UP2 : Apprentissage Statistique & Automatique

Apprentissage et décision automatique

ANIS S. HOAYEK

Novembre 2020

Plan du cours

1. Arbres de décision.
2. Validité : Sensibilité, Spécificité, ROC, AUC etc.
3. Forêts aléatoires.
4. Lien avec Bagging et Boosting.
5. Applications classiques + TP (R/WEKA).
6. Évaluation : TP noté (compte rendu à 2 où à 3 personnes).

Arbres de Décision

Introduction

1. Algorithme de classification/régression supervisé.
2. Méthode statistique non-paramétrique.
3. Permet de classer un ensemble d'individus décrits par des variables qualitatives et quantitatives.
4. Produit des classes les plus homogènes possibles.
5. Classifications compréhensibles pour l'utilisateur (dans les méthodes classiques (hiérarchique, k-means,...) l'information est perdue dans les classes).

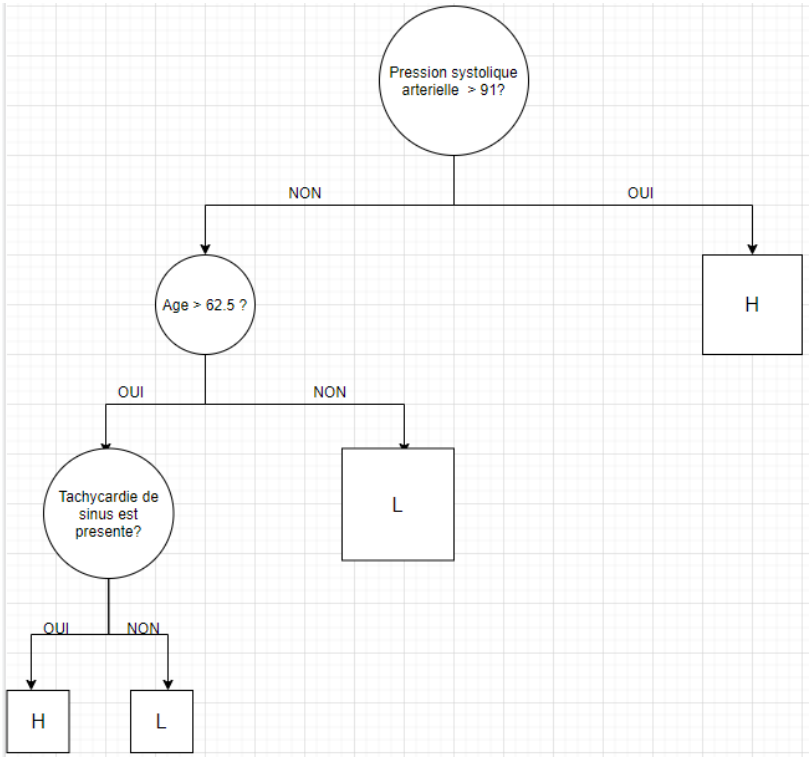
Introduction

Domaines d'applications les plus populaires :

1. Secteur bancaire : Identification du risque de crédit d'un client en fonction de sa probabilité de défaut de paiement.
2. Médecine : Identification des patients à risque et les tendances de la maladie.
3. Marketing : Identification du taux de désabonnement des clients.
4. Problèmes de décision environnementaux, économiques, financiers, etc.

Introduction

1. Dans un hôpital, pour chaque nouveau patient avec une crise cardiaque, on mesure 15 variables pendant les premières 18 heures. Parmi les variables : la pression artérielle, l'age et 13 autres caractéristiques résumant les différents symptômes.
2. L'objectif de l'étude est d'identifier les patients à haut risque (ceux qui ne survivrons pas au moins 30 jours).



Introduction

- Représentation :
  - La racine : χ.
  - Un nœud : sous ensemble de χ (représenté par un cercle).
  - Nœuds terminaux : sous-ensembles qui ne sont plus divisés (représentés par des boites).
  - Chaque nœud terminal est marqué par une classe cible qui est une des valeurs y d'un attribut cible Y.
- Construction d'un arbre de décision :
  - Obtenu par des coupes répétées de χ en des sous-ensembles.
  - On sélectionne la variable qui sépare "le mieux" les données.
  - Le processus se répète pour chaque sous-groupe.
  - On s'arrête quand les sous-groupes atteignent la taille minimale, ou quand il n'y plus d'amélioration.

Ainsi, la construction d'un arbre de décision nécessite :

- Sélectionner les coupes.
- Décider de déclarer un nœud terminal ou continuer de le scinder à nouveau.
- Affecter une classe à chaque nœud terminal.

Notations

- n : taille de l'échantillon (nombre des observations).
- K : nombre de classes de la variable cible Y.
- $N(t)$: nombre d'observations dans le nœud t.
- $N_k(t)$: nombre d'observations de la classe $k \in \{1, 2, \ldots, K\}$ dans le nœud t.
- $p(k|t)$: proportion d'observations dans le nœud t appartenant à la classe $k \in \{1, 2, \ldots, K\}$

$$p(k|t) = \frac{N_k(t)}{N(t)}.$$

- $p(t)$: vecteur de proportions correspondant au nœud t

$$p(t) = [p(1|t), p(2|t), \ldots, p(K|t)].$$

- $Y(t)$: classe attribuée au nœud t

$$Y(t) = \arg \max_{k=1,\ldots,K} p(k|t).$$

Définition 1.

