Partie 2 - Apprentissage supervisé (régression et classification)

Anne-Sophie Charest

École interdisciplinaire outils et méthodes
Cheminement 2 - Science des données

Apprentissage statistique (*machine learning*)

28-30 août 2024

Plan

Concepts de base

L'apprentissage supervisé

Il s'agit en fait de faire la prévision d'une ou de plusieurs variables à l'aides d'autres variables.

On dit que l'apprentissage est supervisé quand on travaille à faire cette prédiction à partir d'un jeu de données pour lequel on connait toutes les variables : celle à prédire (variable réponse) et celles à utiliser pour le faire.

Termes habituellement utilisés en apprentissage statistique :

Régression - si la variable réponse est continue.

Classification - si la variable réponse est catégorique.
(même si la régression logistique convient tout à fait ici !)

Les grandes questions

- Comment obtenir un modèle (algorithme) qui permet de bien prédire la variable d'intérêt ?
  Défis : ne pas se limiter à des relations linéaires, inclure des interactions entre des variables, avoir un modèle qui s'interprète bien, etc.

Les grandes questions

- Comment obtenir un modèle (algorithme) qui permet de bien prédire la variable d'intérêt ?
  Défis : ne pas se limiter à des relations linéaires, inclure des interactions entre des variables, avoir un modèle qui s'interprète bien, etc.

- Comment savoir quel est le meilleur modèle à utiliser pour faire des prévisions ?
  Et ce pour des nouvelles données, pas simplement celles utilisées pour faire le modèle !

Un peu de formalisme

Considérons une variable réponse numérique.

Soit Y la variable que l'on souhaite prédire, et X un ensemble de prédicteurs.

On fait l'hypothèse très générale que

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

où $E(\varepsilon) = 0$ et $Var(\varepsilon) = \sigma^2$, une constante.

On dénote par $\hat{f}(x)$ notre prévision pour la valeur de $f(x)$.

Erreur de prévision

On peut prouver que si $Y = f(X) + \varepsilon$ où $E(\varepsilon) = 0$ et $Var(\varepsilon) = \sigma^2$, une constante, l'erreur quadratique attendue pour la prévision en x_0 s'écrit :

$$E(Y_0 - \hat{f}(x_0)^2) = \underbrace{[Biais(\hat{f}(x_0))]^2 + Var(\hat{f}(x_0))}_{\text{Erreur réductible}} + \underbrace{\sigma^2}_{\text{Erreur irréductible}}$$

L'erreur réductible peut être réduite par le choix de $\hat{f}(x_0)$ et en augmentant le nombre d'observations pour l'estimation.

L'erreur irréductible sera présente même si $\hat{f} = f$; elle est dû à l'aspect aléatoire de Y.

Compromis biais-variance

$$E(Y_0 - \hat{f}(x_0)^2) = \underbrace{[Biais(\hat{f}(x_0))]^2 + Var(\hat{f}(x_0))}_{\text{Erreur réductible}} + \underbrace{\sigma^2}_{\text{Erreur irréductible}}$$

L'erreur réductible dépend du biais et de la variance de $\hat{f}(x_0)$ en tant qu'estimateur de $f(x_0)$.

Dans le choix du $\hat{f}$, il y a généralement un compromis entre le biais et la variance de cet estimateur. C'est-à-dire qu'un estimateur avec un petit biais tend à avoir une grande variance et vice-versa.

L'idée est de choisir l'estimateur approprié pour balancer le biais et la variance de façon à ce que l'EQM (erreur quadratique moyenne) soit minimale.

Mesurer l'erreur de prévision

On peut évaluer la qualité de notre modèle de prévision en regardant s'il prédit bien pour les observations utilisées pour l'estimer, en utilisant

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad \text{ou} \quad \frac{1}{n}\sum_{i=1}^{n} I[h(x_i) \neq y_i].$$

Ici, $\hat{y}_i$ dénote la valeur prédite pour la variable continue y pour l'observation i, et $h(x_i)$ dénote la valeur prédite pour une variable catégorique y pour l'observation i.

Mais, on a alors de fortes chances de sous-estimer les vraies erreurs, car $\hat{f}$ et h ont été estimés à l'aide des données, et s'adaptent donc à celles-ci.

