



Introduction à l'Apprentissage Statistique

Arbres de décision

M2 Actuariat – ISFA – 2021/2022

Pierrick Piette
Actuaire à Seyna
pierrick.piette@gmail.com

Seyna.

In God we trust, all others bring data
- William Edwards Deming





Introduction

Première brique d'IA

7 principes éthiques de la Commission Européenne pour l'IA

- Contrôle/supervision humaine : l'IA n'a pas de conscience !
- Résistance et sécurité des algorithmes : fiabilité pour gérer les erreurs et incohérences
- Gestion des données, protection de la vie privée : utilisateurs en mesure de contrôler leurs propres données
- Transparence algorithmique : expliquer ce que fait l'IA, traçabilité
- Diversité, non-discrimination et équité
- Bien-être social et environnemental : l'IA doit être mise au service de la société dans son ensemble
- L'Accountability : principe de responsabilité, mise en place de procédure internes à l'entreprise pour démontrer le respect des règles relatives à la protection des données

Objectif de l'arbre : classifier des individus

Regrouper des individus hétérogènes en classes homogènes de risque pour résumer l'info d'une grande base de donnée

Classification non supervisée

- algorithme des k -plus proches voisins
- Classification ascendante hiérarchique
- model-based clustering

Classification supervisée

- modèle paramétrique linéaire : Logit, Tobit, etc.
- Réseaux de neurones, SVM
- Arbres descendants (CART, CHAID)

Premiers éléments de l'arbre

De quoi est composé un arbre ?

Une racine

- contient l'ensemble de la population (portefeuille global)

Un tronc et des branches

- contiennent les règles de division qui permettent de segmenter la population

Des feuilles

- contiennent les sous-populations homogènes (sur leurs caractéristiques et la réponses)
- fournissent l'estimation de la quantité d'intérêt

Règles et lecture d'un arbre CART

Un arbre de classification / régression se lit de la racine vers les feuilles
(l'inverse d'une CAH)

A chaque ramification, un règle de division apparaît

- cette règle admet une réponse binaire (oui/non) ,
- elle n'est basée que sur un facteur de risque

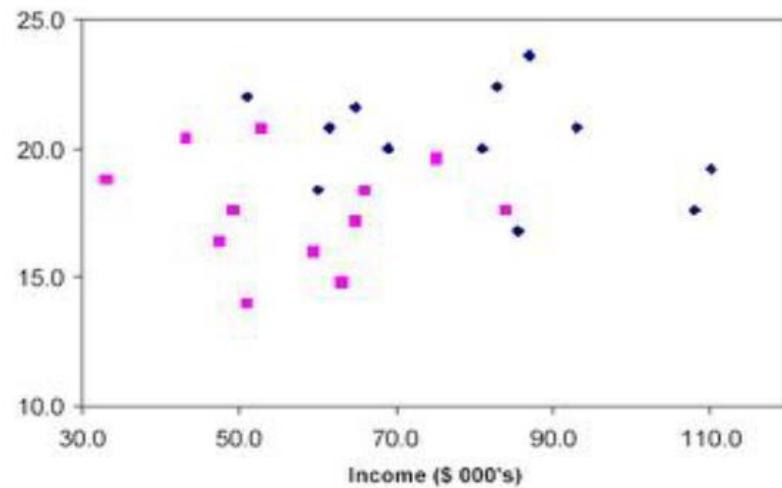
Un nœud est l'intersection d'un ensemble de règles. L'estimation de la quantité d'intérêt se lit dans les nœuds terminaux (feuilles)

N'importe quel individu de la population initiale appartient à une unique feuille

Exemple 1 : arbre de classification

A travers cet exemple, on veut intuiter comment un arbre se construit...
Cherchons à prévoir *propriétaire ~ salaire + surface*

Income (\$ 000's)	Lot Size (000's sq. ft.)	Owners=1, Non-owners=2
60	18.4	
85.5	16.8	
64.8	21.6	
61.5	20.8	
87	23.6	
110.1	19.2	
108	17.6	
82.8	22.4	
69	20	
93	20.8	
51	22	
81	20	
75	19.6	
52.8	20.8	
64.8	17.2	
43.2	20.4	
84	17.6	
49.2	17.6	
59.4	16	
66	18.4	
47.4	16.4	
33	18.8	
51	14	
63	14.8	



Choisir la segmentation de l'espace (1/2)

Choisir une variable explicative j donnée à m valeurs :

- numérique ou catégorielle ordonnée : partitionnement entre 2 valeurs successives → $m - 1$ possibilités
- catégorielle non ordonnée : partitionnement des combinaisons de modalité → $2^m - 1$ possibilité

On teste tous ces partitionnements

- estimation d'un critère d'homogénéité par rapport à ma quantité d'intérêt

On choisit le partitionnement qui conduit à la plus grande homogénéité dans les sous-espaces créés.

