

CART

Méthodes
d'agrégation

Bagging

Boosting

Références

CART, Bagging & Boosting

Vincent Lefieux



Plan

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

CART

Introduction

Exemple jouet

Principes

Cas de la classification
supervisée

Cas de la régression

Compléments

Méthodes
d'agrégation

Bagging

Boosting

Références

Plan

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

- ▶ **CART** (*Classification And Regression Tree*) est une méthode d'apprentissage supervisée : (Breiman et collab., 1984).
- ▶ On construit un **arbre** de manière récursive :
 - ▶ A la **racine** se trouve tout l'échantillon.
 - ▶ Chaque noeud de l'arbre divise l'échantillon en 2 **branches**, selon une covariable :
 - ▶ quantitative : discrète ou continue,
 - ▶ qualitative : ordinaire (seuil) ou nominale (ensemble de catégories).
 - ▶ Un noeud terminal est appelé **feuille**.
- ▶ On parle ici d'algorithme « **greedy** » (gourmand).

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Aperçu II

- ▶ On obtient une **partition de l'espace en pavés** (pavage dyadique) de manière itérative.
- ▶ On ajuste sur chaque pavé un modèle simple :
 - ▶ Cas de la **classification supervisée** : **vote majoritaire**.
 - ▶ Cas de la **régression** : **moyenne**.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Plan