Une mesure d'impureté d'un nœud t dans un arbre de décision ayant une variable cible Y de K classes est une fonction ayant la forme :

$$Imp(t) = \phi(p(t)),$$

où ϕ est une fonction non-négative de $p(t)$ qui satisfait les conditions suivantes :

1. ϕ atteint son maximum unique en $p(t) = \left[\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}\right]$.
2. ϕ atteint le minimum en $[1, 0, \ldots, 0], [0, 1, \ldots, 0], \ldots, [0, 0, \ldots, 1]$.
3. ϕ est une fonction symétrique de $p(1|t), p(2|t), \ldots, p(k|t)$.

Remarque.

$Imp(t)$ est maximale quand toutes les classes sont mélangées avec des "parts égales" (Distribution uniforme/équiprobable) et est minimale quand le nœud ne contient qu'une seule classe (certitude totale).

Exemples des mesures d'impureté :

1. Entropie :

$$Imp(t) = \mathcal{H}(t) = -\sum_{k=1}^{K} p(k|t) \log_2 p(k|t),$$

avec : $0 \log_2 0 = 0$.

2. Indice de Gini :

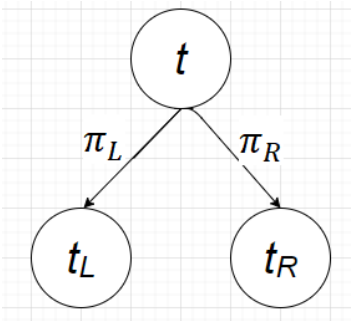
$$Imp(t) = \mathcal{G}(t) = 1 - \sum_{k=1}^{K} p^2(k|t).$$

Remarque.

Un nœud est pure s'il contient des données d'une seule classe. Dans ce cas $\mathcal{H}(t) = \mathcal{G}(t) = 0$.

Couper ou ne pas couper ?

On considère un nœud t avec deux nœuds fils t_L et t_R. On note cette opération de coupe par $\mathcal{S}$.



avec :

- π_L = proportion d'observations de t qui vont vers t_L.
- π_R = proportion d'observations de t qui vont vers t_R.

Couper ou ne pas couper ?

Qualité de la coupe $\mathcal{S}$ est définie par la variation de la mesure d'impureté :

$$\Phi(\mathcal{S}, t) = \Delta Imp(t) = Imp(t) - \pi_L Imp(t_L) - \pi_R Imp(t_R).$$

L'idée est de choisir une coupe $\mathcal{S}$ qui maximise $\Phi(\mathcal{S}, t)$.

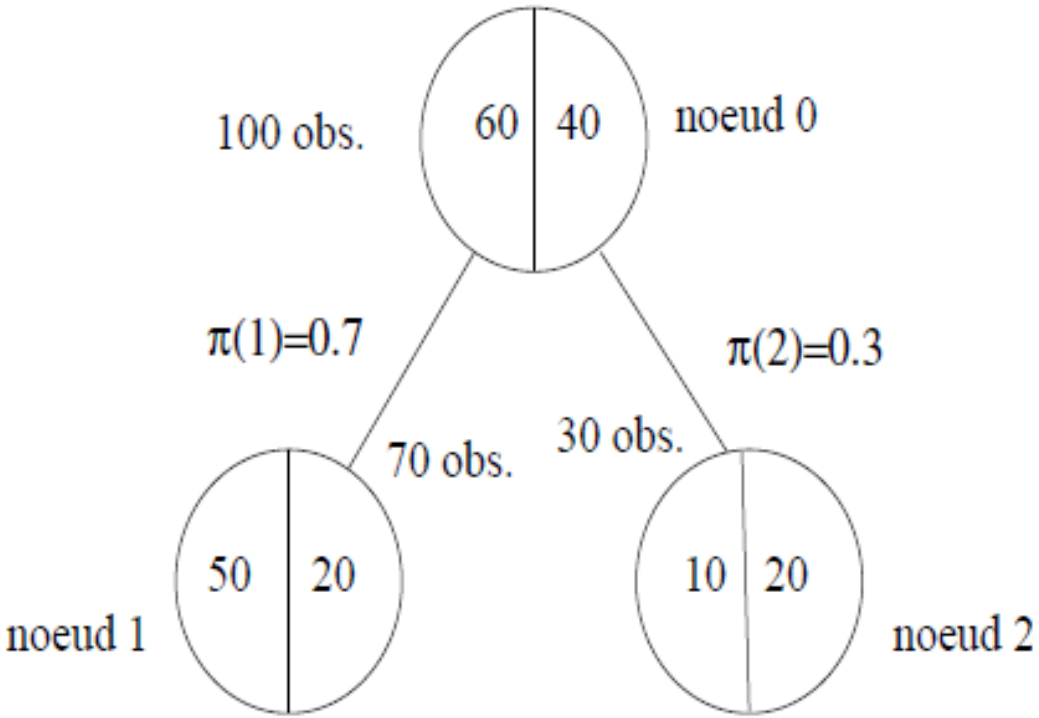
Définition 2.

