UNIVERSITÉ
PARIS-EST CRÉTEIL
VAL DE MARNE

Apprentissage Automatique

ESIPE Créteil

J-B Salomond (jean-bernard.salomond@u-pec.fr)

Années 2018-2019

Chap.1

Introduction

Chap.1 L'apprentissage Automatique

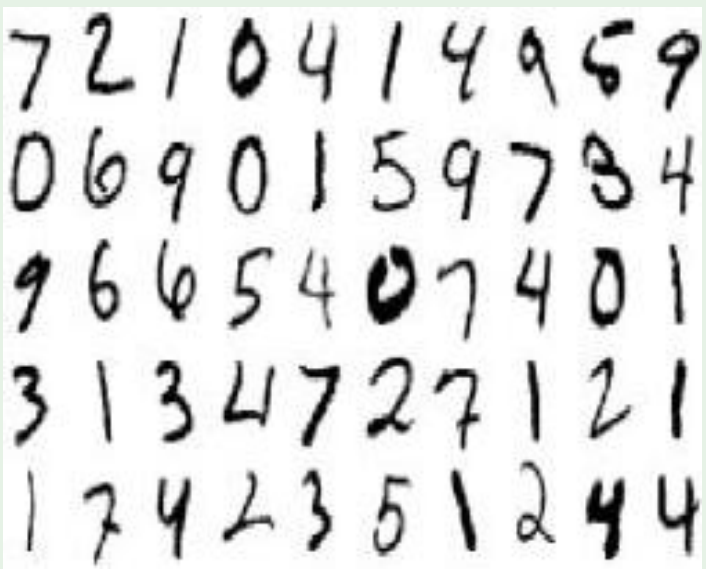
L'apprentissage automatique consiste à prendre une décision optimale à partir des données observées. Il peut s'agir d'un problème de classification (classer des données en groupe), de problème de prédiction (prédire une valeur en fonction des données observées), etc.

Chap.1 L'apprentissage Automatique

L'apprentissage automatique consiste à prendre une décision optimale à partir des données observées. Il peut s'agir d'un problème de classification (classer des données en groupe), de problème de prédiction (prédire une valeur en fonction des données observées), etc.

Exemple

Un problème classique est la lecture de code postaux



En fonction des données disponibles, il existe plusieurs techniques d'apprentissage :

- Apprentissage supervisé

En fonction des données disponibles, il existe plusieurs techniques d'apprentissage :

- Apprentissage supervisé
- Apprentissage non-supervisé

En fonction des données disponibles, il existe plusieurs techniques d'apprentissage :

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage semi-supervisé

En fonction des données disponibles, il existe plusieurs techniques d'apprentissage :

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage semi-supervisé
- Apprentissage par renforcement

En fonction des données disponibles, il existe plusieurs techniques d'apprentissage :

- Apprentissage supervisé
- Apprentissage non-supervisé
- Apprentissage semi-supervisé
- Apprentissage par renforcement

Apprentissage Statistique

Quelques notations

On se place dans un cadre probabiliste. Cela permet notamment de modéliser notre incertitude sur les données, notre ignorance de certains phénomènes, etc.

Apprentissage Statistique

Quelques notations

On se place dans un cadre probabiliste. Cela permet notamment de modéliser notre incertitude sur les données, notre ignorance de certains phénomènes, etc.
On se fixe les notations suivantes (pour le moment)

- X Les données observées (input)
- Y L'output
- $\mathbb{P}_Z$ La loi d'une variable aléatoire Z
- $\hat{f}$ un prédicteur

Apprentissage Statistique

Quelques notations

On se place dans un cadre probabiliste. Cela permet notamment de modéliser notre incertitude sur les données, notre ignorance de certains phénomènes, etc.
On se fixe les notations suivantes (pour le moment)

- X Les données observées (input)
- Y L'output
- $\mathbb{P}_Z$ La loi d'une variable aléatoire Z
- $\hat{f}$ un prédicteur

But

Le but est de construire un prédicteur $\hat{f}$ qui prédise au mieux l'output Y quand on lui donne un input X.

On rappelle les définitions suivantes

Espérance

- Pour une variable aléatoire discrète :

$$\mathbb{E}(X) = \sum_{x \in \Omega} xP(X = x)$$

- Pour une variable aléatoire continue de densité p,

$$\mathbb{E}(X) = \int_{\Omega} X d\mathbb{P}(X) = \int_{\mathbb{R}} xp(x)dx$$

Apprentissage Statistique

Pourquoi estimer les paramètres ?

Il y a deux raisons principales pour lesquelles on peut vouloir estimer les paramètres d'un modèle statistique :

- Faire des prédictions
- Faire une inférence sur le modèle

Prenons l'exemple du modèle suivant

$$Y = f(X) + \epsilon$$

Prédiction

On dispose d'observation X pour lesquelles on ne connaît pas Y et on souhaite le prédire

Inférence

On veut comprendre l'effet de X sur Y.

Apprentissage Statistique

Pourquoi estimer les paramètres ?

Il y a deux raisons principales pour lesquelles on peut vouloir estimer les paramètres d'un modèle statistique :

- Faire des prédictions
- Faire une inférence sur le modèle

Prenons l'exemple du modèle suivant

$$Y = f(X) + \epsilon$$

Prédiction

On dispose d'observation X pour lesquelles on ne connaît pas Y et on souhaite le prédire

Inférence

On veut comprendre l'effet de X sur Y.

Jusqu'à présent les cours précédents se sont intéressé à l'inférence (Stat. inférentielle, régression, régression multiple etc.), ici nous allons nous concentrer plutôt sur la prédiction.

Apprentissage Statistique

Pourquoi estimer les paramètres ?

Notre but est de prédire l'output Y à partir de l'input X. On modélise pour chaque individu

$$(X_i, Y_i) \sim \mathbb{P}_{(X,Y)},$$

et on cherche à prédire Y en observant $X = x$. Il s'agit de trouver une fonction $f \in \mathcal{H}$ qui permet d'expliquer/prédire Y en fonction de X. En général on modélise $Y|X \sim \mathbb{P}_{f(X)}$ ($\mathcal{H}$ est appelé espace des hypothèses).

Apprentissage Statistique

Pourquoi estimer les paramètres ?

Notre but est de prédire l'output Y à partir de l'input X. On modélise pour chaque individu

$$(X_i, Y_i) \sim \mathbb{P}_{(X,Y)},$$

et on cherche à prédire Y en observant $X = x$. Il s'agit de trouver une fonction $f \in \mathcal{H}$ qui permet d'expliquer/prédire Y en fonction de X. En général on modélise $Y|X \sim \mathbb{P}_{f(X)}$ ($\mathcal{H}$ est appelé espace des hypothèses).

Oracle

L'oracle connaît la relation qui est donnée par la loi conditionnelle $\mathbb{P}(Y|X)$.

Risque

Définitions

Une fois choisi le prédicteur f, on va va évaluer l'erreur.

Fonction de perte

On appelle perte associé à la décision $f(x)$ pour une output y une fonction $L(f(x), y)$. Exemples

- Pour des outputs continus
  - $L(f(x), y) = (f(x) - y)^2$
  - $L(f(x), y) = |f(x) - y|$
  - $L(f(x), y) = |f(x) - y|^\alpha$ avec $\alpha \geq 1$
- Pour des output discrets $L(f(x), y) = \mathbb{I}(f(x) = y)$

Une fois choisi le prédicteur f, on va va évaluer l'erreur.

Fonction de perte

On appelle perte associé à la décision $f(x)$ pour une output y une fonction $L(f(x), y)$. Exemples

- Pour des outputs continus
  - $L(f(x), y) = (f(x) - y)^2$
  - $L(f(x), y) = |f(x) - y|$
  - $L(f(x), y) = |f(x) - y|^\alpha$ avec $\alpha \geq 1$
- Pour des output discrets $L(f(x), y) = \mathbb{I}(f(x) = y)$

Risque

On appelle risque associé à un prédicteur pour une fonction de perte L la donnée

$$R(f) = \mathbb{E}_{X,Y}(L(f(x), y)) = \int L(f(x), y) d\mathbb{P}(x, y)$$

En général on ne connaît pas la loi $\mathbb{P}_{X,Y}$. On estime donc le risque par

$$\hat{R}_n(f) = \frac{1}{N} \sum_{i=1}^{n} L(f(x_i), y_i) \to R(f)$$

Cette quantité est appelée risque empirique.

En général on ne connaît pas la loi $\mathbb{P}_{X,Y}$. On estime donc le risque par

$$\hat{R}_n(f) = \frac{1}{N} \sum_{i=1}^{n} L(f(x_i), y_i) \to R(f)$$

Cette quantité est appelée risque empirique. Si les x_i, y_i sont ceux utilisés pour *estimer* f alors on parle de risque d'apprentissage (training risk).

En général on ne connaît pas la loi $\mathbb{P}_{X,Y}$. On estime donc le risque par

$$\hat{R}_n(f) = \frac{1}{N} \sum_{i=1}^{n} L(f(x_i), y_i) \to R(f)$$

Cette quantité est appelée risque empirique. Si les x_i, y_i sont ceux utilisés pour *estimer* f alors on parle de risque d'apprentissage (training risk).

Remarque

On est pas forcément intéressé par le training risk. On cherche a évaluer la qualité du prédicteur pour des observations que l'on a auparavant jamais rencontrées !

Risque

Définitions

Pour le modèle $Y = f(X) + \epsilon$, on dispose d'un prédicteur $\hat{f}$. Pour un input X notre prédiction est donc $\hat{Y} = \hat{f}(X)$. On peut regarde le *risque quadratique* $\mathbb{E}(Y - \hat{Y})^2$. Supposons que X et $\hat{f}$ soient fixés, on a

$$\begin{aligned}\mathbb{E}(Y - \hat{Y})^2 &= \mathbb{E}(f(X) - \hat{f}(X) + \epsilon)^2 \\ &= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Erreur réductible}} + \underbrace{\mathbb{V}(\epsilon)}_{\text{Erreur irréductible}}\end{aligned}$$

Dans l'exemple précédent, supposons que l'on dispose d'observation x_0 et y_0 n'ayant pas servies à construire le prédicteur. On aura

$$\mathbb{E}(\hat{f}(x_0) - f(x_0))^2 = \text{Biais}^2(\hat{f}(x_0)) + \mathbb{V}(\hat{f}(x_0))$$

Dans l'exemple précédent, supposons que l'on dispose d'observation x_0 et y_0 n'ayant pas servies à construire le prédicteur. On aura

$$\mathbb{E}(\hat{f}(x_0) - f(x_0))^2 = \text{Biais}^2(\hat{f}(x_0)) + \mathbb{V}(\hat{f}(x_0))$$

- $\mathbb{E}(\hat{f}(x_0) - f(x_0))^2$ représente le MSE moyen
- $\text{Biais}^2(\hat{f}(x_0))$ représente l'erreur d'approximation dans $\mathcal{H}$
- $\mathbb{V}(\hat{f}(x_0))$ représente la variance du prédicteur

En général plus $\mathcal{H}$ est grand, plus de biais est faible et plus la variance est grande.

- Plus le prédicteur $\hat{f}$ est complexe/flexible plus le biais va être faible...

Risque

Biais et Variance

- Plus le prédicteur $\hat{f}$ est complexe/flexible plus le biais va être faible...
- ... mais plus la variance va être élevé

Risque

Biais et Variance

- Plus le prédicteur $\hat{f}$ est complexe/flexible plus le biais va être faible...
- ... mais plus la variance va être élevé

- Plus le prédicteur $\hat{f}$ est complexe/flexible plus le biais va être faible...
- ... mais plus la variance va être élevé

On parle d'équilibre biais variance.

Exemple

Prédire le nombre de click d'une vidéo :

Modèle simple nombre d'abonnés, lolcat

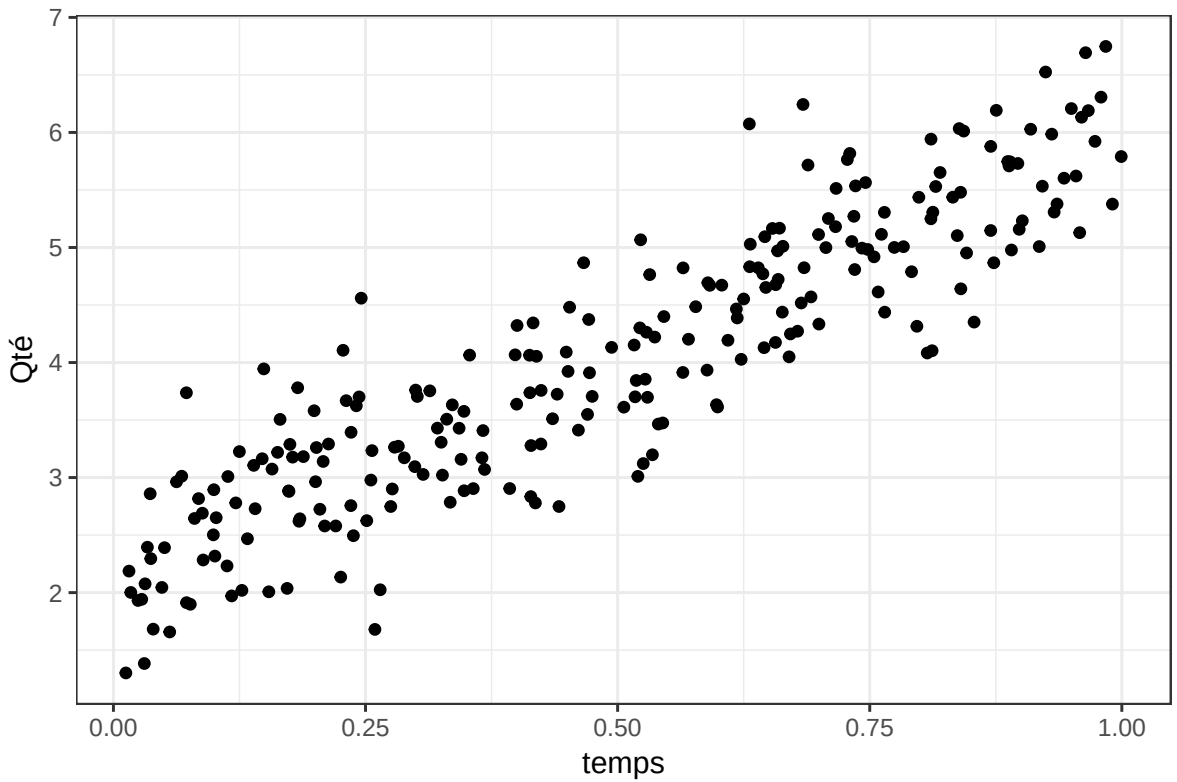
Modèle complexe nombre d'abonnés, lolcat, durée, couleurs, musique, etc.

Exemple

On regarde la quantité de données échangée par un utilisateur d'un site web en fonction du temps passé.

Exemple

On regarde la quantité de données échangée par un utilisateur d'un site web en fonction du temps passé. On observe les données suivantes



Exemple

On va essayé de représenter les données par une fonction affine $f(x) = a + bx$ et on va minimiser le risque quadratique empirique

Problème

On cherche donc a, b solution de

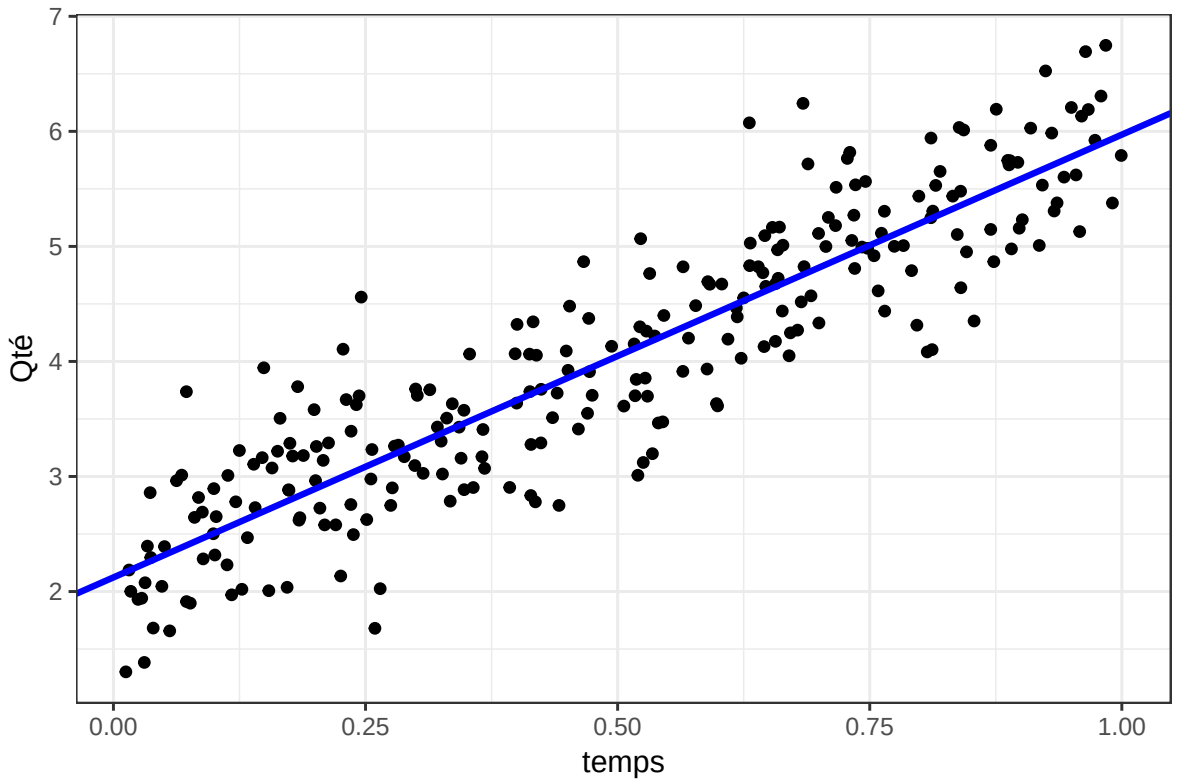
$$\min_{a,b \in \mathbb{R}^2} \frac{1}{n} \sum_{i=1}^{n} (a + bX_i - Y_i)^2$$

Exercice (Estimation)

Résoudre ce problème de minimisation

Exemple

On obtient la fonction de prédiction suivante



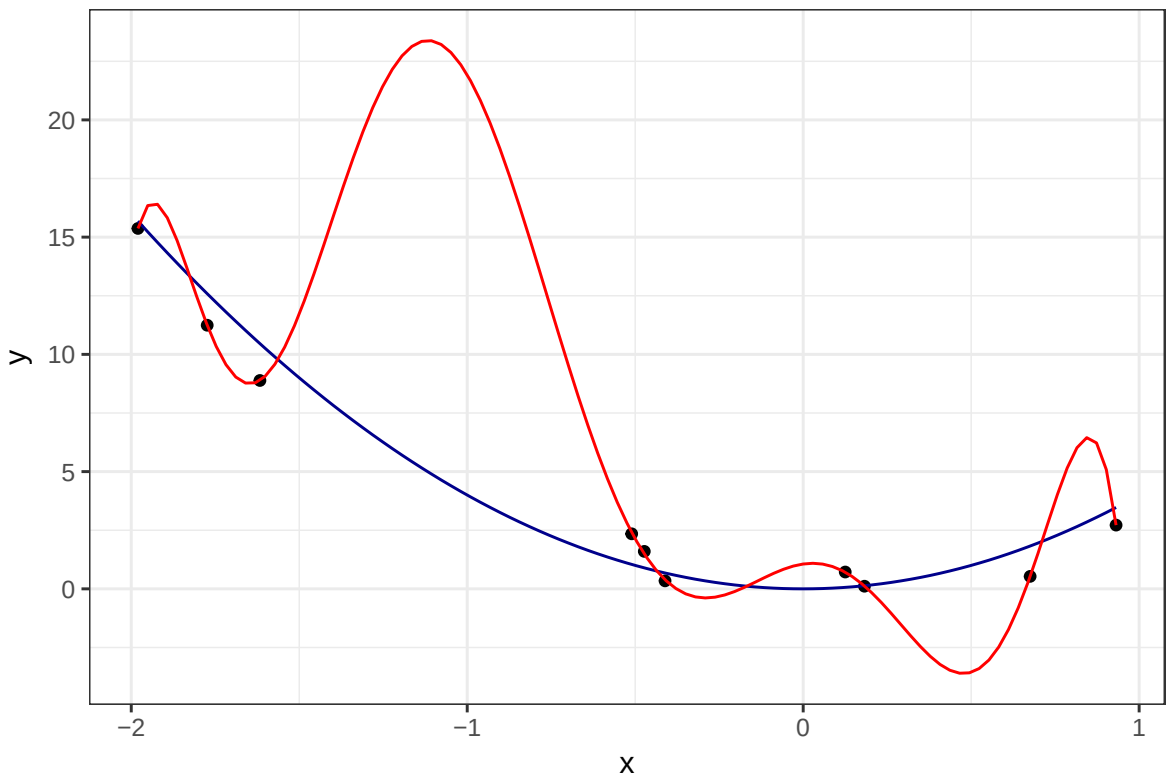
On obtient le prédicteur $\hat{f}(x) = 2.123 + 3.849x$. Calculer le risque empirique pour le risque quadratique ?