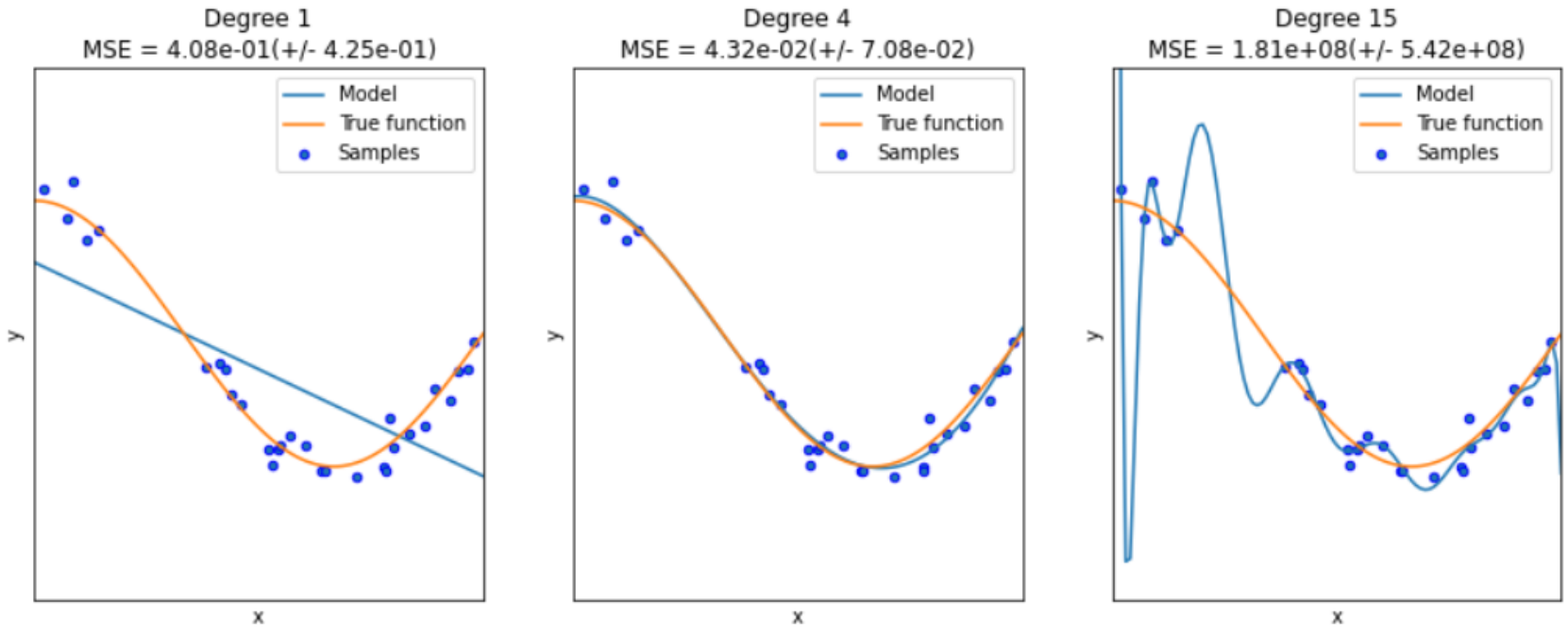
Sur-ajustement (*overfitting*)

Un modèle/classifieur plus complexe/flexible pourra mieux s'ajuster aux données observées, et mènera à une plus petite EQM, si calculée sur le jeu de données originales.

Mais si le modèle/classifieur s'adapte trop aux données observées, il risque de ne pas être aussi bon si on l'applique sur de nouvelles données.