L'impureté globale d'un arbre de décision T est définie par :

$$Imp(T) = \sum_{t \in \widetilde{T}} \pi(t)\, Imp(t),$$

où, $\widetilde{T}$ = l'ensemble des nœuds terminaux
et $\pi(t)$ = la proportion de la population globale en nœud t.

Exemple Numérique



Règles de coupe

Soit $X = [X_1, X_2, \ldots, X_p]$ le vecteur de variables explicatives qui se présentent dans un contexte donné.
Les coupes des nœuds d'un arbre de décision doivent vérifier les conditions suivantes :

- Chaque coupe ne dépend que d'une seule variable.
- Pour les X_i quantitatives, le critère de la coupe est de la forme :

$$\text{Est-ce que } X_i \leq c, \text{avec } c \in \mathbb{R}.$$

- Si X_i est catégorielle à valeurs dans $B = \{b_1, b_2, \ldots, b_m\}$, le critère de la coupe est de la forme :

$$\text{Est-ce que } X_i \in A, \text{avec } A \subseteq B.$$

- A chaque nœud on considère les variables X_i une par une : 1) On trouve la meilleure coupe de chaque X_i, 2) On choisit la meilleure variable en terme de qualité de la coupe.

Règles d'arrêt et d'attribution

Arrêter les coupes si :

- La variation de la mesure d'impureté d'un nœud est inférieure à un certain seuil.
- Profondeur de l'arbre est supérieure à une valeur prédéfinie.
- En utilisant l'élagage au lieu de règles d'arrêt.

On attribue la classe :

$$Y(t) = \arg \max_{k=1,\ldots,K} p(k|t),$$

à un nœud terminal.

Règles d'arrêt et d'attribution

- La démarche de construction précédente fournit l'arbre maximal T_{max}.
- T_{max} peut conduire à un modèle de prévision très instable car fortement dépendant des échantillons qui ont permis la construction de l'arbre.
- C'est une situation de sur-ajustement .
- Solution : procédure d'élagage de l'arbre.

Définition 3.

Taux d'erreurs de classification du nœud t :

$$R(t) = \sum_{k=1, k \neq Y(t)}^{K} p(k|t),$$

avec

$$Y(t) = \arg \max_{k=1,\ldots,K} p(k|t) = \text{classe attribuée au noeud } t \ .$$

Définition 4.

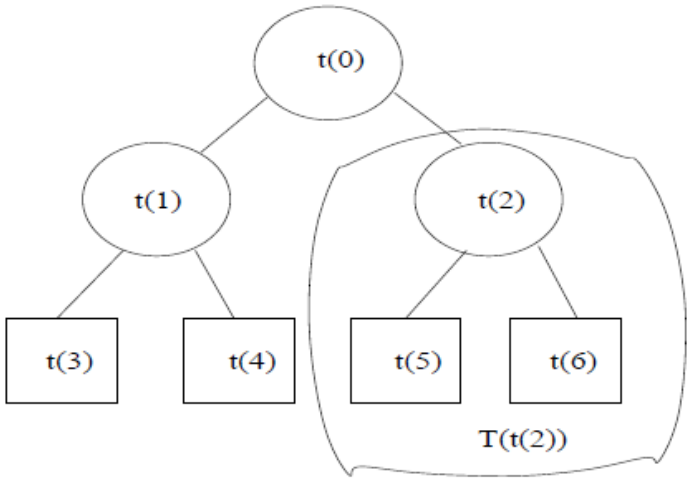
Soit $\widetilde{T} = \{t_1, t_2, \ldots, t_m\}$ les nœuds terminaux d'un arbre T. Le taux d'erreurs de classification de T est :

$$R(T) = \sum_{i=1}^{m} \frac{N(t_i)}{n} R(t_i).$$

De plus, on définit $\text{taille}(T) = \text{Card}\left(\widetilde{T}\right)$ et $\alpha > 0$ un paramètre de complexité (CP) qui nous aide à imposer une certaine pénalité pour les grands arbres. Ainsi, le taux d'erreurs pénalisé de T est donné par :

$$R_\alpha(T) = R(T) + \alpha \text{taille}(T).$$

Élagage et cout de complexité



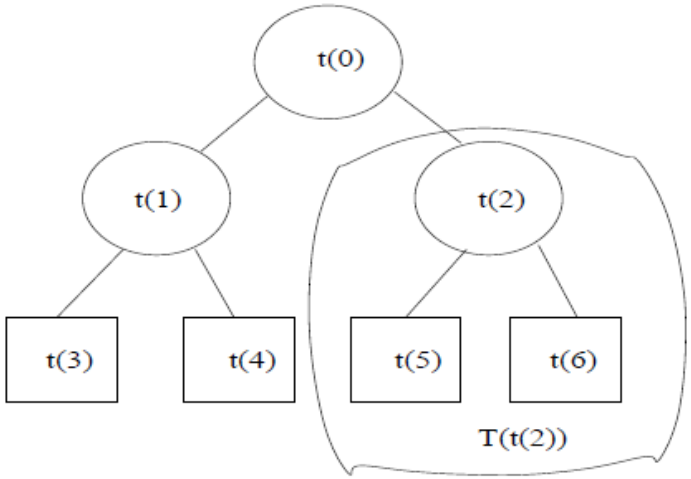
- On note par $T(t)$ le sous-arbre avec la racine en t.
- L'erreur du sous-arbre $T(t_2)$ et l'erreur du nœud t_2 sont respectivement :

$$R_\alpha(T(t_2)) = R(T(t_2)) + \alpha \text{taille}(T(t_2)),$$
$$= \frac{1}{n'}[N(t_5)R(t_5) + N(t_6)R(t_6)] + 2\alpha,$$

avec, n' = taille de l'échantillon dans le sous-arbre de racine t_2.