Généralisation ?

Minimiser $\hat{R}_n(f)$ n'est pas la même chose que minimser $R(f)$ en général. Cela peut avoir des conséquences dramatiques de confondre les deux risques !

Généralisation ?

Minimiser $\hat{R}_n(f)$ n'est pas la même chose que minimser $R(f)$ en général. Cela peut avoir des conséquences dramatiques de confondre les deux risques !



Pour se prémunir de ce type de problèmes il existe plusieurs approches

Approche Théorique Montrer que dans certains cas les minimiseurs du risque empirique sont des bonnes approximation des minimiseurs du risques

Régularisation Eviter de prendre des prédicteurs trop complexe en les pénalisant dans le risque.

Approche numérique Essayer de trouver une estimation du risque par validation croisée par exemple

Estimation du risque

Pour tout f, $\hat{R}_n(f)$ est un estimateur convergent de $R(f)$ d'après la loi des grands nombres. Mais

$$\hat{R}_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\hat{f}(X_i), Y_i)}_{\text{pas indépendant}}$$

On va construire un estimateur convergent $\tilde{R}(f)$ de $R(f)$.

Principe de la validation croisée

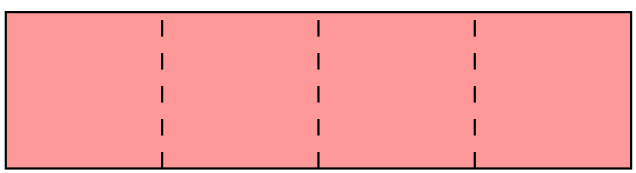
Estimation du risque

Pour tout f, $\hat{R}_n(f)$ est un estimateur convergent de $R(f)$ d'après la loi des grands nombres. Mais

$$\hat{R}_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\hat{f}(X_i), Y_i)}_{\text{pas indépendant}}$$

On va construire un estimateur convergent $\tilde{R}(f)$ de $R(f)$.

Principe de la validation croisée



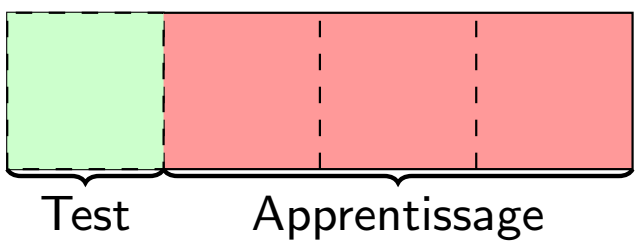
Estimation du risque

Pour tout f, $\hat{R}_n(f)$ est un estimateur convergent de $R(f)$ d'après la loi des grands nombres. Mais

$$\hat{R}_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\hat{f}(X_i), Y_i)}_{\text{pas indépendant}}$$

On va construire un estimateur convergent $\tilde{R}(f)$ de $R(f)$.

Principe de la validation croisée



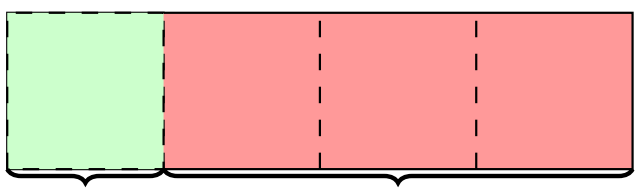
Estimation du risque

Pour tout f, $\hat{R}_n(f)$ est un estimateur convergent de $R(f)$ d'après la loi des grands nombres. Mais

$$\hat{R}_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\hat{f}(X_i), Y_i)}_{\text{pas indépendant}}$$

On va construire un estimateur convergent $\tilde{R}(f)$ de $R(f)$.

Principe de la validation croisée



$$\tilde{R}^1(f) = \sum_{i \in T} L(\hat{f}(X_i), Y_i) \qquad \hat{f}$$

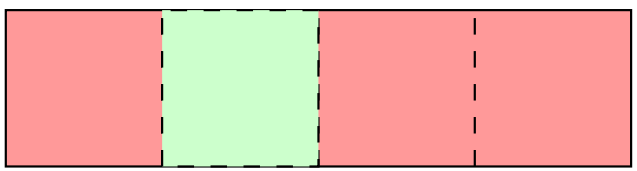
Estimation du risque

Pour tout f, $\hat{R}_n(f)$ est un estimateur convergent de $R(f)$ d'après la loi des grands nombres. Mais

$$\hat{R}_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\hat{f}(X_i), Y_i)}_{\text{pas indépendant}}$$

On va construire un estimateur convergent $\tilde{R}(f)$ de $R(f)$.

Principe de la validation croisée



$\tilde{R}^2(f)$

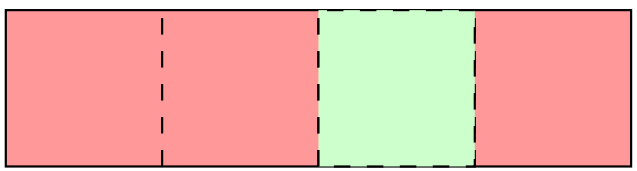
Estimation du risque

Pour tout f, $\hat{R}_n(f)$ est un estimateur convergent de $R(f)$ d'après la loi des grands nombres. Mais

$$\hat{R}_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\hat{f}(X_i), Y_i)}_{\text{pas indépendant}}$$

On va construire un estimateur convergent $\tilde{R}(f)$ de $R(f)$.

Principe de la validation croisée



$\tilde{R}^3(f)$

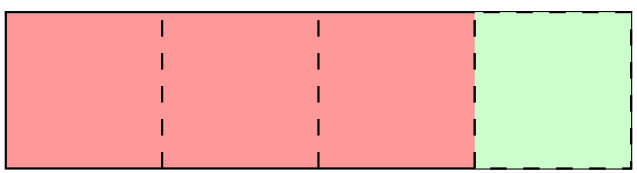
Estimation du risque

Pour tout f, $\hat{R}_n(f)$ est un estimateur convergent de $R(f)$ d'après la loi des grands nombres. Mais

$$\hat{R}_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\hat{f}(X_i), Y_i)}_{\text{pas indépendant}}$$

On va construire un estimateur convergent $\tilde{R}(f)$ de $R(f)$.

Principe de la validation croisée



$\tilde{R}^4(f)$

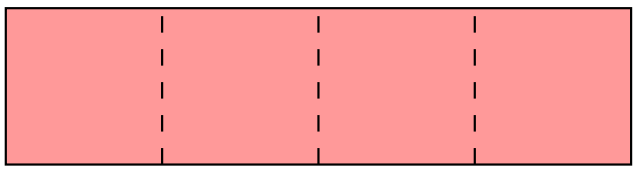
Estimation du risque

Pour tout f, $\hat{R}_n(f)$ est un estimateur convergent de $R(f)$ d'après la loi des grands nombres. Mais

$$\hat{R}_n(\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\hat{f}(X_i), Y_i)}_{\text{pas indépendant}}$$

On va construire un estimateur convergent $\tilde{R}(f)$ de $R(f)$.

Principe de la validation croisée



On estime le risque par $\tilde{R}(f) = \frac{1}{N}\sum_{t=1}^{N} \tilde{R}^t(f)$

Risque et Généralisation
ROC

Dans le cas de la classification, le seul critère du risque n'est pas toujours bien adapté, en particulier dans le cas de classes déséquilibrées.

Risque et Généralisation
ROC

Dans le cas de la classification, le seul critère du risque n'est pas toujours bien adapté, en particulier dans le cas de classes déséquilibrées.

Exemple

On dispose de données (X_i, Y_i), $i = 1, \ldots, 100$ où les Y_i sont binaires $\{0, 1\}$. On suppose que 95 individus sont de la classe 0, le reste étant de la classe 1. Le classifieur $\hat{f}(X) = 0$ aura un risque de 0.05, mais est inutile...

On peut corriger le risque en pondérant les observations, ou en regardant des risques dissymétriques ou encore forcer le modèle à prendre en compte toutes les classes.

Risque et Généralisation
ROC

On va s'intéresser au cas particulier de la classification à deux classes.

Risque et Généralisation
ROC

On va s'intéresser au cas particulier de la classification à deux classes. La plupart des méthodes estiment pour chaque individu un score où une probabilité $\xi(X)$ d'appartenir à une classe (on verra plus loin).

On va s'intéresser au cas particulier de la classification à deux classes. La plupart des méthodes estiment pour chaque individu un score où une probabilité $\xi(X)$ d'appartenir à une classe (on verra plus loin). Le score est ensuite comparé à un seuil s

$$\hat{f}(X) = \mathbb{I}_{\xi(X)>s}$$

Matrice de confusion

Pour chaque seuil s on peut construire une matrice de confusion

	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	$n_{00}(s)$	$n_{01}(s)$
$Y=1$	$n_{10}(s)$	$n_{11}(s)$

	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	$n_{00}(s)$	$n_{01}(s)$
$Y=1$	$n_{10}(s)$	$n_{11}(s)$

A partir de la table de confusion on peut calculer

- Sensibilité $\frac{n_{11}(s)}{n_{1.}}$ C'est le taux d'individus classés positifs à raison
- Spécificité $\frac{n_{00}(s)}{n_{0.}}$ C'est le taux d'individus classés négatifs à raison

On peut aussi regarder le taux de faux positifs $= 1 -$ Spécificité.

Courbe ROC

C'est taux dépendent du seuil s choisi. Plus s est grand, plus la sensibilité diminue, tandis que la spécificité augmente.

	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	$n_{00}(s)$	$n_{01}(s)$
$Y=1$	$n_{10}(s)$	$n_{11}(s)$

A partir de la table de confusion on peut calculer

- Sensibilité $\frac{n_{11}(s)}{n_{1.}}$ C'est le taux d'individus classés positifs à raison
- Spécificité $\frac{n_{00}(s)}{n_{0.}}$ C'est le taux d'individus classés négatifs à raison

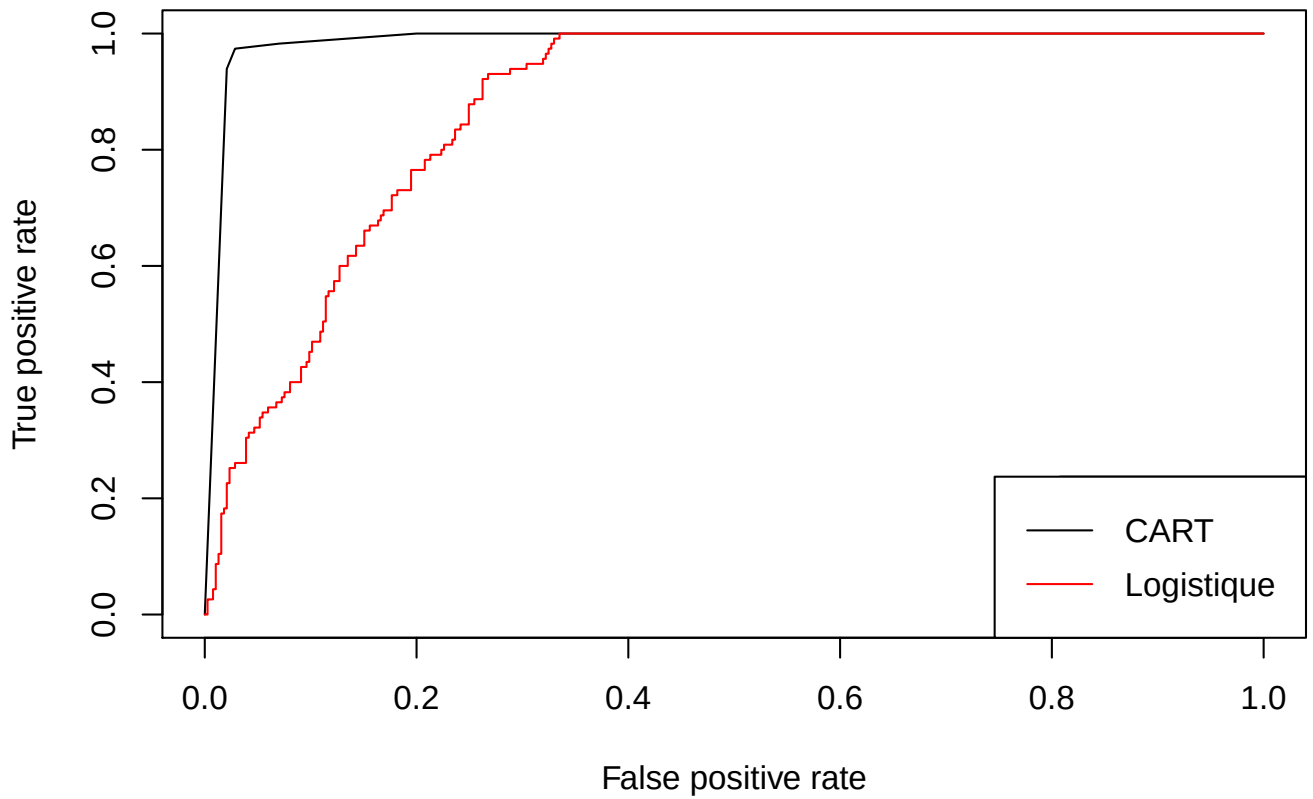
On peut aussi regarder le taux de faux positifs $= 1 -$ Spécificité.

Courbe ROC

C'est taux dépendent du seuil s choisi. Plus s est grand, plus la sensibilité diminue, tandis que la spécificité augmente. Un bon modèle a à la fois une *grande sensibilité* et une *grande spécificité*. On peut représenter ce lien par la courbe ROC qui trace la sensibilité en fonction du taux de faux positifs.

Risque et Généralisation
ROC

Plus la courbe est proche du carré, meilleur est le classifieur.



Chap.2

La Régression

Lorsque l'on souhaite prédire un output *continu* $Y \in \mathbb{R}$ en fonction de co-variables $X \in \mathcal{X}$ (continues ou non), on peut utiliser un modèle de régression.

Lorsque l'on souhaite prédire un output *continu* $Y \in \mathbb{R}$ en fonction de co-variables $X \in \mathcal{X}$ (continues ou non), on peut utiliser un modèle de régression.

Le modèle de régression

On appelle modèle de régression un modèle de la forme suivante

$$\underbrace{Y}_{\text{output}} = \underbrace{f(X)}_{\text{predicteur}} + \underbrace{\epsilon}_{\text{erreur}}$$

Lorsque l'on souhaite prédire un output *continu* $Y \in \mathbb{R}$ en fonction de co-variables $X \in \mathcal{X}$ (continues ou non), on peut utiliser un modèle de régression.

Le modèle de régression

On appelle modèle de régression un modèle de la forme suivante

$$\underbrace{Y}_{\text{output}} = \underbrace{f(X)}_{\text{predicteur}} + \underbrace{\epsilon}_{\text{erreur}}$$

Le but va être *d'apprendre* le prédicteur f à partir de l'observation de X et Y. On prédira pour l'observation de x la valeur $f(x)$.

Lorsque l'on souhaite prédire un output *continu* $Y \in \mathbb{R}$ en fonction de co-variables $X \in \mathcal{X}$ (continues ou non), on peut utiliser un modèle de régression.

Le modèle de régression

On appelle modèle de régression un modèle de la forme suivante

$$\underbrace{Y}_{\text{output}} = \underbrace{f(X)}_{\text{predicteur}} + \underbrace{\epsilon}_{\text{erreur}}$$

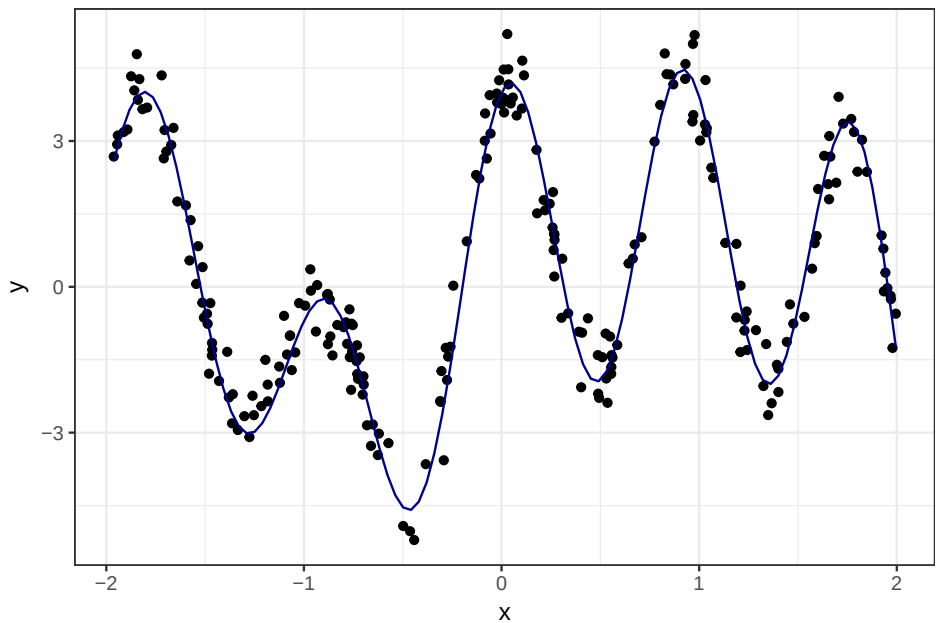
Le but va être *d'apprendre* le prédicteur f à partir de l'observation de X et Y. On prédira pour l'observation de x la valeur $f(x)$.

Note sur le terme d'erreur

Le terme ϵ sert *à la fois* à modéliser les possibles erreurs de mesure, mais aussi à représenter la partie de l'output qui ne peut pas être expliquée par notre prédicteur f.

Le modèle de régression est en général bien adapté lorsque les données fluctuent autour d'une tendance moyenne.

Le modèle de régression est en général bien adapté lorsque les données fluctuent autour d'une tendance moyenne. Mais la tendance moyenne peut être assez complexe...



De plus en général la co-variable X est multi-dimensionnelle ce qui peut rendre les représentations graphiques hasardeuses...

Hypothèses sur les erreurs

On supposera en général que les erreurs sont indépendantes des co-variables X, ainsi le seul lien entre X et Y est par le biais du prédicteur f.

Hypothèses sur les erreurs

On supposera en général que les erreurs sont indépendantes des co-variables X, ainsi le seul lien entre X et Y est par le biais du prédicteur f.

Apprentissage

On va se donner une classe de fonction $\mathcal{H}$ et on va chercher le meilleur prédicteur au sein de cette classe

- prédicteur linéaires
- enrichissement dans le cadre de la régression linéaire

Hypothèses sur les erreurs

On supposera en général que les erreurs sont indépendantes des co-variables X, ainsi le seul lien entre X et Y est par le biais du prédicteur f.