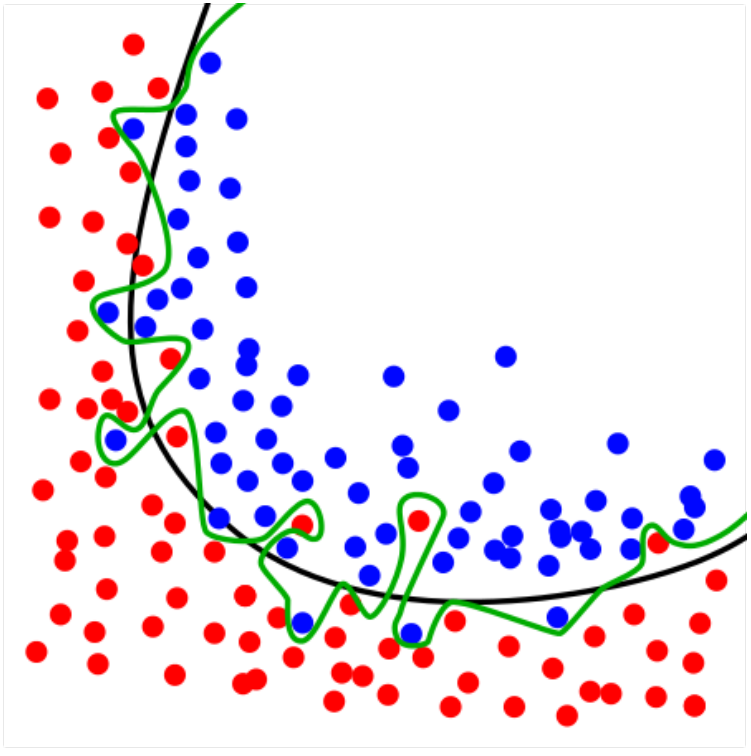
Il y a sur-ajustement si on choisit à l'aide du jeu de données d'entrainement un modèle/classifieur trop complexe/flexible, qui s'ajuste à la partie aléatoire des données de sorte que l'EQM sur un jeu de données de validation est plus grande qu'elle ne l'aurait été avec un modèle/classifieur moins complexe/flexible.

Illustration - régression



Source : `https://datascience.foundation/sciencewhitepaper/underfitting-and-overfitting-in-machine-learning`

Illustration - classification



Source : Wikipedia, overfitting

Solutions au sur-ajustement

- Utiliser un jeu de données de validation
- Utiliser la validation croisée

Utiliser un jeu de données de validation

On divise le jeu de données initial en deux parties :

1. Jeu d'apprentissage
   On estime $\hat{f}$ ou h à l'aide de ce jeu de données.
2. Jeu de validation
   On estime l'erreur quadratique moyenne espérée ou le risque du classifieur sur ce jeu de données.

Utiliser un jeu de données de validation

On divise le jeu de données initial en deux parties :

1. Jeu d'apprentissage
   On estime $\hat{f}$ ou h à l'aide de ce jeu de données.
2. Jeu de validation
   On estime l'erreur quadratique moyenne espérée ou le risque du classifieur sur ce jeu de données.

Attention :

- La qualité de l'estimation de l'erreur dépendra de la taille du jeu de validation.
- La qualité de $\hat{f}$ ou h dépendra de la taille du jeu d'apprentissage.
- Il faut faire un compromis entre ces deux aspects quand on divise le jeu de données original. Requiert en général un grand jeu de données original.

Validation croisée

Extension de l'idée du jeu de validation, mais chacune des observations sera tour à tour dans le jeu d'apprentissage et dans le jeu de validation.

1. Diviser les données en k groupes.
2. Ajuster un modèle sur $k-1$ des groupes.
   L'utiliser pour prédire les valeurs dans le dernier échantillon.
   Calculer l'erreur quadratique moyenne ou le risque de classification.
3. Répéter l'étape 2 k fois en utilisant un groupe différent comme jeu de validation à chaque fois.
4. Calculer la moyenne des k erreurs obtenues.

Choix du nombre de groupes k (1/2)

Si k est grand (disons $k = n$) :

- L'estimateur de l'erreur de prévision espérée est approximativement sans biais pour la vraie erreur de prévision espérée.
- Mais, sa variance peut être grande car les $k = n$ jeux d'apprentissage sont très corrélés.

  Note : il faut penser à la variabilité en terme de différents jeux de données observés, pas simplement en terme de répétition de la validation croisée.

Choix du nombre de groupes k (2/2)

Si k est petit (disons $k = 5$) :

- La variance de l'estimateur de l'erreur de prévision espérée est diminuée car les jeux d'apprentissage sont moins corrélés.
- L'estimateur sera toutefois biaisé, si la performance de notre modèle/classifieur varie beaucoup avec la taille d'échantillon, car on apprend sur un jeu de données de taille inférieure à n.

En pratique :
On choisit en général $k = 5$ ou $k = 10$.
Le choix de k peut dépendre du temps de calcul requis.

Méthode des plus proches voisins

Général

On utilise les observations les plus proches de celle d'intérêt, en termes de variables explicatives, pour prédire la variable réponse.

Peut être utilisé pour la classification ou la régression.