$$R_\alpha(t_2) = R(t_2) + \alpha,$$
$$= \sum_{k=1, k \neq Y(t_2)}^{K} p(k|t_2) + \alpha.$$

Élagage et cout de complexité



第一个是t2这个点本身

第二个是2这个节点以及子节点

L'élagage en vaut la peine si :

$$R_\alpha(t_2) \leq R_\alpha(T(t_2)) \Leftrightarrow g(t_2, T) \equiv \frac{R(t_2) - R(T(t_2))}{\text{taille}(T(t_2)) - 1} \leq \alpha.$$

La fonction $g(t, T)$ peut être calculée pour chaque nœud interne de l'arbre.

Autre application d'élagage (Estimation de α)

- Algorithme de coupe du maillon faible :
  - Commencer avec l'arbre complet T.
  - Pour chaque nœud non-terminal $t \in T$ calculer $g(t, T)$, et trouver $t_1 = \arg\min_{t \in T} g(t, T)$. On pose $\alpha_1 = g(t_1, T)$.
  - Définir un nouvel arbre T_1 en enlevant la branche partant de t_1.
  - Trouver le maillon faible de T_1 et procéder comme pour T.
- Le résultat : une suite décroissante d'arbres et l'arbre final sera déterminé en fonction de son pouvoir de prédiction.

Application 1 :
Données “weather”

Application 1

Le tableau est composé de 14 observations, il s'agit d'expliquer le comportement des individus par rapport à un jeu {jouer, ne pas jouer} à partir des prévisions météorologiques :

Numéro	Ensoleillement	Température	Humidité	Vent	Jouer
1	soleil	Chaude	Elevee	non	non
2	soleil	Chaude	Elevee	oui	non
3	couvert	Chaude	Elevee	non	oui
4	pluie	Tiede	Elevee	non	oui
5	pluie	Fraiche	Normale	non	oui
6	pluie	Fraiche	Normale	oui	non
7	couvert	Fraiche	Normale	oui	oui
8	soleil	Tiede	Elevee	non	non
9	soleil	Fraiche	Normale	non	oui
10	pluie	Tiede	Normale	non	oui
11	soleil	Tiede	Normale	oui	oui
12	couvert	Tiede	Elevee	oui	oui
13	couvert	Chaude	Normale	non	oui
14	pluie	Tiede	Elevee	oui	non

Application 1

- On commence par calculer les gains d'information pour chaque attribut :

Variable	Variation d'impureté
Ensoleillement	
Température	
Humidité	
Vent	

- Donc, la racine de l'arbre de décision est la variable "Ensoleillement".
- Maintenant, l'attribut "Ensoleillement" peut prendre trois valeurs. On refait le calcul de l'étape précédente pour chacune des différentes valeurs.

Numéro	Ensoleillement	Température	Humidité	Vent	Jouer
1	soleil	Chaude	Elevee	non	non
2	soleil	Chaude	Elevee	oui	non
8	soleil	Tiede	Elevee	non	non
9	soleil	Fraiche	Normale	non	oui
11	soleil	Tiede	Normale	oui	oui

因为对ensoleiilemnt分类之后，temperature的chaude和fraiche对应的jouer只有一个（对应的树叶terminal只有一个）那么就没有分类的必要了，直接舍弃，所以只有tiede，那么就看humidite，对应有elevee和normal

Application 1

- Les gains d'information pour la valeur "soleil" :

Variable	Variation d'impureté
Température	
Humidité	
Vent	

- Pour la valeur "couvert" on a un nœud pure :

Numéro	Ensoleillement	Température	Humidité	Vent	Jouer
3	couvert	Chaude	Elevee	non	oui
7	couvert	Fraiche	Normale	oui	oui
12	couvert	Tiede	Elevee	oui	oui
13	couvert	Chaude	Normale	non	oui

- La distribution de la variable dépendante pour la valeur "pluie" est :

Numéro	Ensoleillement	Température	Humidité	Vent	Jouer
4	pluie	Tiede	Elevee	non	oui
5	pluie	Fraiche	Normale	non	oui
6	pluie	Fraiche	Normale	oui	non
10	pluie	Tiede	Normale	non	oui
14	pluie	Tiede	Elevee	oui	non

Ainsi,

Variable	Variation d'impureté
Température	
Humidité	
Vent	

Application 1

首先对ensoleillement，tem，hum，vent，分别对下属的分类对应的jouer分类求值（课间截图片有样式）求出最大的为树的第一层（求出ensol最大，越大说明分类越好），接着根据ensol的种类在接着上面的步骤再来一次，确定第二层

