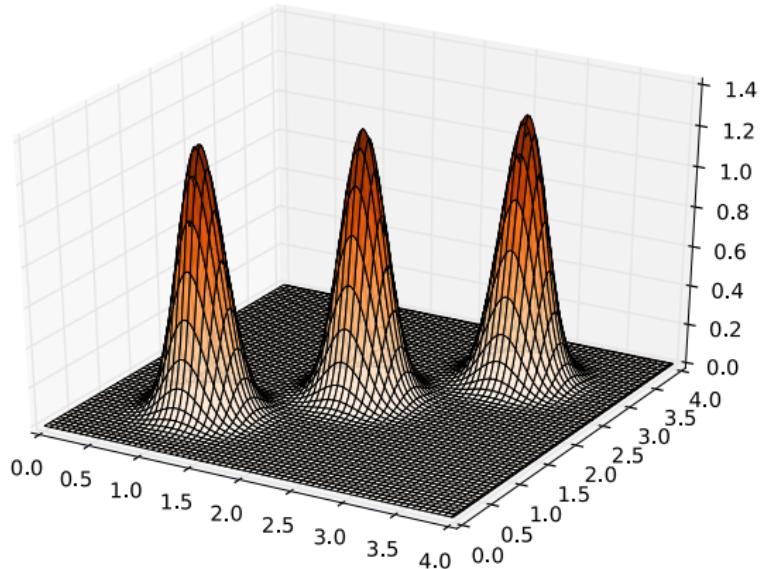
Mélanges : on tire avec probabilité  $\pi_k$  une étiquette  $Y = k$ , qui indique si  $X$  est tiré selon la loi  $f_k$ . La densité du mélange est donc

$$f(\mathbf{x}) = \sum_{k=0}^{K-1} \pi_k f_k(\mathbf{x})$$

où la densité  $p$ -dimensionnelle de la loi  $\mathcal{N}_p(\mu_k, \Sigma_k)$  est

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} \sqrt{|\Sigma_k|}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu_k)^\top \Sigma_k^{-1} (\mathbf{x} - \mu_k) \right\} .$$

## Exemple de mélange ( $K = 3, p = 2$ )



## Analyse discriminante linéaire (II)

La formule de Bayes donne :

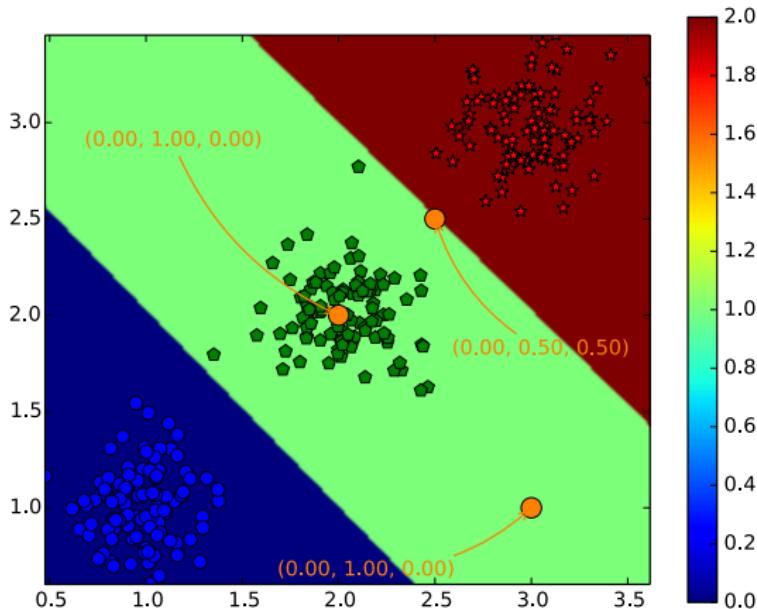
$$\mathbb{P}(Y = k | X = \mathbf{x}) = \frac{\pi_k f_k(\mathbf{x})}{\sum_{k'=0}^{K-1} \pi_{k'} f_{k'}(\mathbf{x})}$$

Rapport de log-vraisemblance sous l'hypothèse de **covariances identiques**  $\Sigma_k = \Sigma, \quad \forall k \in \{0, \dots, K-1\}$

$$\begin{aligned} \log \left( \frac{\mathbb{P}(Y = k | X = \mathbf{x})}{\mathbb{P}(Y = l | X = \mathbf{x})} \right) &= \mathbf{x}^\top \Sigma^{-1} (\mu_k - \mu_l) + \frac{1}{2} (\mu_l^\top \Sigma^{-1} \mu_l - \mu_k^\top \Sigma^{-1} \mu_k) \\ &\quad + \log \left( \frac{\pi_k}{\pi_l} \right) \end{aligned}$$

Les séparatrices sont **linéaires** (affines) : on affecte  $\mathbf{x}$  à la classe  $k$  si  $\mathbf{x}^\top \Sigma^{-1} (\mu_l - \mu_k) + \frac{1}{2} (\mu_k^\top \Sigma^{-1} \mu_k - \mu_l^\top \Sigma^{-1} \mu_l) + \log \left( \frac{\pi_k}{\pi_l} \right) > 0$

# Un exemple : LDA



Données avec trois classes : les triplets indiquent les probabilités estimées des classes 0, 1 et 2 aux points indiqués

## Analyse discriminante linéaire (II)

Classifieur par LDA pour un nouveau point  $X_{n+1}$

$$\hat{Y}_{n+1}^{\text{LDA}} = \arg \max_{k \in \{0, \dots, K-1\}} \left( -X_{n+1}^\top \Sigma^{-1} \mu_k + \frac{1}{2} (\mu_k^\top \Sigma^{-1} \mu_k) + \log(\pi_k) \right)$$

En pratique, remplacer les quantités théoriques (inconnues)  $\pi_k, \mu_k$  et  $\Sigma$  par les contreparties empiriques :

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{Y_i=k\}} \text{ et } n_k = \sum_{i=1}^n \mathbb{1}_{\{Y_i=k\}}$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i=1}^n X_i \mathbb{1}_{\{Y_i=k\}}$$

$$\hat{\Sigma} = \frac{1}{n} \sum_{k=0}^{K-1} n_k \times \frac{1}{n_k} \sum_{i=1}^n \mathbb{1}_{\{Y_i=k\}} (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^\top$$

Rem: noter que le fait que  $\hat{\Sigma}$  soit inversible n'est en rien garantie !

## Autre interprétation

$$\hat{Y}_{n+1}^{\text{LDA}} = \arg \min_{k \in \{0, \dots, K-1\}} \frac{1}{2} (X_{n+1} - \hat{\mu}_k)^\top \hat{\Sigma}^{-1} (X_{n+1} - \hat{\mu}_k) - \log(\hat{\pi}_k)$$

### Interprétation

Centrer et réduire les donnés ("blanchir") :

$$\begin{aligned}(X_{n+1} - \hat{\mu}_k)^\top \hat{\Sigma}^{-1} (X_{n+1} - \hat{\mu}_k) &= \|\hat{\Sigma}^{-1/2} (X_{n+1} - \hat{\mu}_k)\|^2 \\ &= \|\tilde{X}_{n+1} - \tilde{\mu}_k\|^2\end{aligned}$$

Avec la décomposition spectrale  $\hat{\Sigma} = UDU^\top$  et

$$\tilde{X}_{n+1} = D^{-1/2} U^\top \hat{X}_{n+1}, \quad \tilde{\mu}_k = D^{-1/2} U^\top \hat{\mu}_k$$

Rem: Si  $\text{Var}(\mathbf{x}) = \hat{\Sigma}$  alors

$$\text{Var}(D^{-1/2} U^\top \mathbf{x}) = D^{-1/2} U^\top \hat{\Sigma} (D^{-1/2} U^\top)^\top = \text{Id}_p,$$

# Avantages / Inconvénients : LDA

## Avantages

- ▶ optimal pour (certaines) gaussiennes
- ▶ pas d'effet masque

## Inconvénients

- ▶ Inversion de la covariance : besoin éventuellement de régulariser  $\hat{\Sigma}$  (quand l'inversion est impossible)
- ▶ Les points très loin des frontières ont tous la même influence
- ▶ Robustesse aux hypothèses gaussiennes...

**TP:** Faire la partie TP associée

# Sommaire

Prérequis

Cadre et notations

## Quelques méthodes de classification

Prédiction linéaire et indicatrices

Analyse discriminante linéaire (LDA)

Analyse discriminante quadratique (QDA)

Bayésien Naïf

Régression logistique

K-nn

Régularisation

# Analyse discriminante quadratique (I)

## Hypothèse provisoire

Pour tout  $k$ , la loi conditionnelle de  $X$  sachant  $Y = k$  est gaussienne  $\mathcal{N}_p(\mu_k, \Sigma_k)$  (on note  $f_k$  leur densités respectives)

Hypothèse identique que sous le cadre LDA :

$$f(\mathbf{x}) = \sum_{k=0}^{K-1} \pi_k f_k(\mathbf{x})$$

Cette fois on ne fait plus l'hypothèse :  $\forall k = 0, \dots, K-1, \hat{\Sigma}_k = \Sigma$

## Analyse discriminante quadratique (II)

Rapport de log-vraisemblance : sans

$$\Sigma_k^{-1} = \Sigma^{-1}, \quad \forall k \in \{0, \dots, K-1\}$$

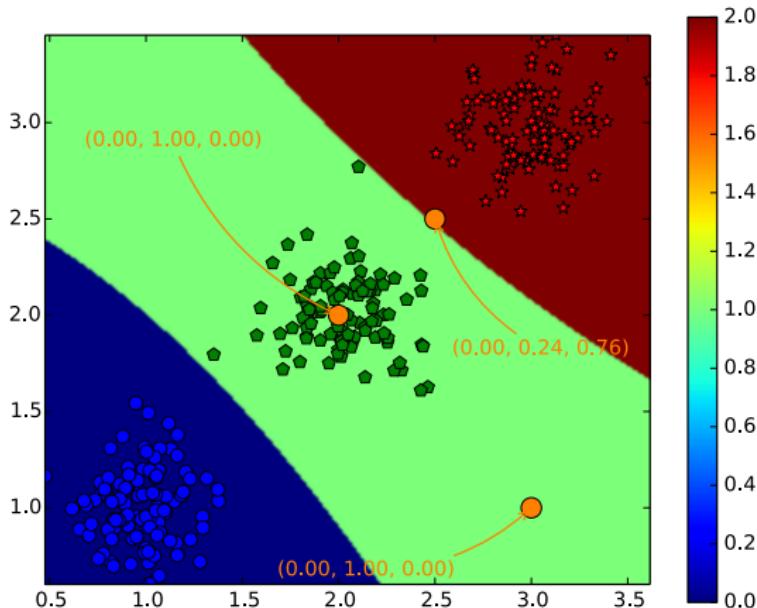
$$\begin{aligned} \log \left( \frac{\mathbb{P}(Y=k|X=\mathbf{x})}{\mathbb{P}(Y=l|X=\mathbf{x})} \right) &= \mathbf{x}^\top (\Sigma_k^{-1} \boldsymbol{\mu}_k - \Sigma_l^{-1} \boldsymbol{\mu}_l) + \\ &\quad \frac{1}{2} (\boldsymbol{\mu}_l^\top \Sigma_l^{-1} \boldsymbol{\mu}_l - \boldsymbol{\mu}_k^\top \Sigma_k^{-1} \boldsymbol{\mu}_k) \\ &\quad + \log \left( \frac{\pi_k}{\pi_l} \right) \\ &\quad - \boxed{\log(|\Sigma_k|/|\Sigma_l|) + \mathbf{x}^\top (\Sigma_k^{-1} - \Sigma_l^{-1}) \mathbf{x}} \end{aligned}$$