Méthode très simple, mais qui donne parfois de bons résultats.

Régression par la méthode des k plus proches voisins :

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_k} y_i$$

où N_k est l'ensemble des k plus proches voisins de x_0.

Compromis biais-vairance

Le choix de k aura un impact important sur le biais et la variance de l'estimateur.

Si k augmente :

- $\hat{f}(x_0)$ dépend de données plus loin de x_0, donc le biais augmente.
- $\hat{f}(x_0)$ est la moyenne d'un plus grand nombre d'observations, donc la variance diminue.

Classification par la méthode des k plus proches voisins :

$$h(x_0) = \text{argmax}\, y \sum_{x_i \in N_k} (y_i = y)$$

où N_k est l'ensemble des k plus proches voisins de x_0.

C'est à dire qu'on prédit la valeur de y la plus fréquente parmi les voisins de x_0. En cas d'égalité, on peut tirer au hasard parmi les deux valeurs les plus fréquentes.

Note

Cette méthode requiert de calculer les distances entre les observations du jeu de données. La distance euclidienne est souvent utilisée, mais d'autres distances peuvent être plus appropriées dans certains cas.

Classification : fonction `knn` dans la librairie `class`.

Régresssion : fonction `knn.reg` dans la librairie `FNN`.

Vous trouverez facilement en ligne des tutoriels sur comment les utiliser.

Arbres de régression / classification

Origine

CART = Classification And Regression Trees

Livre source :
Breiman, Leo ; Friedman, J. H., Olshen, R. A., Stone, C. J. (1984). Classification and regression trees. Monterey, CA : Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8.

Classification : variable dépendante catégorique
Régression : cariable dépendante continue

Variables indépendantes catégoriques ou continues.

Produire l'arbre

1. Divisions binaires en fonction de la valeur d'une variable.

   Choix des variables et des limites pour la division en fonction d'un certain critère (ex. index de Gini)

2. Chaque feuille (noeud terminal) correspond à une région à l'intérieur de laquelle on prédit la même catégorie.

   Prévision dans une feuille :
   Classification : vote à majorité (c.-à-d. prédire la classe la plus fréquente)
   Régression : moyenne des observations de la région