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

CART

Introduction

Exemple jouet

Principes

Cas de la classification
supervisée

Cas de la régression

Compléments

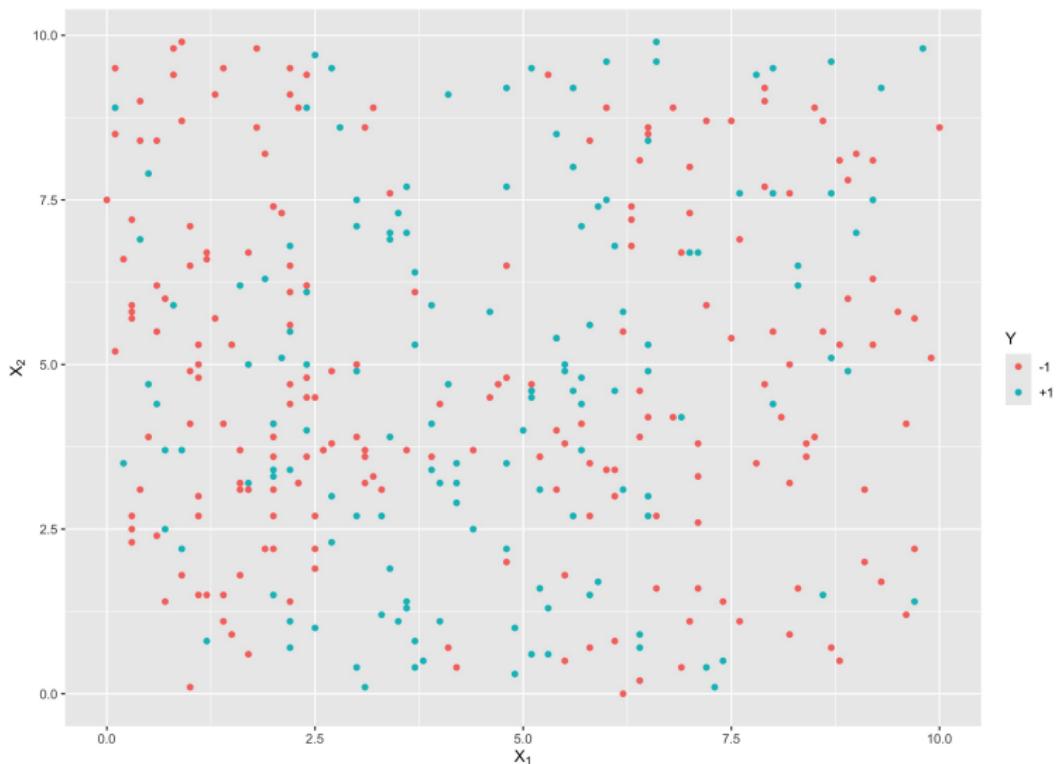
Méthodes
d'agrégation

Bagging

Boosting

Références

Exemple (Classification) I



CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Y
-1
+1

Exemple (Classification) II

207
143
0.483

Nombre de données labelées +1 x
Nombre de données labelées +1 x
Indice de Gini

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

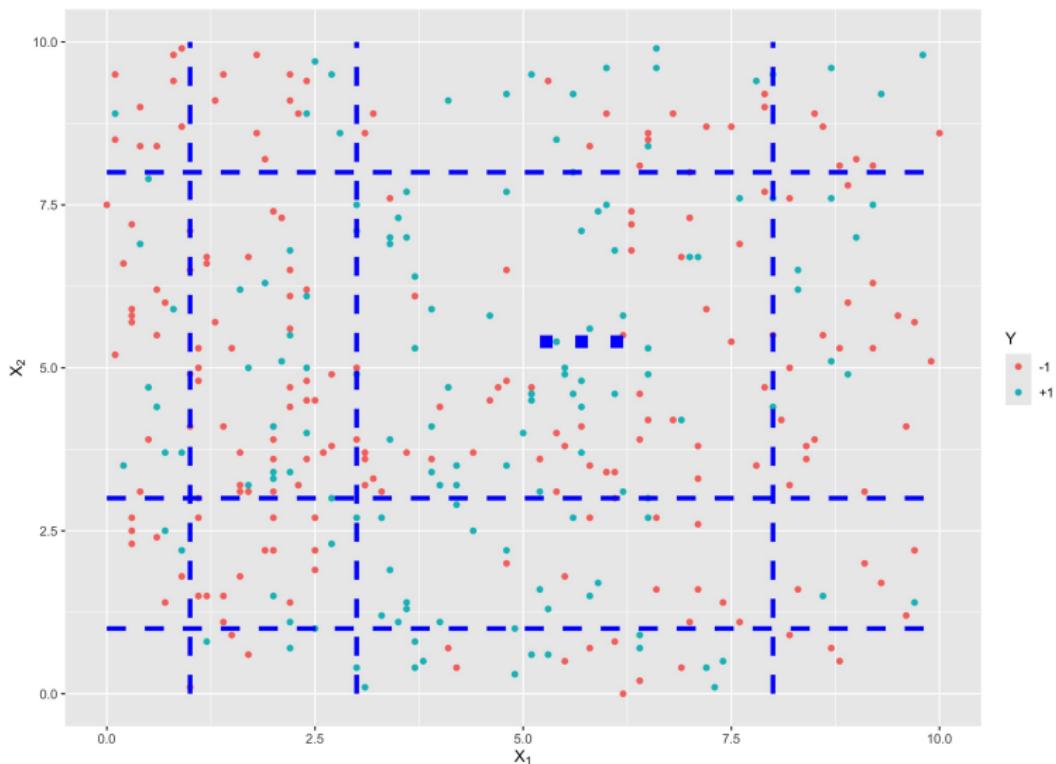
Méthodes d'agrégation

Bagging

Boosting

Références

Exemple (Classification) III



CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

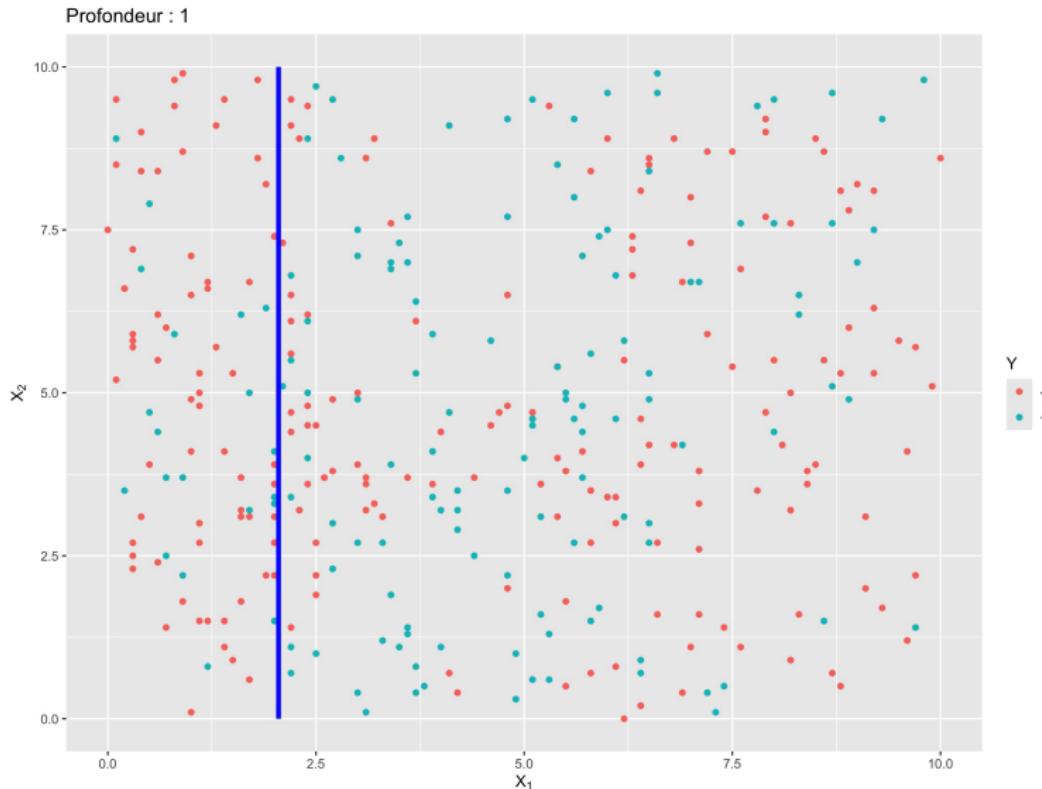
Bagging

Boosting

Références

Y
-1
+1

Exemple (Classification) IV



CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

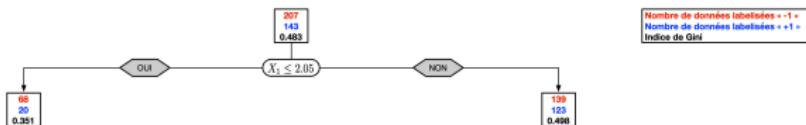
Méthodes d'agrégation

Bagging

Boosting

Références

Exemple (Classification) V



CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

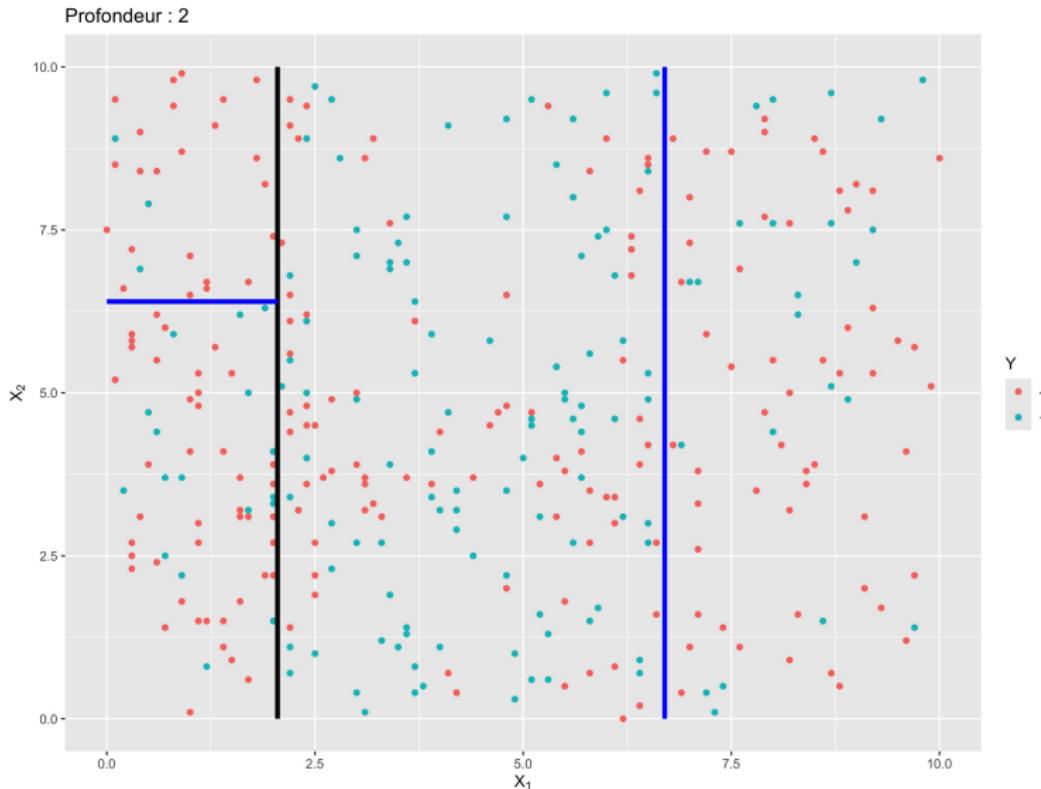
Méthodes d'agrégation

Bagging

Boosting

Références

Exemple (Classification) VI



CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

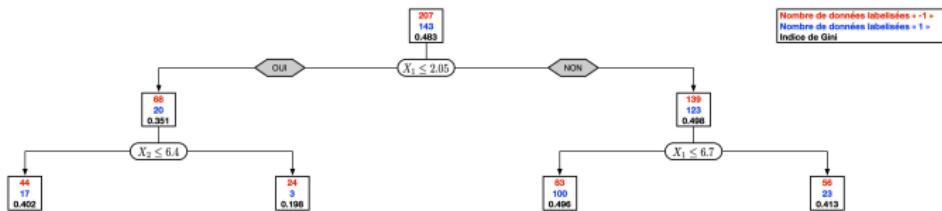
Méthodes d'agrégation

Bagging

Boosting

Références

Exemple (Classification) VII



CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

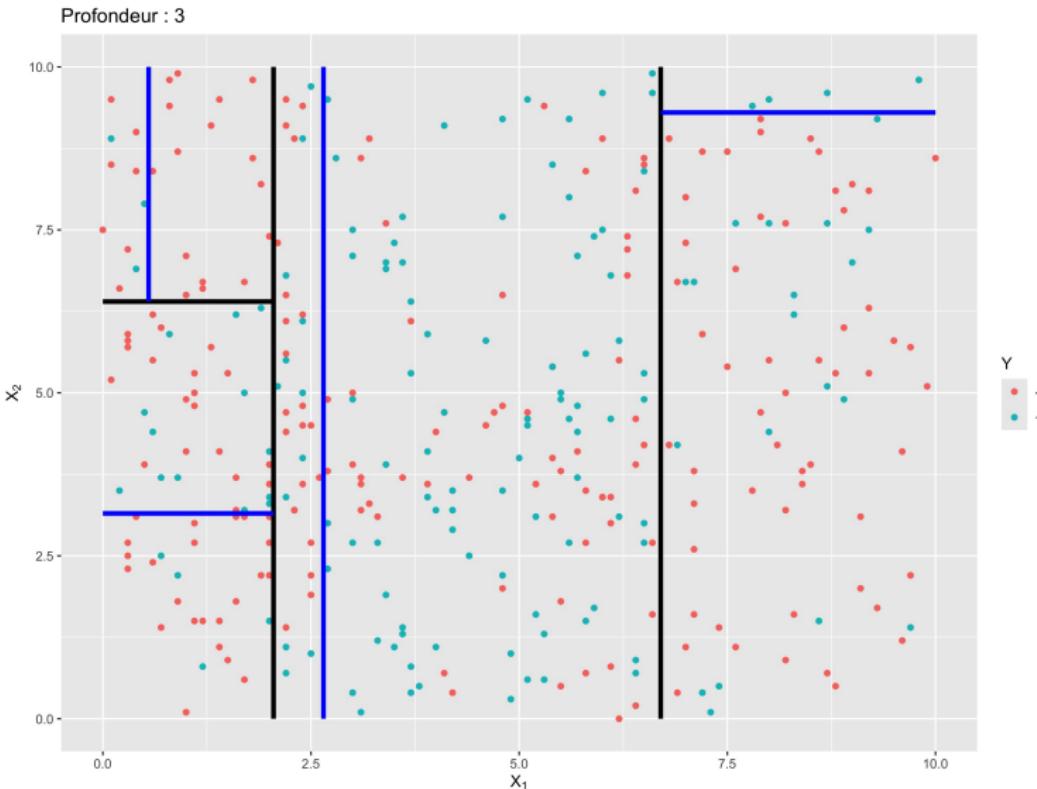
Méthodes d'agrégation

Bagging

Boosting

Références

Exemple (Classification) VIII



CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

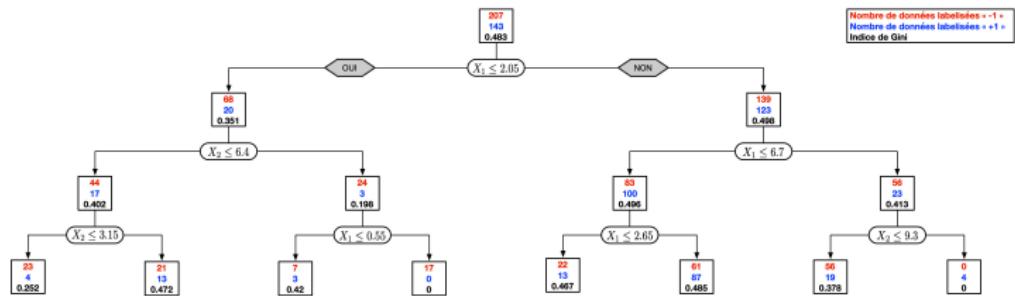
Méthodes d'agrégation

Bagging

Boosting

Références

Exemple (Classification) IX



Introduction

Exemple jouet

Principles

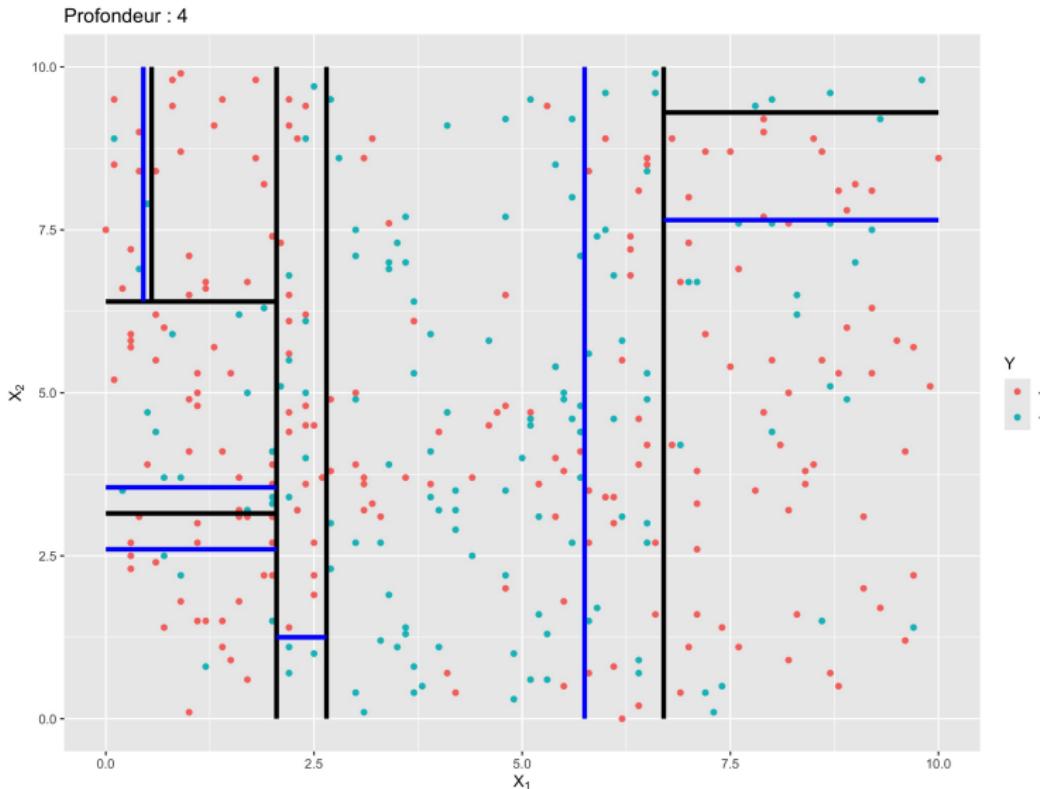
Cas de la classification supervisée

Cas de la régression

Compléments

Références

Exemple (Classification) X



CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Exemple (Classification) XI

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

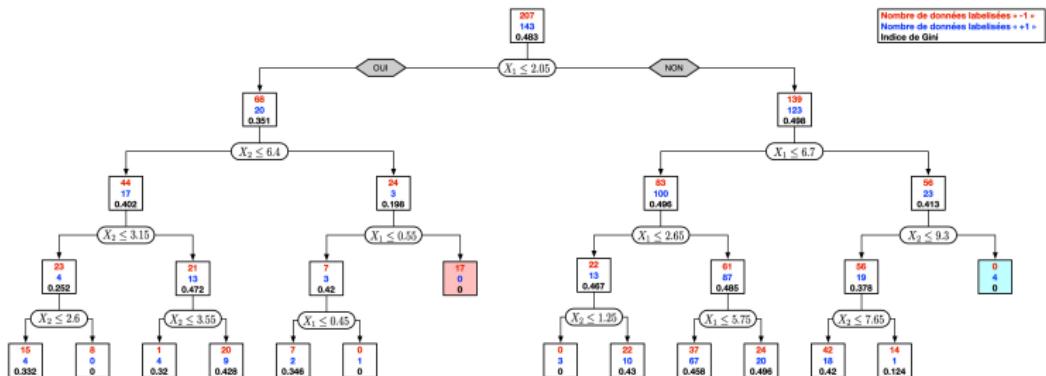
Compléments

Méthodes d'agrégation

Bagging

Boosting

Références



Exemple (Classification) XII

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

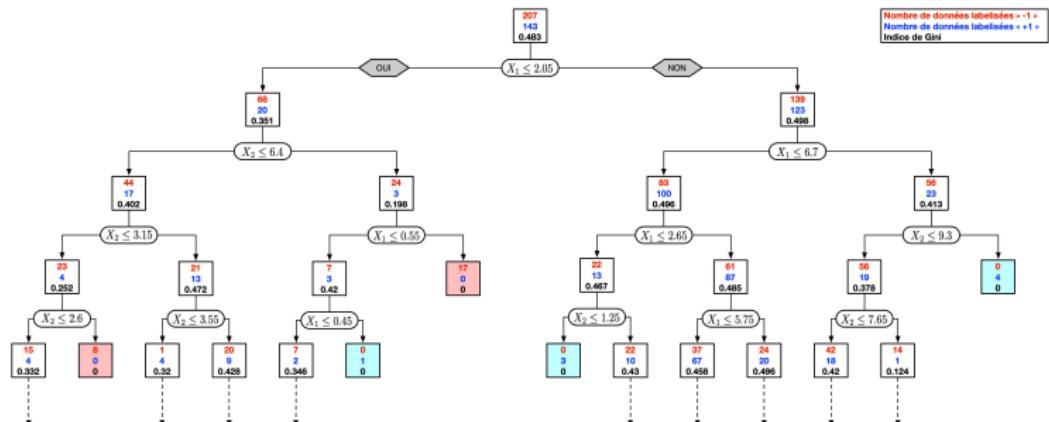
Compléments

Méthodes d'agrégation

Bagging

Boosting

Références



Plan

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Le nombre de divisions possibles

- ▶ Pour une covariable quantitative : $\ell - 1$ où $\ell \leq n$ est le nombre de valeurs distinctes prises par la covariable.
- ▶ Pour une covariable ordinaire : $\ell - 1$ où ℓ est le nombre de modalités.
- ▶ Pour des covariable nominales : $2^\ell - 1$ où ℓ est le nombre de modalités.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Cas d'une covariable nominale

- ▶ L'algorithme tend à favoriser les covariables avec beaucoup de modalités.
- ▶ Il est recommandé de réduire le nombre de modalités par fusion de certaines d'entre elles.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Enjeux techniques

- ▶ Choisir la **meilleure division pour chaque covariable**.
- ▶ Déterminer la **meilleure covariable séparatrice**.
- ▶ Décider qu'un noeud est une **feuille**.
- ▶ Estimer un **modèle de prévision dans chaque pavé**.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Sur-apprentissage

- ▶ Un arbre trop grand sur-apprend les données.
- ▶ Un arbre trop petit risque de ne pas apprendre suffisamment.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Plan

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Objectif

- ▶ **Classifie une variable qualitative Y à K classes :**

$$\{1, \dots, K\} ,$$

à partir de p covariables :

$$(X_1, \dots, X_p) .$$

- ▶ Pour simplification, considère des covariables quantitatives ici.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Prévision de la feuille