Nouvelle règle :

$$\begin{aligned} \hat{Y}_{n+1}^{\text{QDA}} &= \\ \arg \min_{k \in \{0, \dots, K-1\}} & \left[ \frac{(X_{n+1} - \hat{\boldsymbol{\mu}}_k)^\top \hat{\Sigma}_k^{-1} (X_{n+1} - \hat{\boldsymbol{\mu}}_k)}{2} - \log(\hat{\pi}_k) + \log(|\hat{\Sigma}_k|) \right] \end{aligned}$$

Rem: les séparatrices sont **quadratiques**

# Un exemple : QDA



Données avec trois classes : les triplets indiquent les probabilités estimées des classes 0, 1 et 2 aux points indiqués

# Avantages / Inconvénients : QDA

## Avantages

- ▶ Modèle plus riche/flexible (complexité plus grande)

## Inconvénients

- ▶ plus lourd à calculer
- ▶ Les séparatrices ne sont plus linéaires

**TP:** Faire la partie TP associé

# Sommaire

Prérequis

Cadre et notations

## Quelques méthodes de classification

Prédiction linéaire et indicatrices

Analyse discriminante linéaire (LDA)

Analyse discriminante quadratique (QDA)

**Bayésien Naïf**

Régression logistique

K-nn

Régularisation

# Bayésien Naïf gaussien (I)

Retour sur le Bayésien Naïf gaussien :

$$\mathbb{P}(Y = k | X = \mathbf{x}) = \frac{\pi_k f_k(\mathbf{x})}{\sum_{k'=0}^{K-1} \pi_{k'} f_{k'}(\mathbf{x})}$$

Supposons que cette fois les densité de chaque classe sont indépendantes sur toutes les dimensions :

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} \sqrt{|\Sigma_k|}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu_k)^\top \Sigma_k^{-1} (\mathbf{x} - \mu_k) \right\} .$$

avec  $\Sigma_k = \begin{bmatrix} \sigma_{1,k}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{2,k}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{p,k}^2 \end{bmatrix}$

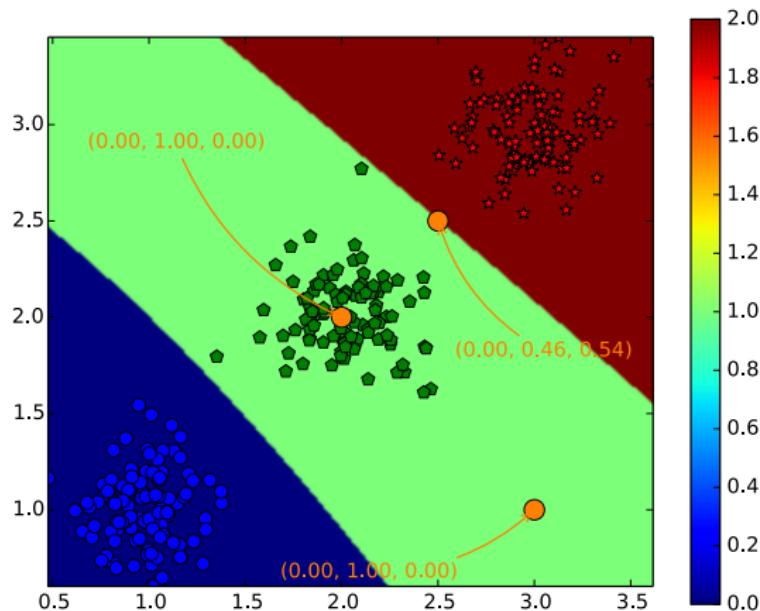
# Règle naïve de Bayes (gaussien)

Nouvelle règle :

$$\hat{Y}_{n+1}^{\text{NB}} = \arg \min_{k \in \{0, \dots, K-1\}} \left[ \frac{(X_{n+1} - \hat{\mu}_k)^\top \Sigma_k^{-1} (X_{n+1} - \hat{\mu}_k)}{2} - \log(\hat{\pi}_k) + \log(|\hat{\Sigma}_k|) \right]$$

Rem: les séparatrices sont **quadratiques**

# Un exemple : Bayésien Naïf



Données avec trois classes : les triplets indiquent les probabilités estimées des classes 0, 1 et 2 aux points indiqués

# Avantages / Inconvénients : Bayésien Naïf

## Avantages

- ▶ En pratique cela facilite les calculs : il ne faut plus calculer des matrices de covariance mais simplement les variances sur les  $p$  directions (par “nuage de points” correspondant aux classes)
- ▶ inversion facile des matrices diagonales !
- ▶ rapide pour de la grande dimension (e.g., très grands textes / spams)

## Inconvénients

- ▶ les séparatrices ne sont plus linéaires
- ▶ connu pour avoir des probabilités estimées mauvaises
- ▶ “renforcement de rumeur” attention à ne pas rajouter des variables très corrélées (besoin d'un pré-écrémage)

# Sommaire

Prérequis

Cadre et notations

## Quelques méthodes de classification

Prédiction linéaire et indicatrices

Analyse discriminante linéaire (LDA)

Analyse discriminante quadratique (QDA)

Bayésien Naïf

Régression logistique

K-nn

Régularisation

## Régression logistique : cas binaire

On suppose  $K = 2$  et l'on souhaite modéliser les probabilités conditionnelles des classes, ou plutôt leur log-ratio, par des quantités linéaires (affines) :

$$\log \left( \frac{\mathbb{P}(Y = 0 | X = \mathbf{x})}{\mathbb{P}(Y = 1 | X = \mathbf{x})} \right) = \alpha + \beta^\top \mathbf{x}$$

où  $\alpha \in \mathbb{R}$  et  $\beta \in \mathbb{R}^p$

Sous une telle hypothèse la séparatrice est **linéaire**, la règle étant simplement :

$$\alpha + \langle \beta, \mathbf{x} \rangle > 0$$

signifiant que l'on préfère la classe 0 à la classe 1 pour le point  $\mathbf{x}$

# Régression logistique (0')

On peut alors estimer les probabilités conditionnelles facilement :

$$\mathbb{P}(Y = 0 | X = \mathbf{x}) = \frac{\exp(\alpha + \langle \beta, \mathbf{x} \rangle)}{1 + \exp(\alpha + \langle \beta, \mathbf{x} \rangle)}$$

$$\mathbb{P}(Y = 1 | X = \mathbf{x}) = \frac{1}{1 + \exp(\alpha + \langle \beta, \mathbf{x} \rangle)}$$

# Régression logistique : numériquement

“Maximisation de la (log-)vraisemblance”

Notant la (log-)vraisemblance la fonction  $\ell$  des paramètres :

$$\begin{aligned}\ell(\alpha, \beta) &= \sum_{i=1}^n \log(\mathbb{P}(Y = Y_i | X = X_i, \alpha, \beta)) \\ &= \sum_{i=1}^n \sum_{k=0}^1 \mathbb{1}_{\{Y_i=k\}} \log(\mathbb{P}(Y = k | X = X_i, \alpha, \beta))\end{aligned}$$

on cherche alors une solution

$$(\hat{\alpha}, \hat{\beta}) \in \arg \max_{\alpha, \beta} \ell(\alpha, \beta)$$

---

**Exo:** montrer la formule qui suit :

$$\ell(\alpha, \beta) = \sum_{i=1}^n \left( Y_i(\alpha + \langle \beta, X_i \rangle) - \log[1 + \exp(\alpha + \langle \beta, X_i \rangle)] \right)$$

# Régression logistique et méthode Newton

La Hessienne est calculable : on peut donc appliquer la méthode de Newton

Ceux intéressés par les détails techniques, peuvent trouver les calculs de la Hessienne [Hastie et al. \(2009, page 120\)](#)

<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>

# Régression logistique (I)

On tente ici de modéliser les probabilités conditionnelles des classes, ou plutôt leur log-ratio, par des quantités linéaires (affines) :

$$\log \left( \frac{\mathbb{P}(Y = k | X = \mathbf{x})}{\mathbb{P}(Y = K - 1 | X = \mathbf{x})} \right) = \alpha_k + \langle \beta_k, \mathbf{x} \rangle$$

où  $\alpha_k \in \mathbb{R}$  et  $\beta_k \in \mathbb{R}^p$  pour tout  $k \in \{0, \dots, K - 1\}$

Sous une telle hypothèse les séparatrices sont **linéaires**, la règle étant simplement :

$$\alpha_k + \langle \beta_k, \mathbf{x} \rangle > 0$$

signifiant que l'on préfère la classe  $k$  à la classe  $K - 1$  pour le point  $\mathbf{x}$

## Régression logistique (II)

On peut alors estimer les probabilités conditionnelles facilement :

Pour  $k = 0, \dots, K - 2$  :

$$\mathbb{P}(Y = k | X = \mathbf{x}) = \frac{\exp(\alpha_k + \langle \beta_k, \mathbf{x} \rangle)}{1 + \sum_{l=0}^{K-2} \exp(\alpha_l + \langle \beta_l, \mathbf{x} \rangle)},$$