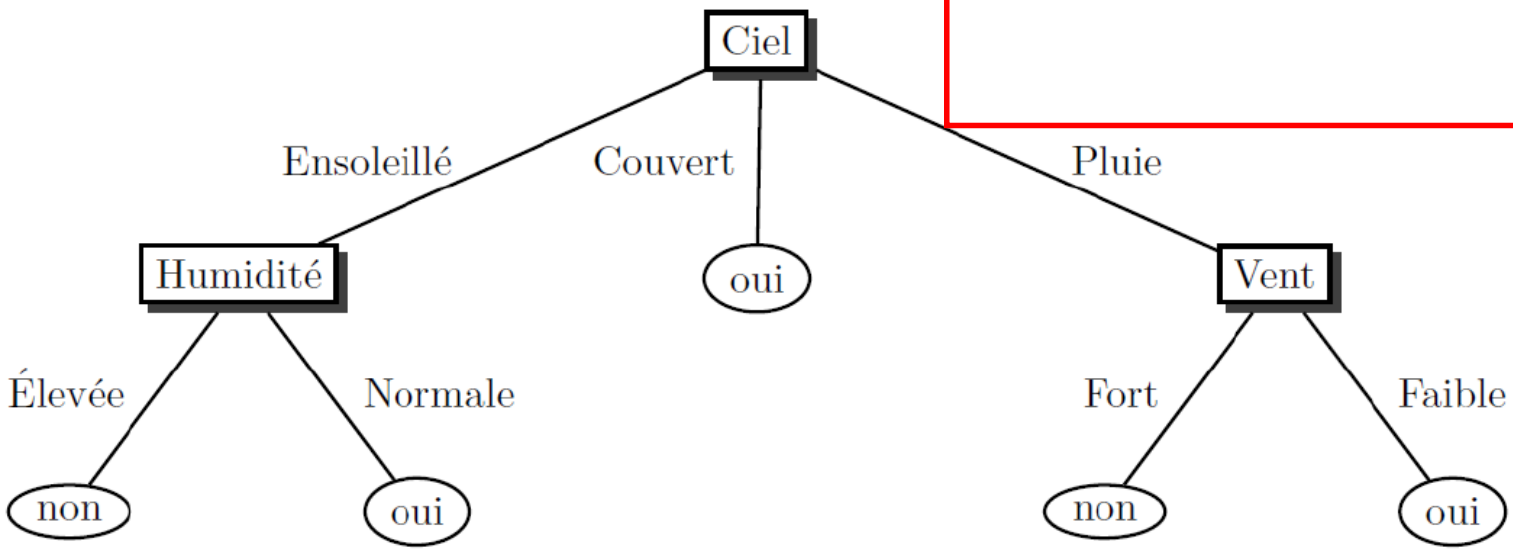


Figure 1 – Arbre de décision obtenu pour l'exemple "Jouer ou ne pas Jouer ?"

Une autre application (live demo) : `http://live.yworks.com/demos/complete/decisiontree/index.html`

Application demo

Application 1

- L'arbre de décision qui vient d' être construit donne des informations sur le niveau de significativité des attributs vis-à-vis de la classification de la variable dépendante.
- L'attribut "Température" n'étant pas utilisé dans l'arbre ; ceci indique que cet attribut n'est pas statistiquement significatif pour déterminer la classe de la variable dépendante.
- Si l'attribut "Ensoleillement" vaut "soleil", l'attribut "Vent" n'est plus significatif. Si l'attribut "Ensoleillement" vaut "pluie", c'est l'attribut "Humidité" qui ne l'est pas.

Application 1 bis

Cas d'un attribut numérique :

Numéro	soleilleme	Température	Humidité	Vent	Jouer
1	soleil	27.5	85	non	non
2	soleil	25	90	oui	non
3	couvert	26.5	86	non	oui
4	pluie	20	96	non	oui
5	pluie	19	80	non	oui
6	pluie	17.5	70	oui	non
7	couvert	17	65	oui	oui
8	soleil	21	95	non	non
9	soleil	16.5	70	non	oui
10	pluie	22.5	80	non	oui
11	soleil	22.5	70	oui	oui
12	couvert	21	90	oui	oui
13	couvert	25.5	75	non	oui
14	pluie	20.5	91	oui	non

Application 1 bis

Afin de préciser le seuil pour lequel on peut couper une variable numérique :

- Trier la variable numérique dans l'ordre croissant :

Numéro	Ensoleillement	Température	Humidité	Vent	Jouer
9	soleil	16.5	70	non	oui
7	couvert	17	65	oui	oui
6	pluie	17.5	70	oui	non
5	pluie	19	80	non	oui
4	pluie	20	96	non	oui
14	pluie	20.5	91	oui	non
8	soleil	21	95	non	non
12	couvert	21	90	oui	oui
10	pluie	22.5	80	non	oui
11	soleil	22.5	70	oui	oui
2	soleil	25	90	oui	non
13	couvert	25.5	75	non	oui
3	couvert	26.5	86	non	oui
1	soleil	27.5	85	non	non

- Ne pas séparer deux observations successives ayant la même classe.
- Si on coupe entre deux valeurs v et w $(v < w)$ le seuil s est fixe à v ou aussi $s = \frac{v+w}{2}$.
- Choisir s de telle manière que le gain d'information soit maximal.