Choisir la segmentation de l'espace (2/2)

On répète les trois premières étapes pour chacune des covariables

- liste de k homogénéité maximales

On choisit la covariable et son partitionnement qui maximise l'homogénéité globalement

On répète le processus jusqu'à l'obtention de l'arbre maximal

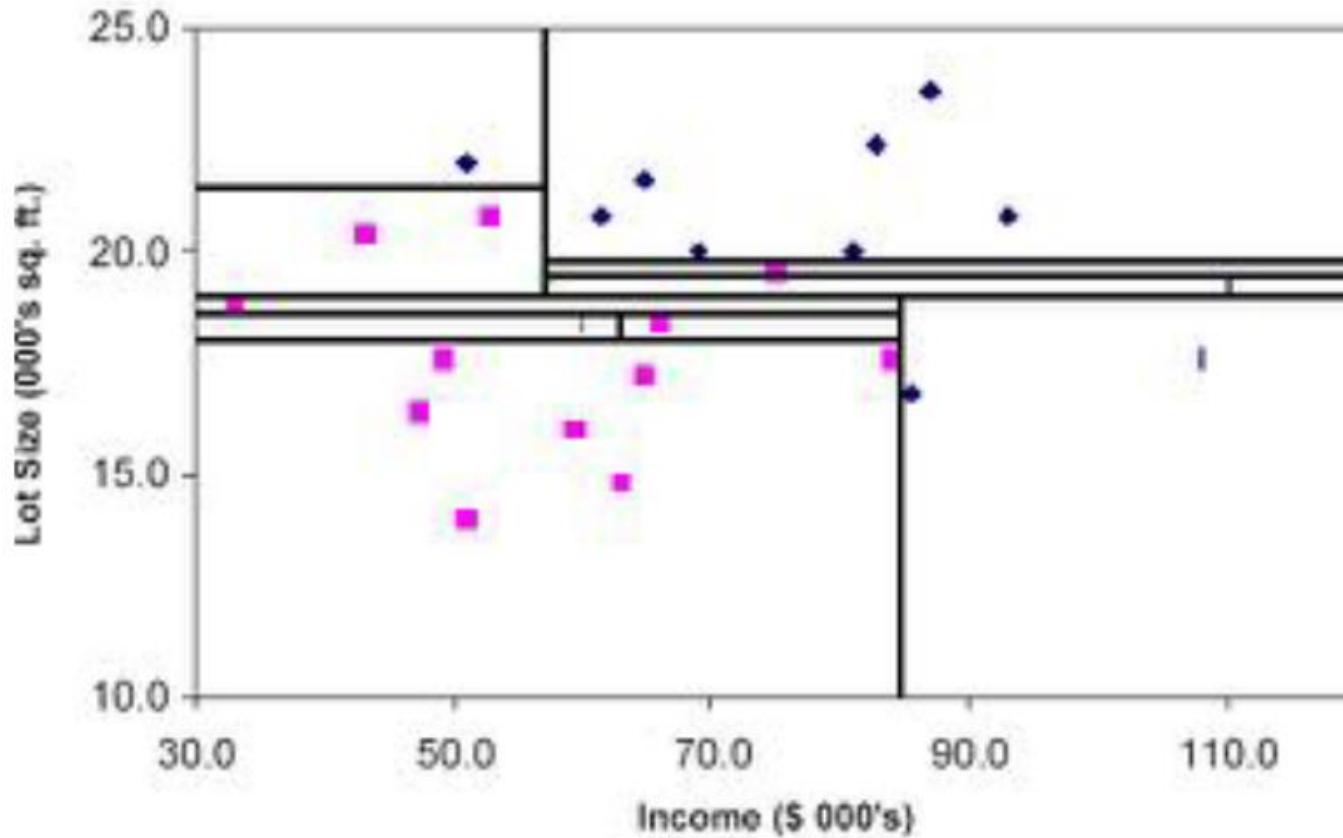
Exemple: Tenter un arbre non maximal

Risque de surapprendre si: arbre trop profond
Car arbre deviendrait le base de dommages.

$$y = f(x) + \text{bruit}$$

Si on surapprend
on apprend aussi le bruit
mauvaise chose.

Partitionnement de l'espace





Notations

Notations

$i \in [1, n]$: identifiant de l'individu

$j \in [1, k]$: identifiant du facteur de risque

Y_i : réponse observée du i ème individu

$\mathbf{X}_i = (X_{i1}, \dots, X_{ik})$: vecteur des facteurs de risque de l'individu i

\mathcal{X} : espace des covariables

$l \in [1, L]$: identifiant des feuilles de l'arbre

\mathcal{X}_l : ensemble de la partition correspondant à la feuille l

Arbre de régression avec Y continue

En régression, la quantité d'intérêt est

$$\pi_0(\mathbf{x}) = \mathbb{E}_0[Y \mid \mathbf{X} = \mathbf{x}]$$

En supposant une relation linéaire, on a

$$\hat{\pi}(\mathbf{x}) = \hat{\beta}_0 + \mathbf{x}^T \hat{\beta}$$

et on estime les paramètres de régression par OLS

Pour les arbres, la classe de l'estimation $\hat{\pi}$ sont **les fonctions constantes par morceaux.**

Construire un arbre maximal génère une suite d'estimateurs selon une procédure spécifique : **divisions successives de l'espace \mathcal{X}**

Critère de division

La ramifications de l'arbre est basée sur la définition d'un critère d'homogénéité, cohérent avec l'estimation de la quantité d'intérêt

En régression, la solution OLS est donnée par

$$\pi_0(\mathbf{x}) = \arg \min \mathbb{E}_0[\phi(Y, \pi(\mathbf{x})) \mid X = \mathbf{x}]$$

où $\phi(Y, \pi(\mathbf{x})) = (Y - \pi(\mathbf{x}))^2$

La fonction de perte ϕ correspond à **l'erreur quadratique**, et le critère est la minimisation de la **MSE (mean squared error)**

La différence est que l'on va estimer $\pi_0(\mathbf{x})$ en **plusieurs étapes** !