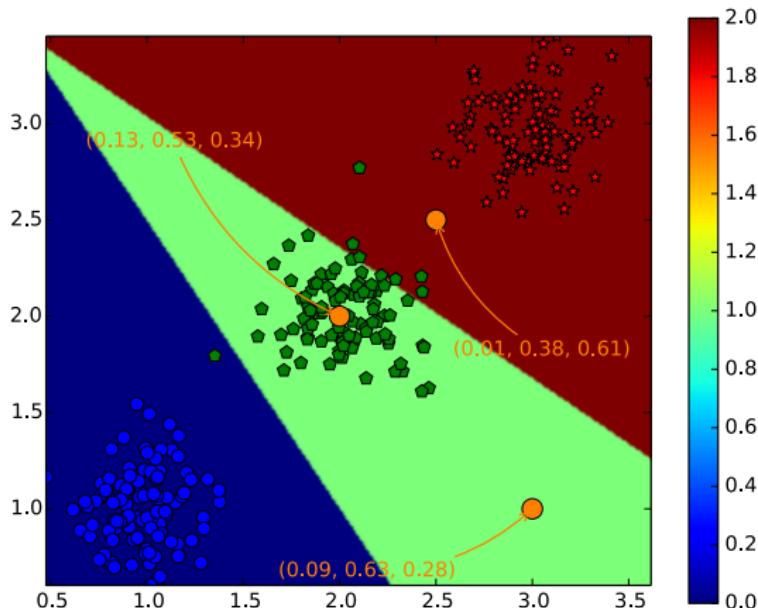
Pour  $k = K - 1$  :

$$\mathbb{P}(Y = K - 1 | X = \mathbf{x}) = \frac{1}{1 + \sum_{l=0}^{K-2} \exp(\alpha_l + \langle \beta_l, \mathbf{x} \rangle)}$$

Règle : on choisit la classe qui a la plus grande probabilité

Rem: Numériquement : le problème devient beaucoup plus dur (à écrire et à traiter) qu'en binaire, cf. [Hastie et al. \(2009\)](#)

# Un exemple : régression logistique



Données avec trois classes : les triplets indiquent les probabilités estimées des classes 0, 1 et 2 aux points indiqués

# Avantages / Inconvénients : régression logistique

## Avantages

- ▶ connu pour avoir des probabilités estimées bonnes
- ▶ séparations linéaires

## Inconvénients

- ▶ Classification binaire plus facile
- ▶ Problème d'optimisation plus complexe (temps de calcul)
- ▶ En pratique le cadre multi-classe est parfois géré par la technique du “un contre tous” et non par le cas logistique multinomiale (surtout si  $K$  est petit)

**TP:** Faire la partie TP associé

# Sommaire

Prérequis

Cadre et notations

## Quelques méthodes de classification

Prédiction linéaire et indicatrices

Analyse discriminante linéaire (LDA)

Analyse discriminante quadratique (QDA)

Bayésien Naïf

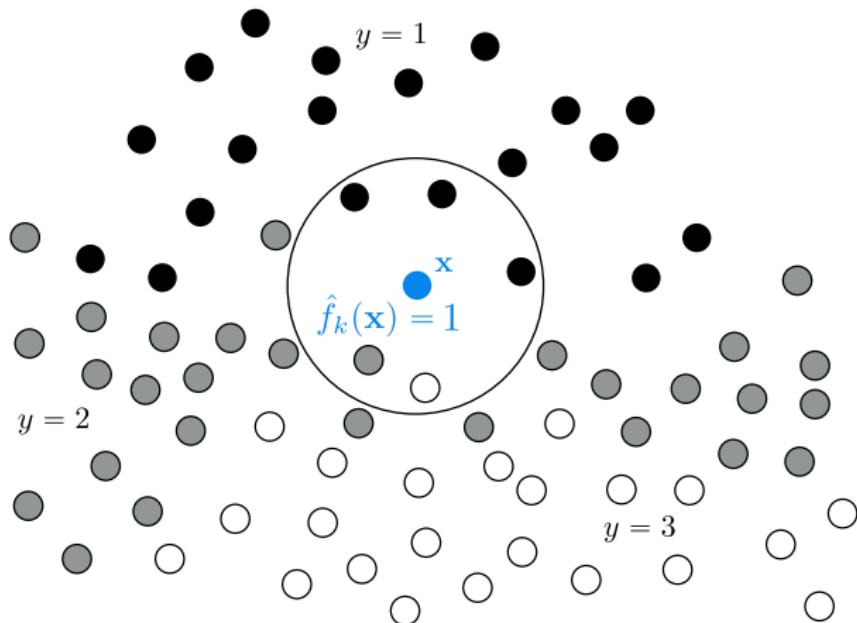
Régression logistique

K-nn

Régularisation

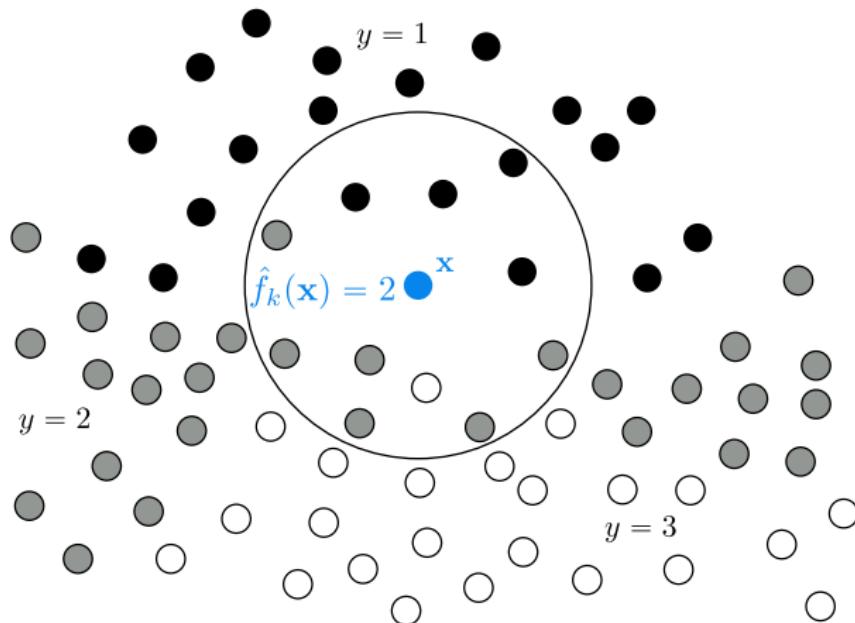
## K-nn

Méthode des  $k$ -plus proches voisins pour des valeurs du paramètres  $k = 5$  et  $k = 11$ . pour  $K = 3$  classes noir ( $y = 1$ ), gris ( $y = 2$ ), blanc ( $y = 3$ ).

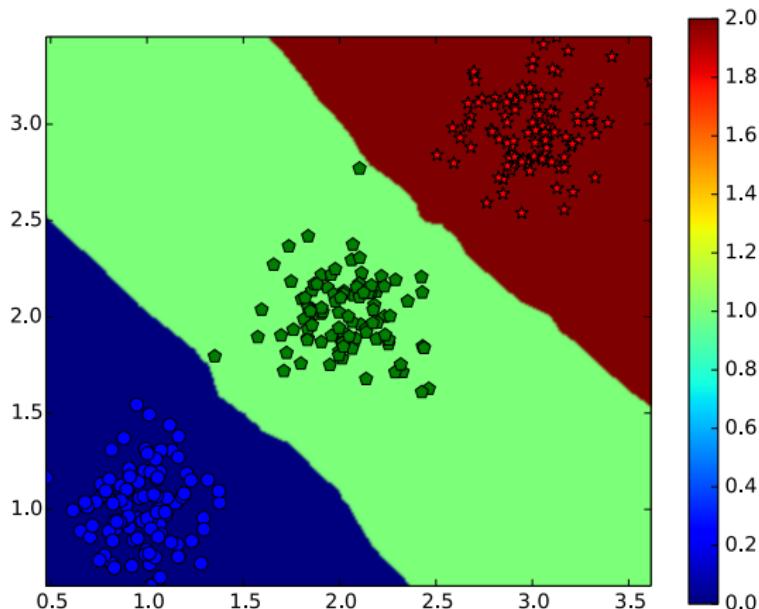


## K-nn

Méthode des  $k$ -plus proches voisins pour des valeurs du paramètres  $k = 5$  et  $k = 11$ . pour  $K = 3$  classes noir ( $y = 1$ ), gris ( $y = 2$ ), blanc ( $y = 3$ ).



## Un exemple : K-nn



Données avec trois classes : ici on pas directement accès aux probabilités estimées des classes

# Avantages / Inconvénients : K-nn

## Avantages

- ▶ séparations non-convexe en général
- ▶ s'adapte avec tout type de distance
- ▶ Multi-classe par défaut.

## Inconvénients

- ▶ temps de calcul peut-être long (calculer toute les distances deux à deux, est-ce vraiment utile?)

# Sommaire

Prérequis

Cadre et notations

Quelques méthodes de classification

Régularisation

Rappel : moindres carrés ordinaires

Régularisation avec  $\|\cdot\|_2^2$

Validation Croisée (CV)

Régularisation avec  $\|\cdot\|_1$

## Avertissement

L'aspect régularisation/pénalisation est illustrée ici en régression, mais il s'adapte de la même manière en classification (e.g., régression logistique)

# Définition des moindres carrés

Un estimateur des moindres carrées est solution du problème d'optimisation :

$$\hat{\boldsymbol{\theta}} \in \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \left( \frac{1}{2} \| Y - X\boldsymbol{\theta} \|_2^2 \right)$$

$$\hat{\boldsymbol{\theta}} \in \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^n \left[ y_i - \left( \theta_0 + \sum_{j=1}^p \theta_j x_{i,j} \right) \right]^2$$

$$\hat{\boldsymbol{\theta}} \in \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^n [y_i - (\langle x_i, \boldsymbol{\theta} \rangle)]^2$$

Rem: le minimiseur n'est pas toujours unique !