Application 1 bis

Pour la variable "Température" les valeurs possibles de s sont :

$$17.25; 18.25; 20.25; 21; 23.75; 25.25; 27$$

- Pour $s = 17.25$ le gain de l'information est :

$$\Phi_{\text{Temperature}}(s = 17.25, t_0) = \Delta Imp(t_0) = Imp(t_0) - \pi_L Imp(t_L) - \pi_R Imp(t_R),$$

- De la même manière, en fonction du seuil, le gain d'information est alors :

Seuil $= s$	$\Phi_{\text{Temperature}}(s, t_0)$
17.25	??
18.25	...
20.25	...
21	...
23.75	...
25.25	...
27	...

Application 1 bis

Remarque.

Nous avons montré comment choisir le seuil pour un attribut numérique donné. Ainsi, on applique cette méthode pour chaque attribut numérique et on détermine pour chacun un seuil produisant un gain d'information maximal. Alors, Le gain d'information associé à chacun des attributs numériques est celui pour lequel le seuil entraine un maximum. Finalement, l'attribut choisi pour effectuer la coupe est celui, parmi les numériques et les catégoriels, qui produit un gain d'information maximal.

Application 2 :
Cancer de prostate en stage C

Application 2

Données : On considère 7 variables sur 146 patients au stade C du cancer de prostate :

- pgtime : dernier suivi (années).
- pgstat : le statut au dernier suivi (1 = progression, 0 = pas de progression).
- age : l'age du patient.
- eet : thérapie endocrinienne précoce (1=no, 2=yes).
- grade : grade de la tumeur (1–4) - Farrow system.
- gleason : grade de la tumeur - Gleason system.
- ploidy : l'état de la tumeur diploid/tetraploid/aneuploid.

Application 2

Paramètres par défaut :

- minsplit = 20 : nombre minimal d'observations dans un nœud pour lequel la coupe est calculée.
- minbucket = minsplit/3 : nombre minimal d'observation dans un nœud terminal.
- cp = 0.01 : paramètre de complexité.

Application 2

Sous R avec la fonction rpart on obtient :

- Le tableau de CP est imprimé du plus petit arbre (0 coupes) au plus grand (5 coupes pour les données de cancer).
- rel error : l'erreur relative $R(T)$.
- xerror : l'erreur calculée par validation croisée.
- xstd : l'écart-type de l'erreur calculée par validation croisée.

Généralités

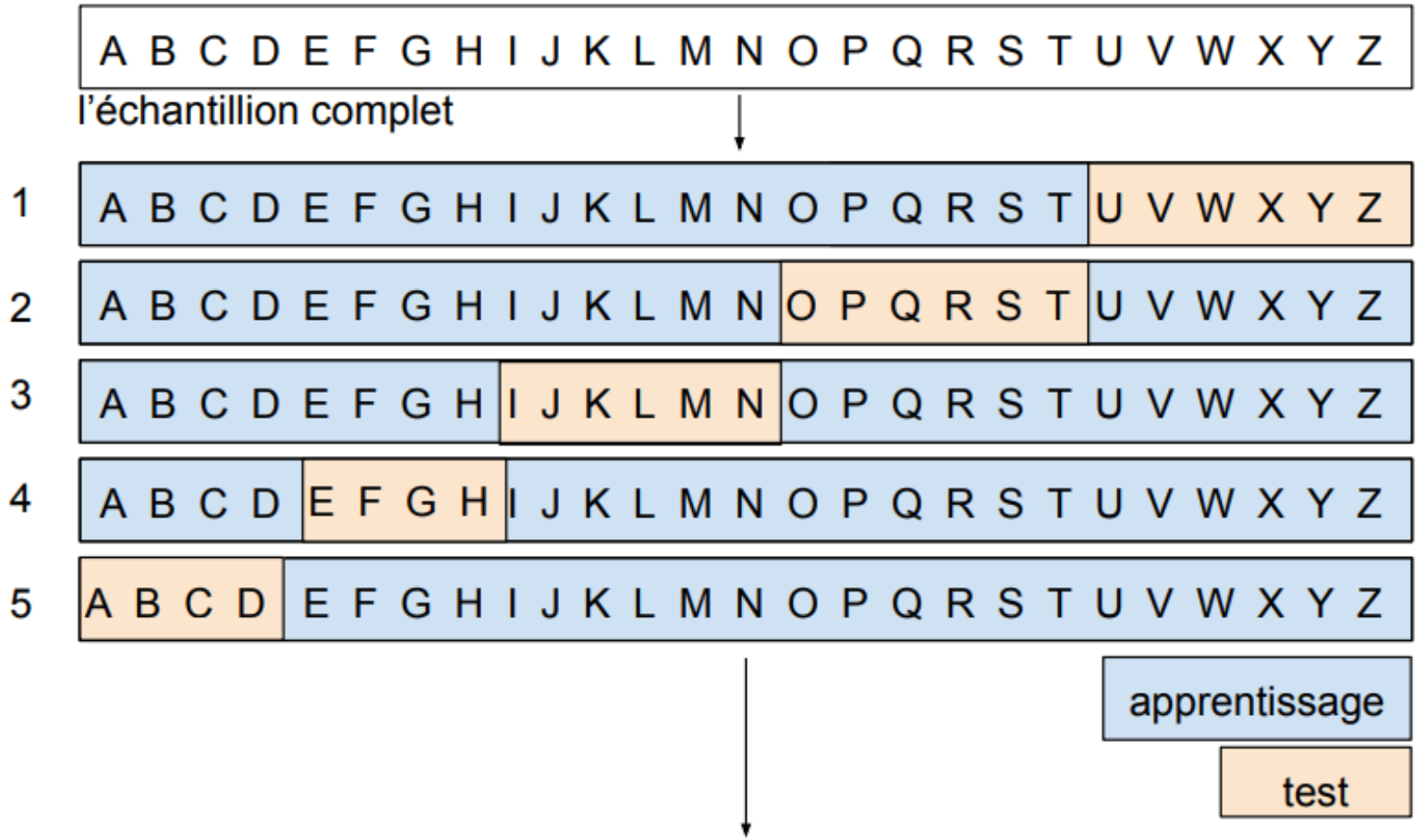
- Les différents algorithmes d'arbre de décision vont différer par les trois opérations suivantes :
  - Décider si un nœud est terminal (tous les individus sont dans la même classe ou il y a moins d'un certain nombre d'erreurs).
  - Sélectionner un test à associer à un nœud (aléatoirement ou en utilisant des critères statistiques).
  - Affecter une classe à une feuille (nœud terminal) - la classe majoritaire !
- Problèmes :
  - L'algorithme ne revient pas en arrière et ne remet pas en question ses choix.
  - On peut obtenir une erreur faible sur l'ensemble d'apprentissage, mais aussi un faible pouvoir prédictif (Phénomène de sur-apprentissage).
- Solutions pour le phénomène de sur-apprentissage :
  - Élagage pour essayer de diminuer l'erreur réelle (de validation).
  - Découpage en ensemble d'apprentissage et de test $\Longrightarrow$ Validation croisée.