Etapes de construction

Enchainement des étapes de construction de l'arbre maximal

1. On part de la racine
2. On cherche la meilleure première segmentation
3. On segmente
4. On itère sur chacun des 2 nœuds fils
5. On itère sur les fils des nœuds fils
6. Etc.

Par construction l'hétérogénéité diminue à chaque segmentation, pour atteindre sa valeur minimale sur l'arbre maximal

Lien entre régression et arbre

Un arbre est un ensemble de règles. Pour chaque nœud m une règle R_m est associée à un sous-ensemble $\mathcal{X}_m \subseteq \mathcal{X}$

Notation : dans la suite $\mathbb{E}_n[Y]$ désigne la moyenne empirique de Y et $\mathcal{X}_{pa(m)}$ est le sous-ensemble associé au nœud parent de m

L'arbre est associé à la fonction de régression

$$\hat{\pi}(\mathbf{x}) = \sum_{m=1}^M \hat{\beta}_m R_m(\mathbf{x})$$

où $\hat{\beta}_m = \mathbb{E}_n[Y \mid \mathbf{x} \in \mathcal{X}_m] - \mathbb{E}_n[Y \mid \mathbf{x} \in \mathcal{X}_{pa(m)}]$ si m n'est pas la racine
et $\hat{\beta}_m = \mathbb{E}_n[Y]$ sinon

Lien entre régression et arbre

Cela équivaut en régression classique à chercher

$$\hat{\beta} = \arg \min_{\beta} \mathbb{E}_n \left[\left(Y - \sum \beta_m R_m(\mathbf{x}) \right)^2 \right]$$

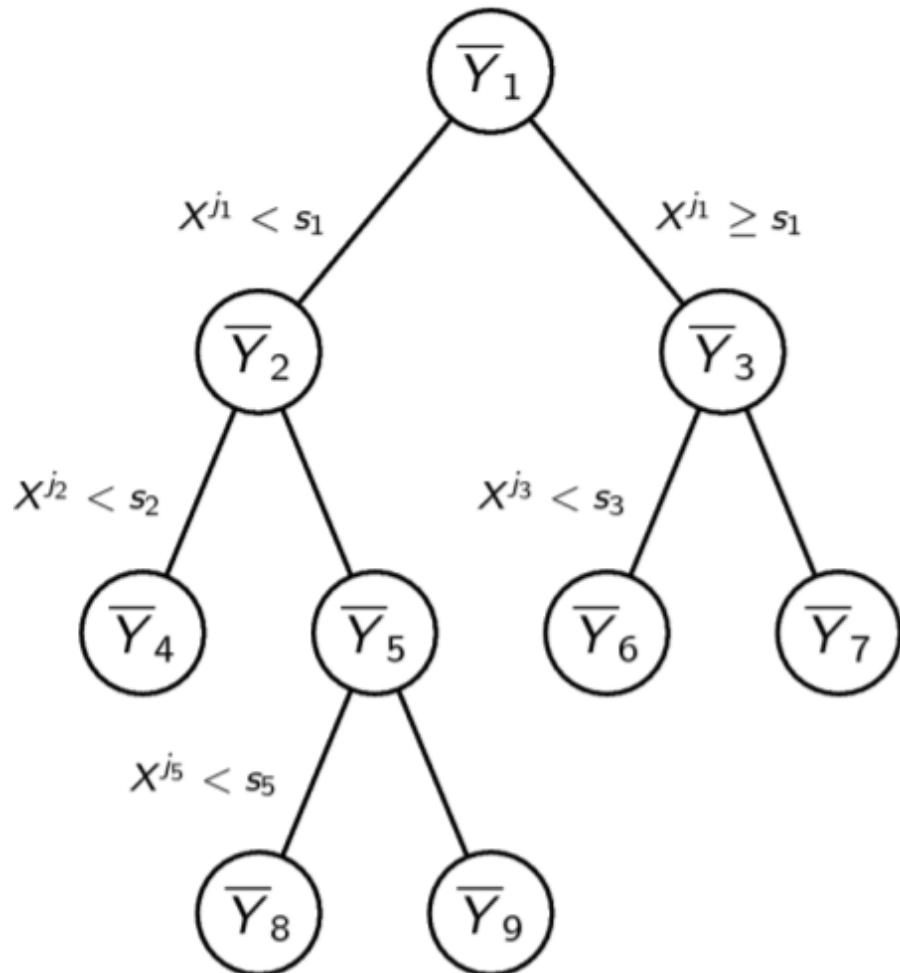
En sommant sur tous les nœuds, il reste les feuilles

$$\hat{\pi}(\mathbf{x}) := \hat{\pi}^L(\mathbf{x}) = \sum_{l=1}^L \hat{\gamma}_l R_l(\mathbf{x})$$