Rem: le terme  $\frac{1}{2}$  ne change rien au problème de minimisation, mais facilite certains calculs

# Formule des moindres carrés

Formule pour le cas d'un noyau non trivial

Si la matrice  $X$  est de plein rang (i.e., si  $X^\top X$  inversible) alors

$$\hat{\theta} = (X^\top X)^{-1} X^\top Y$$

Rem: on retrouve pour la moyenne pour le cas simple  $X = \mathbf{1}_n$  :

$$\hat{\theta} = (\langle \mathbf{1}_n, \mathbf{1}_n \rangle)^{-1} \langle \mathbf{1}_n, Y \rangle = \bar{y}_n$$

Rem: dans le cas simple  $X = \mathbf{x} = (x_1, \dots, x_n)^\top$  :  $\hat{\theta} = \langle \frac{\mathbf{x}}{\|\mathbf{x}\|^2}, Y \rangle$

**ATTENTION** : en pratique éviter de calculer l'inverse de  $X^\top X$  :

- ▶ cela est coûteux en temps de calcul
- ▶ une matrice  $(p+1) \times (p+1)$  peut être volumineuse, si " $p \gg n$ " (e.g., en biologie  $n$  patients,  $p$  gènes... )

# Sommaire

Prérequis

Cadre et notations

Quelques méthodes de classification

Régularisation

Rappel : moindres carrés ordinaires

Régularisation avec  $\|\cdot\|_2^2$

Validation Croisée (CV)

Régularisation avec  $\|\cdot\|_1$

## Ridge / Tikhonov : la définition pénalisée

$$\hat{\theta}_\lambda^{\text{rdg}} = \arg \min_{\theta \in \mathbb{R}^p} \left( \underbrace{\|Y - X\theta\|_2^2}_{\text{attaché aux données}} + \underbrace{\lambda \|\theta\|_2^2}_{\text{régularisation}} \right)$$

- ▶ Noter que l'estimateur *Ridge* est **unique** pour un  $\lambda$  fixé
- ▶ Cas limites :

$$\lim_{\lambda \rightarrow 0} \hat{\theta}_\lambda^{\text{rdg}} = \hat{\theta}^{\text{MCO}}$$

$$\lim_{\lambda \rightarrow +\infty} \hat{\theta}_\lambda^{\text{rdg}} = 0 \in \mathbb{R}^p$$

- ▶ Intérêt : assure l'unicité de la solution, rend le problème plus simple à résoudre numériquement (meilleur conditionnement)

Rem: version codée dans `sklearn`

# Interprétation contrainte

Un problème de la forme “Lagrangienne” suivante :

$$\arg \min_{\theta \in \mathbb{R}^p} \left( \underbrace{\frac{1}{2} \|Y - X\theta\|_2^2}_{\text{attaché aux données}} + \underbrace{\frac{\lambda}{2} \|\theta\|_2^2}_{\text{régularisation}} \right)$$

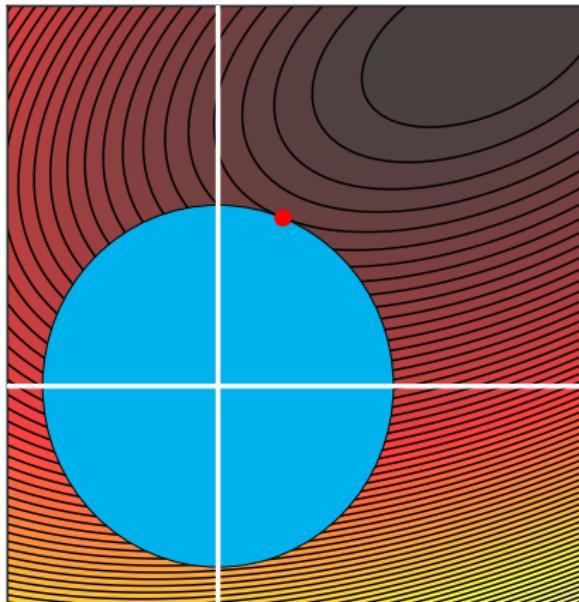
admet pour un certain  $T > 0$  la même solution que :

$$\begin{cases} \arg \min_{\theta \in \mathbb{R}^p} \|Y - X\theta\|_2^2 \\ \text{t.q. } \|\theta\|_2^2 \leq T \end{cases}$$

Rem: le lien  $T \leftrightarrow \lambda$  n'est pas explicite !

- ▶ Si  $T \rightarrow 0$  on retrouve le vecteur nul :  $0 \in \mathbb{R}^p$
- ▶ Si  $T \rightarrow \infty$  on retrouve  $\hat{\theta}^{\text{MCO}}$  (non contraint)

# Lignes de niveau et ensemble de contraintes



Optimisation sous contraintes  $\ell_2$

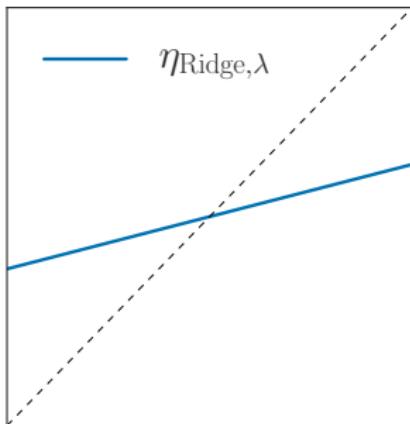
## Le cas orthogonal

Retour sur un cas simple  $X^\top X = \text{Id}_p$

$$\hat{\theta}_\lambda^{\text{rdg}} = (\lambda \text{Id}_p + X^\top X)^{-1} X^\top Y$$

$$\hat{\theta}_\lambda^{\text{rdg}} = (\lambda \text{Id}_p + \text{Id}_p)^{-1} X^\top Y = \frac{1}{\lambda + 1} X^\top Y$$

$$\hat{Y} = \frac{1}{\lambda + 1} Y = (\eta_{\text{rdg}, \lambda}(Y_i))_{i=1, \dots, n}$$



Rem: la fonction réelle  $\eta_{\text{rdg}, \lambda}$  est une contraction linéaire (shrinkage)

# Prédiction associée

Partant du coefficient *Ridge* :

$$\hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = (\lambda \text{Id}_p + X^T X)^{-1} X^T Y$$

la prédiction associée s'obtient ainsi :

$$\hat{Y} = X \hat{\boldsymbol{\theta}}_{\lambda}^{\text{rdg}} = X(\lambda \text{Id}_p + X^T X)^{-1} X^T Y$$

Rem: l'estimateur  $\hat{Y}$  est toujours linéaire en  $Y$

## Astuce du noyau (Kernel trick)

*Astuce du noyau* : Selon si  $n > p$  ou  $n \leq p$ , une méthode qui cherche à trouver une solution des moindres carrés par inversion peut préférer l'une des deux formulations suivantes :

$$X^\top (XX^\top + \lambda \text{Id}_n)^{-1} Y = (X^\top X + \lambda \text{Id}_p)^{-1} X^\top Y$$

- ▶ membre de gauche : on inverse une matrice  $n \times n$
- ▶ membre de droite : on inverse une matrice  $p \times p$

Rem: cette propriété est aussi très utile pour les méthodes à noyaux de type SVM

# Sommaire

Prérequis

Cadre et notations

Quelques méthodes de classification

Régularisation

Rappel : moindres carrés ordinaires

Régularisation avec  $\|\cdot\|_2^2$

**Validation Croisée (CV)**

Régularisation avec  $\|\cdot\|_1$

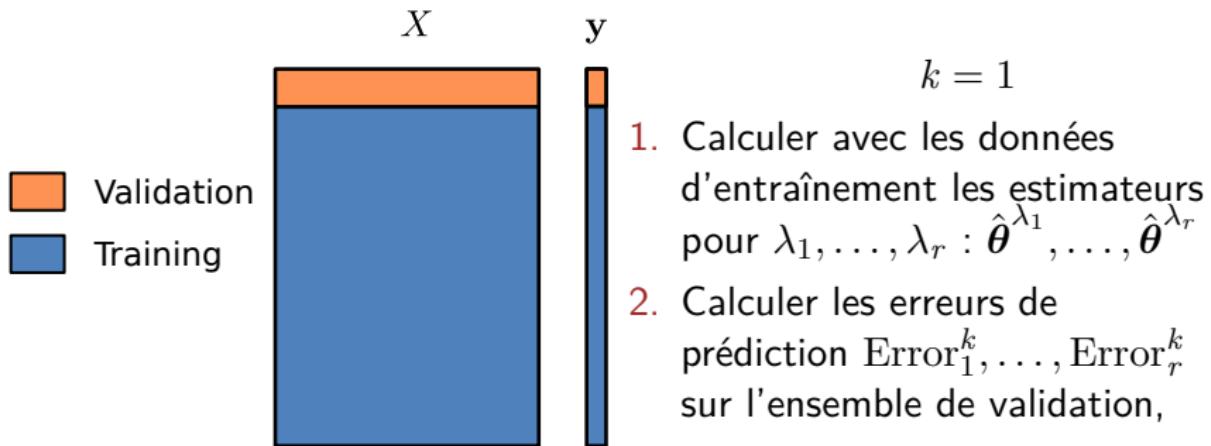
## Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



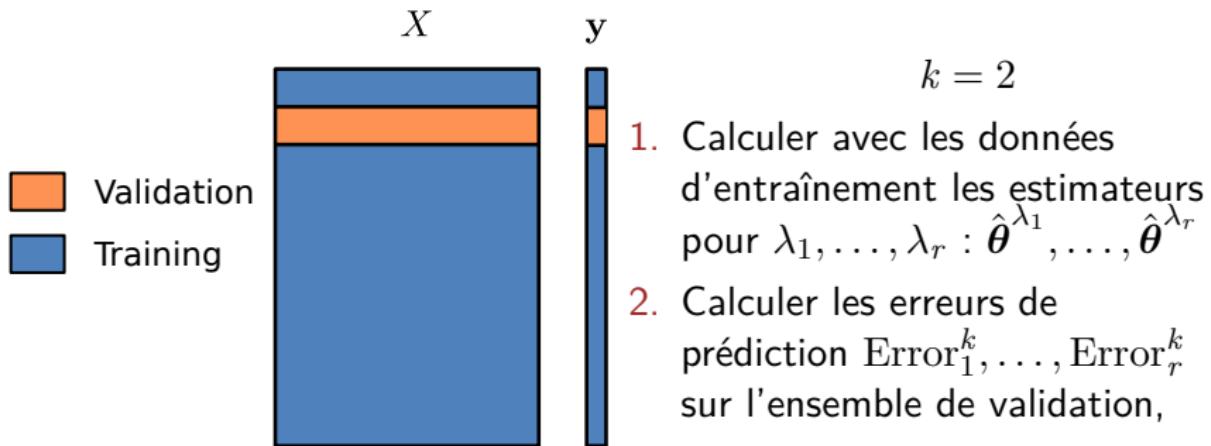
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



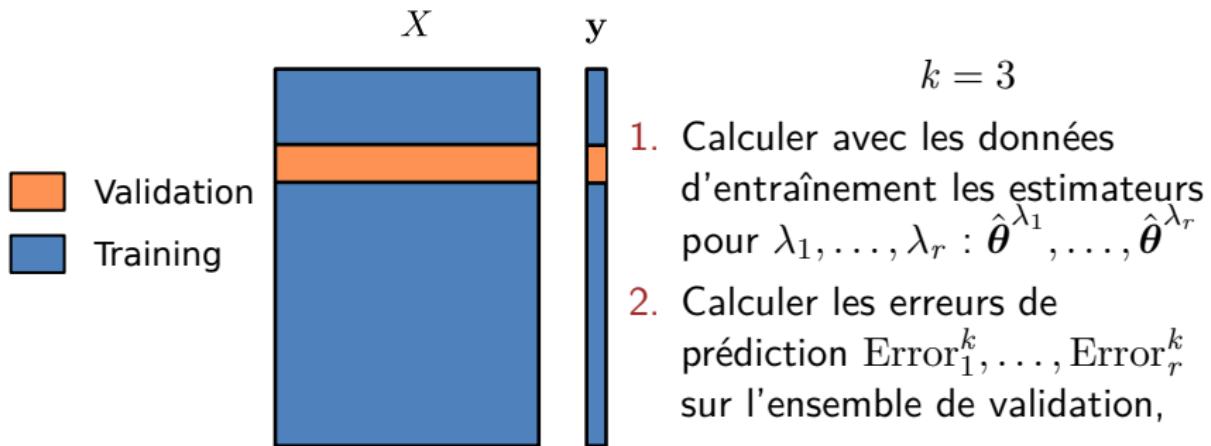
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