- ▶ Soit une partition en M pavés $\{R_1, \dots, R_M\}$.
- ▶ Soit C_m la classe du m -ème pavé.
- ▶ Pour $k \in \{1, \dots, K\}$, on estime $\mathbb{P}(C_m = k)$ par :

$$\hat{p}_k^m = \frac{1}{\text{Card} \{x_i \in R_m\}} \sum_{x_i \in R_m} \mathbb{1}_{y_i=k} .$$

- ▶ La classe prévue pour le m -ème pavé est la classe la plus présente dans ce pavé :

$$\hat{C}_m = \arg \max_{k \in \{1, \dots, K\}} \hat{p}_k^m .$$

CART

Introduction
Exemple jouet
Principes
Cas de la classification supervisée
Cas de la régression
Compléments

Méthodes d'agrégation

Bagging
Boosting
Références

Critère de division

- ▶ Considérons la partition binaire pour la j -ème covariable et le seuil s :

$$R_1(j, s) = \{X / X_j \leq s\} ,$$

$$R_2(j, s) = \{X / X_j > s\} .$$

- ▶ On choisit la j -ème covariable et le seuil s qui minimisent (par exemple) l'erreur de classification sur les 2 pavés ainsi constitués :

$$\arg \min_{j \in \{1, \dots, p\}} \min_s \left[\left(1 - \hat{p}_{\hat{C}_1}^1\right) + \left(1 - \hat{p}_{\hat{C}_2}^2\right) \right] .$$

- ▶ Une fois cette division optimale déterminée, on réitère l'étape de division sur les deux pavés obtenus, et ainsi de suite.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Fonction d'hétérogénéité

- ▶ On a utilisé ici une **fonction d'hétérogénéité** (« impurity ») *i* particulière : l'erreur de classification.
- ▶ Plus généralement, une fonction d'hétérogénéité doit vérifier :
 - ▶ *i* est minimale, et égale à 0, pour les configurations avec une seule classe :

$$(1, 0, 0, \dots, 0)$$

$$(0, 1, 0, \dots, 0)$$

⋮

$$(0, 0, 0, \dots, 1) .$$

- ▶ *i* est maximale pour la configuration équi-répartie :

$$\forall i \in \{1, \dots, K\} : p_i = \frac{1}{K} .$$

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Fonctions d'hétérogénéité

- ▶ Pour un pavé R_m :

- ▶ Erreur de classification :

$$i(R_m) = 1 - \hat{p}_{\widehat{C}_m}^m .$$

- ▶ Indice de Gini :

$$i(R_m) = \sum_{k=1}^K \hat{p}_k^m (1 - \hat{p}_k^m) .$$

- ▶ Entropie de Shannon :

$$i(R_m) = - \sum_{k=1}^K \hat{p}_k^m \ln(\hat{p}_k^m) .$$

CART

Introduction
Exemple jouet
Principes
Cas de la classification supervisée
Cas de la régression
Compléments
Méthodes d'agrégation
Bagging
Boosting
Références

Retour sur l'algorithme de division I

- ▶ Considérons la partition binaire pour la j -ème covariable et le point seuil s :

$$R_1(j, s) = \{X / X_j \leq s\},$$

$$R_2(j, s) = \{X / X_j > s\},$$

- ▶ On choisit la j -ème covariable et le seuil s qui minimisent (par exemple) l'**erreur de classification** sur les 2 pavés ainsi constitués :

$$\min_{j \in \{1, \dots, p\}} \min_s \left[\left(1 - \hat{p}_{\widehat{C}_1}^1\right) + \left(1 - \hat{p}_{\widehat{C}_2}^2\right) \right].$$

- ▶ Une fois cette division optimale déterminée, on **réitère** l'étape de division sur les deux pavés obtenus, et ainsi de suite.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Retour sur l'algorithme de division II

- ▶ Considérons la partition binaire pour la j -ème covariable et le point seuil s :

$$R_1(j, s) = \{X / X_j \leq s\} ,$$

$$R_2(j, s) = \{X / X_j > s\} ,$$

- ▶ On choisit la j -ème covariable et le seuil s qui minimisent la fonction d'hétérogénéité sur les 2 pavés ainsi constitués :

$$\arg \min_{j \in \{1, \dots, p\}} \min_s [i(R_1(j, s)) + i(R_2(j, s))] .$$

- ▶ Une fois cette division optimale déterminée, on réitère l'étape de division sur les deux pavés obtenus, et ainsi de suite.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Plan

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Objectif

- ▶ Régresser une variable quantitative $Y \in \mathbb{R}$: à partir de p covariables :

$$(X_1, \dots, X_p) .$$

- ▶ Pour simplification, considère des covariables quantitatives ici.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Prévision de la feuille

- ▶ Soit une partition en M pavés $\{R_1, \dots, R_M\}$.
- ▶ Soit c_m la valeur (constante) du m -ème pavé.
- ▶ On considère ici la fonction de régression suivante :

$$m(x) = \sum_{m=1}^M c_m \mathbf{1}_{x \in R_m} .$$

- ▶ Avec le critère des moindres carrés, on obtient la fonction de prévision suivante :

$$\hat{m}(x) = \sum_{m=1}^M \hat{c}_m \mathbf{1}_{x \in R_m} .$$

où :

$$\hat{c}_m = \frac{1}{\text{Card } \{x_i \in R_m\}} \sum_{x_i \in R_m} y_i .$$

CART

- Introduction
- Exemple jouet
- Principes
- Cas de la classification supervisée
- Cas de la régression**
- Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Critère de division

- ▶ Considérons la partition binaire pour la j -ème covariable et le point seuil s :

$$R_1(j, s) = \{X / X_j \leq s\},$$

$$R_2(j, s) = \{X / X_j > s\},$$

- ▶ On choisit la j -ème covariable et le seuil s qui minimisent la **variance** sur les 2 pavés ainsi constitués :

$$\arg \min_{j \in \{1, \dots, p\}} \min_s \left(\sum_{x_i \in R_1(j, s)} (y_i - \hat{c}_1)^2 + \sum_{x_i \in R_2(j, s)} (y_i - \hat{c}_2)^2 \right).$$

- ▶ Une fois cette division optimale déterminée, on **réitère** l'étape de division sur les deux pavés obtenus, et ainsi de suite.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Fonction d'hétérogénéité

- ▶ Pour un pavé R_m , on considère la **variance** comme fonction d'hétérogénéité :

$$i(R_m) = \sum_{x_i \in R_m(j,s)} (y_i - \hat{c}_m)^2 .$$

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Plan

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Critères d'arrêt

- ▶ Diminution de l'hétérogénéité inférieure à un seuil.
- ▶ Nombre de points dans une feuille (classiquement entre 1 et 5).
- ▶ Tests d'hypothèses.

CART

Introduction

Exemple jouet

Principes

Cas de la classification supervisée

Cas de la régression

Compléments

Méthodes
d'agrégation

Bagging

Boosting

Références

Elagage (*pruning*)

1. **Construire l'arbre maximal** à l'aide d'une procédure forward.

A chaque étape, trouver la meilleure division et s'arrêter lorsque toutes les feuilles contiennent moins d'une nombre fixé de points (communément entre 1 et 5) ou ont les mêmes sorties.

2. **Créer une suite imbriquée de sous-arbres**, de complexité décroissante.
3. **Elaguer les branches inutiles** (déterminer le sous-arbre optimal).

CART

- Introduction
- Exemple jouet
- Principes
- Cas de la classification supervisée
- Cas de la régression
- Compléments

- Méthodes d'agrégation
- Bagging
- Boosting
- Références

Coût de la complexité

- ▶ Pour un sous-arbre \mathcal{T} , avec un nombre de feuilles $|\mathcal{T}|$, on considère :

$$C(\mathcal{T}) = \sum_{i=1}^{|T|} i(R_i) + \lambda |\mathcal{T}| .$$

- ▶ On désigne par \mathcal{T}_{max} l'arbre maximal.
- ▶ Afin de déterminer le sous-arbre optimal $\mathcal{T}_{opt} \subset \mathcal{T}_{max}$ qui minimise C , on élague récursivement la feuille la plus faible (au sens de l'hétérogénéité i).
- ▶ Dans CART, l'hyperparamètre λ est choisi par validation croisée.

CART

Introduction
Exemple jouet
Principes
Cas de la classification supervisée
Cas de la régression
Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

Avantages et inconvénients

► Avantages :

- ▶ Aucune hypothèse sur la loi et sur la fonction de lien.
- ▶ Facilité d'implémentation.

► Inconvénients :

- ▶ Nécessité de disposer de gros jeux de données.
- ▶ Pas de prise en compte directe de l'interaction entre les variables.
- ▶ Représentation graphique agréable mais fallacieuse.
- ▶ Variance élevée (une légère modification de l'échantillon peut conduire à des résultats très différents).

CART

- Introduction
- Exemple jouet
- Principes
- Cas de la classification supervisée
- Cas de la régression
- Compléments

Méthodes d'agrégation

Bagging

Boosting

Références

En guise de conclusion

- ▶ Les arbres CART ne sont pas utilisés en pratique, à cause de leur variance élevée.
- ▶ Mais les arbres CART sont la base des méthodes d'agrégation : « grands » arbres dans le **bagging** (et les **random forests**) et « petits » arbres dans le **boosting**.

CART

Introduction
Exemple jouet
Principes
Cas de la classification supervisée
Cas de la régression
Compléments

Méthodes d'agrégation

Bagging
Boosting
Références

Plan

Méthodes d'agrégation

CART

Méthodes
d'agrégation

Bagging

Boosting

Références

Données considérées

- ▶ On dispose d'un échantillon de (X_1, \dots, X_p, Y) :

$$d_n = (x_{i1}, \dots, x_{ip}, y_i)_{i \in \{1, \dots, n\}} .$$

- ▶ On considère dans la suite que :

- ▶ $\forall j \in \{1, \dots, p\} : X_j \in \mathbb{R}$.

*Toutes les covariables sont considérés quantitatives.
Mais il est également possible de considérer des covariables qualitatives.*

- ▶ $Y \in \{-1, 1\}$ dans le cas de la classification supervisée.

On se place dans le cadre d'une classification supervisé binaire

Mais il est également possible de considérer des classifications supervisées avec K classes.

- ▶ $Y \in \mathbb{R}$ dans le cas de la régression.

CART

Méthodes
d'agrégation

Bagging

Boosting

Références

Principe de l'agrégation de modèles

1. Estimer plusieurs modèles (sur plusieurs échantillons) :

- ▶ Cas de la **classification supervisée** : $(\hat{g}_b)_{b \in \{1, \dots, B\}}$.
- ▶ Cas de la **régression** : $(\hat{m}_b)_{b \in \{1, \dots, B\}}$.

2. Agréger ces modèles :

- ▶ Cas de la **classification supervisée binaire** (à valeurs dans $\{-1, 1\}$) :

$$\hat{g} = \text{signe} \left(\sum_{b=1}^B \alpha_b \hat{g}_b \right).$$

- ▶ Cas de la **régression** :

$$\hat{m} = \sum_{b=1}^B \alpha_b \hat{m}_b.$$

CART

Méthodes d'agrégation

Bagging

Boosting

Références

Différentes méthodes d'agrégation

CART

Méthodes
d'agrégation

Bagging

Boosting

Références

► Bagging

- ▶ Du bagging « de base » : (Breiman, 1996).
Ajuster des modèles (LASSO, arbres...) sur des échantillons bootstrappés, et les agréger.
- ▶ Aux random forests : (Breiman, 2001).
Ajuster des arbres décorrélés sur des échantillons bootstrappés, et les agréger.
- ▶ Boosting : (Freund et Schapire, 1997).
Ajuster des règles faibles (ex : petits arbres) sur un échantillon répondéré de manière récursive, et les agréger.

Similitudes et différences

CART

Méthodes
d'agrégation

Bagging

Boosting

Références

- ▶ Ces agrégations sont réputées pour leur efficacité numérique.
- ▶ Ces agrégations ne sont pas adaptées aux mêmes modèles :
 - ▶ Le **bagging** est adapté pour des modèles à **forte variance** et **faible biais**.
 - ▶ Le **boosting** est adapté pour des modèles à **faible variance** et **fort biais**.