Apprentissage

On va se donner une classe de fonction $\mathcal{H}$ et on va chercher le meilleur prédicteur au sein de cette classe

- prédicteur linéaires
- enrichissement dans le cadre de la régression linéaire

Une difficulté va être de trouvé la bonne classe de fonctions $\mathcal{H}$.

Dans le modèle de régression linéaire on suppose qu'il existe une relation **linéaire** entre les co-variables et l'output.

Les données

On dispose de n couples (X_i, Y_i) de co-variables et output. On suppose que les co-variables sont sous forme d'un vecteur de dimension p. Dans un premier temps on supposera que les co-variables sont continues (réelles)

$$X_i = (X_i^1, X_i^2, \ldots, X_i^p)$$

Soit $\mathbf{X}$ la matrice des co-variables

$$\mathbf{X} = \begin{pmatrix} 1 & X_1^1 & X_1^2 & \ldots & X_1^p \\ 1 & X_2^1 & X_2^2 & \ldots & X_2^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_n^1 & X_n^2 & \ldots & X_n^p \end{pmatrix}$$

Les outputs Y sont aussi représentés sous la forme d'un vecteur $Y = (Y_1, \ldots, Y_n)'$

On va chercher des predicteurs de la forme $f(x) = x'\theta$ où $\theta \in \mathbb{R}^p$. Cela revient à choisir $\mathcal{H}$ l'ensemble des fonction affines de $\mathbb{R}^p$ dans $\mathbb{R}$.

On va chercher des predicteurs de la forme $f(x) = x'\theta$ où $\theta \in \mathbb{R}^p$. Cela revient à choisir $\mathcal{H}$ l'ensemble des fonction affines de $\mathbb{R}^p$ dans $\mathbb{R}$.

Hypothèses sur les données

On va supposer que la Matrice des données $\mathbf{X}$ est de plein rang (i.e. $\mathbf{X}'\mathbf{X}$ est inversible). Sinon cela signifie que l'une des variable (ou une observation) peut être reconstruite par une combinaison linéaire des autres.

On va chercher des predicteurs de la forme $f(x) = x'\theta$ où $\theta \in \mathbb{R}^p$. Cela revient à choisir $\mathcal{H}$ l'ensemble des fonction affines de $\mathbb{R}^p$ dans $\mathbb{R}$.

Hypothèses sur les données

On va supposer que la Matrice des données $\mathbf{X}$ est de plein rang (i.e. $\mathbf{X}'\mathbf{X}$ est inversible). Sinon cela signifie que l'une des variable (ou une observation) peut être reconstruite par une combinaison linéaire des autres. Cette hypothèse nous oblige à nous limiter au cas $p \leq n$. On ne peut pas avoir plus de variables que d'observations.

On va chercher des predicteurs de la forme $f(x) = x'\theta$ où $\theta \in \mathbb{R}^p$. Cela revient à choisir $\mathcal{H}$ l'ensemble des fonction affines de $\mathbb{R}^p$ dans $\mathbb{R}$.

Hypothèses sur les données

On va supposer que la Matrice des données $\mathbf{X}$ est de plein rang (i.e. $\mathbf{X}'\mathbf{X}$ est inversible). Sinon cela signifie que l'une des variable (ou une observation) peut être reconstruite par une combinaison linéaire des autres. Cette hypothèse nous oblige à nous limiter au cas $p \leq n$. On ne peut pas avoir plus de variables que d'observations.

Cas des co-variables discrètes

Si on a une co-variables qualitative Z prenant des valeurs $(z_1, \dots z_k)$, on peut créer k variables réelles $\tilde{Z}^j$ telles que $\tilde{Z}^j = \mathbb{I}_{Z=z_j}$. Pour que la matrice $\mathbf{X}$ soit bien de plein rang, on n'ajoutera dans les colonnes que $k-1$ de ces nouvelles variables. La modalité que l'on retirera sera prise comme modalité de référence.

On va chercher à minimiser le risque empirique pour une perte quadratique

Moindres Carrés

On cherche à minimiser

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i'\theta)^2 = \frac{1}{n}(Y - \mathbf{X}\theta)'(Y - \mathbf{X}\theta)$$

On va chercher à minimiser le risque empirique pour une perte quadratique

Moindres Carrés

On cherche à minimiser

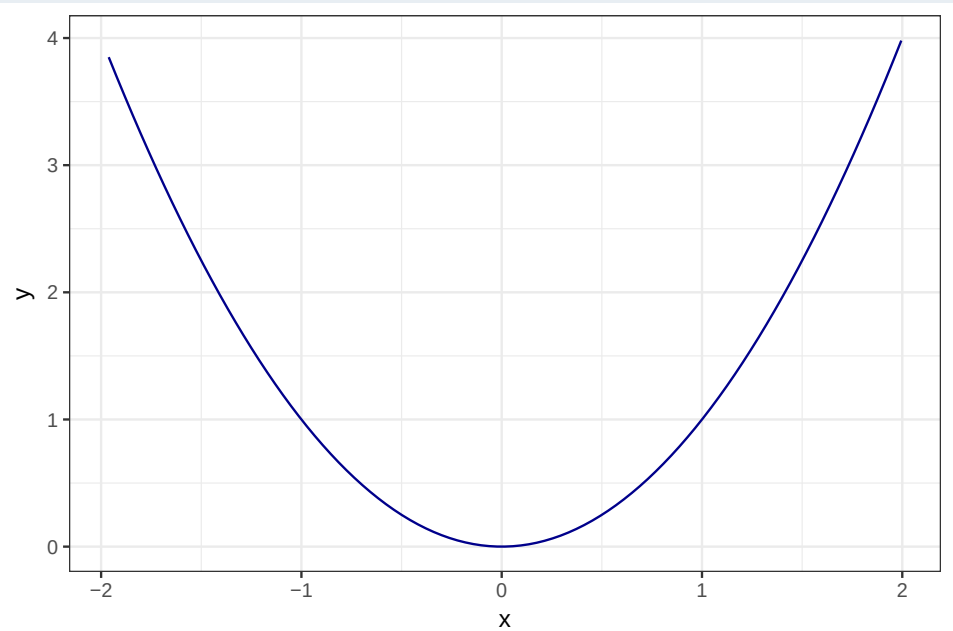
$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i'\theta)^2 = \frac{1}{n}(Y - \mathbf{X}\theta)'(Y - \mathbf{X}\theta)$$

Mais pourquoi les moindres carrés ?

- Pour des raisons pratiques : C'est une perte convexe ce qui facilite le problème d'optimisation.
- Cette perte à des liens avec l'estimation par maximum de vraisemblance pour des erreurs Gaussiennes.

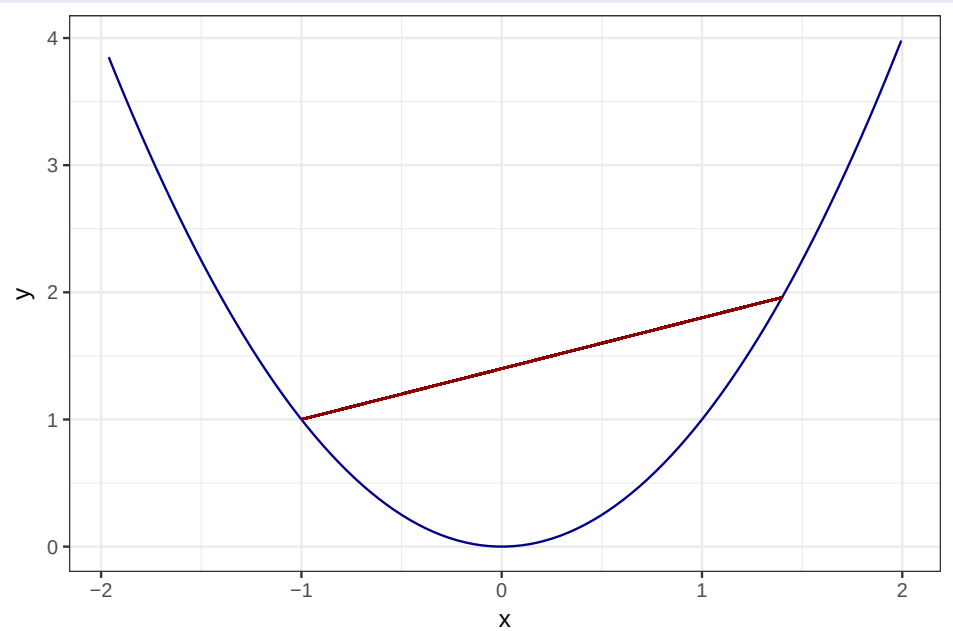
Fonction convexe

Intuitivement une fonction convexe est une fonction qui "regarde vers le haut"



Fonction convexe

Intuitivement une fonction convexe est une fonction qui "regarde vers le haut"



f est en dessous de ses cordes i.e.

$$\forall x, y \in \mathcal{X}, \forall \lambda \in [0,1] \; f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

Propriétés des fonctions convexes

- Une fonction convexe admet au moins un minimum global
- La somme de deux fonction convexe est convexe
- La composé de fonctions convexe est convexe

Exercice (Risque quadratique)

Montrer que le risque quadratique est une fonction convexe en θ

On va maintenant calculer le minimiseur du risque quadratique. Pour cela on va utiliser les outils classique d'optimisation.

Rappels d'optimisation convexe

Soit g une fonction de $\mathbb{R}^p$ dans $\mathbb{R}$ $\mathcal{C}^2$. $g : (x_1, \ldots, x_p) \to g(x_1, \ldots, x_p)$

- Le laplacien (ou gradient) est le vecteur

$$\nabla g := \left(\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}, \ldots, \frac{\partial g}{\partial x_p} \right)'$$

- La Hessienne de g est la matrice donnée par

$$H(g) := \begin{pmatrix} \frac{\partial^2 g}{\partial x_1^2} & \frac{\partial^2 g}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 g}{\partial x_1 \partial x_p} \\ \frac{\partial^2 g}{\partial x_2 \partial x_1} & \frac{\partial^2 g}{\partial x_2^2} & \cdots & \frac{\partial^2 g}{\partial x_2 \partial x_p} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 g}{\partial x_p \partial x_1} & \frac{\partial^2 g}{\partial x_p \partial x_2} & \cdots & \frac{\partial^2 g}{\partial x_p^2} \end{pmatrix}$$

Optimisation d'une fonction convexe

Pour une fonction convexe son minimum est donnée atteint au point $(x_1^*, \dots x_p^*)$ tel que

- $\nabla g(x_1^*, \dots, x_p^*) = 0$
- $H(g)(x_1^*, \dots, x_p^*)$ est définie positive

Optimisation d'une fonction convexe

Pour une fonction convexe son minimum est donnée atteint au point $(x_1^*, \dots x_p^*)$ tel que

- $\nabla g(x_1^*, \dots, x_p^*) = 0$
- $H(g)(x_1^*, \dots, x_p^*)$ est définie positive

Pour la méthode des moindres carrés on cherche à minimiser

$$g(\theta) = (Y - \mathbf{X}\theta)'(Y - \mathbf{X}\theta) = Y'Y + \theta'\mathbf{X}'\mathbf{X}\theta - 2\theta'\mathbf{X}'Y$$

On calcule le gradient

$$\nabla g(\theta) = 2\mathbf{X}'\mathbf{X}\theta - 2\mathbf{X}'Y$$

Optimisation d'une fonction convexe

Pour une fonction convexe son minimum est donnée atteint au point $(x_1^*, \dots x_p^*)$ tel que

- $\nabla g(x_1^*, \dots, x_p^*) = 0$
- $H(g)(x_1^*, \dots, x_p^*)$ est définie positive

Pour la méthode des moindres carrés on cherche à minimiser

$$g(\theta) = (Y - \mathbf{X}\theta)'(Y - \mathbf{X}\theta) = Y'Y + \theta'\mathbf{X}'\mathbf{X}\theta - 2\theta'\mathbf{X}'Y$$

On calcule le gradient

$$\nabla g(\theta) = 2\mathbf{X}'\mathbf{X}\theta - 2\mathbf{X}'Y$$

On en déduit

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

Le minimiseur du risque empirique pour les prédicteur de la forme $f(x) = x'\theta$ est

$$\hat{f}(x) = x'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y.$$

Le minimiseur du risque empirique pour les prédicteur de la forme $f(x) = x'\theta$ est

$$\hat{f}(x) = x'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y.$$

Biais et variance

Sous l'hypothèse que $\mathbb{E}(\epsilon|X) = 0$ on a

$$R(\hat{f}) = \mathbb{E}_X\left((f(X) - \hat{f}(X))^2\right) + \mathbb{E}(\epsilon^2)$$

Régression linéaire

Lien avec le maximum de vraisemblance

Repartons du modèle de régression linéaire

$$Y = \mathbf{X}\beta + \epsilon$$

et supposons $\epsilon \sim \mathcal{N}(0, \sigma I)$. On suppose de plus $\mathbf{X}$ fixé

Régression linéaire

Lien avec le maximum de vraisemblance

Repartons du modèle de régression linéaire

$$Y = \mathbf{X}\beta + \epsilon$$

et supposons $\epsilon \sim \mathcal{N}(0, \sigma I)$. On suppose de plus $\mathbf{X}$ fixé
Vraisemblance du modèle

$$\mathcal{L}(Y, \beta, \sigma) = \frac{1}{(\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2}(Y-\mathbf{X}\beta)'(Y-\mathbf{X}\beta)}$$

Régression linéaire

Lien avec le maximum de vraisemblance

Repartons du modèle de régression linéaire

$$Y = \mathbf{X}\beta + \epsilon$$

et supposons $\epsilon \sim \mathcal{N}(0, \sigma I)$. On suppose de plus $\mathbf{X}$ fixé
Vraisemblance du modèle

$$\mathcal{L}(Y, \beta, \sigma) = \frac{1}{(\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2}(Y-\mathbf{X}\beta)'(Y-\mathbf{X}\beta)}$$

Estimateur du maximum de vraisemblance

Soit $\tilde{\beta} = \arg\max_\beta \mathcal{L}(Y, \beta, \sigma)$. Maximiser la vraisemblance en β est équivalent à minimiser

$$(Y - \mathbf{X}\beta)'(Y - \mathbf{X}\beta)$$

Régression linéaire

Lien avec le maximum de vraisemblance

Repartons du modèle de régression linéaire

$$Y = \mathbf{X}\beta + \epsilon$$

et supposons $\epsilon \sim \mathcal{N}(0, \sigma I)$. On suppose de plus $\mathbf{X}$ fixé
Vraisemblance du modèle

$$\mathcal{L}(Y, \beta, \sigma) = \frac{1}{(\sigma^2)^{n/2}} e^{-\frac{1}{2\sigma^2}(Y-\mathbf{X}\beta)'(Y-\mathbf{X}\beta)}$$

Estimateur du maximum de vraisemblance

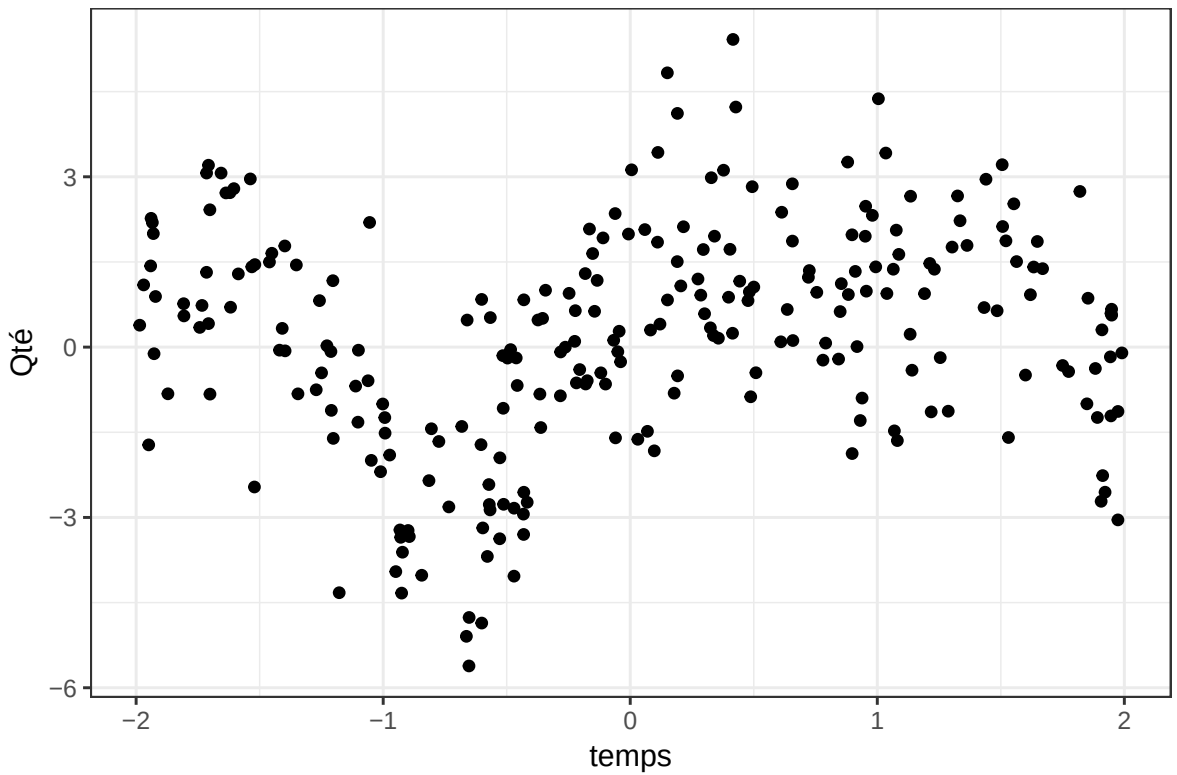
Soit $\tilde{\beta} = \arg\max_\beta \mathcal{L}(Y, \beta, \sigma)$. Maximiser la vraisemblance en β est équivalent à minimiser

$$(Y - \mathbf{X}\beta)'(Y - \mathbf{X}\beta)$$

donc $\tilde{\beta} = \hat{\beta}$.

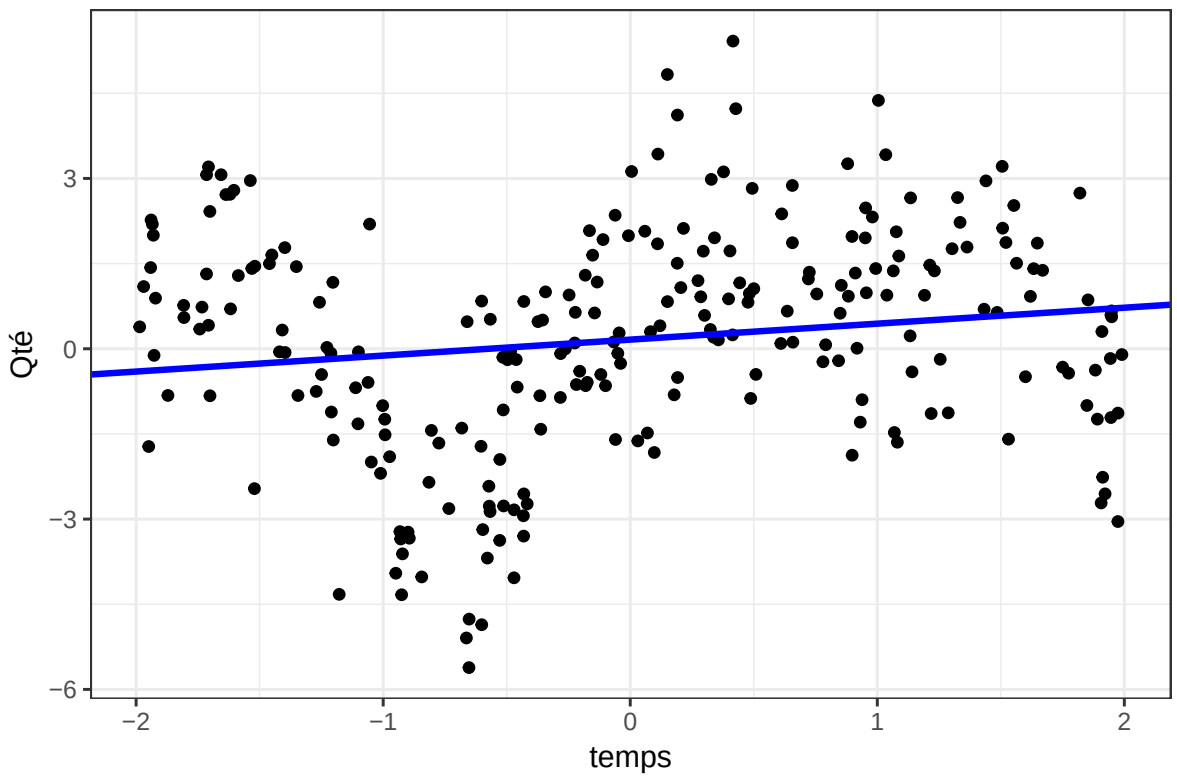
Enrichissement
Des prédicteurs plus flexibles

Le modèle de régression linéaire est un cadre facile à utiliser mais assez peu flexible. On est souvent confronté à des *effets* qui sont non-linéaires.