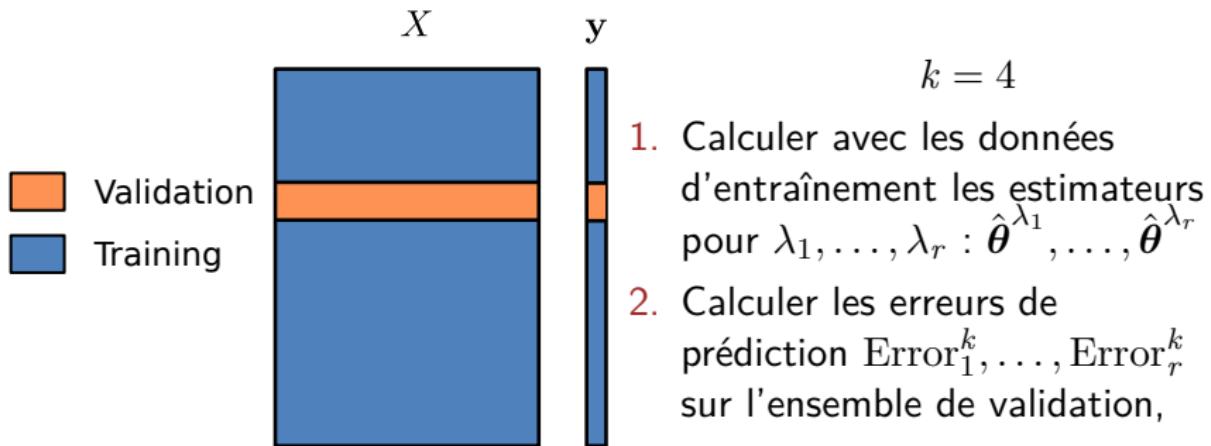
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



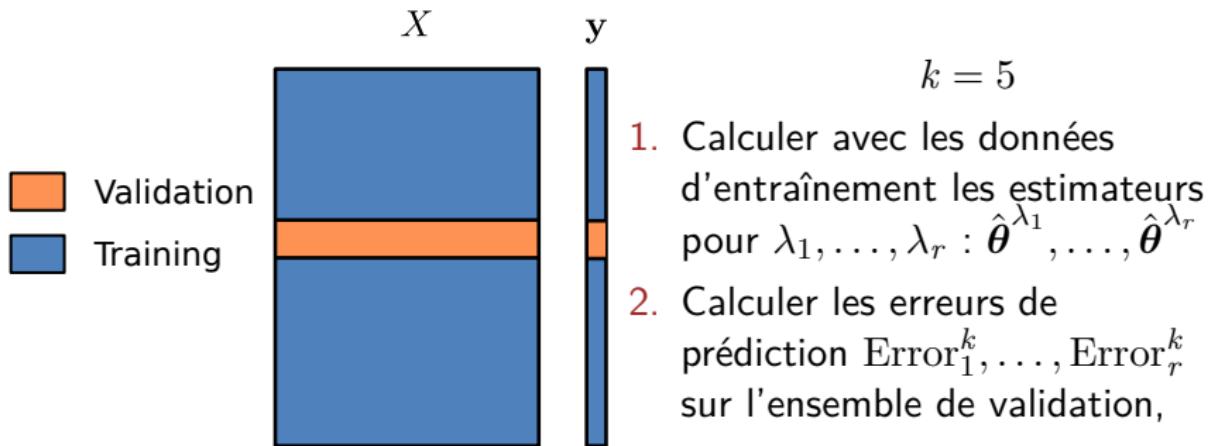
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



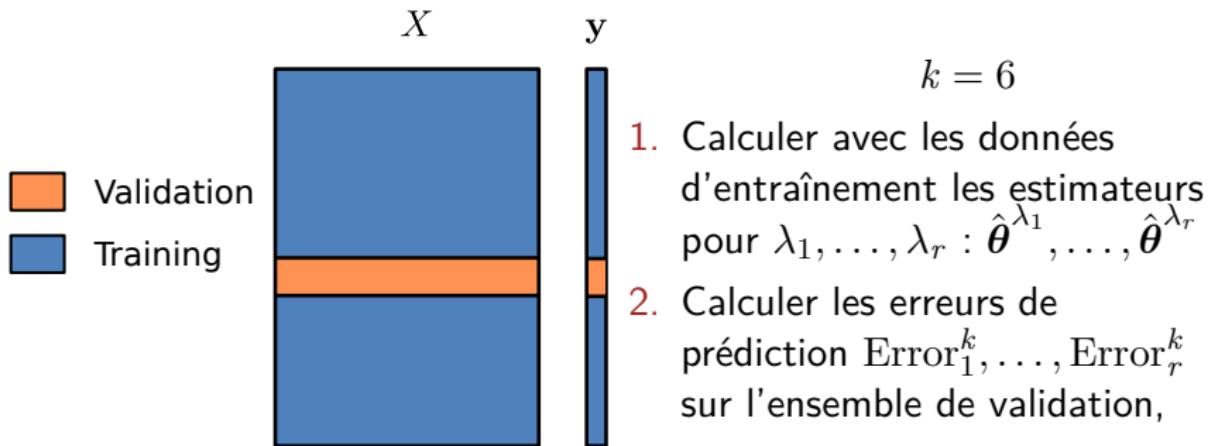
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



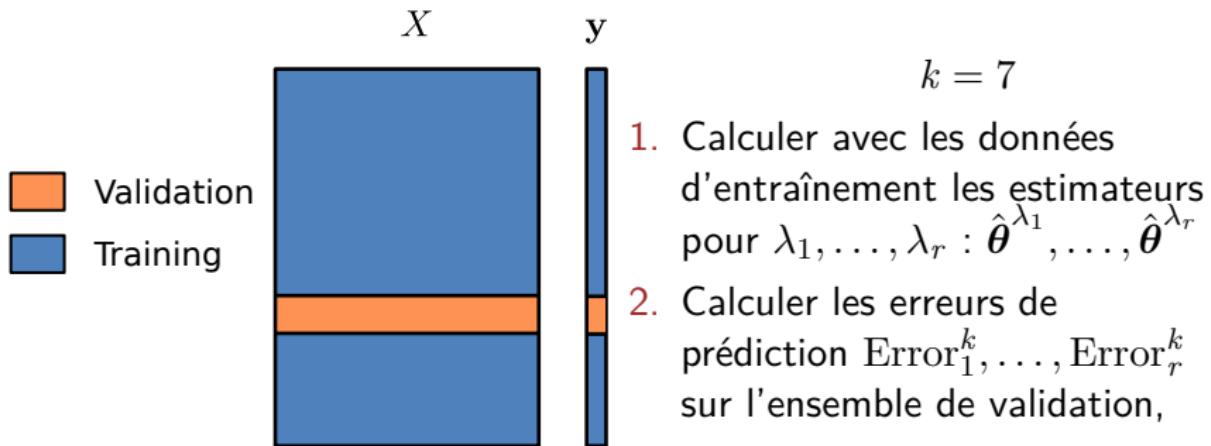
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



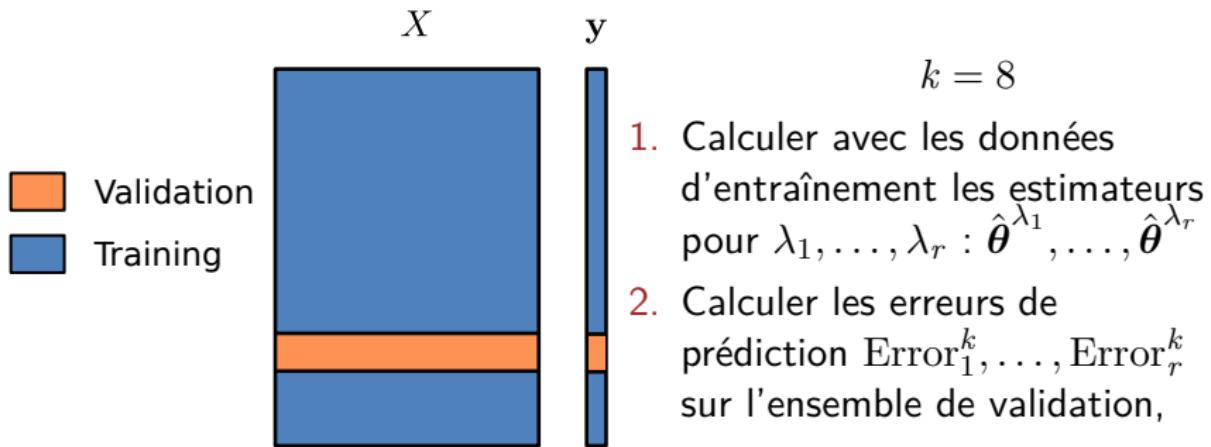
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



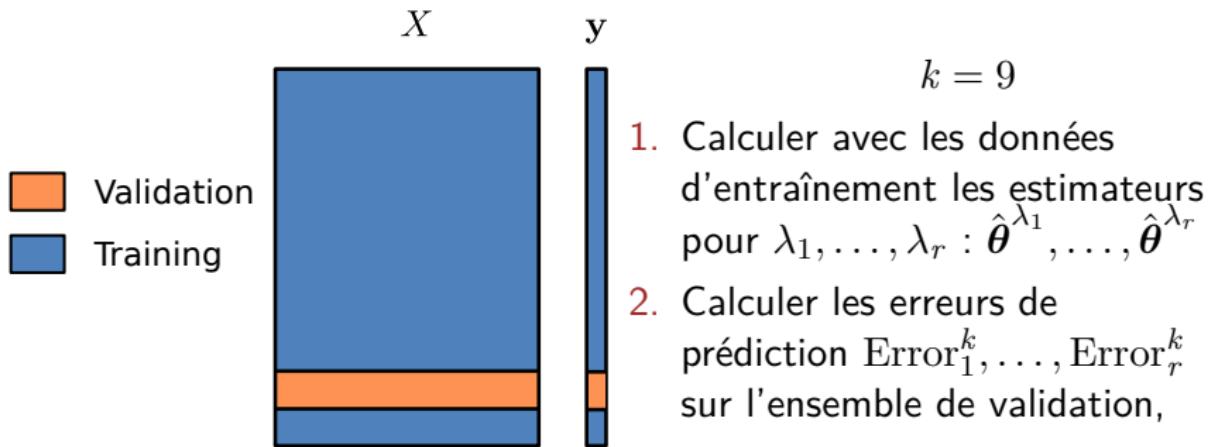
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