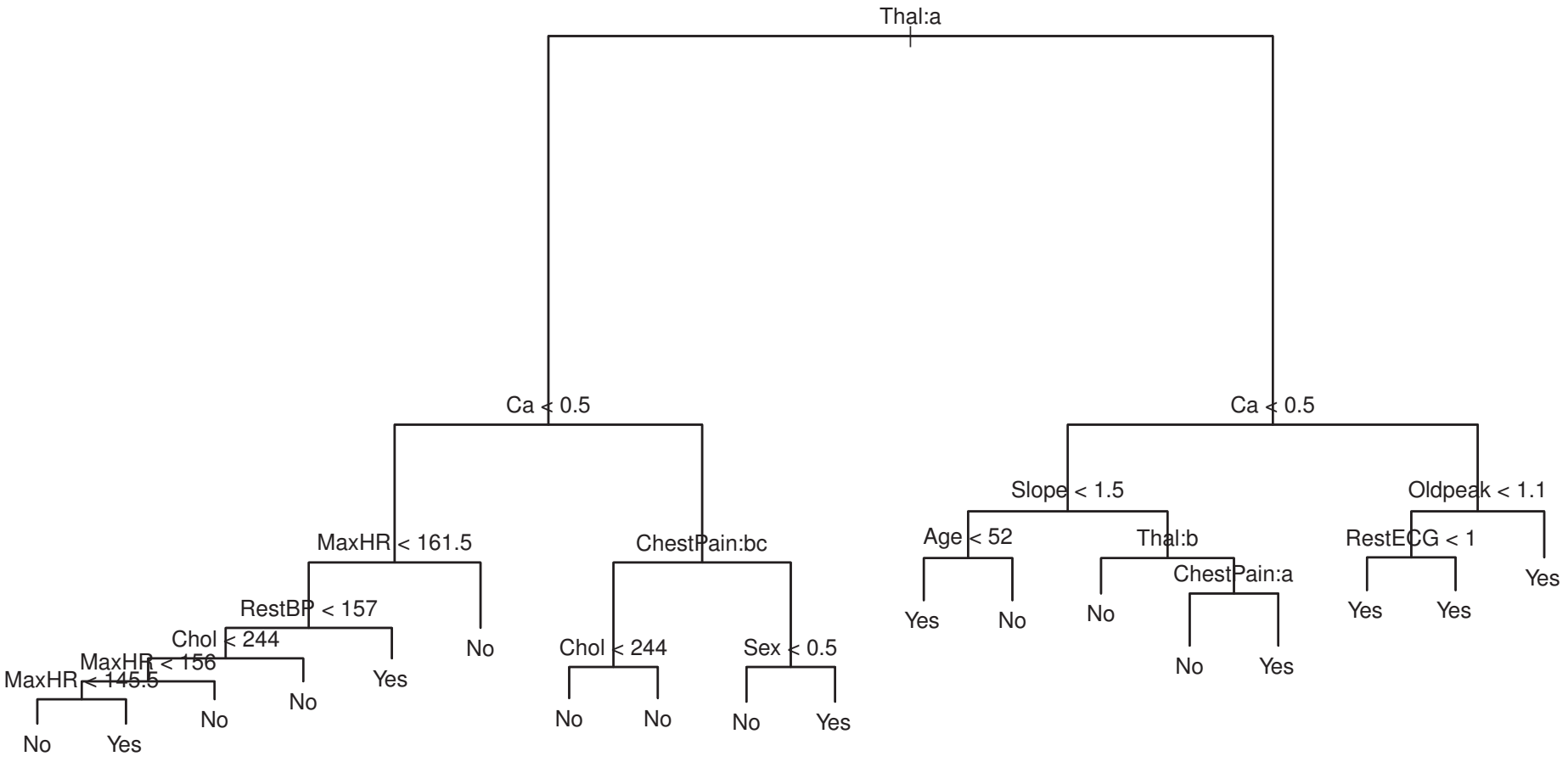
Exemple - données

303 patients atteints de douleur à la poitrine

Y : Indicateur d'un problème cardiaque (oui ou non)

13 variables indépendantes : âge, sexe,cholestérol, ...
(continues et catégoriques)

Exemple - arbre obtenu



L'algorithme

C'est un algorithme glouton, car on choisit la meilleure division à chaque niveau. On n'est pas certain d'obtenir l'arbre optimal.

À chaque étape, on utilise un critère pour choisir sur quelle variable diviser, et quelle valeur utiliser pour la division.

En R, disponible dans la librairie `rpart` et la librairie `tree`, entre autres.

Critère de coupure - Régression

Pour une variable X_j donnée et un point de coupure s donné, on obtiendra les deux régions suivantes :

$$R_1(j,s) = \{X | X_j < s\} \quad \text{et} \quad R_2(j,s) = \{X | X_j \geq s\}$$

Si X_j est une variable catégorique, on forme les régions en divisant en deux groupes les différents niveaux de la variable.

On choisira j et s de façon à minimiser la somme des carrés des erreurs

$$\sum_{i:x_i \in \mathbb{R}_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in \mathbb{R}_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

où $\hat{y}_{R_k} = \sum_{i:x_i \mathbb{R}_k(j,s)} y_i$ pour $k = 1, 2$.

Critères de coupure - Classification

Soit $\hat{p}_{mk}$ la proportion d'observations dans la région m qui font partie de la classe k.

On choisit habituellement les divisions pour minimiser un des trois critères suivants :

1. Taux d'erreur de classification :

$$E_m = 1 - \max_k(\hat{p}_{mk})$$

2. Index de Gini :

$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

3. Entropie croisée :

$$D_m = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk})$$

Nombre de divisions

Quand arrêter de faire des divisions ?
Habituellement basé sur le critère utilisé pour choisir les divisions. On peut aussi limiter la taille des feuilles (ex. au moins 5 observations par feuille).

Importance du nombre de divisions :

Pas assez de divisions : notre modèle n'est pas assez flexible ; le biais sera donc important.

Trop de divisions : chaque prévision ne dépend que de peu d'observations ; la variance sera donc importante.

Élagage

Soit T_0 l'arbre obtenu par l'algorithme précédent.
On l'élague en enlevant des branches (divisions) dans le but d'obtenir un meilleur sous-arbre.

On utilise souvent l'élagage coût-complexité (*Cost-complexity pruning*). Étant donné un paramètre α, on choisit le sous-arbre $T \subset T_0$ tel que le critère suivant est minimal :

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

où $|T|$ dénote le nombre de feuilles dans l'arbre T.