Validation croisée

Technique utilisée dans le domaine de sélection de modèle. e.g. paramètre de complexité d'un arbre de décision (cp) :

- Séparer les données d'apprentissage et de test.
- Construire l'estimateur sur l'échantillon d'apprentissage.
- Utiliser l'échantillon test pour calculer un risque de prédiction.
- Répéter plusieurs fois et moyenner les risques de prédiction obtenus.

Validation croisée

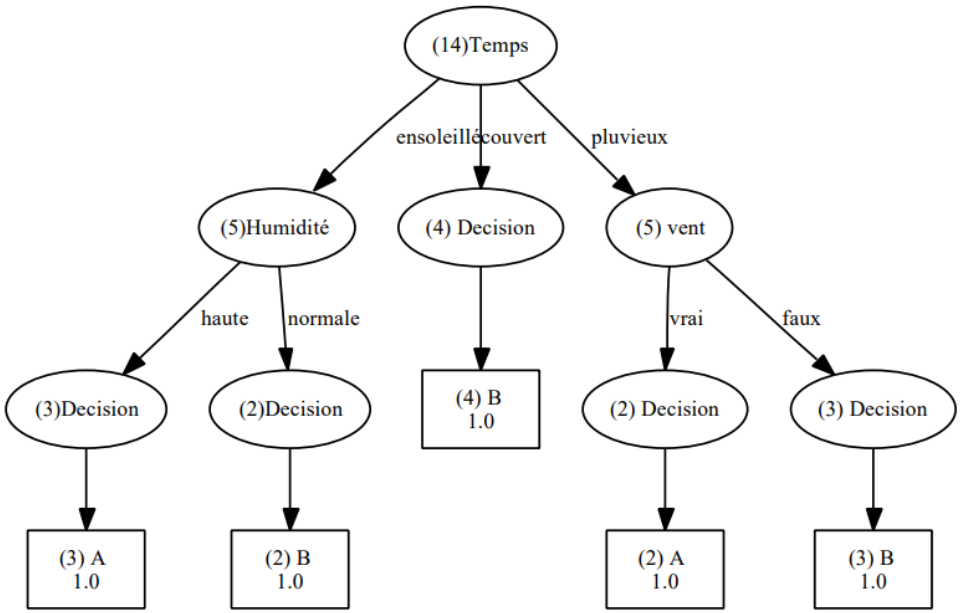


$$CV(\alpha) = \frac{1}{5}\sum_{v=1}^{5} MSE_v^{(test)}$$

- Valeurs d'attributs manquantes :
  - Certaines valeurs des variables Indépendantes sont manquantes.
  - Certaines valeurs de la variable à classer sont manquantes.
  - Pendant la phase de classement si la valeur d'un attribut est manquante, il est impossible de décider quelle branche on doit choisir pour classer l'objet.
- Les solutions les plus populaires :
  - On laisse de coté les instances ayant des valeurs manquantes.
  - Imputation : Moyenne, Mode, Médiane, aide d'un expert, etc.
  - Utiliser l'arbre de décision pour déterminer la valeur manquante d'un attribut (Méthode Quinlan).

Exemple : Valeurs manquantes

Temps	Température	Humidité	Vent	Classe
Ensoleillé	basse	?	Faux	?



Validité : Sensibilité, Spécificité, ROC, AUC

Validité : Sensibilité, Spécificité, VPP, VPN

Test	Covid-19	Non-Covid-19	Total
Positif	56	49	105
Négatif	14	461	475
Total	70	510	580

- Faux positifs : 49.
- Faux négatifs : 14.
- Vrais positifs : 56.
- Vrais négatifs : 461.
- Sensibilité (Se) : proportion de vrais positifs parmi les malades : $^{56}/_{70} = 80\%$.
- Spécificité (Sp) : proportion de vrais négatifs chez les non-malades : $^{461}/_{510} = 90\%$.
- Un bon test doit être sensible et spécifique.