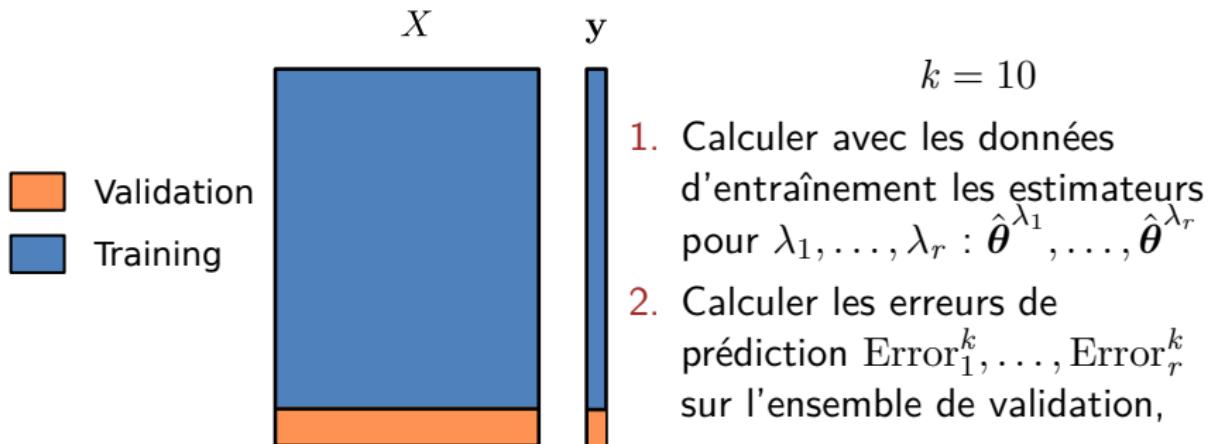
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



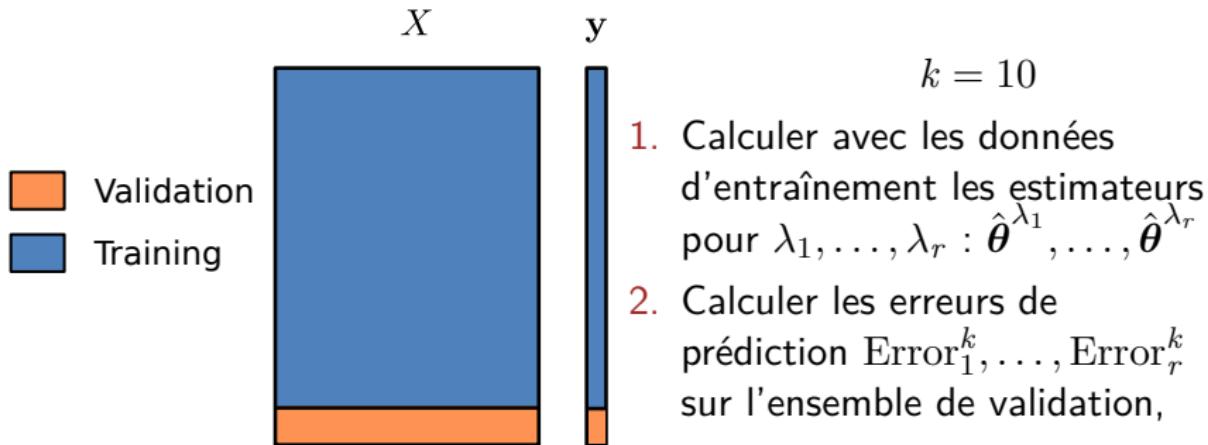
# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



# Validation croisée $K$ -fold ( $K = 10$ )

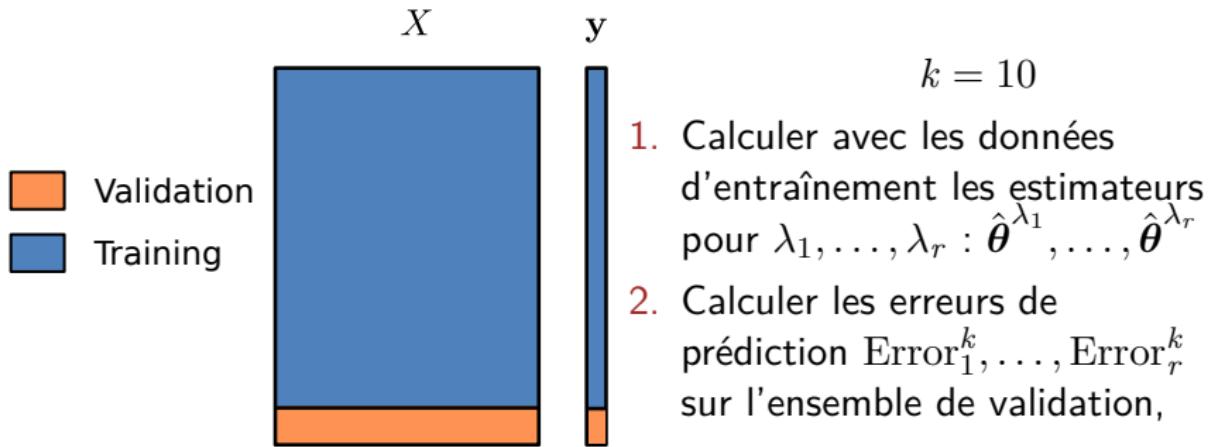
- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



**Choix du paramètre** : calculer  $\widehat{\text{Error}}_1, \dots, \widehat{\text{Error}}_r$ , moyennes des erreurs et choisir  $\hat{i}^{\text{CV}} \in \llbracket 1, r \rrbracket$  atteignant la plus petite

# Validation croisée $K$ -fold ( $K = 10$ )

- ▶ Choisir une grille de taille  $r$  de  $\lambda$  à tester :  $\lambda_1, \dots, \lambda_r$
- ▶ Diviser  $(X, Y)$  selon les observations en  $K$  blocs :



**Choix du paramètre** : calculer  $\widehat{\text{Error}}_1, \dots, \widehat{\text{Error}}_r$ , moyennes des erreurs et choisir  $\hat{i}^{\text{CV}} \in \llbracket 1, r \rrbracket$  atteignant la plus petite

**Re-calibration** : calculer  $\hat{\theta}^{\lambda_{\hat{i}^{\text{CV}}}}$  sur l'ensemble des observations

# CV en pratique

Cas extrême de validation croisée *cross-validation*

- ▶  $K = 1$  impossible, au moins  $K = 2$
- ▶  $K = n$ , stratégie “*leave-one-out*” (*cf. Jackknife*) : autant de blocs que de variables
  - Rem:  $K = n$  efficace computationnellement mais instable

Conseils pratiques :

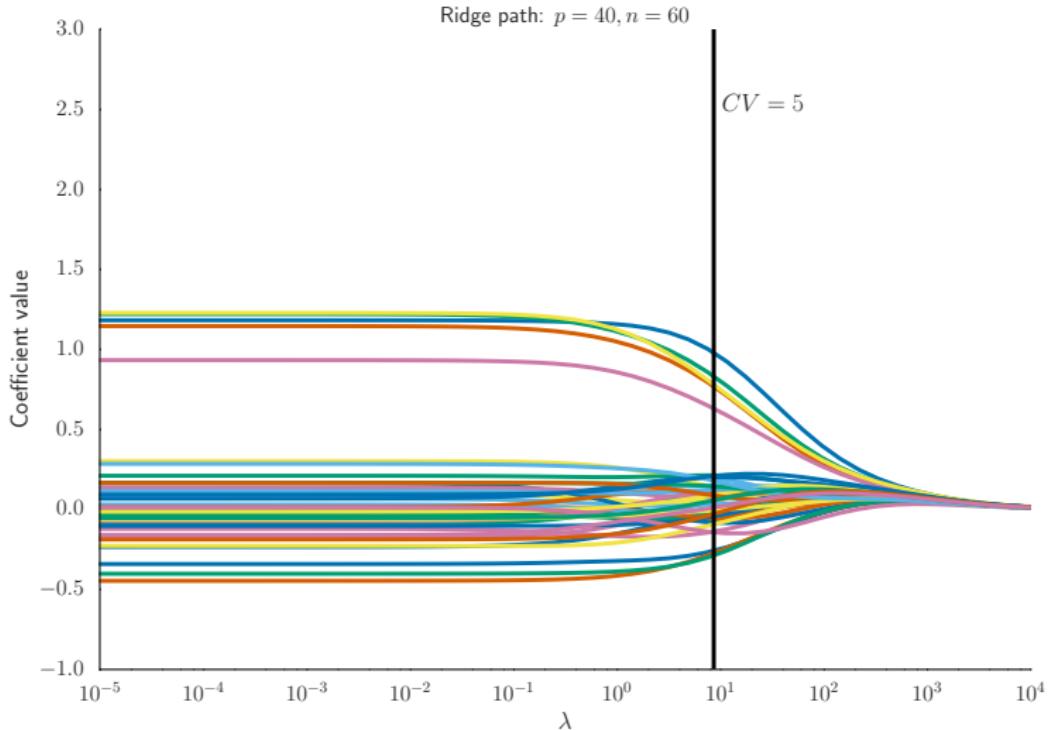
- ▶ “randomiser les observations” : observations dans un ordre aléatoire, évite des blocs de données trop similaires (chaque sous-bloc doit être représentatif de l’ensemble)
- ▶ choix habituels :  $K = 5, 10$

Alternatives : partition aléatoire entre ensemble d’apprentissage et validation, version pour séries temporelles, etc.