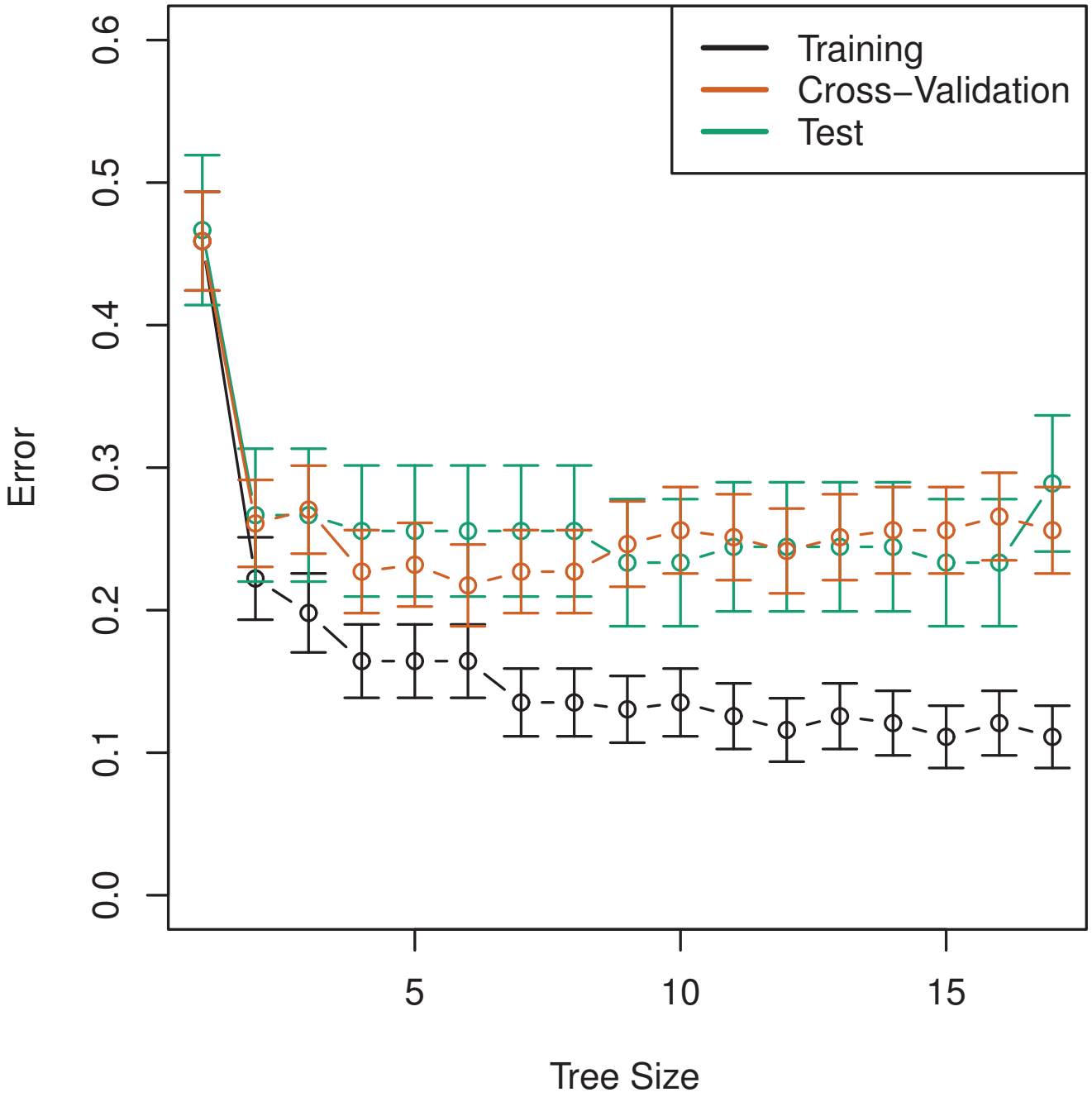
Élagage - choix de α

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

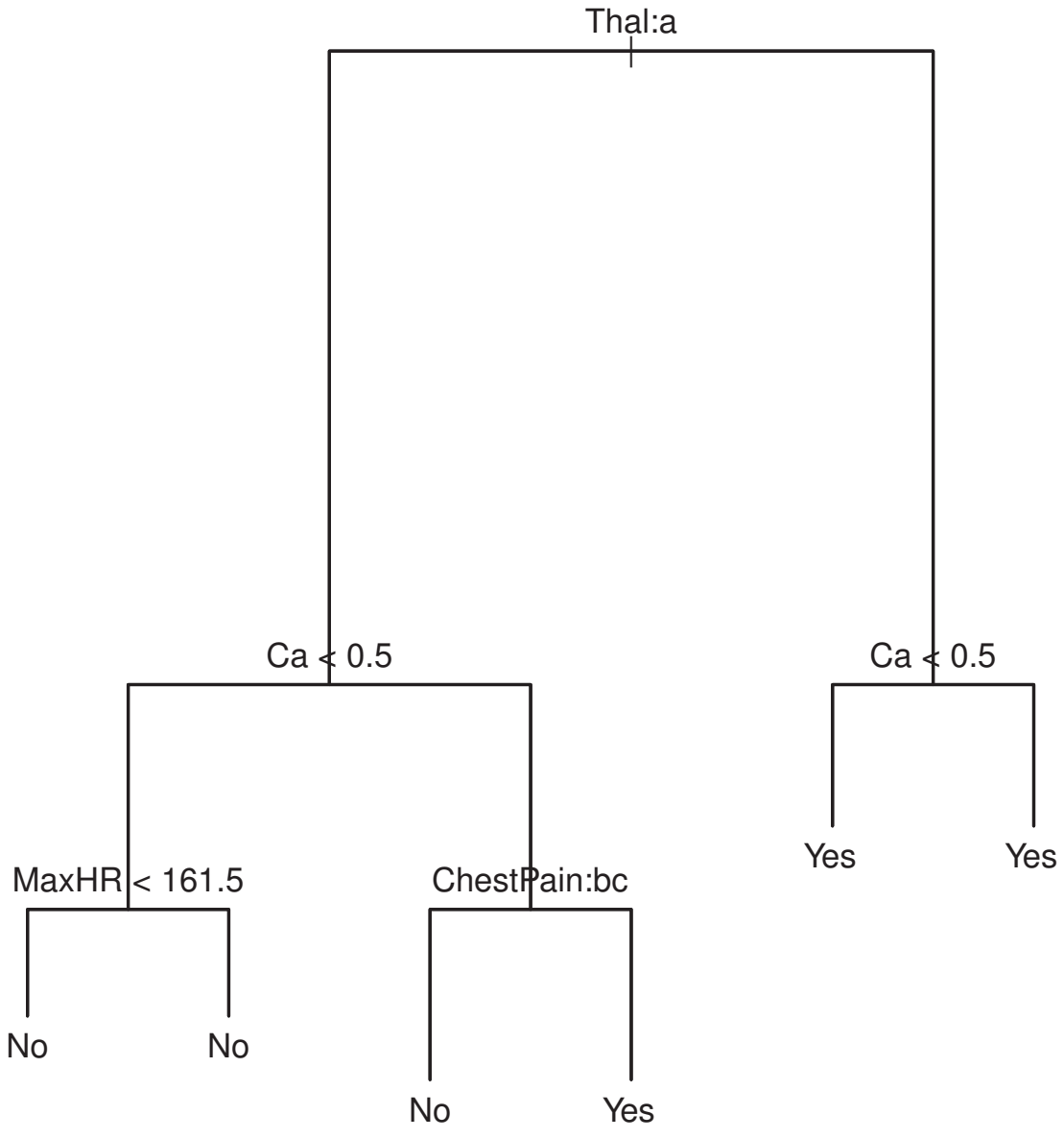
Impact de α :
Si $\alpha = 0$, on ne limite pas la taille de l'arbre.
Plus α augmente plus on favorise les petits arbres.

Choix de α : Généralement pas validation croisée.