Décomposition en base fonctionnelle de \mathbf{x} avec :

- $R_l = \mathbb{I}(\mathbf{x} \in \mathcal{X}_l)$: règle d'appartenance au sous-ensemble \mathcal{X}_l
- $\hat{\gamma}_l = \mathbb{E}_n[Y \mid \mathbf{x} \in \mathcal{X}_l]$: moyenne empirique de Y dans la feuille l
- \mathcal{X}_l forme une partition : disjoints et exhaustif

CART



Généralisation

Tout arbre peut être vu comme un estimateur par morceaux, quelque soit la quantité d'intérêt.

Interprétation

- chaque morceau est une feuille, dont la valeur est la moyenne empirique des valeurs de Y de cette feuille (dans le cas quantitatif)
- chaque division d'un nœud m minimise la somme des variances intra-nœuds résultantes
- ce qui permet de maximiser la décroissance de l'hétérogénéité

Pas besoin des fonct^o Lipschitzennes car but d'apprendre en gros pas de sensibilité



Récursivité

La construction étant récursive, on génère une suite d'estimateur depuis le nœud racine : soit une suite $\{\Pi^K\}$ de sous-espaces

$$\Pi^K = \left\{ \pi^L(\cdot) = \sum_{l=1}^L \gamma_l R_l(\cdot) \mid L \in \mathbb{N}^*, L \leq K \right\}$$

Pour K fixé, on cherche donc

$$\pi_0(x) = \arg \min \{ \mathbb{E}_0[\phi(Y, \pi(x)) \mid X = x] \mid \pi(x) \in \Pi^K \}$$

L'algorithme CART ne cherche pas tous les estimateurs possibles avec $L \leq K$, il approche ce minimum petit à petit.

Arrêt de la procédure

L'algorithme CART ne fixe pas de règle d'arrêt arbitraire pour la procédure de division de l'espace

L'algorithme arrête ainsi de diviser les feuilles quand

- il n'y a qu'une observation dans la feuille
- ou quand les individus de la feuille ont les mêmes valeurs de facteurs de risque

On construit ainsi l'arbre maximal, qui sera ensuite élagué.

- c'est l'estimateur par morceaux le plus complexe de la suite d'estimateurs construits

Exemple 2 : Modélisation de la mortalité

Objectifs : prévision de décès et modélisation des taux de mortalité. Résultats issus de l'article de Olbricht (2012), publié dans l'EAJ.

Portefeuille de SwissRe avec les caractéristiques suivantes

- comprenant 1 463 964 enregistrements
- couvrant une période de 4 ans
- variables explicatives : sexe et âge

Les résultats obtenus par CART sont comparés à la table de mortalité en vigueur à l'époque « German standard life table DAV 2008 T »

Arbre (élagué)