[http://scikit-learn.org/stable/modules/cross\\_validation.html](http://scikit-learn.org/stable/modules/cross_validation.html)

Rem: en prédiction on peut aussi moyenner les meilleurs estimateurs obtenus plutôt que de re-calibrer sur toutes les données

## Choix de $\lambda$ : exemple avec $CV = 5$ (I)



# Sommaire

Prérequis

Cadre et notations

Quelques méthodes de classification

## Régularisation

Rappel : moindres carrés ordinaires

Régularisation avec  $\|\cdot\|_2^2$

Validation Croisée (CV)

Régularisation avec  $\|\cdot\|_1$

# Motivation

Utilité des estimateurs  $\hat{\theta}$  avec beaucoup de coefficients nuls :

- ▶ pour l'interprétation
- ▶ pour l'efficacité computationnelle si  $p$  est énorme

L'idée sous-jacente : **sélectionner des variables**

# Méthodes de sélection de variables

- ▶ Méthodes de type **écrémage screening** : on supprime les  $x_j$  dont la corrélation avec  $Y$  est faible
  - avantages : rapide (+++), coût :  $p$  produits scalaires de taille  $n$ , intuitive (+++)
  - défauts : néglige les interactions entre variables  $x_j$ , résultats théoriques faibles (- - -)
- ▶ Méthodes **gloutonnes greedy** ou **pas à pas stagewise/stepwise**
  - avantages : rapide (++) , intuitive (++)
  - défauts : propagation mauvaises sélections de variables aux étapes suivantes ; résultats théoriques faibles (-)
- ▶ Méthodes **pénalisées** favorisant la parcimonie (e.g., Lasso)
  - avantages : résultats théoriques bons (++)
  - défauts : encore lent (on y travaille !) (-),

# La pseudo-norme $\ell_0$

## Définitions

Le **support** du vecteur  $\theta$  est l'ensemble des indices des coordonnées non nulles :

$$\text{supp}(\theta) = \{j \in \llbracket 1, p \rrbracket, \theta_j \neq 0\}$$

La **pseudo-norme**  $\ell_0$  d'un vecteur  $\theta \in \mathbb{R}^p$  est son nombre de coordonnées non-nulles :

$$\|\theta\|_0 = \text{card}\{j \in \llbracket 1, p \rrbracket, \theta_j \neq 0\}$$

Rem:  $\|\cdot\|_0$  n'est pas une norme,  $\forall t \in \mathbb{R}^*, \|t\theta\|_0 = \|\theta\|_0$

Rem:  $\|\cdot\|_0$  n'est pas non plus convexe,  $\theta_1 = (1, 0, 1, \dots, 0)$   
 $\theta_2 = (0, 1, 1, \dots, 0)$  et  $3 = \|\frac{\theta_1 + \theta_2}{2}\|_0 \geq \frac{\|\theta_1\|_0 + \|\theta_2\|_0}{2} = 2$

# La pénalisation $\ell_0$

Première tentative de méthode pénalisée pour introduire de la parcimonie : utiliser  $\ell_0$  pour la pénalisation / régularisation

$$\hat{\boldsymbol{\theta}}_\lambda = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \left( \underbrace{\frac{1}{2} \|Y - X\boldsymbol{\theta}\|_2^2}_{\text{attache aux données}} + \underbrace{\lambda \|\boldsymbol{\theta}\|_0}_{\text{régularisation}} \right)$$

## Problème combinatoire !!!

La résolution exacte nécessite de considérer tous les sous-modèles, i.e., calculer les estimateurs pour tous les supports possibles ; il y en a  $2^p$ , ce qui requiert le calcul de  $2^p$  moindres carrés !

### **Exemple:**

$p = 10$  possible :  $\approx 10^3$  moindres carrés

$p = 30$  impossible :  $\approx 10^{10}$  moindres carrés

# Le Lasso : la définition pénalisée

Lasso : *Least Absolute Shrinkage and Selection Operator*

Tibshirani (1996)

$$\hat{\boldsymbol{\theta}}_{\lambda}^{\text{Lasso}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \left( \underbrace{\frac{1}{2} \|Y - X\boldsymbol{\theta}\|_2^2}_{\text{attache aux données}} + \underbrace{\lambda \|\boldsymbol{\theta}\|_1}_{\text{régularisation}} \right)$$

où  $\|\boldsymbol{\theta}\|_1 = \sum_{j=1}^p |\theta_j|$  (somme des valeurs absolues des coefficients)

- On retrouve de nouveau les cas limites :

$$\lim_{\lambda \rightarrow 0} \hat{\boldsymbol{\theta}}_{\lambda}^{\text{Lasso}} = \hat{\boldsymbol{\theta}}^{\text{MCO}}$$

$$\lim_{\lambda \rightarrow +\infty} \hat{\boldsymbol{\theta}}_{\lambda}^{\text{Lasso}} = \mathbf{0} \in \mathbb{R}^p$$

**Attention** : l'estimateur Lasso n'est pas toujours **unique** pour un  $\lambda$  fixé (prendre par exemple deux colonnes identiques)

# Interprétation contrainte

Un problème de la forme :

$$\hat{\boldsymbol{\theta}}_{\lambda}^{\text{Lasso}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \left( \underbrace{\frac{1}{2} \|Y - X\boldsymbol{\theta}\|_2^2}_{\text{attache aux données}} + \underbrace{\lambda \|\boldsymbol{\theta}\|_1}_{\text{régularisation}} \right)$$

admet la même solution qu'une version contrainte :

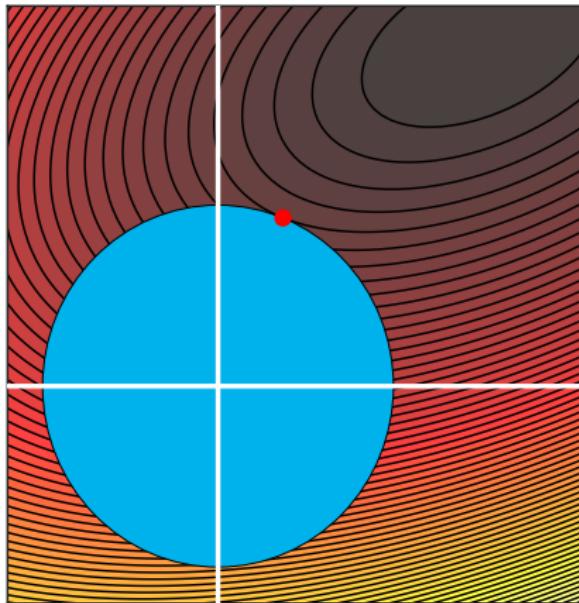
$$\begin{cases} \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^p} \|Y - X\boldsymbol{\theta}\|_2^2 \\ \text{t.q. } \|\boldsymbol{\theta}\|_1 \leq T \end{cases}$$

pour un certain  $T > 0$ .

Rem: hélas le lien  $T \leftrightarrow \lambda$  n'est pas explicite

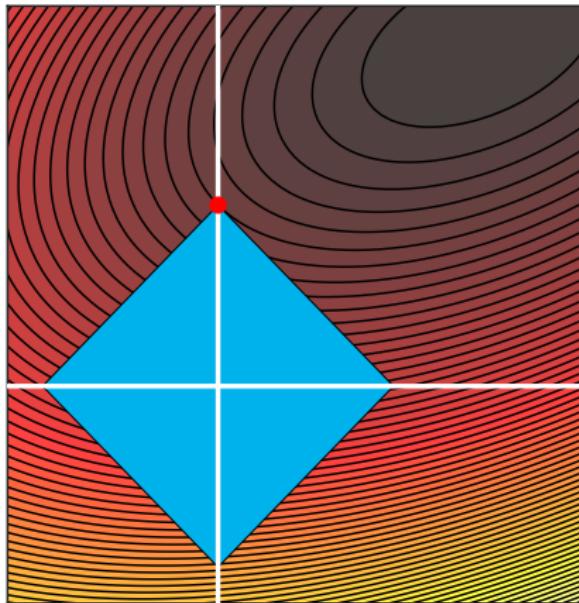
- ▶ Si  $T \rightarrow 0$  on retrouve le vecteur nul :  $0 \in \mathbb{R}^p$
- ▶ Si  $T \rightarrow \infty$  on retrouve  $\hat{\boldsymbol{\theta}}^{\text{MCO}}$  (non contraint)

# Mise à zéro de certains coefficients



Optimisation sous contrainte  $\ell_2$  : solution non parcimonieuse

# Mise à zéro de certains coefficients

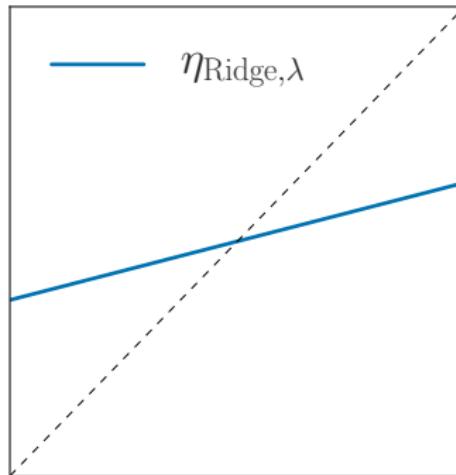


Optimisation sous contrainte  $\ell_1$  : solution parcimonieuse

# Régularisation en 1D : Ridge

Solution de :  $\eta_\lambda(z) = \arg \min_{x \in \mathbb{R}} x \mapsto \frac{1}{2}(z - x)^2 + \frac{\lambda}{2}x^2$

$$\eta_\lambda(z) = \frac{z}{1 + \lambda}$$

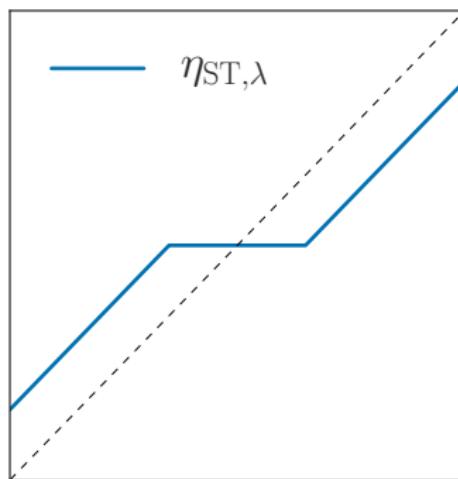


Contraction  $\ell_2$  : Ridge

## Régularisation en 1D : Lasso

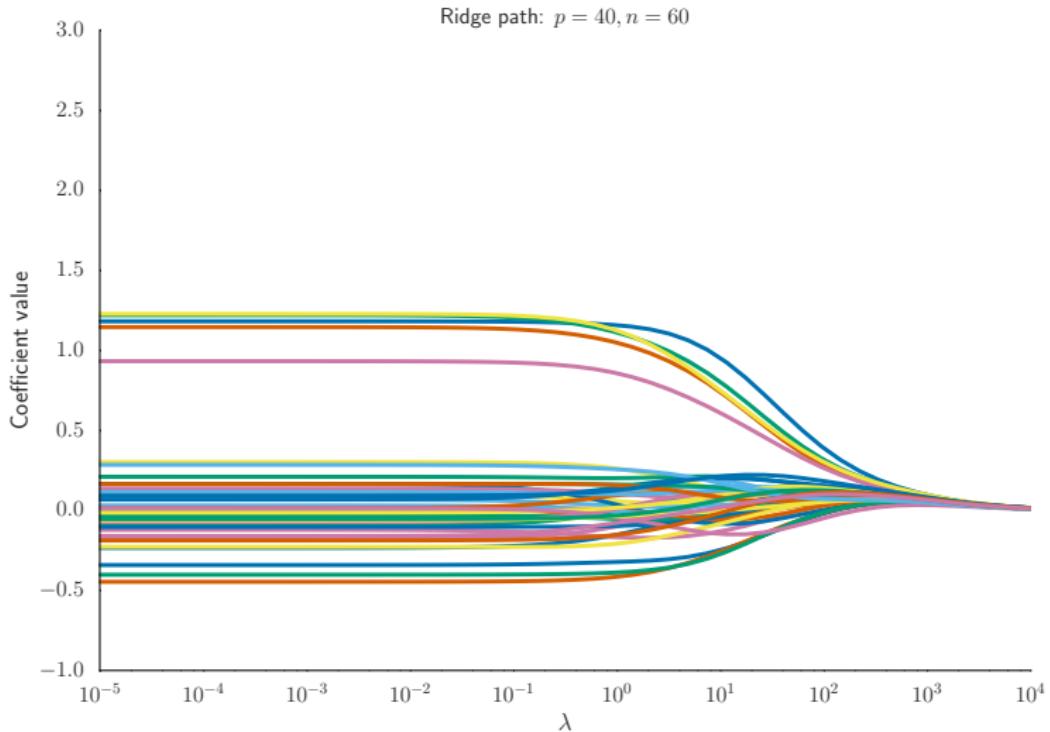
Solution de :  $\eta_\lambda(z) = \arg \min_{x \in \mathbb{R}} x \mapsto \frac{1}{2}(z - x)^2 + \lambda|x|$