Illustration (avec l'exemple de tout à l'heure)



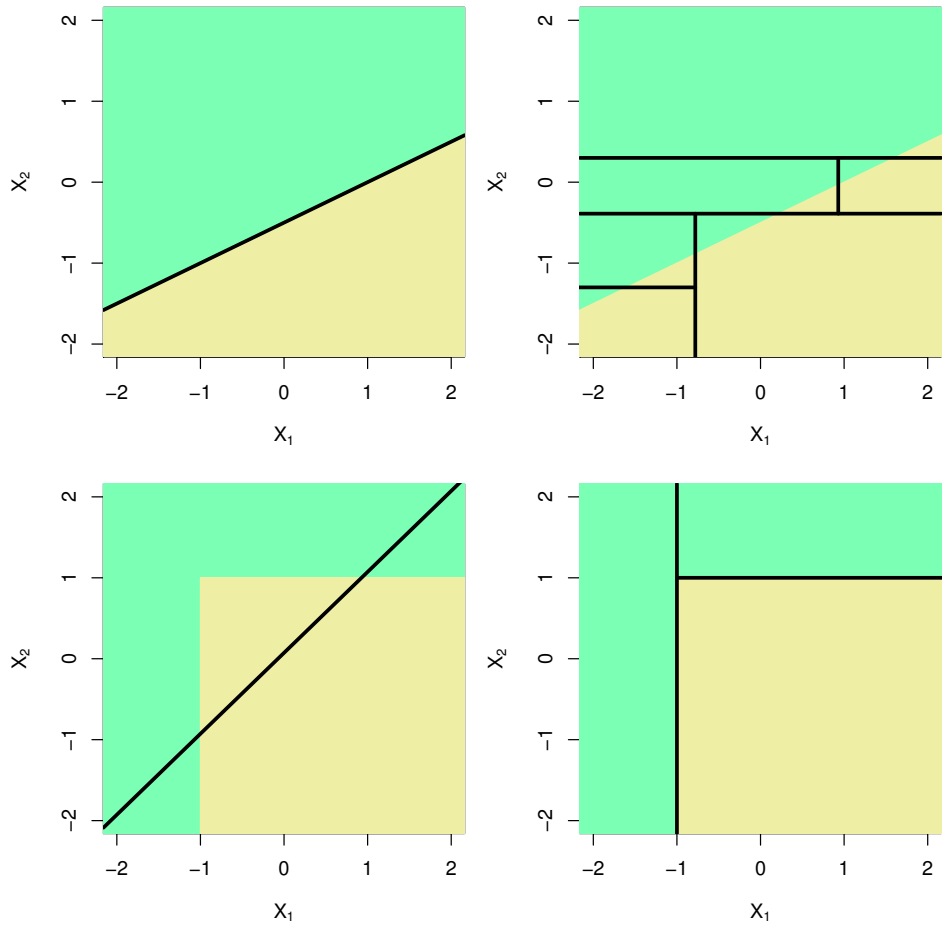
Exemple - arbre optimal post élagage



Avantages des arbres de classification

- Méthode non paramétrique :
  aucune hypothèse *a priori* sur la distribution des données
- Résistante aux données atypiques
- Facile à interpréter
- Modélise indirectement des interactions
- Sélection de variable implicite
- Modèle non linéaire

Non-linéarité du modèle



Désavantages des arbres de classification

- Besoin d'un grand nombre de données.
- Peuvent être instable.

Améliorations possibles :

- Bagging
- Forêts aléatoires
- Boosting

Autres méthodes

Très variées !

Il existe une multitude méthodes pour l'apprentissage supervisé, que ce soit pour la régression ou la classification.

L'important dans tous les cas, c'est de s'assurer de choisir un modèle approprié en faisant attention au sur-ajustement.

Stratégie souvent utilisée :

1. Choisir les hyperparamètres d'un modèle par validation croisée (ex. nombre de variables à conserver, valeur de k pour k-nn, etc.).
2. Choisir entre les différents modèles à l'aide d'un jeu de données de validation.

Méthodes plus statistiques

- Régression linéaire, logistique, Poisson, multinomiale, polynomiale, etc.
- Régression pénalisée (régularisée), LASSSO
- Régression non-paramétrique
- Régression linéaire locale
- Régression avec splines
- *Generalized additive models*
- Etc.

Méthodes plus ML

- Classifieur bayesien naif
- Forêts aléatoires
- Machine à vecteur de supports (SVM)
- Réseaux de neurones
- Apprentissage profond (CNN, RNN, GAN, etc.)
- Etc.

Source

Certaines figures de cette présentation viennent du livre *An Introduction to Statistical Learning with Applications in R.*

”Some of the figures in this presentation are taken from ”An Introduction to Statistical Learning, with applications in R” (Springer, 2013) with permission from the authors : G. James, D. Witten, T. Hastie and R. Tibshirani ”