Enrichissement
Des prédicteurs plus flexibles

Utiliser des prédicteurs linéaires dans ce cas revient à introduire un biais important (et donc un risque de prédiction plus élevé...)



Enrichissement

Enrichir l'espace d'hypothèses

On va enrichir l'espace des hypothèses $\mathcal{H}$ c'est a dire qu'on va chercher une famille de prédicteurs plus grande. Dans le cadre de la régression linéaire, on peut regarder des prédicteurs de la forme

$$f(x) = \sum_{k=1}^{K} \theta_k \phi_k(x)$$

où ϕ est une fonction de $\mathbb{R}^p \to \mathbb{R}$

Enrichissement

Enrichir l'espace d'hypothèses

On va enrichir l'espace des hypothèses $\mathcal{H}$ c'est a dire qu'on va chercher une famille de prédicteurs plus grande. Dans le cadre de la régression linéaire, on peut regarder des prédicteurs de la forme

$$f(x) = \sum_{k=1}^{K} \theta_k \phi_k(x)$$

où ϕ est une fonction de $\mathbb{R}^p \to \mathbb{R}$ On a donc le modèle suivant

$$Y = \Phi(\mathbf{X})\beta + \epsilon$$

avec

$$\Phi(\mathbf{X}) = \begin{pmatrix} \phi_1(X_1) & \phi_2(X_1) & \dots & \phi_K(X_1) \\ \phi_1(X_2) & \phi_2(X_2) & \dots & \phi_K(X_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(X_n) & \phi_2(X_n) & \dots & \phi_K(X_n) \end{pmatrix}$$

On a donc $\hat{\beta} = (\Phi(\mathbf{X})'\Phi(\mathbf{X}))^{-1}\Phi(\mathbf{X})'Y$

Quelques exemples de ϕ

- Polynômes $\phi_k(x) = x^k$
- Gaussien $\phi_k(x) = e^{-\frac{1}{2\sigma^2}||x-\mu_k||^2}$
- Splines
- etc.

Le choix de la famille dépend beaucoup du problème que l'on considère.

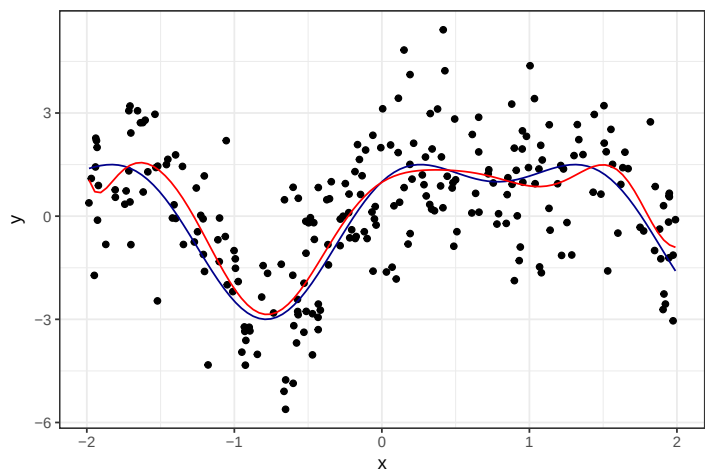


FIGURE: Régression avec une famille de polynomes

Enrichissement
Enrichir l'espace d'hypothèses

Quelques exemples de ϕ

- Polynômes $\phi_k(x) = x^k$
- Gaussien $\phi_k(x) = e^{-\frac{1}{2\sigma^2}||x-\mu_k||^2}$
- Splines
- etc.

Le choix de la famille dépend beaucoup du problème que l'on considère.

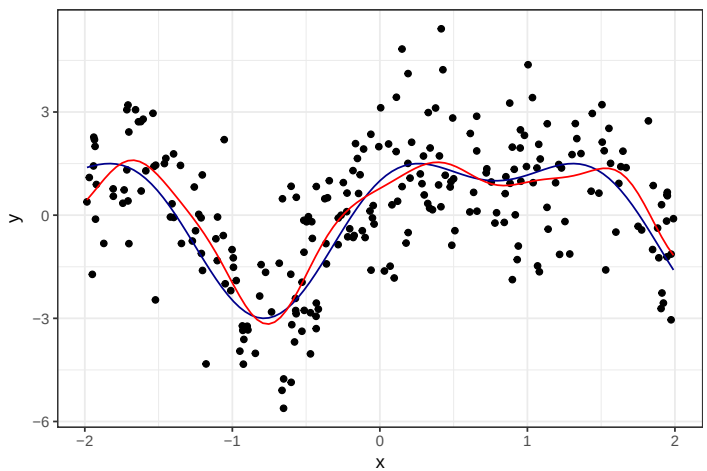
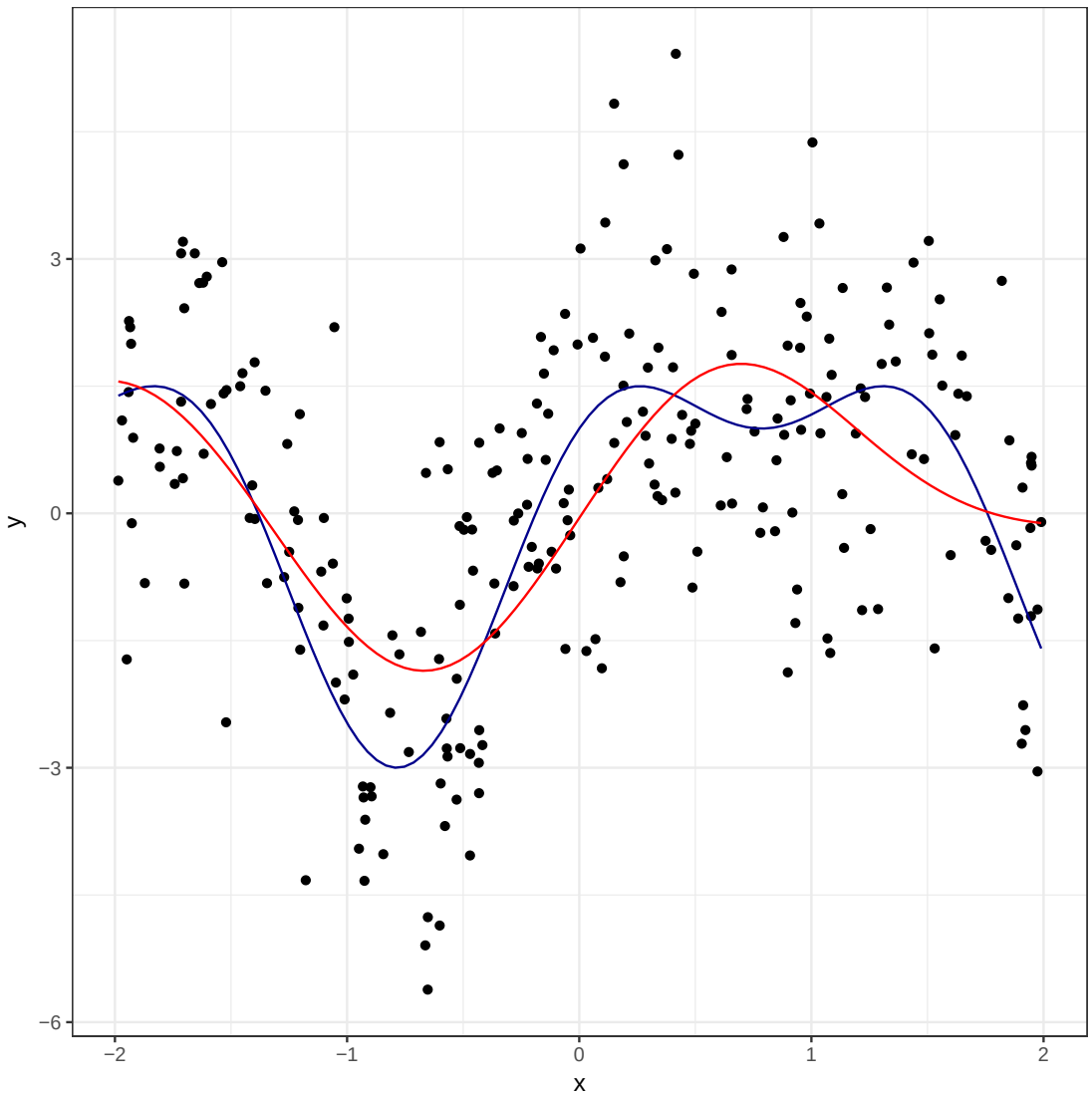


FIGURE: Régression avec des noyaux Gaussiens

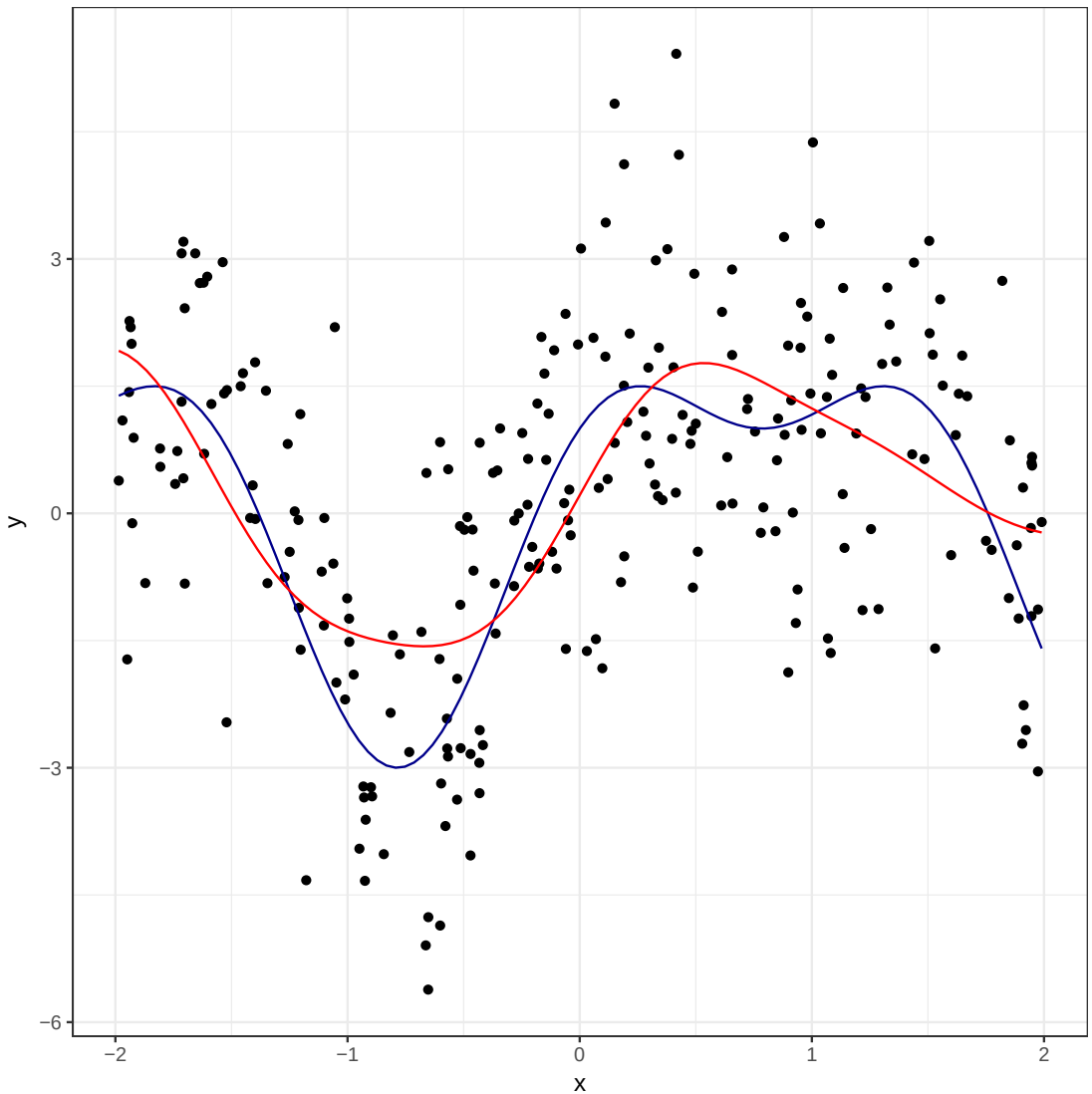
Enrichissement

Comment choisir ?



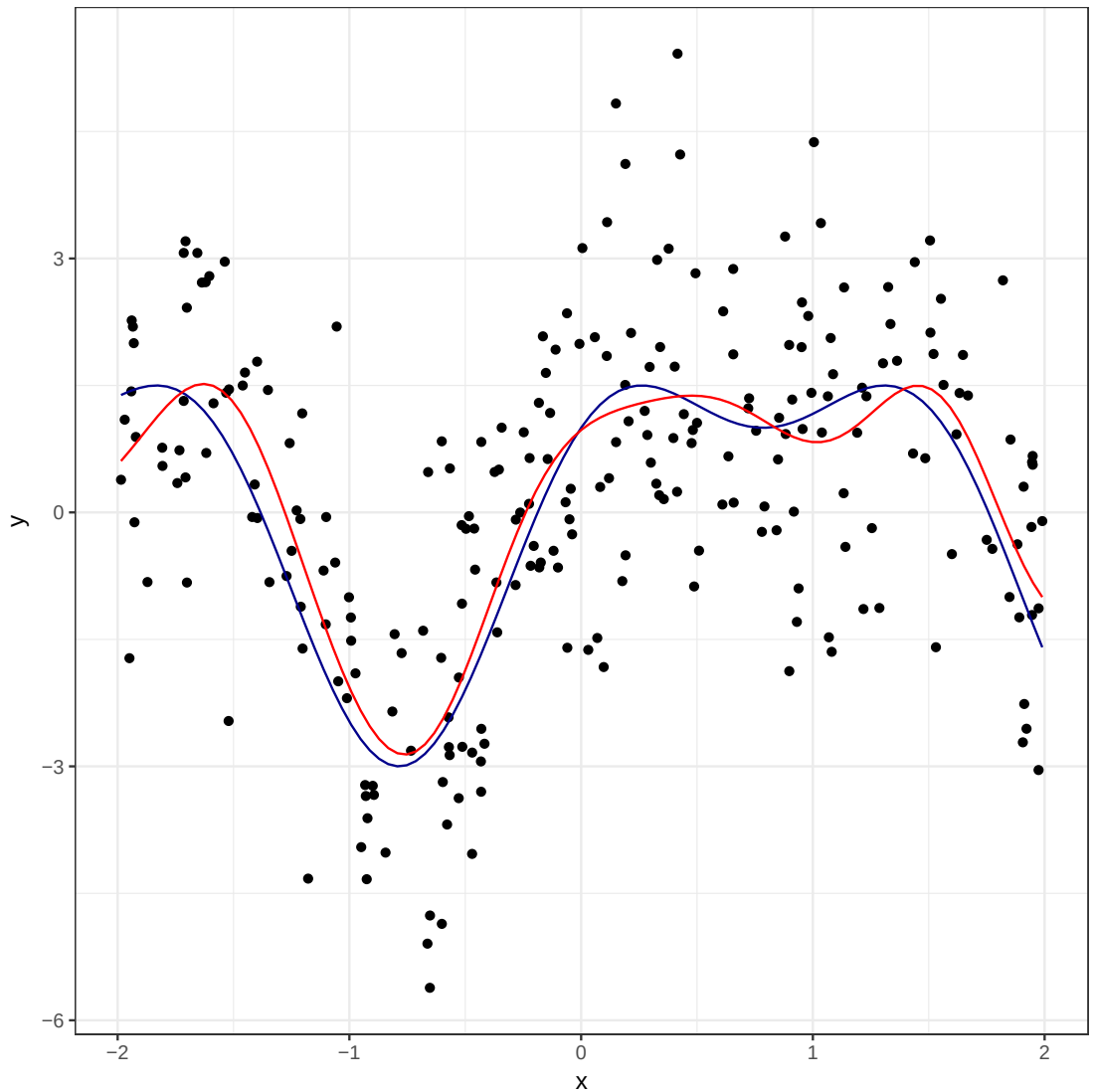
Enrichissement

Comment choisir ?



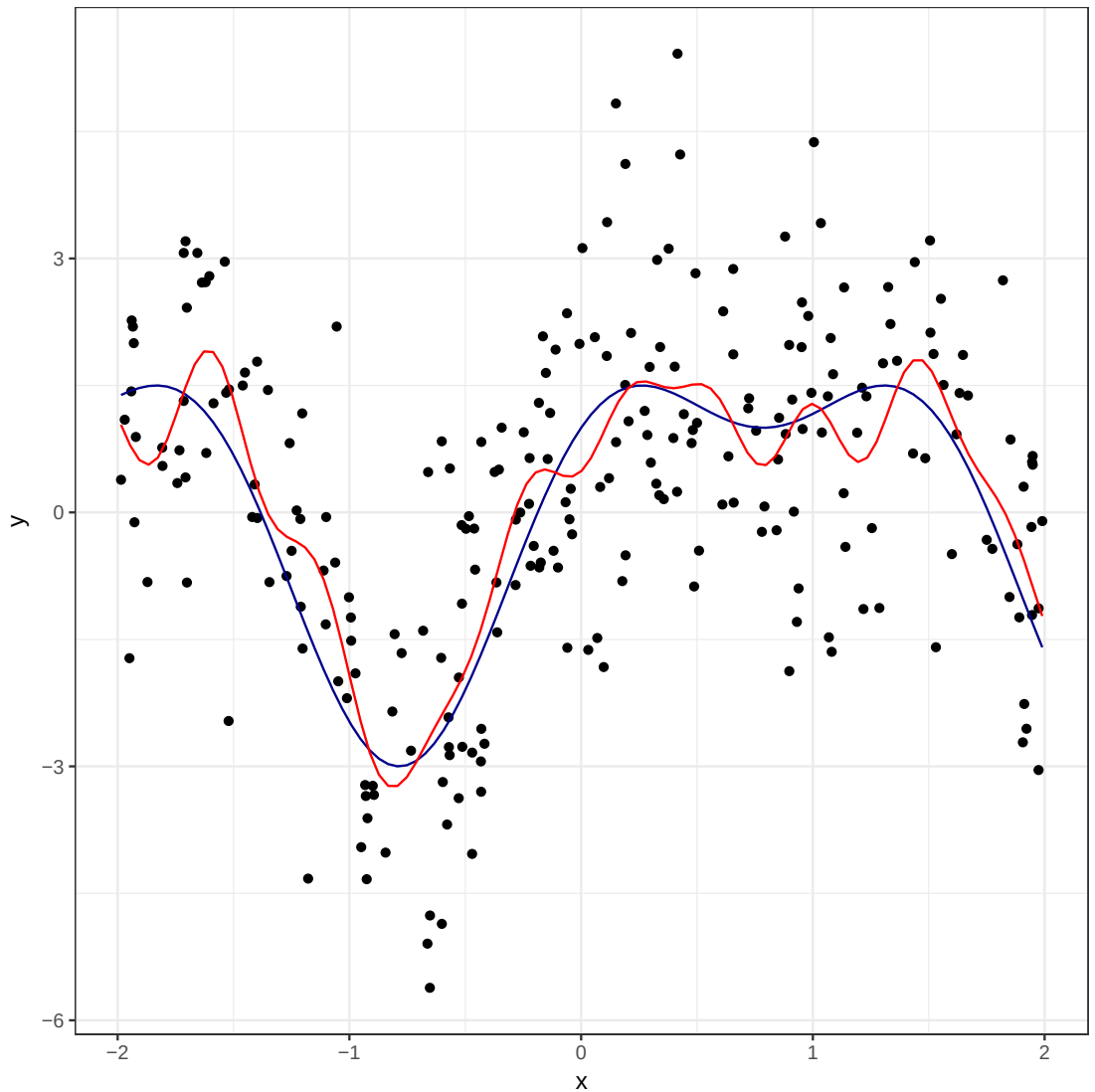
Enrichissement

Comment choisir ?