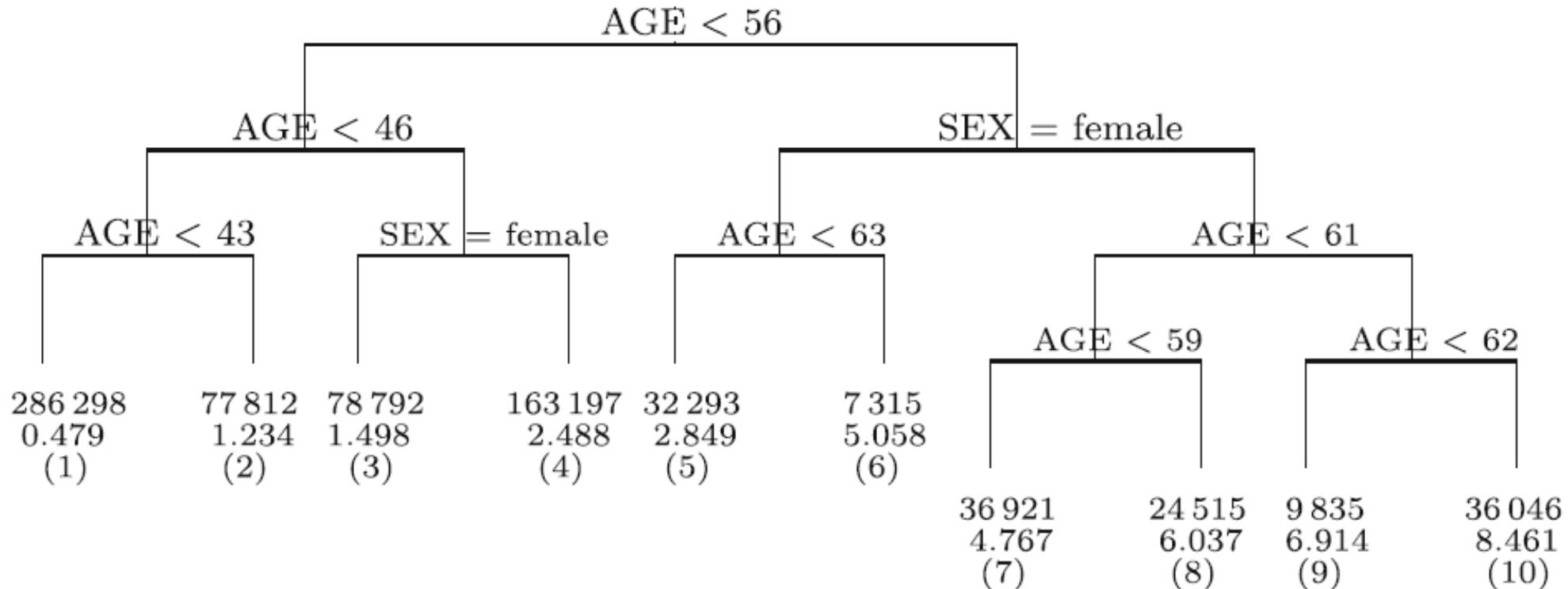
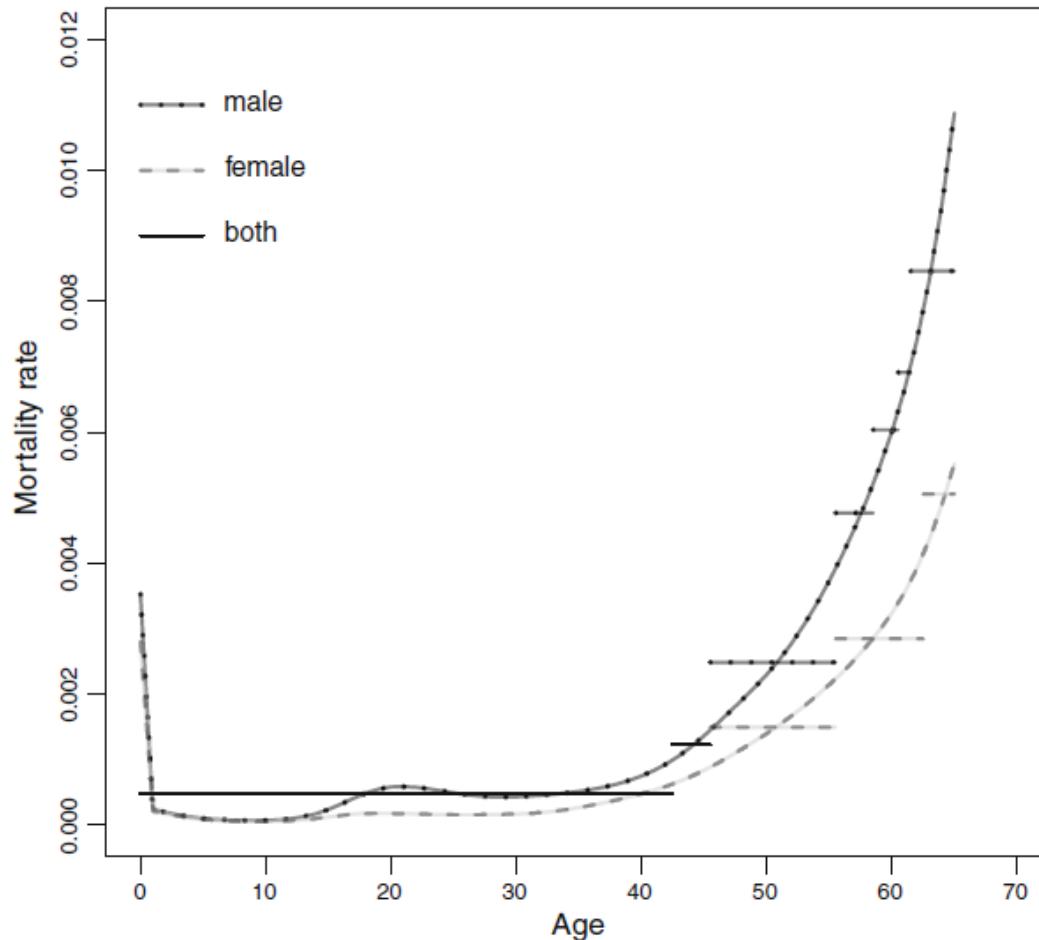


Fig. 8 Final tree for the standard life table example. For each terminal node the number of cases and the mortality rate (per mille) are given (the numbers in *brackets* are the labels for the nodes used in Table 6)

Courbes de mortalité correspondantes



Remarque

Grande différence entre la modélisation CART et une modélisation de type paramétrique

Le CART permet de s'autoriser des formes de **dépendance nettement plus variées**, alors que le modèle paramétrique impose une forme de dépendance bien précise entre Y et X

- Potentiellement inadapté dans de nombreux cas pratiques !
- exemple : fréquence d'un contrat auto en fonction de l'âge avec des classes d'âges

Cependant, dans l'exemple de la mortalité, **un modèle paramétrique serait préférable !**