$$\eta_\lambda(z) = \text{sign}(z)(|z| - \lambda)_+$$

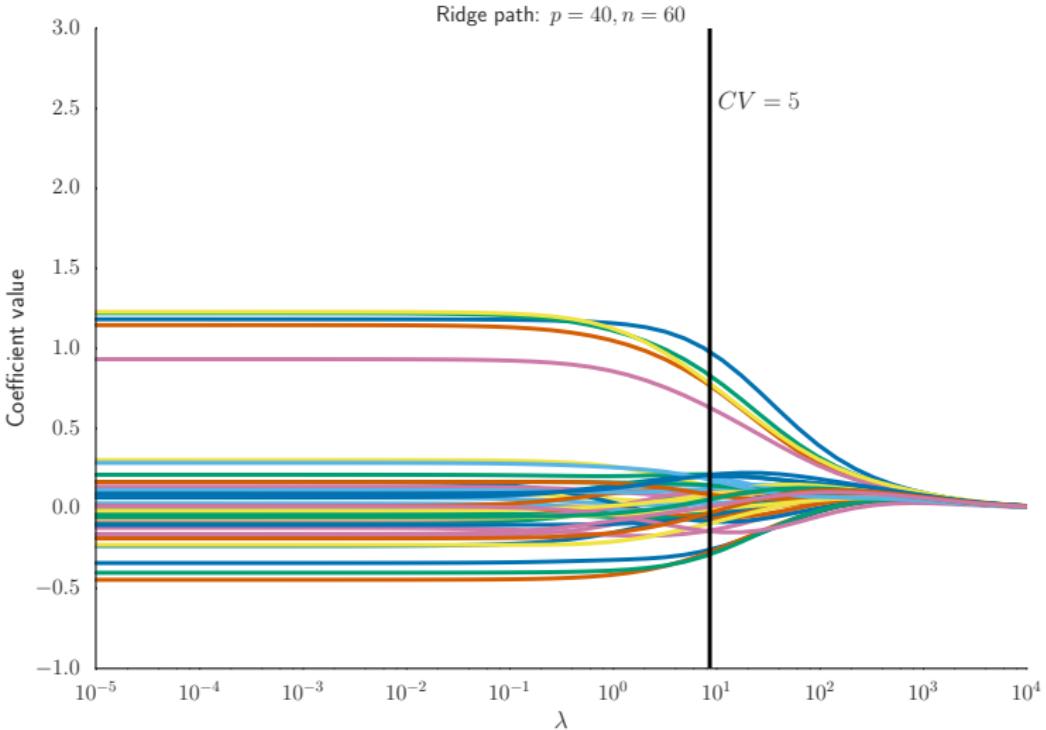


Contraction  $\ell_1$  : Seuillage doux *soft thresholding*

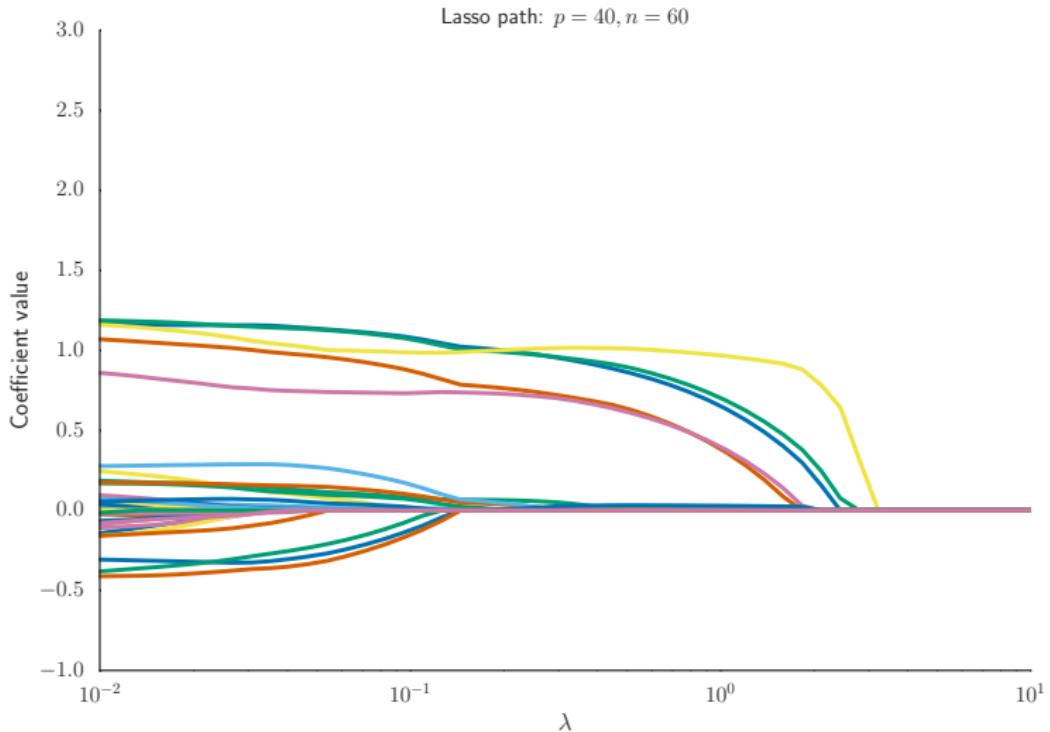
# Lasso vs Ridge



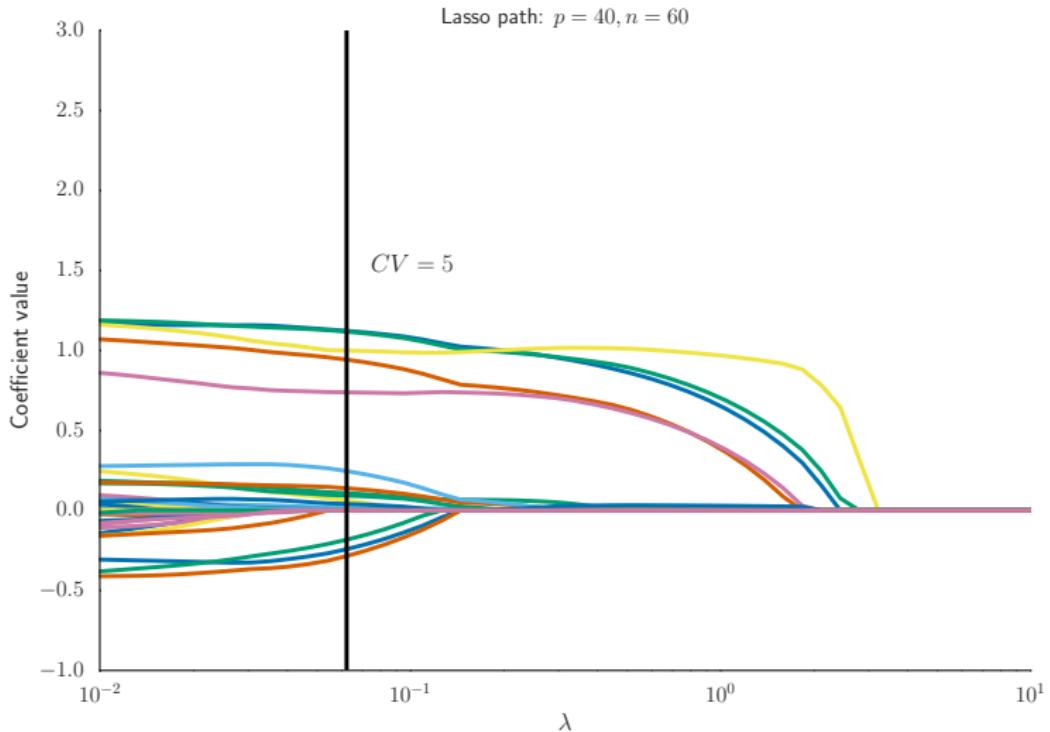
# Lasso vs Ridge



# Lasso vs Ridge



# Lasso vs Ridge



# Intérêt du Lasso

- ▶ Enjeu numérique : le Lasso est un problème **convexe**
- ▶ Sélection de variables / solutions parcimonieuses (sparse) :  
 $\hat{\theta}_\lambda^{\text{Lasso}}$  à potentiellement de nombreux coefficients nuls. Le paramètre  $\lambda$  contrôle le niveau de parcimonie : si  $\lambda$  est grand, les solutions sont très creuses.

**Exemple:** on obtient 17 coefficients non nuls pour LassoCV dans la simulation précédente

Rem: RidgeCV n'a aucun coefficient nul

# Références I

**Bonus** : feuille de route simplifiée pour l'apprentissage automatique :

[http://scikit-learn.org/stable/tutorial/machine\\_learning\\_map/](http://scikit-learn.org/stable/tutorial/machine_learning_map/)

Aspects numériques sur LDA/QDA :

<http://www.stat.cmu.edu/~ryantibs/datamining/>

Divers :

- ▶ T. Hastie, R. Tibshirani, and J. Friedman.

*The elements of statistical learning.*

Springer Series in Statistics. Springer, New York, second edition, 2009.

<http://www-stat.stanford.edu/~tibs/ElemStatLearn/>.

- ▶ R. Tibshirani.

Regression shrinkage and selection via the lasso.

*J. Roy. Statist. Soc. Ser. B*, 58(1) :267–288, 1996.