Enrichissement

Comment choisir ?



Enrichissement
Comment choisir ?

Sur Apprentissage

Il y a un risque de sur-apprentissage d'autant plus que la famille d'hypothèse considérée est flexible. On va donc contrôler le risque par validation croisée.

Enrichissement
Comment choisir ?

Sur Apprentissage

Il y a un risque de sur-apprentissage d'autant plus que la famille d'hypothèse considérée est flexible. On va donc contrôler le risque par validation croisée.

Il existe aussi des résultats théoriques permetant de choisir le nombre de fonction ϕ_k pour garantir un niveau de risque.

Enrichissement
Comment choisir ?

Sur Apprentissage

Il y a un risque de sur-apprentissage d'autant plus que la famille d'hypothèse considérée est flexible. On va donc contrôler le risque par validation croisée.

Il existe aussi des résultats théoriques permetant de choisir le nombre de fonction ϕ_k pour garantir un niveau de risque.

Exercice (Mise en œuvre)

Etudier l'effet du sur apprentissage sur des données simulée en estimant un prédicteur avec un noyau polynomial.

Grande dimension ?

Fléau de la dimension

Les méthodes précédentes sont bien adaptées dans le cas ou les inputs $\mathbf{X}$ sont de petite dimension. Il existe cependant des moyen de contourner le problème notamment en faisant des hypothèses de structure sur les *familles d'hypothèse* $\mathcal{H}$.

Grande dimension ?

Fléau de la dimension

Les méthodes précédentes sont bien adaptées dans le cas ou les inputs $\mathbf{X}$ sont de petite dimension. Il existe cependant des moyen de contourner le problème notamment en faisant des hypothèses de structure sur les *familles d'hypothèse* $\mathcal{H}$.
On se place dans le cas de la régression habituelle

$$Y_i = f(\mathbf{X}_i) + \epsilon_i$$

où les ϵ_i sont iid centrés et de variance σ^2, les $\mathbf{X}_i \in \mathbb{R}^d$.

Grande dimension ?

Modèles additifs généralisés

Une première approche est de considérer des prédicteurs f additifs

$$f(\mathbf{X}_i) = \alpha + f_1(X_{i,1}) + \cdots + f_d(X_{i,d})$$

Pour pouvoir garantir l'unicité d'une telle écriture il faut imposer

$$\int_{\mathbb{R}} f_j(x_j) dP_{X_j}(x_j) = 0$$

On estime chacune des fonctions unidimensionnelle à l'aide d'une des méthodes vues précédemment. On va par exemple minimiser le risque empirique

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(\mathbf{X}_i))^2$$

sous la contrainte $\sum_{i=1}^{n} f_j(X_{i,j}) = 0$.

Grande dimension ?

Modèles additifs généralisés

Une première approche est de considérer des prédicteurs f additifs

$$f(\mathbf{X}_i) = \alpha + f_1(X_{i,1}) + \cdots + f_d(X_{i,d})$$

Pour pouvoir garantir l'unicité d'une telle écriture il faut imposer

$$\int_{\mathbb{R}} f_j(x_j) dP_{X_j}(x_j) = 0$$

On estime chacune des fonctions unidimensionnelle à l'aide d'une des méthodes vues précédemment. On va par exemple minimiser le risque empirique

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(\mathbf{X}_i))^2$$

sous la contrainte $\sum_{i=1}^{n} f_j(X_{i,j}) = 0$. Dans ce cas on a directement $\hat{\alpha} = \bar{Y}$. Une manière de minimiser est de faire une descente de gradient composante par composante.

Grande dimension ?
RKHS

Un exemple élémentaire de *machine à noyau* est le Kernel Regression Least Square

- C'est une méthode qui fourni des prédicteurs non linéaire
- C'est une méthode de régularisation qui s'appuie sur la méthode des moindre carrés pénalisé
- Elle est utile quand les $\mathbf{X}_i$ sont de "grande" dimension.

Grande dimension ?
RKHS

Un exemple élémentaire de *machine à noyau* est le Kernel Regression Least Square

- C'est une méthode qui fourni des prédicteurs non linéaire
- C'est une méthode de régularisation qui s'appuie sur la méthode des moindre carrés pénalisé
- Elle est utile quand les $\mathbf{X}_i$ sont de "grande" dimension.

On se donne un noyau K défini sur $\mathbb{R}^p$ symétrique, semi-défini positif.

$$K(\mathbf{X}_1, \mathbf{X}_2) = K(\mathbf{X}_2, \mathbf{X}_1), \ \sum_{i,j=1}^{n} K(\mathbf{X}_i, \mathbf{X}_j) \geq 0$$

Grande dimension ?
RKHS

Un exemple élémentaire de *machine à noyau* est le Kernel Regression Least Square

- C'est une méthode qui fourni des prédicteurs non linéaire
- C'est une méthode de régularisation qui s'appuie sur la méthode des moindre carrés pénalisé
- Elle est utile quand les $\mathbf{X}_i$ sont de "grande" dimension.

On se donne un noyau K défini sur $\mathbb{R}^p$ symétrique, semi-défini positif.

$$K(\mathbf{X}_1, \mathbf{X}_2) = K(\mathbf{X}_2, \mathbf{X}_1), \ \sum_{i,j=1}^{n} K(\mathbf{X}_i, \mathbf{X}_j) \geq 0$$

On va chercher un prédicteur de la forme

$$f(x) = \sum_{i=1}^{n} c_j K(X_j, x)$$

On notera $\mathbf{c} = (c_1, \ldots, c_n)$

Grande dimension ?
RKHS

Soit $\mathbb{K}$ la matrice définie par $\mathbb{K}_{i,j} = K(\mathbf{X}_i, \mathbf{X}_j)$. On va chercher à minimiser un critère des moindres carrés pénalisé

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - f(\mathbf{X}_i))^2 + \lambda||f||^2_{\mathbb{K}}$$

où $||f||^2_{\mathbb{K}} = \sum_{i,j=1}^{n} c_i c_j K(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{c}'\mathbb{K}\mathbf{c}$.

Grande dimension ?
RKHS

Soit $\mathbb{K}$ la matrice définie par $\mathbb{K}_{i,j} = K(\mathbf{X}_i, \mathbf{X}_j)$. On va chercher à minimiser un critère des moindres carrés pénalisé

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - f(\mathbf{X}_i))^2 + \lambda||f||_{\mathbb{K}}^2$$

où $||f||_{\mathbb{K}}^2 = \sum_{i,j=1}^{n} c_i c_j K(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{c}'\mathbb{K}\mathbf{c}$. On a une solution explicite à ce problème d'optimisation

$$\hat{\mathbf{c}} = (\mathbb{K} + \lambda I_n)^{-1} Y$$

Soit $\mathbb{K}$ la matrice définie par $\mathbb{K}_{i,j} = K(\mathbf{X}_i, \mathbf{X}_j)$. On va chercher à minimiser un critère des moindres carrés pénalisé

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - f(\mathbf{X}_i))^2 + \lambda||f||_{\mathbb{K}}^2$$

où $||f||_{\mathbb{K}}^2 = \sum_{i,j=1}^{n} c_i c_j K(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{c}'\mathbb{K}\mathbf{c}$. On a une solution explicite à ce problème d'optimisation

$$\hat{\mathbf{c}} = (\mathbb{K} + \lambda I_n)^{-1} Y$$

Un intérêt de cette méthode est quelle peut s'appliquer à tout type d'inputs dès que l'on peut définir un noyau sur l'espace $\mathcal{X}$!

Grande dimension ?

Arbres de régression

Il existe plusieurs autres méthodes de construction de prédicteurs $\hat{f}$ pour la régression. Une méthode populaire se base sur les arbrea de décision.

Grande dimension ?

Arbres de régression

Il existe plusieurs autres méthodes de construction de prédicteurs $\hat{f}$ pour la régression. Une méthode populaire se base sur les arbrea de décision.

Arbre de décision

On construit un arbre tel que chaque nœds intérieur sélectionne une variable d'entrée (input). Chaque arête d'un nœud vers un autre correspond à un ensemble de valeurs possible pour cette variable. Chaque feuille ou nœud terminal, correspond à une valeur prédite pour l'output.

Grande dimension ?

Arbres de régression

Il existe plusieurs autres méthodes de construction de prédicteurs $\hat{f}$ pour la régression. Une méthode populaire se base sur les arbrea de décision.

Arbre de décision

On construit un arbre tel que chaque nœds intérieur sélectionne une variable d'entrée (input). Chaque arête d'un nœud vers un autre correspond à un ensemble de valeurs possible pour cette variable. Chaque feuille ou nœud terminal, correspond à une valeur prédite pour l'output.

En général, l'arbre construit une partition de l'espace des input $\mathcal{X}$. On a donc une famille d'hypothèse qui est l'ensemble des fonction constantes par morceaux.

Grande dimension ?
Arbres de régression

La construction d'un arbre de décision repose sur plusieurs critères

- Comment choisir les variables associées à chaque nœud

Grande dimension ?
Arbres de régression

La construction d'un arbre de décision repose sur plusieurs critères

- Comment choisir les variables associées à chaque nœud
- Comment choisir la meilleure partition pour cette variable

Grande dimension ?
Arbres de régression

La construction d'un arbre de décision repose sur plusieurs critères

- Comment choisir les variables associées à chaque nœud
- Comment choisir la meilleure partition pour cette variable
- Selon quel critère optimise-t-on ?

Grande dimension ?
Arbres de régression

La construction d'un arbre de décision repose sur plusieurs critères

- Comment choisir les variables associées à chaque nœud
- Comment choisir la meilleure partition pour cette variable
- Selon quel critère optimise-t-on ?

Exemple (Quelques algorithmes pour construire des arbres)

CART, C4.5, C5, CHAID, MARS, etc.

Grande dimension ?
Arbres de régression

La construction d'un arbre de décision repose sur plusieurs critères

- Comment choisir les variables associées à chaque nœud
- Comment choisir la meilleure partition pour cette variable
- Selon quel critère optimise-t-on ?

Exemple (Quelques algorithmes pour construire des arbres)

CART, C4.5, C5, CHAID, MARS, etc.

Grande dimension ?
Arbres de régression

La construction d'un arbre de décision repose sur plusieurs critères

- Comment choisir les variables associées à chaque nœud
- Comment choisir la meilleure partition pour cette variable
- Selon quel critère optimise-t-on ?

Exemple (Quelques algorithmes pour construire des arbres)

CART, C4.5, C5, CHAID, MARS, etc.

Choix de l'arbre

Une fois choisi la méthode (ou algorithme) pour construire l'arbre, il faut choisir sa profondeur et le "tailler".

Grande dimension ?
CART

CART (Classification And Regression Tree) est un algorithme pour la construction d'arbre de décision. On défini pour chaque feuille c, $m_c = \frac{1}{n_c} \sum_{i \in F_c} y_i$ et $S = \sum_c \sum_{i \in F_c} (y_i - m_c)^2$.

1. Construire un nœud de départ
2. A chaque nœuds terminal
   - 2.1 Pour toutes les variables, trouver la coupure qui réduit le plus S
   - 2.2 Sélectionner la meilleure coupure, et créer deux nouveaux nœuds terminaux potentiels
3. Comparer toutes les partitions potentielles et sélectionner celle qui réduit au plus S
4. Retour à 2 aussi longtemps que nécessaire.

Chap.3

La Classification

Principe de la classification

Principe général

On observe des couples (X_i, Y_i) de données, mais cette fois les Y_i sont discrètes.

Exemple

Clic sur une pub X_i sont les données du visiteur, Y_i "clic" ou "non clic"

Principe de la classification

Principe général

On observe des couples (X_i, Y_i) de données, mais cette fois les Y_i sont discrètes.

Exemple

Clic sur une pub X_i sont les données du visiteur, Y_i "clic" ou "non clic"

Détection de population à risque X_i sont les données médicales d'un patient, Y_i "patient à risque" ou non

Principe de la classification

Principe général

On observe des couples (X_i, Y_i) de données, mais cette fois les Y_i sont discrètes.

Exemple

Clic sur une pub X_i sont les données du visiteur, Y_i "clic" ou "non clic"

Détection de population à risque X_i sont les données médicales d'un patient, Y_i "patient à risque" ou non

Lecture de code postaux X_i est l'image, Y_i est 0,1,2,...,9

Principe de la classification

Principe général

On observe des couples (X_i, Y_i) de données, mais cette fois les Y_i sont discrètes.

Exemple

Clic sur une pub X_i sont les données du visiteur, Y_i "clic" ou "non clic"

Détection de population à risque X_i sont les données médicales d'un patient, Y_i "patient à risque" ou non

Lecture de code postaux X_i est l'image, Y_i est 0,1,2,...,9

Principe de la classification

Principe général

On observe des couples (X_i, Y_i) de données, mais cette fois les Y_i sont discrètes.

Exemple

Clic sur une pub X_i sont les données du visiteur, Y_i "clic" ou "non clic"

Détection de population à risque X_i sont les données médicales d'un patient, Y_i "patient à risque" ou non

Lecture de code postaux X_i est l'image, Y_i est 0,1,2,...,9

Dans ce cas on doit construire des prédicteurs $\hat{f}$ à valeurs dans l'ensemble $\mathcal{Y}$ des valeurs possible de Y.

Principe de la classification

Modélisation probabiliste

Dans la plupart des cas on utilisera une modélisation probabiliste

$$\mathbb{P}(Y_i = y|X_i) = p_y(X_i)$$

Principe de la classification

Modélisation probabiliste

Dans la plupart des cas on utilisera une modélisation probabiliste

$$\mathbb{P}(Y_i = y|X_i) = p_y(X_i)$$

Si on observe que les input X_i, alors la probabilité que $Y_i = y$ est donnée par $p_y(X_i)$.

Principe de la classification

Modélisation probabiliste

Dans la plupart des cas on utilisera une modélisation probabiliste

$$\mathbb{P}(Y_i = y|X_i) = p_y(X_i)$$

Si on observe que les input X_i, alors la probabilité que $Y_i = y$ est donnée par $p_y(X_i)$.

Quelle perte et quel prédicteur

Ici la perte quadratique n'a pas de sens, on va s'intéresser à une perte $0 - 1$

$$L(f(X_i), Y_i) = \mathbb{I}\{f(X_i) \neq Y_i\}$$

Principe de la classification

Modélisation probabiliste

Dans la plupart des cas on utilisera une modélisation probabiliste

$$\mathbb{P}(Y_i = y|X_i) = p_y(X_i)$$

Si on observe que les input X_i, alors la probabilité que $Y_i = y$ est donnée par $p_y(X_i)$.

Quelle perte et quel prédicteur

Ici la perte quadratique n'a pas de sens, on va s'intéresser à une perte $0 - 1$

$$L(f(X_i), Y_i) = \mathbb{I}\{f(X_i) \neq Y_i\}$$

Le prédicteur qui minimise le risque pour cette perte est donné par

$$f^*(X_i) = \arg\max_{y \in \mathcal{Y}} p_y(X_i)$$

Principe de la classification

Modélisation probabiliste

Dans la plupart des cas on utilisera une modélisation probabiliste

$$\mathbb{P}(Y_i = y | X_i) = p_y(X_i)$$

Si on observe que les input X_i, alors la probabilité que $Y_i = y$ est donnée par $p_y(X_i)$.

Quelle perte et quel prédicteur

Ici la perte quadratique n'a pas de sens, on va s'intéresser à une perte $0 - 1$

$$L(f(X_i), Y_i) = \mathbb{I}\{f(X_i) \neq Y_i\}$$

Le prédicteur qui minimise le risque pour cette perte est donné par

$$f^*(X_i) = \arg\max_{y \in \mathcal{Y}} p_y(X_i)$$

Exercice (Meilleur prédicteur)

Prouver le résultat précédent.

Classifieur Bayésien Naïf

Cas général

Une première approche pour construire un prédicteur est de modéliser les co-varialbes sachant la classe

$$Y \sim \mathcal{M}(\mathbf{g}),\ X|Y = y \sim g_y(X)$$

où $\mathbf{g} = (g_y)_{y \in \mathcal{Y}}$.

Classifieur Bayésien Naïf

Cas général

Une première approche pour construire un prédicteur est de modéliser les co-varialbes sachant la classe

$$Y \sim \mathcal{M}(\mathbf{g}),\ X|Y = y \sim g_y(X)$$

où $\mathbf{g} = (g_y)_{y \in \mathcal{Y}}$.

Interprétation

Sachant que l'on est dans la classe y, les co variables suivent la loi g_y.

Classifieur Bayésien Naïf

Cas général

Une première approche pour construire un prédicteur est de modéliser les co-varialbes sachant la classe

$$Y \sim \mathcal{M}(\mathbf{g}),\ X|Y = y \sim g_y(X)$$

où $\mathbf{g} = (g_y)_{y \in \mathcal{Y}}$.

Interprétation

Sachant que l'on est dans la classe y, les co variables suivent la loi g_y.

D'après le théorème de Bayes on a

$$\mathbb{P}(Y = y|X) = \frac{q_y p_y(X)}{\sum_{z \in \mathcal{Y}} q_z g_z(X)}$$

et comme précédemment on prends le prédicteur

$$f = \arg\max_{y \in \mathcal{Y}} \mathbb{P}(Y = y|X) = \arg\max_{y \in \mathcal{Y}} q_y g_y(X).$$

Classifieur Bayésien Naïf

Cas général

On peut choisir une perte plus générale.

Approche décisionnelle

En pratique on peut associer un *coût* $L(z, y)$ de choisir z alors que y est vrais. Dans ce cas le risque s'écrit

$$R(f) = \mathbb{E}_X \sum_{y \in \mathcal{Y}} L(f(X), y) \mathbb{P}(Y = y|X)$$

et

$$f^*(X) = \arg\min_{z \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} L(z, y) \mathbb{P}(Y = y|X)$$

Classifieur Bayésien Naïf

Cas général

On peut choisir une perte plus générale.

Approche décisionnelle

En pratique on peut associer un *coût* $L(z, y)$ de choisir z alors que y est vrais. Dans ce cas le risque s'écrit

$$R(f) = \mathbb{E}_X \sum_{y \in \mathcal{Y}} L(f(X), y)\mathbb{P}(Y = y|X)$$

et

$$f^*(X) = \arg\min_{z \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} L(z, y)\mathbb{P}(Y = y|X)$$

Exemple

Lorsque l'on pose un diagnostique médicale, il peut être plus risqué de dire à un individu malade qu'il est sain que l'inverse. On va avoir des coût différents pour $L(\text{sain}, \text{malade})$ et $L(\text{malade}, \text{sain})$.