Plan

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Plan

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Origines du bootstrap

- ▶ A l'origine du mot, [The surprising adventures of Baron Munchausen](#) de Rudolph Erich Raspe :
*« The Baron had fallen to the bottom of a deep lake.
Just when it looked like all was lost, he thought to pick
himself up by his own **bootstraps**. »*
- ▶ En France, [Cyrano de Bergerac](#) d'Edmond Rostand (pour atteindre la lune) :
*« Enfin, me plaçant sur un plateau de fer,
Prendre un morceau d'aimant et le lancer en l'air !
Ça, c'est un bon moyen : le fer se précipite,
Aussitôt que l'aimant s'envole, à sa poursuite ;
On relance l'aimant bien vite, et cadédis !
On peut monter ainsi indéfiniment. »*
- ▶ Une amélioration de l'idée du jacknife : ([Efron, 1979](#)), ([Efron et Tibshirani, 1994](#)).

CART

Méthodes d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

Boosting

Références

Objectifs et principe du bootstrap

- ▶ Objectif : approcher par simulation la distribution d'un estimateur statistique de loi inconnue.
- ▶ Tandis que la validation croisée considère des ensembles de données indépendantes, le **bootstrap** fonctionne sur des échantillons de la même taille que l'original, en **rééchantillonnant** les observations via un **tirage aléatoire avec remise**.
- ▶ Dans un échantillon bootstrappé, certaines observations peuvent apparaître plusieurs fois, et d'autres ne pas apparaître du tout.
- ▶ Si n est grand, la loi de l'échantillon bootstrappé est proche de la loi de l'échantillon d'origine.

CART

Méthodes
dagrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Plan

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Principe

- ▶ Bagging (**Bootstrap aggregating**) : (Breiman, 1996).
- ▶ Agréger des modèles estimés sur des échantillons bootstrappés.

CART

Méthodes d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

Boosting

Références

Algorithme : classification supervisée (binaire)

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

1. Pour $b \in \{1, \dots, B\}$:

1.1 Tirer un échantillon bootstrappé à partir de l'échantillon \mathcal{D}_n : \mathcal{D}_n^{*b} .

1.2 Estimer un modèle sur l'échantillon bootstrappé \mathcal{D}_n^{*b} : \hat{g}_b .

2. Agréger les modèles :

$$\hat{g} = \text{signe} \left(\frac{1}{B} \sum_{b=1}^B \hat{g}_b \right).$$

Illustration I

Echantillon
d'apprentissage

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Illustration II

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

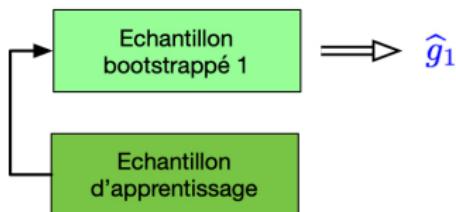


Illustration III

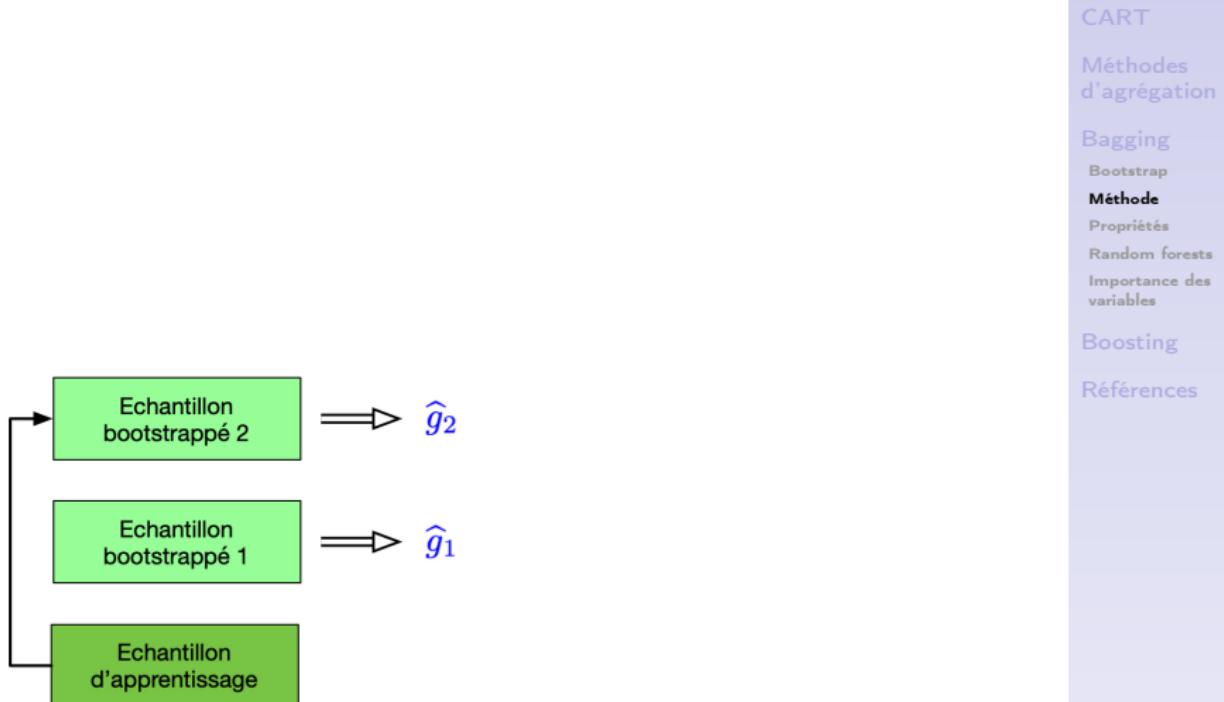
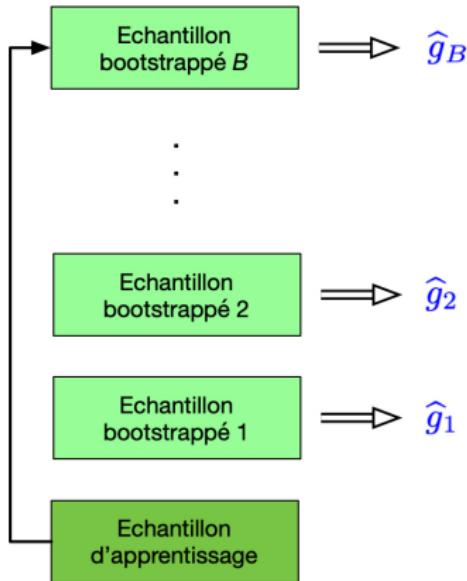


Illustration IV



CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

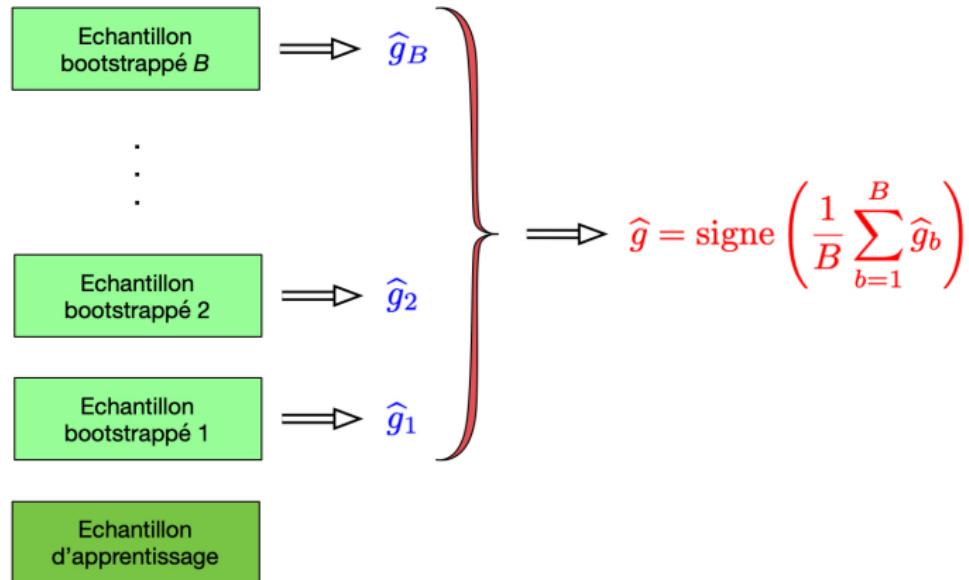
Random forests

Importance des
variables

Boosting

Références

Illustration V



CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Algorithme : régression

1. Pour $b \in \{1, \dots, B\}$:

1.1 Tirer un échantillon bootstrappé à partir de l'échantillon \mathcal{D}_n : \mathcal{D}_n^{*b} .

1.2 Estimer un modèle sur l'échantillon bootstrappé \mathcal{D}_n^{*b} : \hat{m}_b .

2. Agréger les modèles :

$$\hat{m} = \frac{1}{B} \sum_{b=1}^B \hat{m}_b.$$

CART

Méthodes d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

Boosting

Références

Plan

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Cas de la régression

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

- ▶ On se place ici dans le cadre de la régression :

$$Y = m(X) + \varepsilon .$$

- ▶ Considérons \hat{m} l'estimateur de m obtenu en agrégeant B estimateurs $\hat{m}_1, \dots, \hat{m}_B$:

$$\hat{m}(x) = \frac{1}{B} \sum_{b=1}^B \hat{m}_b(x) .$$

Biais et variance : sous hypothèse i.i.d

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

- ▶ Hypothèse : $\hat{m}_1, \dots, \hat{m}_B$ sont i.i.d.

- ▶ Biais :

$$\mathbb{E} [\hat{m}(x)] = \mathbb{E} [\hat{m}_b(x)] .$$

Le bagging ne modifie pas le biais.

- ▶ Variance :

$$\text{Var} [\hat{m}(x)] = \frac{1}{B} \text{Var} [\hat{m}_b(x)] .$$

Le bagging réduit la variance.

Biais et variance : sous hypothèse de loi identique

- ▶ **Hypothèse** : $\hat{m}_1, \dots, \hat{m}_B$ sont de même loi mais corrélées, de corrélation $\rho(x)$ en $x \in \mathbb{R}^p$.

- ▶ **Biais** :

$$\mathbb{E}[\hat{m}(x)] = \mathbb{E}[\hat{m}_b(x)].$$

Le bagging ne modifie pas le biais.

- ▶ **Variance** :

$$\text{Var}[\hat{m}(x)] = \left[\rho(x) \left(1 - \frac{1}{B} \right) + \frac{1}{B} \right] \text{Var}[\hat{m}_b(x)]$$

$$\sim \rho(x) \text{Var}[\hat{m}_b(x)] \text{ pour } B \text{ grand.}$$

Le bagging réduit la variance, d'autant plus que la corrélation entre les modèles est faible.

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Quel type de modèle considérer ?

- ▶ Il faut considérer des estimateurs sensibles à de légères perturbations de l'échantillon.
- ▶ Il faut considérer des estimateurs avec un biais faible (et donc avec une forte variance).
- ▶ Les arbres (CART) ont cette caractéristique.

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Choix du nombre d'itérations (d'estimateurs)

- ▶ Il n'y a **pas de risque de sur-apprentissage** avec le nombre d'itérations B .
- ▶ La variance tend à se stabiliser avec le nombre d'itérations.
- ▶ Il faut choisir un compromis entre le gain sur la prévision et le temps de calcul.

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Avantages du bagging

- ▶ Simplicité de la mise en oeuvre.
- ▶ Qualité de la prévision !

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests
Importance des
variables

Boosting

Références

Inconvénients du bagging

- ▶ Temps de calcul.
Mais le calcul est parallélisable.
- ▶ Aspect « boîte noire ».
Mais il est possible d'estimer empiriquement l'importance d'une variable.
- ▶ Les différents estimateurs obtenus sur des échantillons bootstrappés, ne peuvent pas être considérés comme indépendants.
Mais il existe des solutions comme les random forests qui introduisent une nouvelle source d'aléa pour rendre des arbres plus indépendants.

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Plan

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Algorithmes

- ▶ (Breiman, 2001).
- ▶ Pour $b \in \{1, \dots, B\}$:

1. Tirer un échantillon bootstrappé à partir de l'échantillon \mathcal{D}_n : \mathcal{D}_n^{*b} .
2. Estimer un arbre à partir de d variables tirées au sort parmi les p variables disponibles sur l'échantillon bootstrappé \mathcal{D}_n^{*b} .
3. Agréger les modèles :
 - ▶ Pour la classification supervisée : vote majoritaire.
 - ▶ Pour la régression : moyenne des prévisions obtenues.