Gestion du surapprentissage

Sélection de modèle

L'arbre maximal construit (de taille $K(n)$) génère une suite d'estimateurs $(\hat{\pi}^K(\mathbf{x}))_{K=1,\dots,K(n)}$ qui correspondent à chaque sous-arbre

Objectif : éviter un estimateur trop complexe (surapprentissage)

➤ Trouver le meilleur sous-arbre selon un arbitrage adéquation / prévision

$$R_\alpha(\hat{\pi}^K(\mathbf{x})) = \mathbb{E}_n[\Phi(Y, \hat{\pi}^K(\mathbf{x}))] + \alpha \times K/n$$

où α paramètre de complexité, K dimension de l'estimateur (#feuilles)

Pour α fixé, l'estimateur final optimise un critère coût-complexité

$$\hat{\pi}_\alpha^K(\mathbf{x}) = \arg \min_{(\hat{\pi}^K(\mathbf{x}))_{K=1,\dots,K(n)}} \{R_\alpha(\hat{\pi}^K(\mathbf{x}))\}$$

Résultats

Pour α fixé, l'arbre $\hat{\pi}_\alpha^K(x)$ est unique et le calcul est rapide !

Cas limites

- $\alpha = \infty$ le modèle sélectionné sera la racine ;
- $\alpha = 0$ le modèle sélectionné sera l'arbre maximal

Puisque n'importe quelle suite de sous-arbres emboîtés de l'arbre maximal a au maximum K membres, toutes les valeurs possibles de α peuvent être groupées en m intervalles ($m \leq K$)

$$I_1 = [0, \alpha_1] \quad I_2 = (\alpha_1, \alpha_2] \quad \dots \quad I_m = (\alpha_{m-1}, +\infty]$$

Procédure d'élagage

Raisonnement : impossible de parcourir tous les sous-modèles de l'arbre maximal (# de sous-arbres augmente exponentiellement avec # de feuilles)

1. On part de l'arbre maximal construit ;
2. On considère une 1^{ère} valeur de α : conduit à sélectionner un sous-arbre optimal de l'arbre maximal
3. À partir de ce sous-arbre optimal, on prend une autre valeur de α (plus grande) qui conduit à sélectionner un sous-arbre optimal de ce sous-arbre
4. Et ainsi de suite ...

On crée une suite croissante de α_z !

Procédure d'élagage

Par construction, on obtient une suite décroissante de sous-arbres optimaux emboîtés (de l'arbre maximal vers la racine)

Dans cette liste d'estimateurs, on choisit finalement $\hat{\alpha}$ (et l'arbre correspondant) tel que

$$\hat{\pi}_{\hat{\alpha}}^K(\mathbf{x}) = \arg \min_{\left(\hat{\pi}_{\alpha_Z}^K(\mathbf{x}) \right)_{\alpha=\alpha_1, \dots, \alpha_m}} \left\{ R_{\alpha_Z} \left(\hat{\pi}_{\alpha_Z}^K(\mathbf{x}) \right) \right\}$$

Pratiquement :

- il faut déterminer les valeurs possibles de α
- $\hat{\alpha}$ est choisi par cette erreur, mais moyennée via une validation croisée

Tuning : choix de l'hyperparamètre α

Différence entre tuning et élagage

- Tuning du modèle : sélection du paramètre de complexité α
- Elagage : sélection de modèle pour un α fixé

Application au CART : la validation croisée induit des séquence d'arbres emboîtés différentes

- L'erreur moyenne n'est pas calculée pour chaque sous-arbre avec un nombre de feuille donné,
- mais pour chaque valeur α_Z fixée issue de la séquence produite initialement par tout l'échantillon

Formulation algorithmique (V-fold)

1. Construction de l'arbre maximal T_{max}
2. Construction de la séquence T_K, \dots, T_1 d'arbres emboîtés associée à une séquence de valeurs (α_Z) ;
3. Pour $v = 1, \dots, V$ (où v désigne le segment de l'échantillon servant à la validation)
 1. Pour chaque nouvel échantillon d'apprentissage, construit T_{max} et estimer la séquence d'arbres associée à la séquence des pénalisations α_Z
 2. Estimation de l'erreur sur la partie validation de l'échantillon
4. Calcul de la séquence des moyennes de ces erreurs
5. L'erreur minimale désigne la pénalisation α_{opt} optimale
6. Retenir l'arbre associé à α_{opt} dans la suite initiale T_K, \dots, T_1

Arbre de Classification

Arbre de classification : Y discrète

Supposons que $Y \in \{A, B\}$

Dans le cas discret la quantité d'intérêt est

$$\pi_0 = \mathbb{E}_0[\mathbb{I}_{Y=A} \mid X = x] = \mathbb{P}(Y = A \mid X = x)$$

Il faut adapter le critère d'homogénéité, et donc la fonction de perte ϕ . On considère classiquement