Classifieur Bayésien Naïf

Cas normal

On ne connaît pas a priori les distribution de probabilité p_y...

Classifieur Bayésien Naïf

Cas normal

On ne connaît pas a priori les distribution de probabilité p_y...Dans le cas de variables continues on peut choisir une loi normale

$$p_y(X) = \frac{1}{(2\pi|\Sigma_y|)^{p/2}} e^{-\frac{1}{2}(X-\theta_y)'\Sigma_y^{-1}(X-\theta_y)},$$

où $\theta_y \in \mathbb{R}^p$ est le vecteur des moyenne et $\Sigma_y \in \mathbb{R}^{p\times p}$ est la matrice de variance co-variance.

Classifieur Bayésien Naïf

Cas normal

On ne connaît pas a priori les distribution de probabilité p_y...Dans le cas de variables continues on peut choisir une loi normale

$$p_y(X) = \frac{1}{(2\pi|\Sigma_y|)^{p/2}} e^{-\frac{1}{2}(X-\theta_y)'\Sigma_y^{-1}(X-\theta_y)},$$

où $\theta_y \in \mathbb{R}^p$ est le vecteur des moyenne et $\Sigma_y \in \mathbb{R}^{p\times p}$ est la matrice de variance co-variance.

Prédicteur

On va construire un prédicteur en estimant les paramètres de la loi θ_y et Σ_y sur les données d'apprentissage.

Classifieur Bayésien Naïf

Cas normal

Estimation de la moyenne θ_y pour la classe y :

$$\hat{\theta}_y = \frac{1}{n_y} \sum_{i, Y_i = y} X_i$$

Classifieur Bayésien Naïf

Cas normal

Estimation de la moyenne θ_y pour la classe y :

$$\hat{\theta}_y = \frac{1}{n_y} \sum_{i, Y_i = y} X_i$$

Pour la matrice de variance-covariance Σ_y il existe deux solution. Estimation complète de la matrice

$$\hat{\Sigma}_y = \frac{1}{n_y} \sum_{i, Y_i = y} (X_i - \hat{\theta}_y)(X_i - \hat{\theta}_y)'$$

Classifieur Bayésien Naïf

Cas normal

Estimation de la moyenne θ_y pour la classe y :

$$\hat{\theta}_y = \frac{1}{n_y} \sum_{i, Y_i = y} X_i$$

Pour la matrice de variance-covariance Σ_y il existe deux solution. Estimation complète de la matrice

$$\hat{\Sigma}_y = \frac{1}{n_y} \sum_{i, Y_i = y} (X_i - \hat{\theta}_y)(X_i - \hat{\theta}_y)'$$

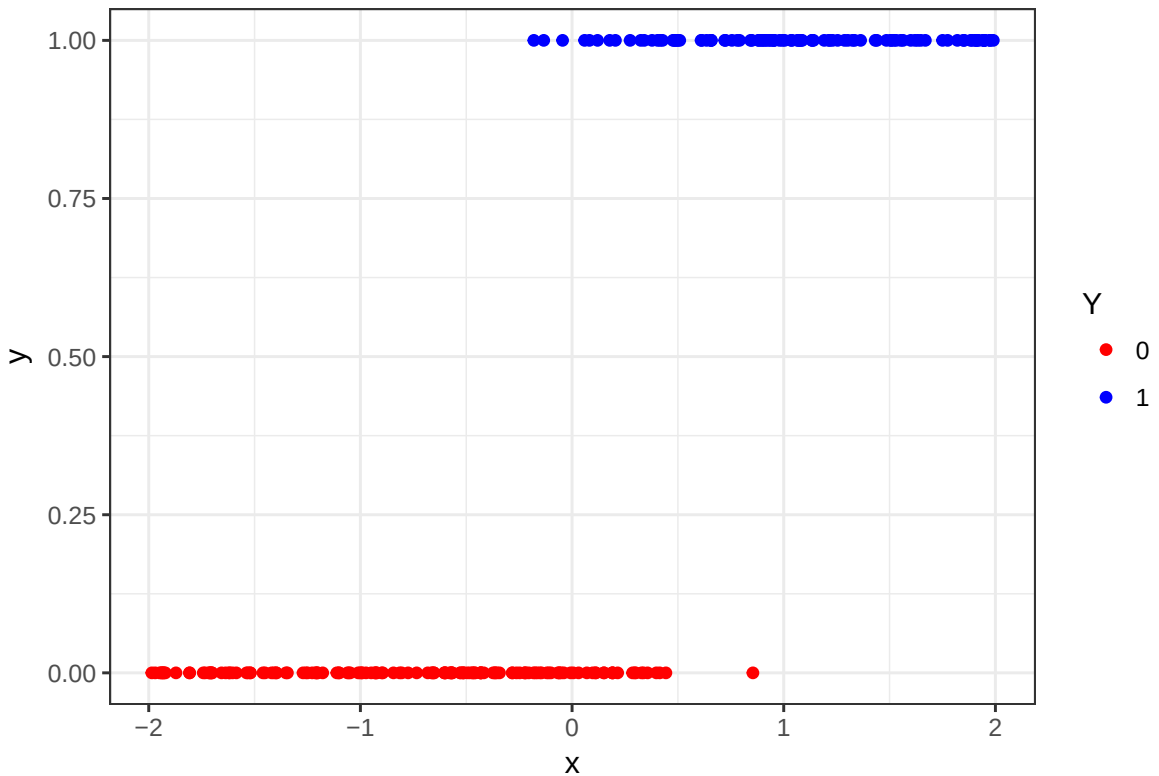
Une autre solution populaire consiste à supposer les co-variables indépendantes

$$\sigma^2_{y,j} = \frac{1}{n_y - 1} \sum_{i, Y_i = y} (X_i^j - \hat{\theta}_{y,j})^2$$

Régression Logistique

Problème

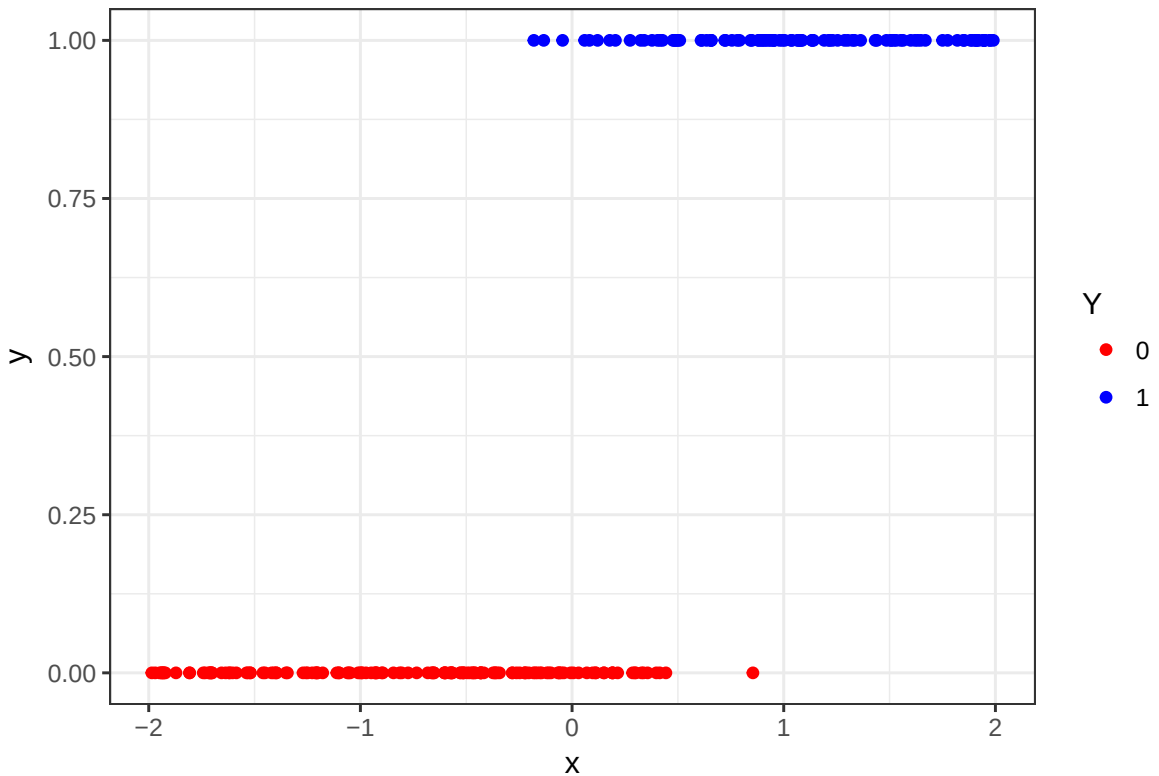
On considère le cas d'un output binaire $\mathcal{Y} = \{0, 1\}$



Régression Logistique

Problème

On considère le cas d'un output binaire $\mathcal{Y} = \{0, 1\}$



On ne peut pas utiliser un modèle de régression pour prédire ces valeurs !

Régression Logistique

Problème

On cherche à prédire $\mathbb{P}(Y|X)$, pour cela on va utiliser le modèle suivant

$$\mathbb{P}(Y = 1|X) = q(X),$$

où q est une fonction de $\mathcal{X} \to [0, 1]$.

Régression Logistique

Problème

On cherche à prédire $\mathbb{P}(Y|X)$, pour cela on va utiliser le modèle suivant

$$\mathbb{P}(Y = 1|X) = q(X),$$

où q est une fonction de $\mathcal{X} \to [0, 1]$.On s'intéresse à une forme particulière pour la fonction q.

Fonction Sigmoïde

On appelle fonction sigmoïde les fonctions

$$x \mapsto \frac{e^{\lambda x}}{1 + e^{\lambda x}}.$$

On s'intéressera aux fonctions q de la forme

$$q(X) = \sigma(X'\theta + \theta_0),$$

où σ est une fonction sigmoïde.

Régression Logistique

Interprétation du modèle

On peut interpréter le modèle de la manière suivante. Supposons qu'il existe une variable Z que l'on observe pas, et qui est telle que

$$\begin{cases} Y = 0 \text{ si } Z \leq 0 \\ Y = 1 \text{ si } Z > 0 \end{cases} .$$

Régression Logistique

Interprétation du modèle

On peut interpréter le modèle de la manière suivante. Supposons qu'il existe une variable Z que l'on observe pas, et qui est telle que

$$\begin{cases} Y = 0 \text{ si } Z \leq 0 \\ Y = 1 \text{ si } Z > 0 \end{cases}.$$

Si l'on choisi de prédire Z à l'aide d'un modèle de régression linéaire

$$Z = X'\theta + \theta_0 + \epsilon$$

où ϵ suit une loi symétrique de fonction de répartition σ, on a

$$\begin{aligned} \mathbb{P}(Y = 1|\ X) &= \mathbb{P}(Z > 0|X) \\ &= \mathbb{P}(X'\theta + \theta_0 + \epsilon > 0) \\ &= \mathbb{P}(-\epsilon < X'\theta + \theta_0) = \sigma(X'\theta + \theta_0) \end{aligned}$$

On considère la perte logistique

Perte logistique

On appelle perte logistique

$$\begin{aligned} L(f(X), Y) &= -\log \mathbb{P}(Y|X) \\ &= -Y \log(\mathbb{P}(Y = 1|X)) - (1 - Y) \log(\mathbb{P}(Y = 0|X)) \end{aligned}$$

On considère la perte logistique

Perte logistique

On appelle perte logistique

$$\begin{aligned} L(f(X), Y) &= -\log \mathbb{P}(Y|X) \\ &= -Y \log(\mathbb{P}(Y=1|X)) - (1-Y) \log(\mathbb{P}(Y=0|X)) \end{aligned}$$

Pour notre modèle on a donc

$$R_n(f_\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left(Y_i \log(\sigma(X_i'\theta + \theta_0)) + (1-Y_i) \log(1 - \sigma(X_i'\theta + \theta_0)) \right).$$

On considère la perte logistique

Perte logistique

On appelle perte logistique

$$\begin{aligned} L(f(X), Y) &= -\log \mathbb{P}(Y|X) \\ &= -Y \log(\mathbb{P}(Y=1|X)) - (1-Y) \log(\mathbb{P}(Y=0|X)) \end{aligned}$$

Pour notre modèle on a donc

$$R_n(f_\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left(Y_i \log(\sigma(X_i'\theta + \theta_0)) + (1-Y_i) \log(1 - \sigma(X_i'\theta + \theta_0)) \right).$$

On a pas de solution simple pour minimiser le risque empirique...

Régression Logistique

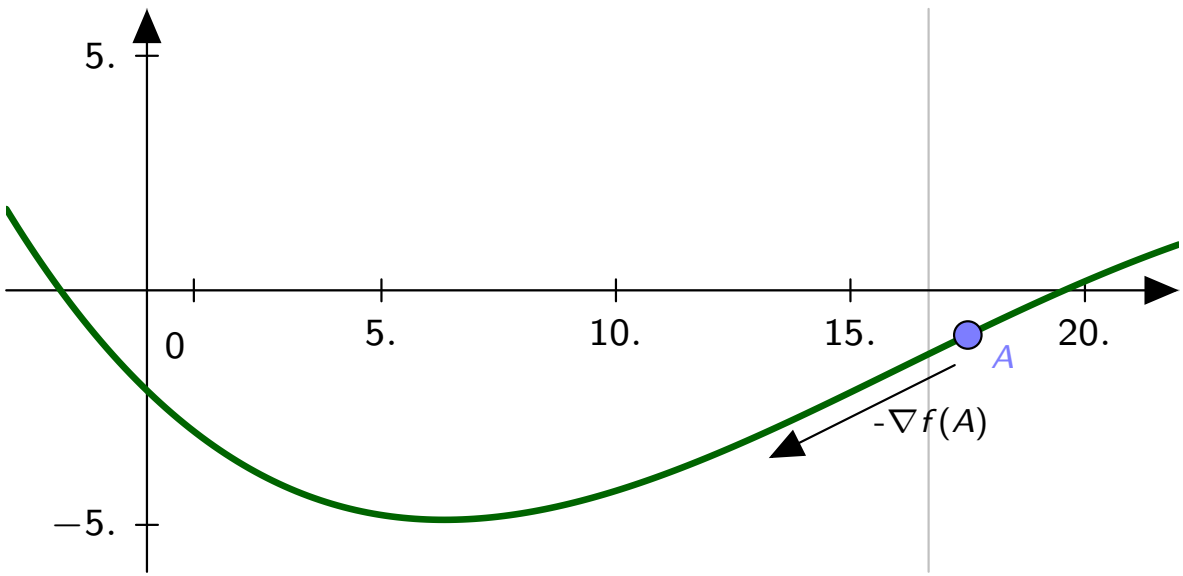
Mise en œuvre

Comme on ne peut pas minimiser le risque empirique *analytiquement* on va chercher une approximation numérique du minimum.

Régression Logistique

Mise en œuvre

Comme on ne peut pas minimiser le risque empirique *analytiquement* on va chercher une approximation numérique du minimum.Une méthode de minimisation numérique est la descente de gradient.



On souhaite minimiser une fonction $f : \mathbb{R}^p \to \mathbb{R}$

Descente de Gradient

Algorithm Descente de Gradient

1: Choisir un point initial x_0, et initialiser $i = 0$
2: **while** $||\nabla f(x_i)|| > \epsilon$ **do**
3: $\quad x_{i+1} \leftarrow x_i - \alpha_i \nabla f(x_i)$
4: $\quad i \leftarrow i + 1$
5: **end while**
6: **return** x_i

α_i est appelé *pas d'apprentissage* et change avec i. En pratique $\alpha_i = \frac{a}{b+ci}$ et on trouve a, b et c par essais.

Régression Logistique

Mise en œuvre

On souhaite minimiser une fonction $f : \mathbb{R}^p \to \mathbb{R}$

Descente de Gradient

Algorithm Descente de Gradient

1: Choisir un point initial x_0, et initialiser $i = 0$
2: **while** $||\nabla f(x_i)|| > \epsilon$ **do**
3: $\quad x_{i+1} \leftarrow x_i - \alpha_i \nabla f(x_i)$
4: $\quad i \leftarrow i + 1$
5: **end while**
6: **return** x_i

α_i est appelé *pas d'apprentissage* et change avec i. En pratique $\alpha_i = \frac{a}{b+ci}$ et on trouve a, b et c par essais.Par exemple si ∇f augmente, on aura tendance à diminuer α_i...

Régression Logistique

Mise en œuvre

Une autre approche est de chercher les zéros de ∇R. Pour cela on peut utiliser l'algorithme de Newton-Raphson.

Algorithme de N-R

Pour trouver le zéro d'une fonction f l'algorithme de Newton-Raphson fonctionne comme suit.

Algorithm Newton-Raphson

1: Choisir un point initial x_0, $i \leftarrow 0$
2: **while** $|\nabla f(x_i) - \nabla f(x_{i+1})| > \epsilon$ **do**
3: $\quad x_{i+1} \leftarrow x_i - [H(f)(x_i)]^{-1} \nabla f(x_i)$
4: $\quad i \leftarrow i + 1$
5: **end while**
6: **return** x_i

Une autre approche est de chercher les zéros de ∇R. Pour cela on peut utiliser l'algorithme de Newton-Raphson.

Algorithme de N-R

Pour trouver le zéro d'une fonction f l'algorithme de Newton-Raphson fonctionne comme suit.

Algorithm Newton-Raphson

1: Choisir un point initial x_0, $i \leftarrow 0$
2: **while** $|\nabla f(x_i) - \nabla f(x_{i+1})| > \epsilon$ **do**
3: $\quad x_{i+1} \leftarrow x_i - [H(f)(x_i)]^{-1} \nabla f(x_i)$
4: $\quad i \leftarrow i + 1$
5: **end while**
6: **return** x_i

Cet algorithme est assez similaire à une descente de gradient, mais avec un pas adaptatif dans chacune des directions. Il nécessite cependant de calculer l'inverse de la matrice Hessienne à chaque pas, ce qui peut être coûteux.

Régression Logistique

Enrichissement

Comme pour la régression linéaire, rien ne nous empêche de choisir une famille d'hypothèses plus riche.

$$\mathbb{P}(Y_i = 1|X) = \sigma(\sum_{k=1}^{K} \Phi_k(X_i)'\theta).$$

Régression Logistique

Enrichissement

Comme pour la régression linéaire, rien ne nous empêche de choisir une famille d'hypothèses plus riche.

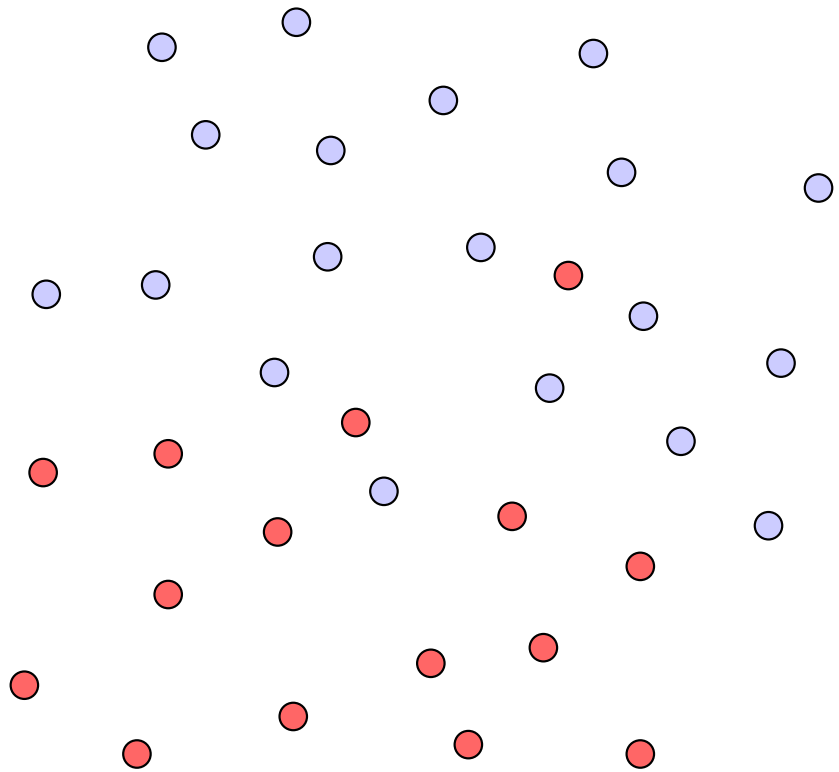
$$\mathbb{P}(Y_i = 1|X) = \sigma(\sum_{k=1}^{K} \Phi_k(X_i)'\theta).$$

L'estimation ce fait de la même manière. Cependant, en choisissant K très grand, en plus du risque de sur apprentissage, on risque d'avoir des problèmes de convergence des algorithmes utilisés pour trouver le paramètre θ.