CART

Méthodes d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

Boosting

Références

Paramètres par défaut

CART

Méthodes
dagrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

- ▶ Nombre minimum d'observations dans les feuilles :
 - ▶ Pour la **classification supervisée** : 1.
 - ▶ Pour la **régression** : 5.
- ▶ Nombre de variables considérées pour chaque arbre :
 - ▶ Pour la **classification supervisée** : \sqrt{p} .
 - ▶ Pour la **régression** : $\frac{p}{3}$.

Quelques références internet

- ▶ Leo Breiman and Adele Cutler :

[http://www.stat.berkeley.edu/~breiman/
RandomForests/](http://www.stat.berkeley.edu/~breiman/RandomForests/)

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

- ▶ Andrej Karpathy :

[http://cs.stanford.edu/people/karpathy/svmjs/
demo/demoforest.html](http://cs.stanford.edu/people/karpathy/svmjs/demo/demoforest.html)

Plan

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

**Importance des
variables**

Boosting

Références

Erreur Out Of Bag : objectif

- ▶ L'**erreur Out Of Bag (OOB)** permet d'estimer l'erreur de prévision (sans procéder par validation croisée).
- ▶ Elle vaut :
 - ▶ Pour la **classification supervisée** :

$$\mathbb{P}(g(X) \neq Y) .$$

- ▶ Pour la **régression** :

$$\mathbb{E} \left[(Y - m(X))^2 \right] .$$

CART

Méthodes
d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des
variables

Boosting

Références

Erreurs Out Of Bag : classification supervisée (binaire)

► Pour l'observation i :

- Soit \mathcal{I}_i les indices des échantillons bootstrappés \mathcal{D}_n^{*b} qui ne contiennent pas (x_i, y_i) .
- On agrège les modèles $b \in \mathcal{I}_i$ pour déterminer la prévision de y_i :

$$\hat{y}_i = \text{signe} \left(\frac{1}{\text{Card}(\mathcal{I}_i)} \sum_{b \in \mathcal{I}_i} \hat{g}_b(x_i) \right).$$

► On calcule l'erreur Out Of Bag (EOOB) :

$$\text{EOOB} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\hat{y}_i \neq y_i}.$$

CART

Méthodes d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

Boosting

Références

Erreurs Out Of Bag : régression

- ▶ Pour l'observation i :

- ▶ Soit \mathcal{I}_i les indices des échantillons bootstrappés \mathcal{D}_n^{*b} qui ne contiennent pas l'observation (x_i, y_i) .
- ▶ On agrège les modèles $b \in \mathcal{I}_i$ pour déterminer la prévision de y_i :

$$\hat{y}_i = \frac{1}{\text{Card}(\mathcal{I}_i)} \sum_{b \in \mathcal{I}_i} \hat{m}_b(x_i) .$$

- ▶ On calcule l'erreur Out Of Bag (EOOB) :

$$\text{EOOB} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 .$$

CART

Méthodes d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

Boosting

Références

Importance des variables : cas de la régression I

- ▶ Pour $b \in \{1, \dots, B\}$:

- ▶ Soit OOB_b l'ensemble Out Of Bag des observations ne figurant pas dans l'échantillon bootstrappé b .
- ▶ On calcule EOOB_b l'erreur Out Of Bag des observations de l'ensemble OOB_b :

$$\text{EOOB}_b = \frac{1}{\text{Card}(\text{OOB}_b)} \sum_{i \in \text{OOB}_b} [\hat{m}_b(x_i) - y_i]^2.$$

- ▶ Pour $j \in \{1, \dots, p\}$:

- ▶ Soit $\text{OOB}_{b,j}$ l'ensemble des observations obtenu à partir de OOB_b en permutant aléatoirement les valeurs de la variable X_j .
- ▶ On calcule $\text{EOOB}_{b,j}$ l'erreur Out Of Bag des observations de l'ensemble $\text{OOB}_{b,j}$:

$$\text{EOOB}_{b,j} = \frac{1}{\text{Card}(\text{OOB}_{b,j})} \sum_{i \in \text{OOB}_{b,j}} [\hat{m}_b(x_i) - y_i]^2.$$

CART
Méthodes d'agrégation
Bagging
Bootstrap
Méthode
Propriétés
Random forests
Importance des variables
Boosting
Références

Importance des variables : cas de la régression II

- ▶ On considère comme critère d'importance de la variable X_j la différence moyenne sur tous les échantillons b entre EOOB_b et $\text{EOOB}_{b,j}$.
- ▶ Plus la différence est élevée, plus on peut considérer que la variable X_j a eu de l'importance pour l'échantillon b .
- ▶ On moyenne ces écarts pour tous les échantillons $b \in \{1, \dots, B\}$, et on obtient l'**importance** de la variable X_j :

$$\text{Imp}(X_j) = \frac{1}{B} \sum_{b=1}^B (\text{EOOB}_{b,j} - \text{EOOB}_b) .$$

CART

Méthodes d'agrégation

Bagging

Bootstrap

Méthode

Propriétés

Random forests

Importance des variables

Boosting

Références

Plan

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Plan

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Idée

- ▶ Le boosting est une méthode d'agrégation de prédicteurs récursive : le prédicteur obtenu à l'itération b dépend de celui obtenu à l'itération $b - 1$.
- ▶ Le boosting consiste à utiliser une **règle faible** (*weak learner*) qui apprend tout d'abord sur l'échantillon le plus simple, celui de départ, puis au fur et à mesure sur des échantillons rendus plus complexes par des pondérations adéquates, récursives.
- ▶ Contrairement au bagging, le boosting peut conduire à du sur-apprentissage : il faudra donc veiller à limiter le nombre d'itérations.
- ▶ C'est une méthode souvent utilisée en pratique au vu des bons résultats obtenus.
- ▶ Le boosting permet de traiter des problématiques de régression et de classification supervisée.

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Principe I



CART

Méthodes
dagrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Principe II

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

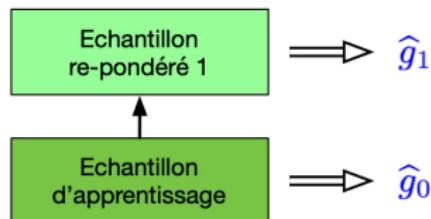
AdaBoost

Gradient boosting

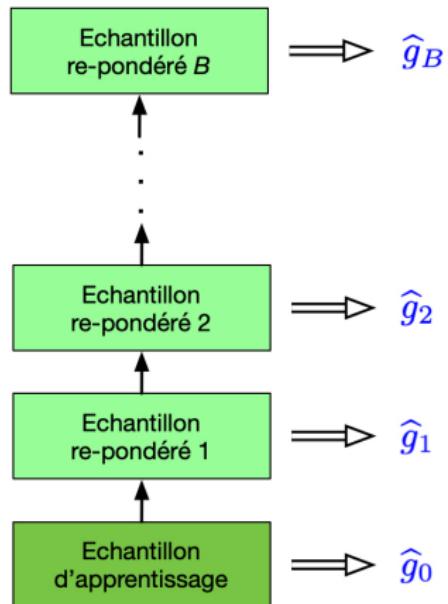
Fonctions de perte
classiques

XGBoost

Références



Principe III



CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

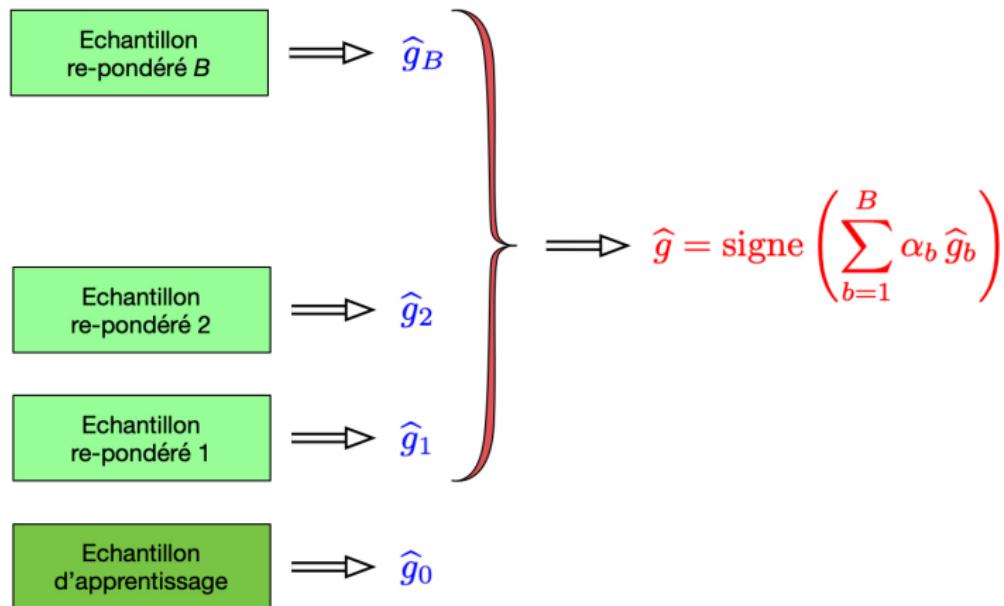
Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Principe IV



CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Enjeux

- ▶ Comment répondre aux observations à chaque étape ?
- ▶ Quels poids accorder aux estimateurs à chaque étape ?

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Plan

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Règle faible

- ▶ On appelle une **règle faible** (*weak learner*) un prédicteur légèrement meilleur que le hasard.
- ▶ Par exemple, dans le cas de classification :

$$\mathbb{P}(g(X) \neq Y) = \frac{1}{2} - \gamma .$$

où $\gamma > 0$.

- ▶ On considère usuellement comme règle faible : 1-plus proche voisin, arbre à 2 feuilles, etc.

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Historiquement : AdaBoost I

(Freund et Schapire, 1997)

1. Initialisation des poids des individus :

$$\forall i \in \{1, \dots, n\} : \omega_i^{(1)} = \frac{1}{n} .$$

2. Pour $b \in \{1, \dots, B\}$:

2.1 Estimer \hat{g}_b avec les poids $(\omega_1^{(b)}, \dots, \omega_n^{(b)})$ pour l'échantillon.

2.2 Calcul du taux d'erreur e_b de \hat{g}_b :

$$e_b = \frac{\sum_{i=1}^n \omega_i^{(b)} \mathbb{1}_{\hat{g}_b(x_i) \neq y_i}}{\sum_{i=1}^n \omega_i^{(b)}} .$$

2.3 Calcul de la pénalité :

$$\alpha_b = \ln \left(\frac{1 - e_b}{e_b} \right) .$$

2.4 Calcul des nouveaux poids :

$$\forall i \in \{1, \dots, n\} : \omega_i^{(b+1)} = \omega_i^{(b)} \exp (\alpha_b \mathbb{1}_{\hat{g}_b(x_i) \neq y_i}) .$$

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Historiquement : AdaBoost II

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

3. Le prédicteur obtenu au final est :

$$\hat{g} = \text{signe} \left(\sum_{b=1}^B \alpha_b \hat{g}_b \right).$$

Remarques

- ▶ L'étape 2.1 implique que la méthode retenue soit en mesure prendre en compte des poids. Dans le cas contraire, l'étape 2.1 d'estimation de \hat{g}_b s'effectue sur un échantillon de dimension n issu du tirage au sort d'observations de d_n avec remise selon les poids $(\omega_1^{(b)}, \dots, \omega_n^{(b)})$.
- ▶ Les poids sont modifiés de manière à accroître l'importance des observations mal classées et diminuer celle des observations bien classées.
- ▶ Le poids α_b du prédicteur \hat{g}_b augmente avec sa performance : α_b augmente lorsque e_b diminue.

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Quelques résultats théoriques

- On a :

$$R_n(\hat{g}) \leq \exp \left(-2 \sum_{b=1}^B \gamma_b^2 \right)$$

où $\gamma_b = \frac{1}{2} - e_b$ est le **gain** de \hat{g}_b par rapport à une décision basée sur le hasard pur.

Si on est meilleur que le hasard alors le risque empirique tend exponentiellement vers 0 avec B.

- On peut montrer que plus B est grand, plus le biais est faible mais plus la variance est élevée.
Il y a un risque de sur-apprentissage si B est trop important. Il faut donc contrôler B par validation croisée.
- On peut montrer qu'AdaBoost est un algorithme qui permet de minimiser le risque empirique « convexifié », et s'inscrit ainsi dans le cadre des méthodes de **gradient boosting**.