- l'indice de Gini,
- l'entropie

Entropie

La **fonction d'entropie** est définie pour $p \in [0,1]$ par

$$f(p) = -p \log(p)$$

Appliquée au CART, dans un problème à deux classes $\{A, B\}$ pour Y , on définit l'hétérogénéité du nœud t comme

$$H_t = - \sum_{I=\{A,B\}} |t| p_t^I \log(p_t^I)$$

Où p_t^I est la proportion de la classe I dans le nœud t

On maximise la diminution de l'hétérogénéité

Indice de Gini

L'indice de Gini est définie pour $p \in [0,1]$ par

$$f(p) = p(1-p)$$

Appliquée au CART, on définit l'hétérogénéité du nœud t comme

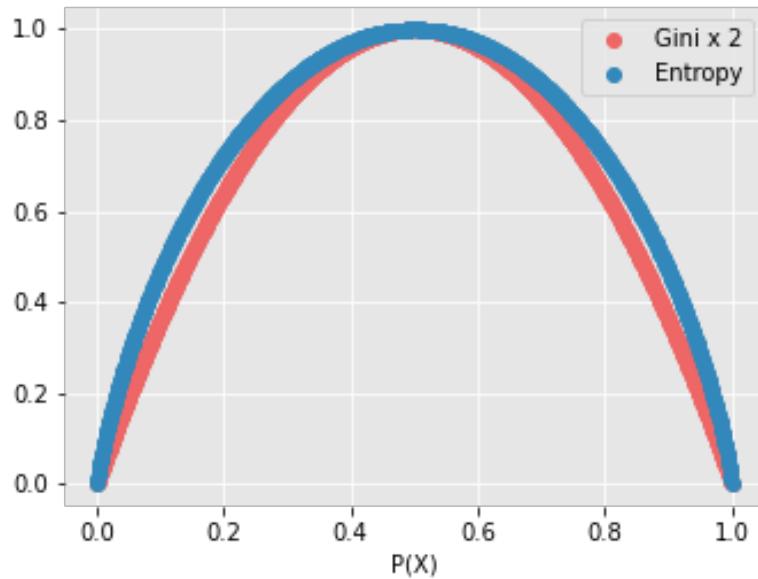
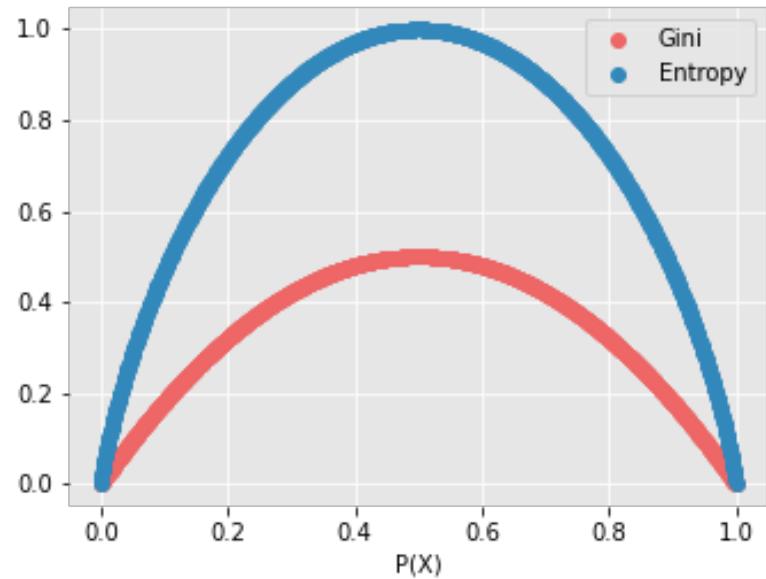
$$H_t = \sum_{I=\{A,B\}} p_t^I(1-p_t^I)$$

Remarque :

- L'indice de Gini est la variance de Bernoulli

Indice de Gini

Dans les deux cas, la quantité à optimiser est convexe/concave



Affectation pour la prévision

Concernant l'affectation de l'observation à prédire à l'une des classes, il y a trois distinctions possibles en fonction de l'information à disposition

1. Soit on affecte la classe **la plus représentée** dans la feuille
2. Soit on affecte la classe **a posteriori la plus probable** si l'on dispose de probabilités a priori des classes (vision bayésienne)
3. Soit on affecte la classe **la moins coûteuse** si des coûts de mauvais classement sont donnés



Mesures de performance

Réponse quantitative

Les mesures classiques de performance d'un modèle si Y est quantitative sont

L'Erreurs Quadratique Moyenne (MSE)

$$MSE(\hat{\pi}^K(\mathbf{x})) = \sum_i (Y_i - \hat{\pi}^K(\mathbf{x}_i))^2$$

L'Erreurs Absolue Moyenne (MAE)

$$MAE(\hat{\pi}^K(\mathbf{x})) = \sum_i |Y_i - \hat{\pi}^K(\mathbf{x}_i)|$$

Remarque : ces erreurs se mesurent sur échantillon test indépendant des échantillons ayant servi à construire et tuner/optimiser le modèle.