Autres approches classiques

K plus proche voisins

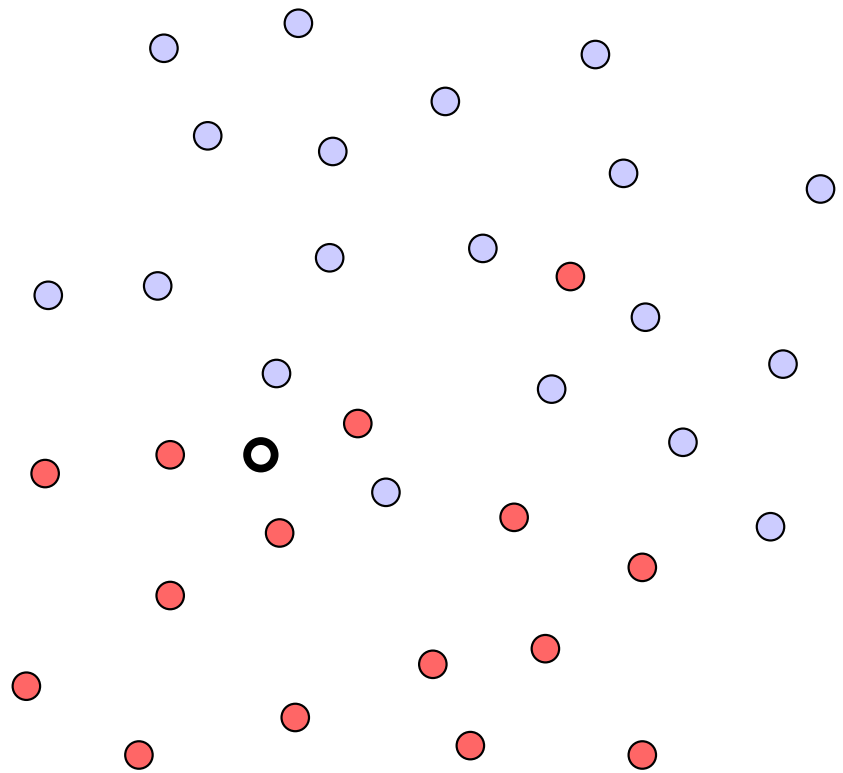
Une méthode simple pour la prédiction. L'idée est d'associer à un nouveau point la classe dominante parmi ses voisins.



Autres approches classiques

K plus proche voisins

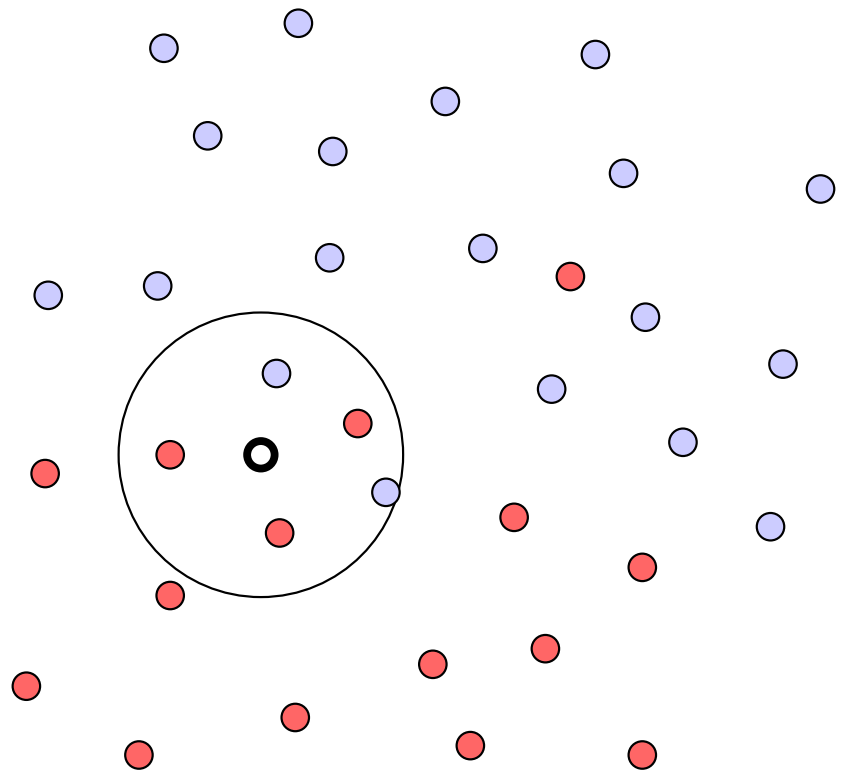
Une méthode simple pour la prédiction. L'idée est d'associer à un nouveau point la classe dominante parmi ses voisins.



Autres approches classiques

K plus proche voisins

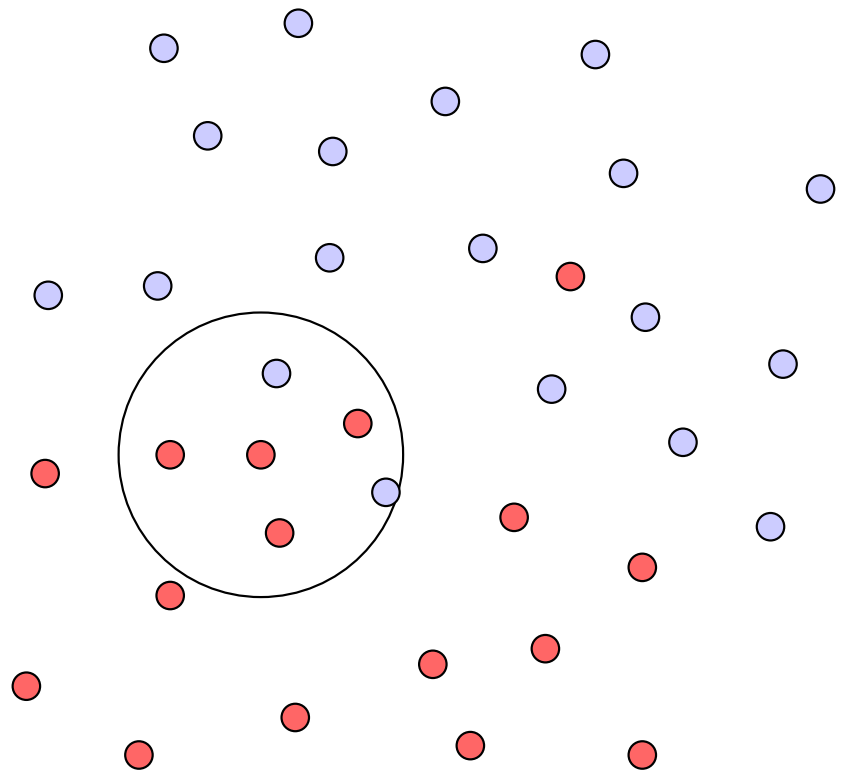
Une méthode simple pour la prédiction. L'idée est d'associer à un nouveau point la classe dominante parmi ses voisins.



Autres approches classiques

K plus proche voisins

Une méthode simple pour la prédiction. L'idée est d'associer à un nouveau point la classe dominante parmi ses voisins.



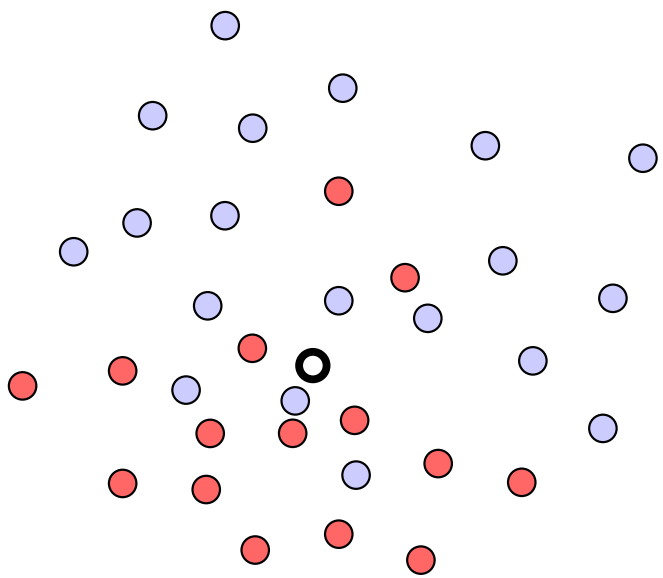
Autres approches classiques

K plus proche voisins

Cette approche est très simple à mettre en œuvre mais soufre de certains problèmes

Drawbacks

- La méthode peut être très instable si les données sont trop bruitées
- Dépend fortement du choix de k...
- ... et de la métrique considérée



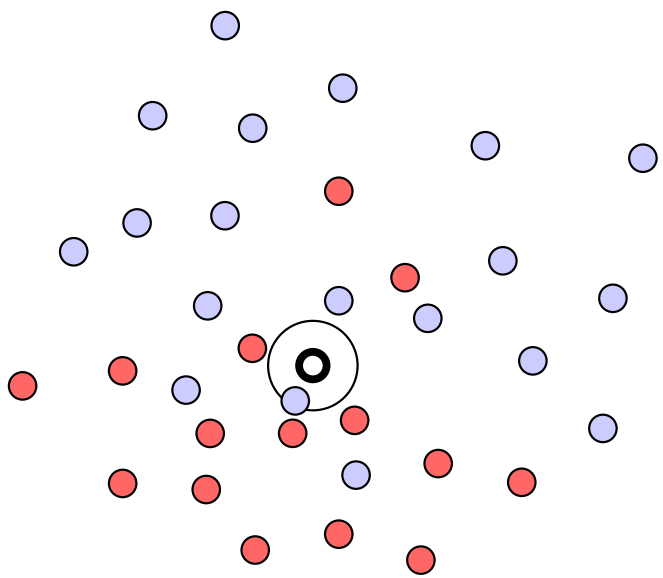
Autres approches classiques

K plus proche voisins

Cette approche est très simple à mettre en œuvre mais soufre de certains problèmes

Drawbacks

- La méthode peut être très instable si les données sont trop bruitées
- Dépend fortement du choix de k...
- ... et de la métrique considérée



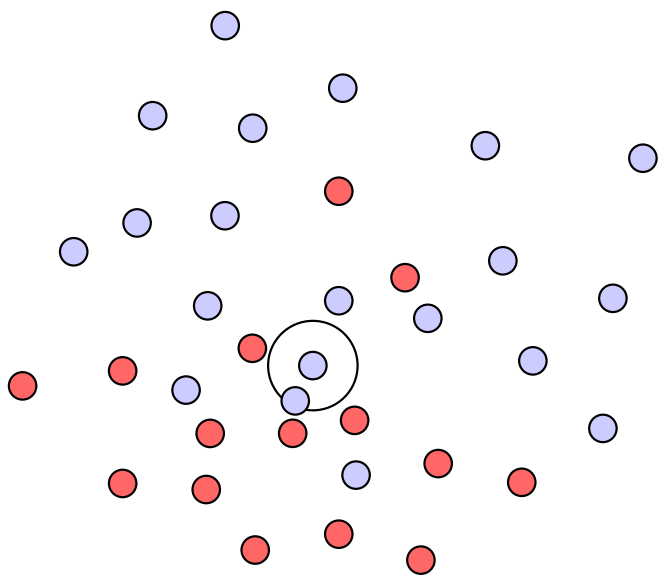
Autres approches classiques

K plus proche voisins

Cette approche est très simple à mettre en œuvre mais soufre de certains problèmes

Drawbacks

- La méthode peut être très instable si les données sont trop bruitées
- Dépend fortement du choix de k...
- ... et de la métrique considérée

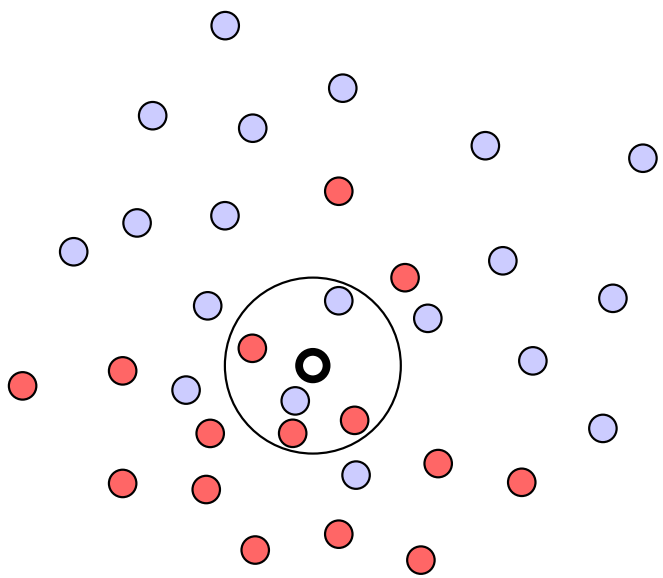


Autres approches classiques
K plus proche voisins

Cette approche est très simple à mettre en œuvre mais soufre de certains problèmes

Drawbacks

- La méthode peut être très instable si les données sont trop bruitées
- Dépend fortement du choix de k...
- ... et de la métrique considérée



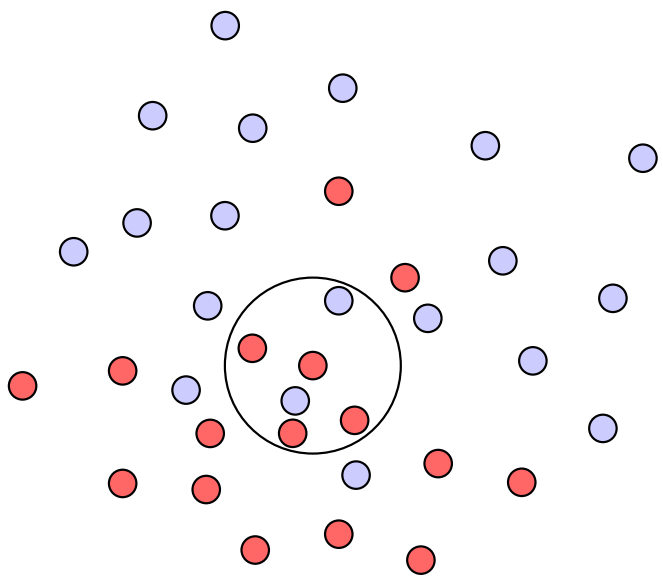
Autres approches classiques

K plus proche voisins

Cette approche est très simple à mettre en œuvre mais soufre de certains problèmes

Drawbacks

- La méthode peut être très instable si les données sont trop bruitées
- Dépend fortement du choix de k...
- ... et de la métrique considérée



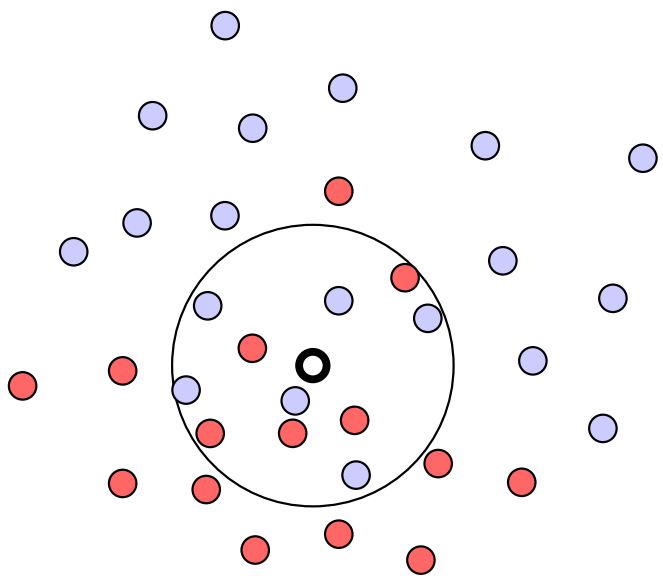
Autres approches classiques

K plus proche voisins

Cette approche est très simple à mettre en œuvre mais soufre de certains problèmes

Drawbacks

- La méthode peut être très instable si les données sont trop bruitées
- Dépend fortement du choix de k...
- ... et de la métrique considérée



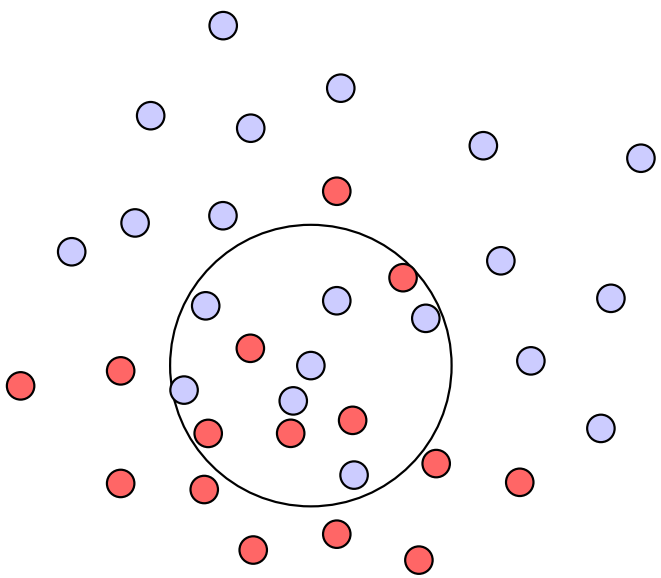
Autres approches classiques

K plus proche voisins

Cette approche est très simple à mettre en œuvre mais soufre de certains problèmes

Drawbacks

- La méthode peut être très instable si les données sont trop bruitées
- Dépend fortement du choix de k...
- ... et de la métrique considérée



Autres approches classiques

Algorithme CART pour la classification

L'algorithme CART peut tout a fait être adapté pour la classification en utilisant la même approche.

Autres approches classiques

Algorithme CART pour la classification

L'algorithme CART peut tout a fait être adapté pour la classification en utilisant la même approche.Il suffit de changer de perte pour choisir la partition optimale. On va chercher à ce que les feuilles de l'arbre soient le plus homogènes possibles.

Critère d'hétérogénéité

Considérons le problème de classification à J classes On appelle critère d'hétérogénéité une fonction $h : \mathbf{q} = (q_1, \ldots, q_J) \in [0,1]^J \mapsto h(\mathbf{q}) \in \mathbb{R}$ tel que

- h soit symétrique en $q_1, \ldots, q_J$
- h est maximal en $(J^{-1}, \ldots, J^{-1})$
- h est minimal en $(1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, $\ldots$, $(0, 0, \ldots, 1)$

Exemples de critères

$$\text{Gini} \quad h(\mathbf{q}) = \sum_{i \neq j} q_i q_j$$

$$\text{Shannon} \quad h(\mathbf{q}) = -\sum_{j=1}^{J} q_j \log(q_j)$$

Chap.4

Bagging, Random Forest et Boosting

Principe

On a vu que pour un problème (Régression, classification) on pouvait avoir plusieurs algorithmes en compétitions.