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Plan

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Une idée générale : la convexification du risque I

- ▶ Si on considère la fonction de perte suivante :

$$\ell(y, y') = \mathbb{1}_{y \neq y'} ,$$

le risque du prédicteur g vaut :

$$R(g) = \mathbb{E}(\mathbb{1}_{g(X) \neq Y}) = \mathbb{P}(g(X) \neq Y) .$$

- ▶ On aimerait trouver l'estimateur qui minimise ce risque, malheureusement ce dernier n'est pas calculable en pratique.
- ▶ On considère alors le risque empirique sur \mathcal{D}_n :

$$R_n(g) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{g(x_i) \neq y_i} .$$

- ▶ Malheureusement, ce risque empirique n'est pas convexe et donc difficile à optimiser.

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Une idée générale : la convexification du risque II

- ▶ On souhaite donc trouver sur une fonction de perte telle que le risque empirique soit convexe, et donc facile à optimiser.
- ▶ Il nous suffit pour cela de considérer une fonction $\ell : (u, y) \mapsto \ell(u, y)$ convexe en u .
- ▶ Dans le cas de la classification supervisée, on peut par exemple choisir :

$$\ell : \mathbb{R} \times \{-1, 1\} \rightarrow \mathbb{R}^+$$

$$(u, y) \mapsto \exp(-y u)$$

qui est bien convexe en u .

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Optimisation

- ▶ On considère donc une fonction de perte telle que le risque empirique soit convexe, et donc facile à optimiser.
- ▶ On recherche la solution, de manière récursive, au problème :

$$\min_g \frac{1}{n} \sum_{i=1}^n \ell(g(x_i), y_i).$$

- ▶ On peut utiliser l'algorithme de Newton-Raphson ou plutôt l'algorithme de descente de gradient fonctionnel.

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Algorithme de Newton-Raphson

- Soit :

$$J(g) = \sum_{i=1}^n \ell(g(x_i), y_i) .$$

- Avec la notation :

$$\hat{g}_b = (\hat{g}_b(x_1), \dots, \hat{g}_b(x_n))^\top ,$$

la formule de récurrence de l'algorithme de Newton-Raphson est :

$$\hat{g}_b = \hat{g}_{b-1} - \eta \nabla J(\hat{g}_{b-1})$$

où η est le taux d'apprentissage (*learning rate*).

- La fonction \hat{g}_b n'est calculées qu'aux points (x_1, \dots, x_n) , c'est notamment pour cela qu'on préfère utiliser l'algorithme de descente de gradient fonctionnel.

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Algorithme de descente de gradient fonctionnel : cas de la classification supervisée

1. Initialisation :

$$\widehat{g}_0 = \arg \min_{c \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \ell(c, y_i).$$

2. Pour $b \in \{1, \dots, B\}$:

2.1 Pour $i \in \{1, \dots, n\}$: calculer les opposés du gradient :

$$U_i = -\frac{\partial}{\partial g(x_i)} \ell(g(x_i), y_i) \Big|_{g(x_i) = \widehat{g}_{b-1}(x_i)}.$$

2.2 Estimer la règle faible sur l'échantillon

$$((x_1, U_1), \dots, (x_n, U_n)).$$

On obtient ainsi \widehat{h}_b .

2.3 Mettre à jour :

$$\widehat{g}_b(x) = \widehat{g}_{b-1}(x) + \eta \widehat{h}_b(x)$$

où η est le taux d'apprentissage (*learning rate*).

3. Le prédicteur obtenu au final est :

$$\widehat{g} = \text{signe}(\widehat{g}_B).$$

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Algorithme de descente de gradient fonctionnel : cas de la régression

1. Initialisation :

$$\hat{m}_0 = \arg \min_{c \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \ell(c, y_i).$$

2. Pour $b \in \{1, \dots, B\}$:

2.1 Pour $i \in \{1, \dots, n\}$: calculer les opposés du gradient :

$$U_i = -\frac{\partial}{\partial m(x_i)} \ell(m(x_i), y_i) \Big|_{m(x_i)=\hat{m}_{b-1}(x_i)}.$$

2.2 Estimer la règle faible sur l'échantillon

$$((x_1, U_1), \dots, (x_n, U_n)).$$

On obtient ainsi \hat{h}_b .

2.3 Mettre à jour :

$$\hat{m}_b(x) = \hat{m}_{b-1}(x) + \eta \hat{h}_b(x)$$

où η est le taux d'apprentissage (*learning rate*).

3. Le prédicteur obtenu au final est :

$$\hat{m} = \hat{m}_B.$$

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Remarques

- ▶ On peut mettre en évidence l'agrégation en remarquant que :

$$\widehat{g}_B(x) = \widehat{g}_0(x) + \eta \sum_{b=1}^B \widehat{h}_b(x).$$

- ▶ La vitesse de minimisation dépend des deux hyperparamètres η et B : si η diminue, B augmente.

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Retour sur AdaBoost

- ▶ AdaBoost est équivalent à un problème de gradient boosting avec $\lambda = 1$ et la fonction de perte :

$$\ell : \mathbb{R} \times \{-1, 1\} \rightarrow \mathbb{R}^+$$

$$(u, y) \mapsto \exp(-y u)$$

- ▶ $\hat{g}(x)$ est un estimateur de :

$$g^*(x) = \frac{1}{2} \ln \left(\frac{\mathbb{P}(Y=1/X=x)}{\mathbb{P}(Y=-1/X=x)} \right).$$

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Justification de la règle faible

- ▶ Le **boosting réduit le biais mais pas la variance.**
- ▶ La règle faible utilisée doit donc avoir une variance faible (et donc un biais élevé), comme les arbres avec très peu de feuilles.
- ▶ Considérer des règles non faibles n'améliore pas les performances de l'algorithme en général.

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

Choix du nombre d'itérations I

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

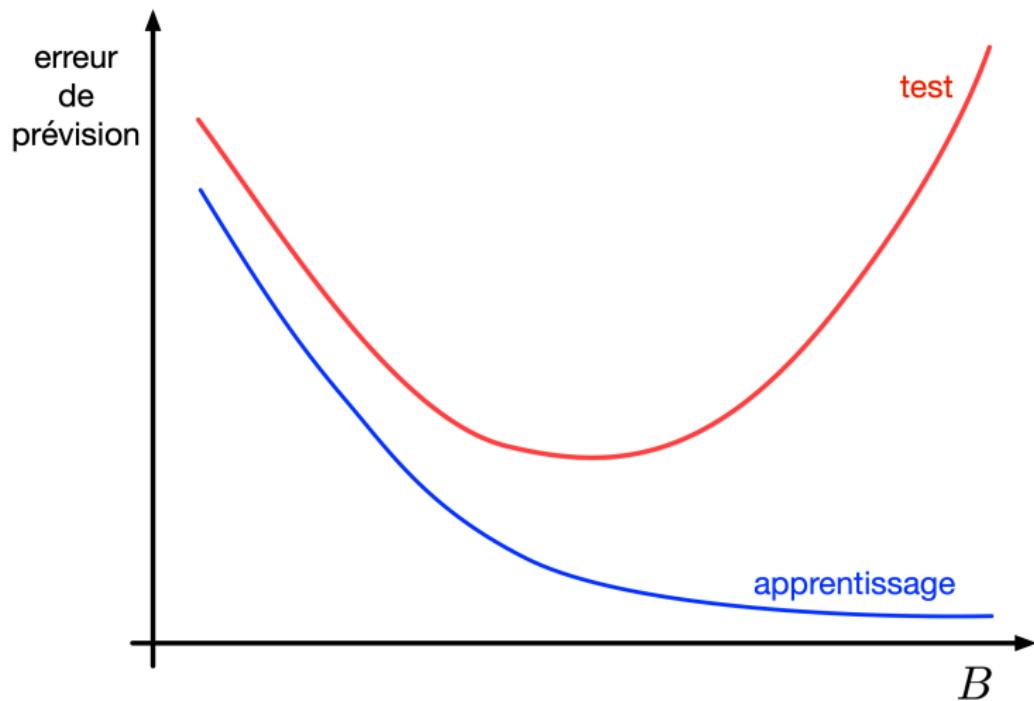
Fonctions de perte
classiques

XGBoost

Références

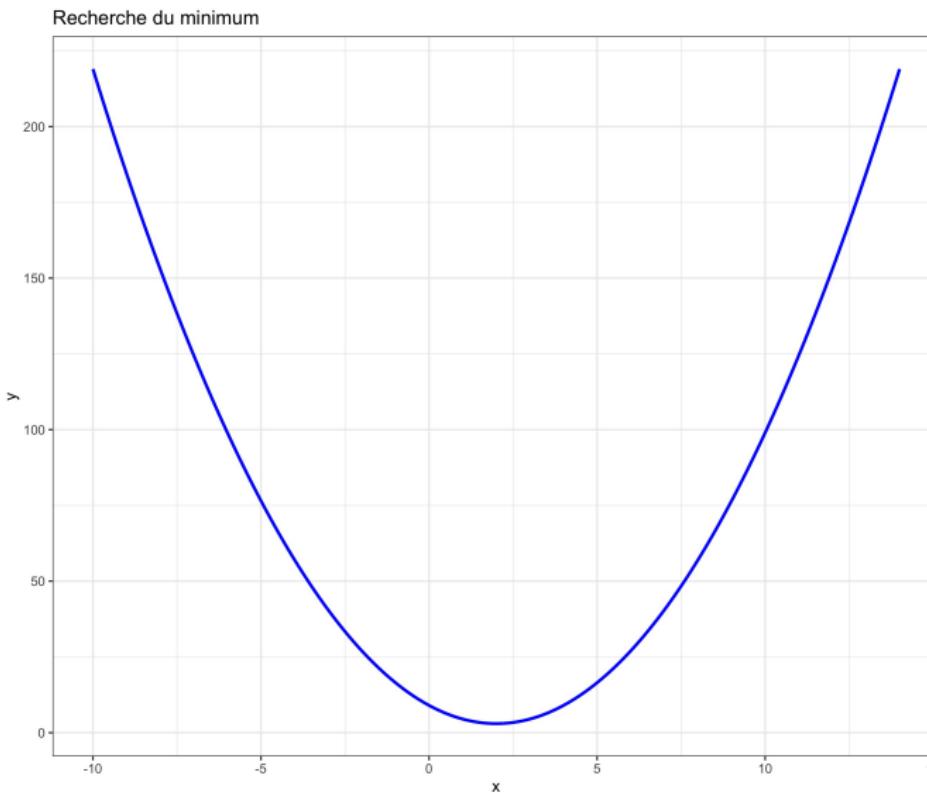
- ▶ Le biais diminue avec B .
- ▶ Mais lorsque B est « trop » grand, la variance est importante : il existe un risque de sur-apprentissage.
- ▶ On choisit B par validation croisée.