Réponse catégorielle : matrice de confusion

Dans un problème de classification, on utilise souvent la [matrice de confusion](#) comme mesure de performance : résume les individus mal classés et ceux bien classés par le modèle

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Remarques

En utilisant cet outil, on peut calculer facilement

Le **taux de mauvaise classification**

$$(FP+FN)/(FP+FN+TP+TN)$$

L'indice de **sensibilité** $TP/(TP+FN)$

L'indice de **spécificité** $TN/(TN+FP)$

Limites de ces mesures

Principalement deux limites à l'utilisation de cette matrice

Elle est dépendante d'un seuil d'affectation

- pour classer les prévisions du modèle, on définit ce seuil
- dans un problème à 2 classes, on utilise 0,5, mais ce n'est pas toujours le seuil optimal

Si le problème a des classes largement disproportionnées

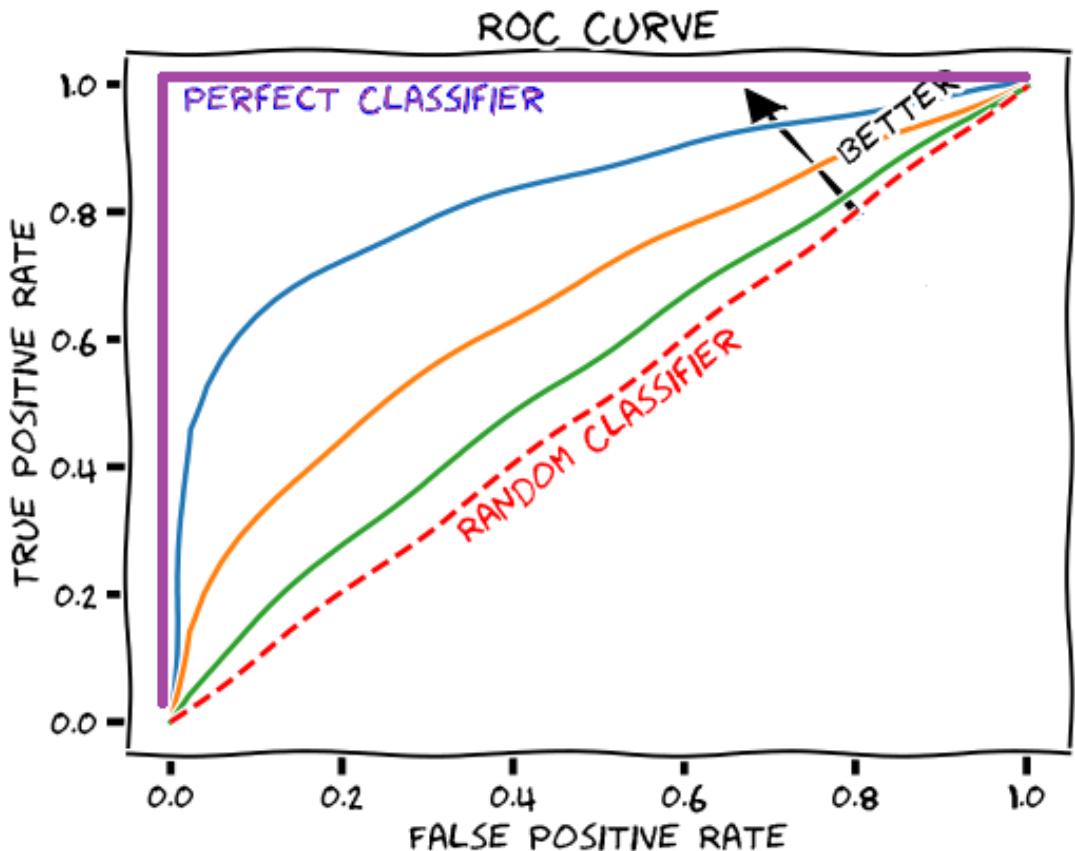
- le modèle prédira toujours la même classe, avec une erreur de classification très faible
- Alors que c'est souvent l'événement rare qui est intéressant à prédire !

Courbe ROC et AU

La courbe ROC (Receiving Operator Curve) résume le taux de TP et FP pour tous les seuils d'affectation

AUC (Area Under Curve)

- modèle aléatoire : 0,5
- modèle parfait : 1



Vision économique

La matrice de confusion ou l'AUC pour mesurer la performance et optimiser l'algorithme

- Vision purement statistique de la problématique
- L'algorithme a une réalité économique

Mesurer la performance d'un point de vue économique

- A quoi sert l'algorithme ?
- Quels sont les gains / profits que je peux en tirer ? → A/B testing

Inclure cette vision économique directement dans l'algorithme

- Pas seulement en fonction de validation mais aussi pour la fonction de perte



Conclusions

Problématiques classiques

Problème de biais de l'estimateur CART

- Si une variable explicative catégorielle contient trop de modalités...
- tendance à attirer la règle de division à cette variable

Problème de données unbalanced

- On se retrouve qu'avec la racine et on ne segmente pas
- Modifier la fonction de perte et/ou les poids des observations

Problème de biais d'observation

- Censure
- Troncature

Pour et contre

- Modèle non paramétrique + partition des données
- Cadre unique pour la **régression** et la **classification**
- Modèles **faciles à interpréter**
- Prédicteur numériques mélangés à des catégoriels
- Principal inconvénient : **manque de stabilité**
- Prédicteur de **base** pour l'**ensemble learning** : bagging, random forests, boosting