Principe

On a vu que pour un problème (Régression, classification) on pouvait avoir plusieurs algorithmes en compétitions.De plus pour un même algorithme, on peut avoir plusieurs choix de paramètres possible.

Principe

On a vu que pour un problème (Régression, classification) on pouvait avoir plusieurs algorithmes en compétitions.De plus pour un même algorithme, on peut avoir plusieurs choix de paramètres possible.

Idée

Faire coopérer plusieurs algorithmes en même temps pour prendre de meilleur décisions et soit *réduire la variance* ou *le biais* de la méthode.

Principe

On a vu que pour un problème (Régression, classification) on pouvait avoir plusieurs algorithmes en compétitions.De plus pour un même algorithme, on peut avoir plusieurs choix de paramètres possible.

Idée

Faire coopérer plusieurs algorithmes en même temps pour prendre de meilleur décisions et soit *réduire la variance* ou *le biais* de la méthode.

Il y a deux approches différentes

- Construire différents modèles sur les mêmes données
- Construire des modèles de même nature sur des données différentes

Bagging

Présentation

Considérons un problème de régression. On dispose d'un échantillon d'apprentissage $d^n = \{(X_1, Y_1), \dots (X_n, Y_n)\}$. Imaginons que l'on puisse construire B prédicteurs indépendants $\hat{f}_1, \dots, \hat{f}_B$. Pour une nouvelle observation X on peut considérer le prédicteur

$$\tilde{f}(X) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i(X)$$

Bagging
Présentation

Considérons un problème de régression. On dispose d'un échantillon d'apprentissage $d^n = \{(X_1, Y_1), \dots (X_n, Y_n)\}$. Imaginons que l'on puisse construire B prédicteurs indépendants $\hat{f}_1, \dots, \hat{f}_B$. Pour une nouvelle observation X on peut considérer le prédicteur

$$\tilde{f}(X) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i(X)$$

Pour le risque quadratique on a, pour un prédicteur optimal f^* et pour tout $x \in \mathcal{X}$

$$\mathbb{E}((f^*(x) - \tilde{f}(x))^2) \leq (\mathbb{E}(\tilde{f}(x)) - f^*(x))^2 + \mathbb{V}(\tilde{f}(x))$$

or si les $\hat{f}_i$ sont indépendants $\mathbb{V}(\tilde{f}(x)) = \frac{1}{B^2} \sum_{i=1}^{B} \mathbb{V}(\hat{f}_i(x))$. En particulier si les $\hat{f}_i$ sont iid, même biais mais la variance diminue.

Bagging
Présentation

Dans les fait les prédicteurs $\hat{f}_i$ étant construits sur les même données, ils ne peuvent pas être indépendants...

Bagging
Présentation

Dans les fait les prédicteurs $\hat{f}_i$ étant construits sur les même données, ils ne peuvent pas être indépendants...

- Mais on réduit quand même la variance si les prédicteurs sont faiblement corrélés !

Bagging
Présentation

Dans les fait les prédicteurs $\hat{f}_i$ étant construits sur les même données, ils ne peuvent pas être indépendants...

- Mais on réduit quand même la variance si les prédicteurs sont faiblement corrélés !

Solution (Breiman)

On va construire des prédicteurs sur des échantillons Bootstrap de d^n

Bagging
Présentation

Dans les fait les prédicteurs $\hat{f}_i$ étant construits sur les même données, ils ne peuvent pas être indépendants...

- Mais on réduit quand même la variance si les prédicteurs sont faiblement corrélés !

Solution (Breiman)

On va construire des prédicteurs sur des échantillons Bootstrap de d^n

Remarque

Le Bagging ne réduit que la variance, il est bien adapté aux cas où les prédicteurs ont un *biais* faible mais sont sensibles à une perturbation de l'échantillon.

Bagging
Présentation

Dans les fait les prédicteurs $\hat{f}_i$ étant construits sur les même données, ils ne peuvent pas être indépendants...

- Mais on réduit quand même la variance si les prédicteurs sont faiblement corrélés !

Solution (Breiman)

On va construire des prédicteurs sur des échantillons Bootstrap de d^n

Remarque

Le Bagging ne réduit que la variance, il est bien adapté aux cas où les prédicteurs ont un *biais* faible mais sont sensibles à une perturbation de l'échantillon.

L'exemple des arbres de régression ou de classification est fondamental pour cette méthode (Random Forests).

Bagging
Algorithme

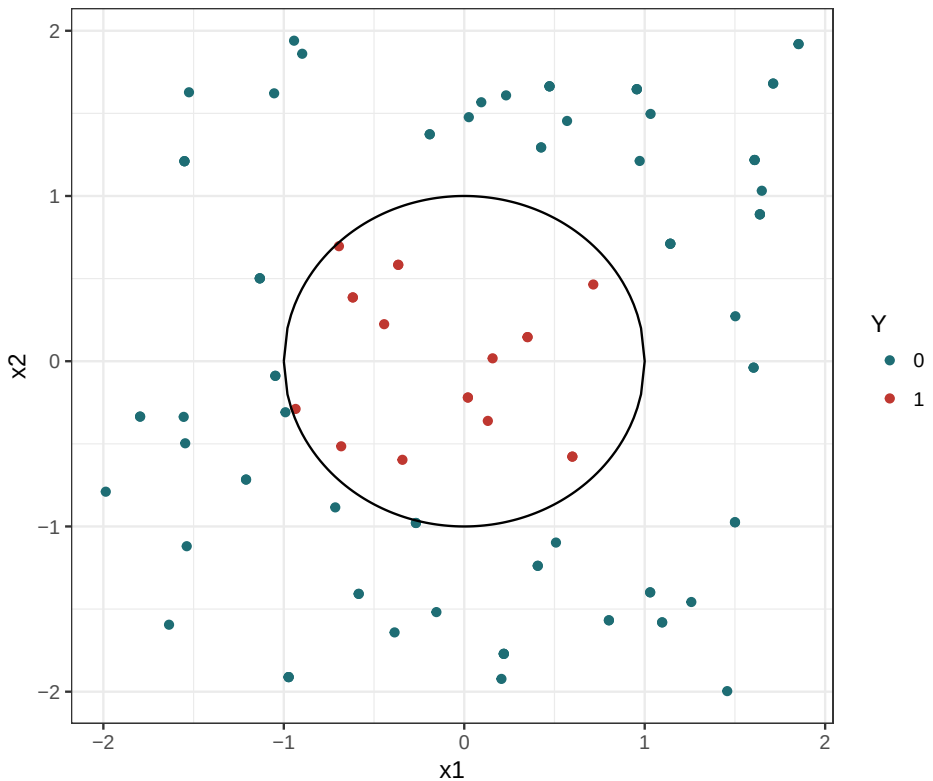
On se donne une méthode de construction de classifieur $\hat{f}$, et un échantillon d'apprentissage d^n, Pour un (grand nombre de répétitions B) et une taille d'échantillon Bootstrap $m_n \leq n$ on suit l'algorithme suivant

Algorithm Bagging

1: **for** $i = 1 : B$ **do**
2: Tirer avec remise un échantillon d_i^n de taille m_n parmi d^n
3: Entraîner le prédicteur $\hat{f}_i$ sur d_i^n
4: **end for**
5: Pour la classification **return** $\tilde{f} = \arg\max_k \sum_{i=1}^{B} \mathbb{I}\{\hat{f}_i = k\}$
6: Pour la régression **return** $\tilde{f} = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i$

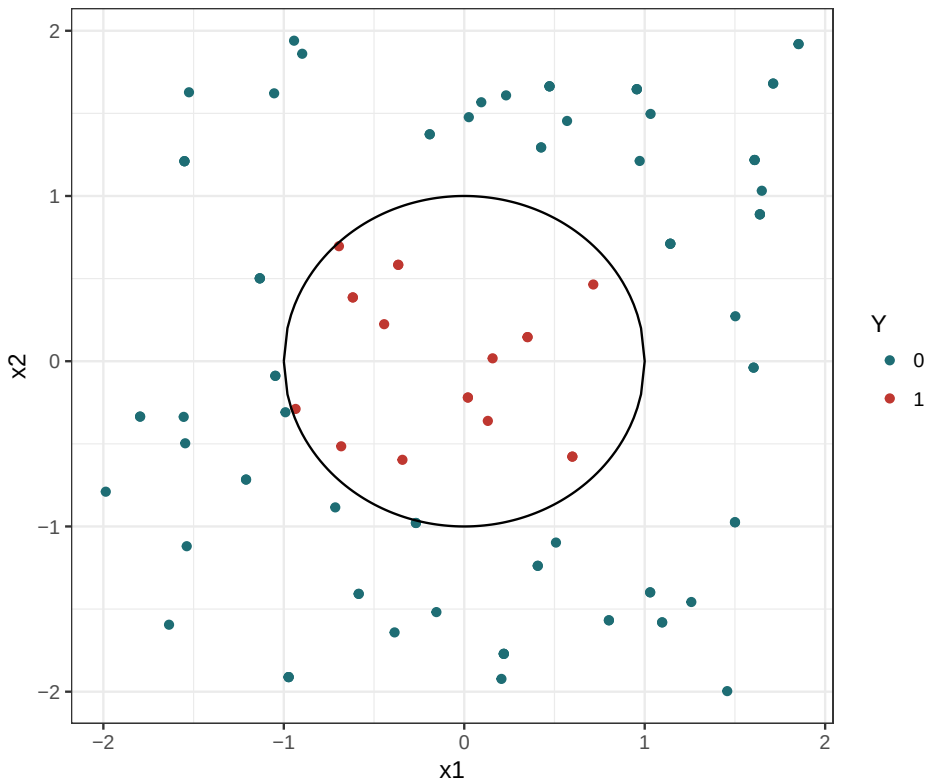
Bagging
Algorithme

On se place de le cas suivant de la classification et l'on dispose des données ci-dessous.



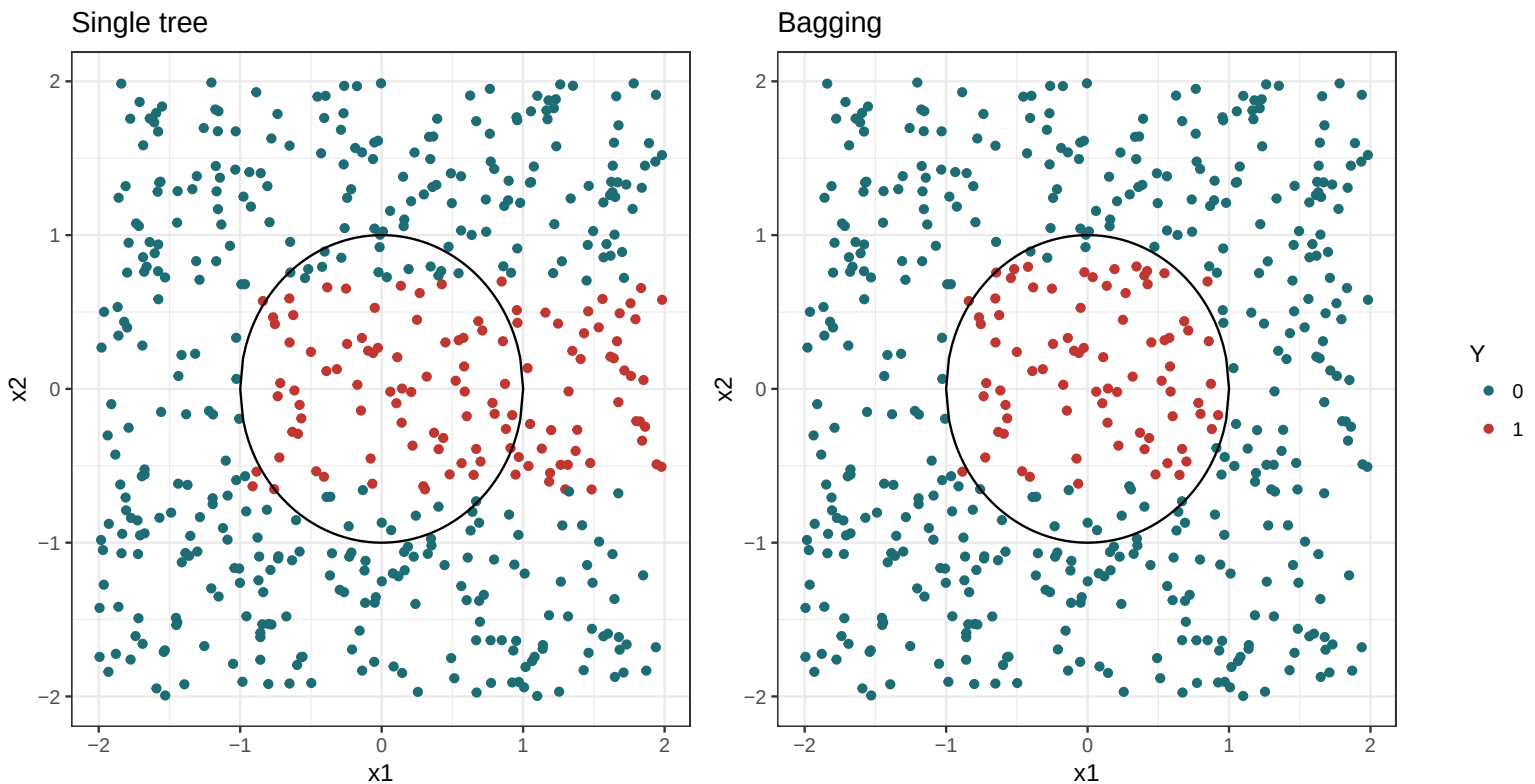
Bagging
Algorithme

On se place de le cas suivant de la classification et l'on dispose des données ci-dessous. On souhaite prédire la classe de nouveaux individus à l'aide de l'algorithme CART.



Bagging
Algorithme

On se place de le cas suivant de la classification et l'on dispose des données ci-dessous. On souhaite prédire la classe de nouveaux individus à l'aide de l'algorithme CART.



Random Forests

Les Foret Aléatoire (ou Random Forest) sont toute méthode de Bagging d'arbre de classification ou de régression.

Random Forests

Les Foret Aléatoire (ou Random Forest) sont toute méthode de Bagging d'arbre de classification ou de régression.Cependant, le terme forêt aléatoire fait le plus souvent référence à une forêt à *inputs aléatoires*.

Random Forests

Les Foret Aléatoire (ou Random Forest) sont toute méthode de Bagging d'arbre de classification ou de régression.Cependant, le terme forêt aléatoire fait le plus souvent référence à une forêt à *inputs aléatoires*.

- On construit des arbres de décision sur des échantillons bootstrap de talle n
- A chaque nœud on effectue une partition sur un échantillon aléatoire de $k < p$ inputs

Random Forests

Les Foret Aléatoire (ou Random Forest) sont toute méthode de Bagging d'arbre de classification ou de régression.Cependant, le terme forêt aléatoire fait le plus souvent référence à une forêt à *inputs aléatoires*.

- On construit des arbres de décision sur des échantillons bootstrap de talle n
- A chaque nœud on effectue une partition sur un échantillon aléatoire de $k < p$ inputs

Conséquence

Cela permet de décorréler encore plus les arbre et ainsi de réduire encore plus la variance.

Algorithm Random Forest

for $i = 1 : B$ **do**
    Tirer avec remise un échantillon d_i^n de taille m_n parmi d^n
    Entraîner le prédicteur $\hat{f}_i$ sur d_i^n de la manière suivante :
    for chaque nœud **do**
        Tirer sans remise k variables explicatives parmi p
        Partitionner le nœud en sélectionnant la *meilleur* partition parmi ces k variables
    end for
end for
Pour la classification **return** $\tilde{f} = \arg\max_k \sum_{i=1}^{B} \mathbb{I}\{\hat{f}_i = k\}$
Pour la régression **return** $\tilde{f} = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i$

Algorithm Random Forest

for $i = 1 : B$ **do**
    Tirer avec remise un échantillon d_i^n de taille m_n parmi d^n
    Entraîner le prédicteur $\hat{f}_i$ sur d_i^n de la manière suivante :
    for chaque nœud **do**
        Tirer sans remise k variables explicatives parmi p
        Partitionner le nœud en sélectionnant la *meilleur* partition parmi ces k variables
    end for
end for
Pour la classification **return** $\tilde{f} = \arg\max_k \sum_{i=1}^{B} \mathbb{I}\{\hat{f}_i = k\}$
Pour la régression **return** $\tilde{f} = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i$

Remarque

Si k est trop petit le biais augmente... il va donc falloir choisir k de manière approprié.

Algorithm Random Forest

for $i = 1 : B$ **do**
    Tirer avec remise un échantillon d_i^n de taille m_n parmi d^n
    Entraîner le prédicteur $\hat{f}_i$ sur d_i^n de la manière suivante :
    for chaque nœud **do**
        Tirer sans remise k variables explicatives parmi p
        Partitionner le nœud en sélectionnant la *meilleur* partition parmi ces k variables
    end for
end for
Pour la classification **return** $\tilde{f} = \arg\max_k \sum_{i=1}^{B} \mathbb{I}\{\hat{f}_i = k\}$
Pour la régression **return** $\tilde{f} = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i$

Remarque

Si k est trop petit le biais augmente... il va donc falloir choisir k de manière approprié. En général on le calibre par validation croisée.

Boosting
Principe

L'idée du boosting est d'apprendre des erreurs, il existe plusieurs variantes, mais la plus utilisée est AdaBoost.

Boosting
Principe

L'idée du boosting est d'apprendre des erreurs, il existe plusieurs variantes, mais la plus utilisée est AdaBoost.

Idée générale

- Apprendre un prédicteur *faible* sur les données
- Repondérer les données en donnant plus de poids au données mal prédites
- Ré-apprendre un classifieur sur les données pondérées et lui associer un poids

Boosting
Principe

L'idée du boosting est d'apprendre des erreurs, il existe plusieurs variantes, mais la plus utilisée est AdaBoost.

Idée générale

- Apprendre un prédicteur *faible* sur les données
- Repondérer les données en donnant plus de poids au données mal prédites
- Ré-apprendre un classifieur sur les données pondérées et lui associer un poids

On va concentrer les efforts sur les données difficiles à prédire.

Boosting
Principe

L'idée du boosting est d'apprendre des erreurs, il existe plusieurs variantes, mais la plus utilisée est AdaBoost.

Idée générale

- Apprendre un prédicteur *faible* sur les données
- Repondérer les données en donnant plus de poids au données mal prédites
- Ré-apprendre un classifieur sur les données pondérées et lui associer un poids

On va concentrer les efforts sur les données difficiles à prédire.

Remarque

Pondérer les données d'apprentissage revient à avoir une pondération dans le risque. Le risque empirique devient

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} w_i L(f(X_i), Y_i)$$

Boosting

Cas de la classification

Considérons le cas simple de la classification à deux classes.

Boosting

Cas de la classification

Considérons le cas simple de la classification à deux classes.

Algorithm AdaBoost

1: Initialiser les poids à $w_i^1 = 1/n$
2: **for** $k = 1 : B$ **do**
3: Construire un prédicteur $\hat{f}_k$ à l'aide des données pondérées
4: Calculer l'erreur du prédicteur

$$E^k \leftarrow \sum_{i=1}^{n} L(\hat{f}_k(X_i), Y_i) w_i$$

5: Calculer le Logit de l'erreur $c_k \leftarrow \log\left(\frac{1-E_k}{E_k}\right)$
6: Actualiser les poids $w_i^{k+1} \leftarrow w_i^k \exp\{c_k \mathbb{I}_{Y_i \neq \hat{f}(X_i)}\}$
7: **end for**
8: **return** $\tilde{f} = \arg\max_{l \in \mathcal{Y}} \sum_{k=1}^{K} c_k \mathbb{I}_{\hat{f}_k = l}$