Choix du nombre d'itérations II



CART
Méthodes d'agrégation
Bagging
Boosting
Introduction
AdaBoost
Gradient boosting
Fonctions de perte classiques
XGBoost
Références

Taux d'apprentissage : exemple 1 |



CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

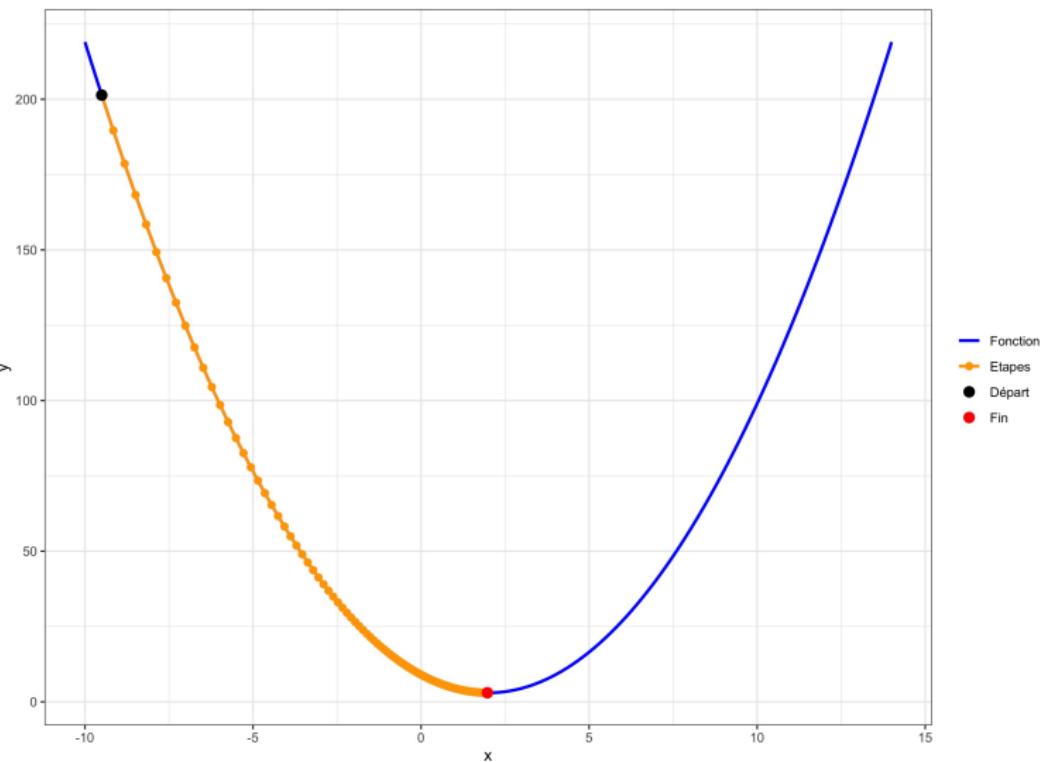
Fonctions de perte
classiques

XGBoost

Références

Taux d'apprentissage : exemple 1 II

Recherche du minimum : $\eta = 0.01$



CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

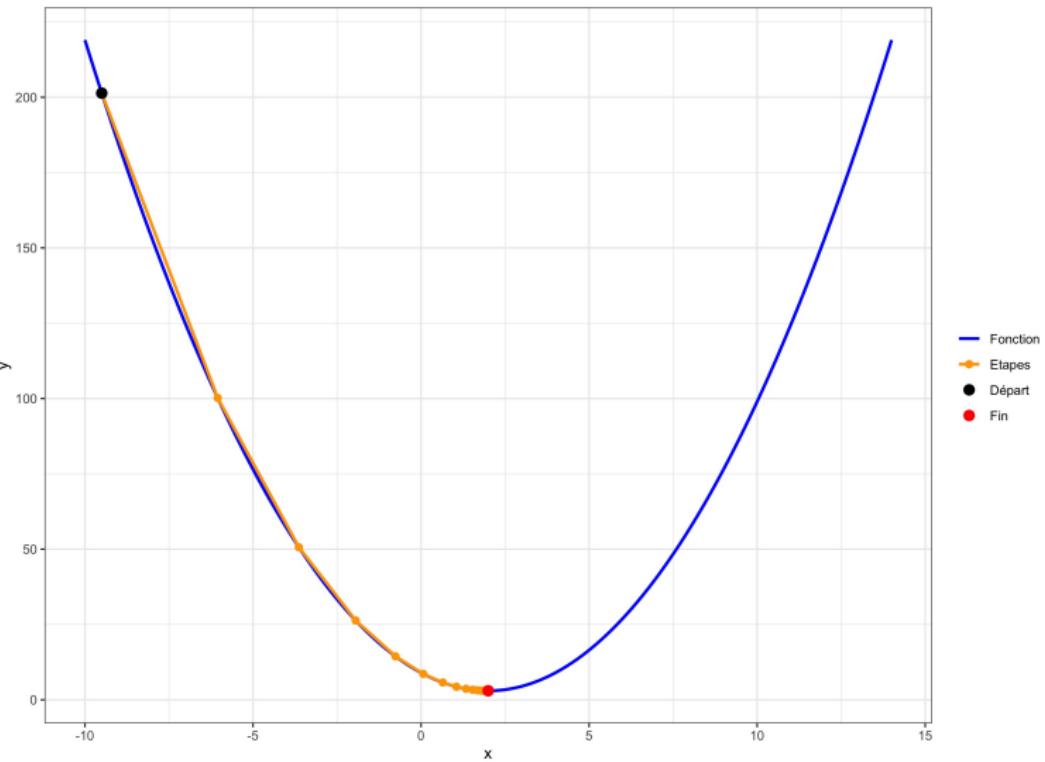
Fonctions de perte classiques

XGBoost

Références

Taux d'apprentissage : exemple 1 III

Recherche du minimum : $\eta = 0.1$



CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

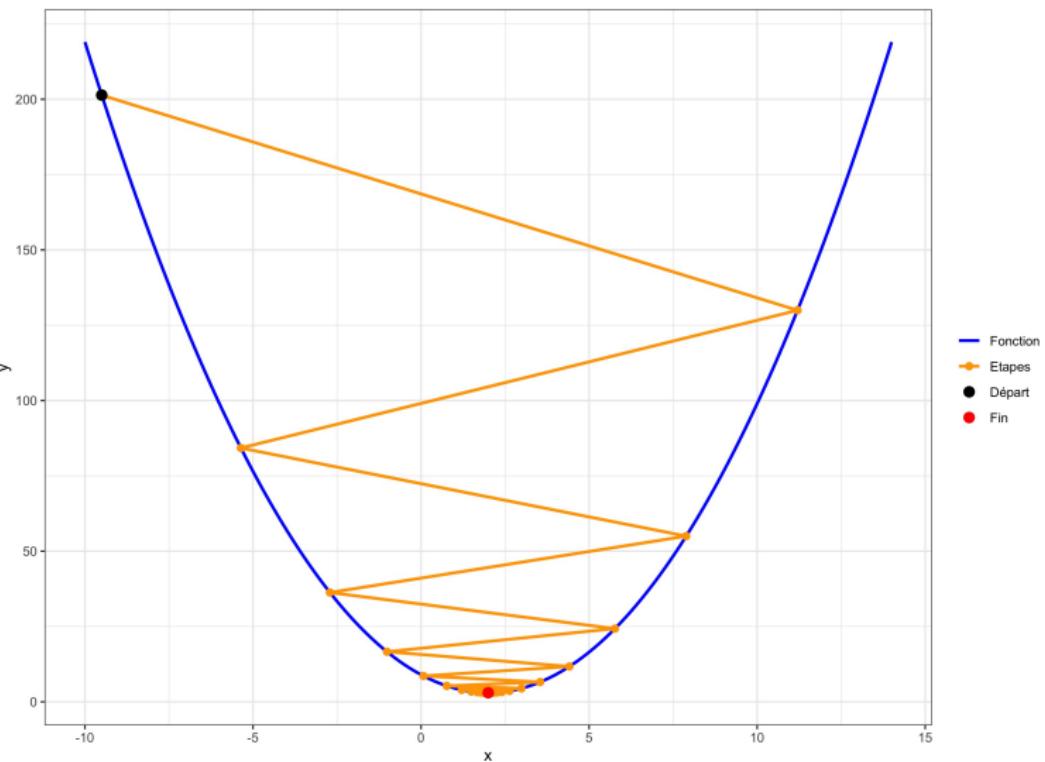
Fonctions de perte classiques

XGBoost

Références

Taux d'apprentissage : exemple 1 IV

Recherche du minimum : $\eta = 0.6$



CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

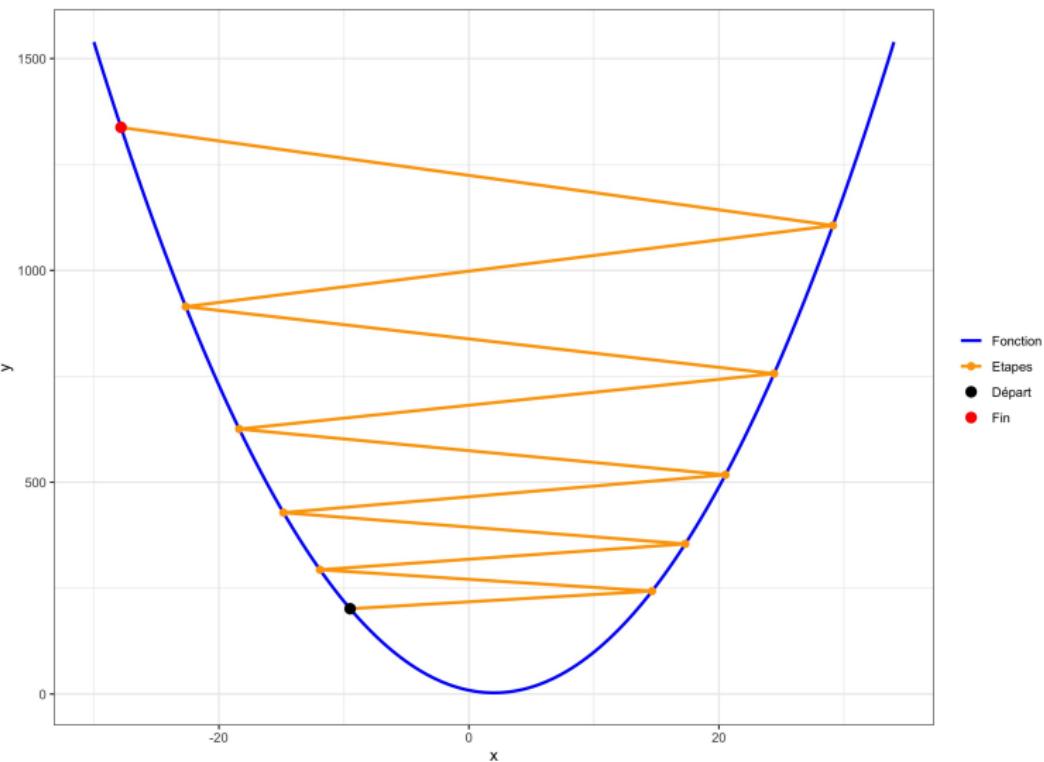
Fonctions de perte classiques

XGBoost

Références

Taux d'apprentissage : exemple 1 V

Recherche du minimum : $\eta = 0.7$



CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

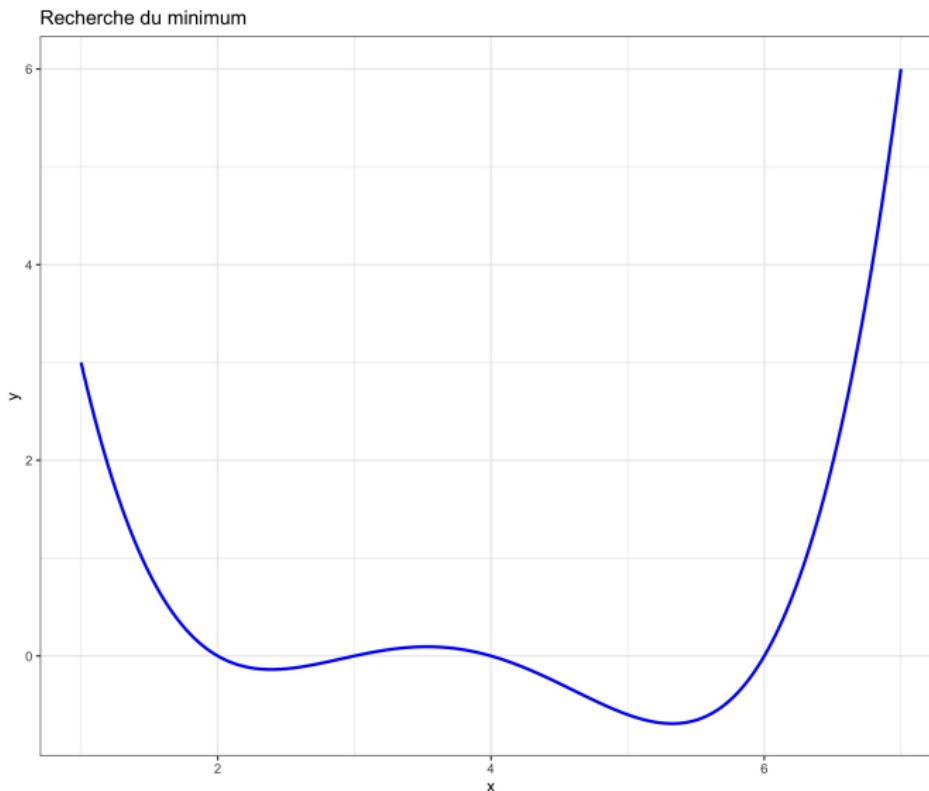
Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Taux d'apprentissage : exemple 2 I



CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

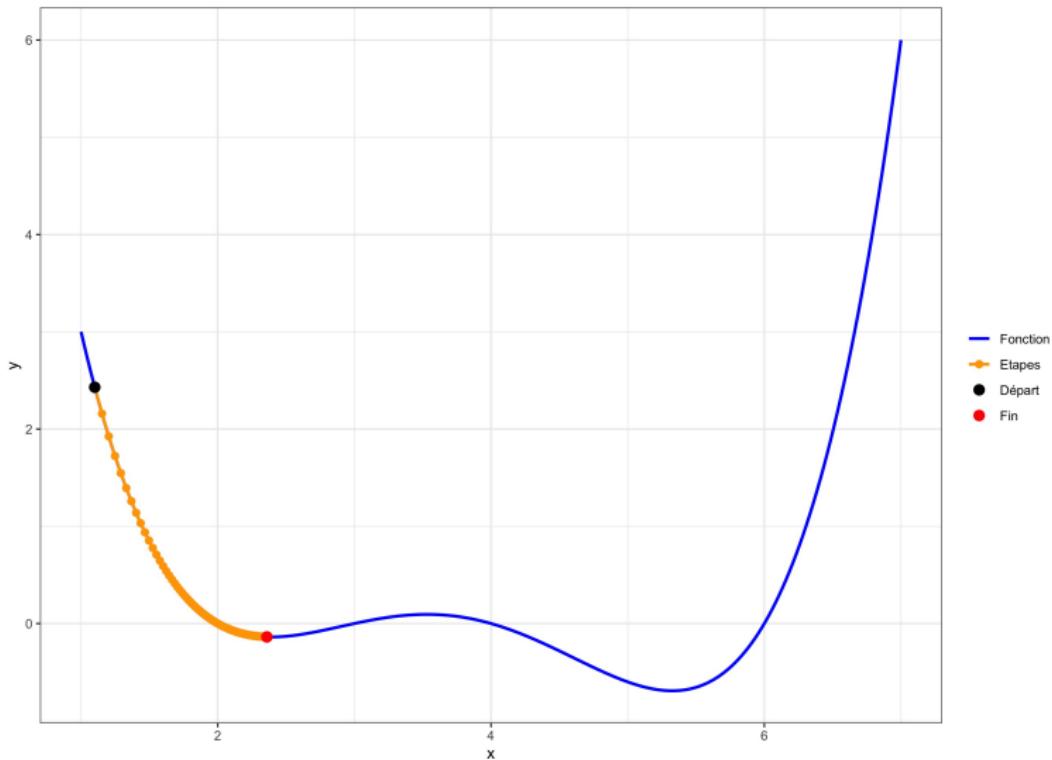
Fonctions de perte
classiques

XGBoost

Références

Taux d'apprentissage : exemple 2 II

Recherche du minimum : $\eta = 0.01$



CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

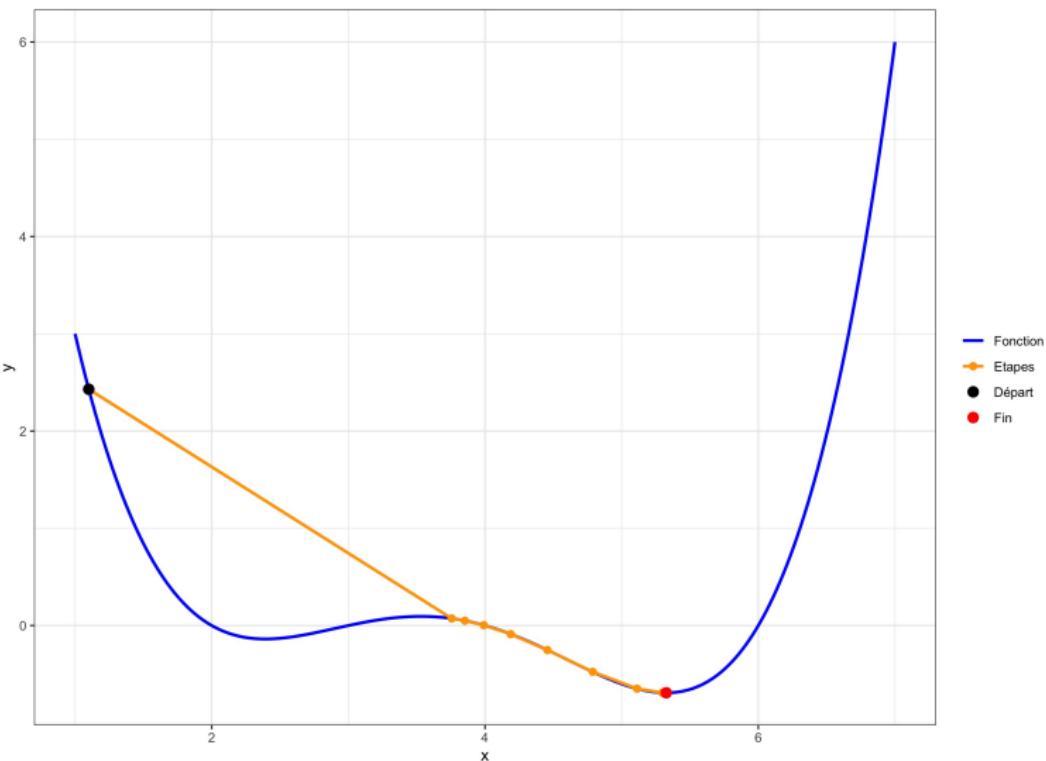
Fonctions de perte classiques

XGBoost

Références

Taux d'apprentissage : exemple 2 III

Recherche du minimum : $\eta = 0.5$



CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Plan

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

**Fonctions de perte
classiques**

XGBoost

Références

LogitBoost I

- Le modèle de régression logistique, entre une variable $Y \in \{0, 1\}$ et des covariables $X = (X_1, \dots, X_p)^\top \in \mathbb{R}^p$, pose :

$$\begin{aligned} p(x) &:= \mathbb{P}(Y = 1 / X = x) \\ &= \frac{\exp(\beta^\top x)}{1 + \exp(\beta^\top x)} = \frac{1}{1 + \exp(-\beta^\top x)} \end{aligned}$$

où $\beta \in \mathbb{R}^p$ est un paramètre inconnu estimé par maximum de vraisemblance.

- L'idée de **LogitBoost** est de supprimer l'hypothèse de linéarité de $p(x)$, en posant :

$$p(x) = \frac{1}{1 + \exp(-2g(x))}$$

où $g : \mathbb{R}^p \rightarrow \mathbb{R}$ est une fonction inconnue.

CART
Méthodes d'agrégation
Bagging
Boosting
Introduction
AdaBoost
Gradient boosting
Fonctions de perte classiques
XGBoost

Références

LogitBoost II

- ▶ La maximisation de :

$$\ln \left[(p(x))^y (1 - p(x))^{1-y} \right] .$$

est équivalente à la minimisation de :

$$\ln [1 + \exp (-2\tilde{y}g(x))]$$

où $\tilde{y} = 2y - 1 \in \{-1, 1\}$.

- ▶ La fonction :

$$\ell : \mathbb{R} \times \{-1, 1\} \rightarrow \mathbb{R}^+$$

$$(u, y) \mapsto \ln (1 + \exp (-2yu))$$

est bien convexe en u .

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

LogitBoost III

- ▶ LogitBoost désigne donc l'algorithme de gradient boosting avec la fonction de perte précédente.
- ▶ Comme pour AdaBoost, la solution $\hat{g}(x)$ de LogitBoost est un estimateur de :

$$g^*(x) = \frac{1}{2} \ln \left(\frac{\mathbb{P}(Y = 1 / X = x)}{\mathbb{P}(Y = -1 / X = x)} \right).$$

- ▶ Une fois la fonction \hat{g} déterminée, on obtient la règle de décision suivante :

$$y = \begin{cases} 1 & \text{si } \hat{p}(x) \geq \frac{1}{2} \\ -1 & \text{sinon} \end{cases},$$

soit

$$y = \begin{cases} 1 & \text{si } \hat{g}(x) \geq 0 \\ -1 & \text{sinon} \end{cases}.$$

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Autres exemples en classification supervisée

- ▶ On peut considérer différentes fonctions φ telles que :

$$\ell(g(x), y) = \varphi(y g(x)) .$$

- ▶ Par exemple :

- ▶ la fonction de perte quadratique :

$$\varphi(x) = (1 - x)^2 ,$$

- ▶ la fonction de perte quadratique tronquée :

$$\varphi(x) = [(1 - x)_+]^2 ,$$

- ▶ la fonction de perte Hinge (SVM) :

$$\varphi(x) = (1 - x)_+ .$$

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

L_2 -Boosting I

- Le L_2 -Boosting s'applique dans le cadre de la régression.
- L_2 -Boosting désigne l'algorithme de gradient boosting avec la fonction de perte :

$$\ell : \quad \mathbb{R}^2 \quad \rightarrow \quad \mathbb{R}^+$$

$$(u, y) \quad \mapsto \quad \frac{1}{2} (u - y)^2$$

- La solution $\hat{m}(x)$ de L_2 -Boosting est un estimateur de :

$$m^*(x) = \mathbb{E}(Y | X = x) .$$

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

L_2 -Boosting II

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

- ▶ Dans l'étape 2.1 du gradient boosting, on obtient :

$$\begin{aligned} U_i &= -\frac{\partial}{\partial m(x_i)} \ell(m(x_i), y_i) \Big|_{m(x_i)=\hat{m}_{b-1}(x_i)} \\ &= y_i - \hat{m}_{b-1}(x_i) \end{aligned}$$

- ▶ Les U_i sont donc les résidus du prédicteur \hat{m}_{b-1} à l'étape $(b-1)$.
- ▶ Le prédicteur \hat{m}_b est élaboré via une régression sur les résidus de l'étape $(b-1)$.
 \hat{m}_{b-1} est amendée en expliquant l'information subsistant dans les résidus de l'étape $(b-1)$.

Plan

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

CART

Méthodes
d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte
classiques

XGBoost

Références

- ▶ XGBoost (eXtreme Gradient Boosting) est une implémentation optimisée de l'algorithme de gradient boosting : ([Chen et Guestrin, 2016](#)).
- ▶ Les principes de XGBoost sont :
 - ▶ Ajout de termes de **régularisation** pour contrôler la complexité du modèle et réduire le sur-apprentissage.
 - ▶ Gestion des **données manquantes**.
 - ▶ Optimisation de la **mémoire** pour accélérer les calculs.
 - ▶ **Parallélisation** des calculs.
 - ▶ **Partition des données** en blocs pour mieux gérer les grands ensembles de données.
- ▶ D'autres algorithmes sont apparus à la suite tels LightGBM et CatBoost.

CART

Méthodes d'agrégation

Bagging

Boosting

Introduction

AdaBoost

Gradient boosting

Fonctions de perte classiques

XGBoost

Références

Références |

- Boyd, S. et L. Vandenberghe. 2003, *Convex optimization*, Cambridge University Press.
- Breiman, L. 1996, «Bagging predictors», *Machine Learning*, vol. 24, p. 123–140.
- Breiman, L. 2001, «Random forests», *Machine Learning*, vol. 45, p. 5–32.
- Breiman, L., J. H. Friedman, C. J. Stone et R. A. Olshen. 1984, *Classification and regression trees*, Taylor & Francis.
- Chen, T. et C. Guestrin. 2016, «Xgboost : a scalable tree boosting system», dans *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, p. 785–794.
- Efron, B. 1979, «Bootstrap methods : another look at the jackknife», *The Annals of Statistics*, vol. 7, n° 1, p. 1–26.

CART

Méthodes d'agrégation

Bagging

Boosting

Références

Références II

- Efron, B. et R. Tibshirani. 1994, *An introduction to the bootstrap*, Chapman & Hall.
- Freund, Y. et R. E. Schapire. 1997, «A decision-theoretic generalization of on-line learning and an application to boosting», *Journal of Computer and System Sciences*, vol. 55, n° 1, p. 119–139.
- Hastie, T., R. Tibshirani et J. H. Friedman. 2009, *The elements of statistical learning. Data Mining, inference, and prediction*, 2^e éd., Springer Series in Statistics, Springer.
- James, G., D. Witten, T. Hastie et R. Tibshirani. 2021, *An introduction to statistical learning with applications in R*, 2^e éd., Springer Texts in Statistics, Springer.
- James, G., D. Witten, T. Hastie, R. Tibshirani et J. Taylor. 2023, *An introduction to statistical learning with applications in Python*, Springer Texts in Statistics, Springer.

CART

Méthodes d'agrégation

Bagging

Boosting

Références

Références III

CART

Méthodes
d'agrégation

Bagging

Boosting

Références

Schapire, R. E. et Y. Freund. 2012, *Boosting. Foundations and algorithms*, Adaptive Computation and Machine Learning, MIT Press.