Validité : Sensibilité, Spécificité, VPP, VPN

- Sachant que le test est positif, quelle est la probabilité d'être vraiment malade ? **Valeur Prédictive Positive ou VPP :**

$$VPP = \mathbb{P}[M|+] = \frac{p \times \mathbf{Se}}{p \times \mathbf{Se} + (1-p) \times (1-\mathbf{Sp})}.$$

- Nécessite de connaitre la prévalence p.
- VPP doit être $> p$.
- On définit également la **Valeur Prédictive Négative VPN.**

Validité : Sensibilité, Spécificité, VPP, VPN

En utilisant les données de l'exemple précédent :

1. Calculer la VPP et la VPN.
2. Calculer les IC à 95% des : Sensibilité, Spécificité et VPN.

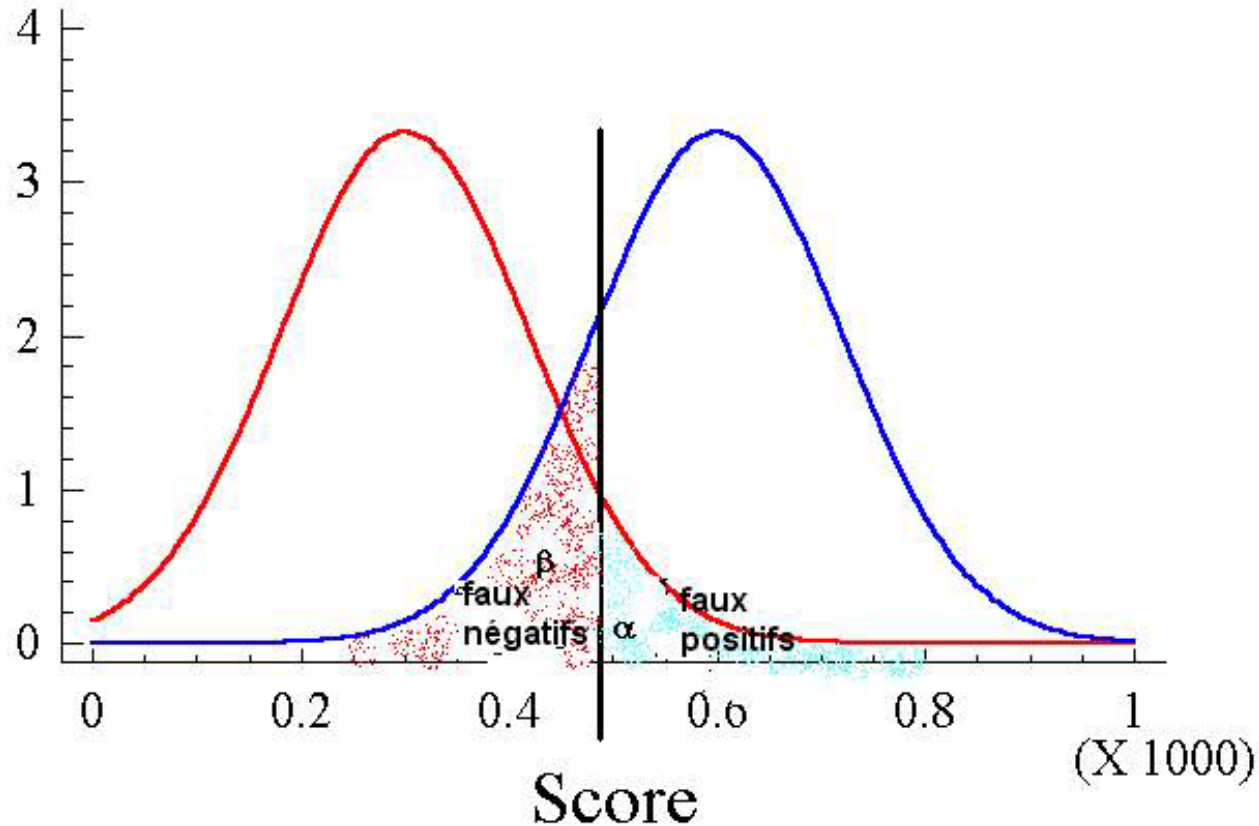
Remarque.

- En pratique, un gain de sensibilité est obtenu contre une perte de spécificité, et vice-versa.
- La connaissance de la Se et Sp d'un test, n'aide pas à décider si un individu a la condition ou non, une fois que le résultat du test est connu. Cette information est donnée par les valeurs prédictives. Ces valeurs dépendent de la prévalence de la condition dans la population étudiée.

La courbe ROC (Receiver Operating Characteristic)

Un test s'appuie souvent sur une mesure quantitative continue S :

- Au-delà d'un seuil s, on est déclaré positif.
- Comment choisir le seuil s ?
- Comment varient la sensibilité et la spécificité en fonction du seuil s ?
- Risque de première espèce $= \alpha$ (taux de faux positifs).
- Risque de deuxième espèce $= \beta$ (taux de faux négatifs).



La courbe ROC (Receiver Operating Characteristic)

- Sensibilité : $1-\beta = \mathbb{P}\left[S > s | M\right]$ (puissance du test).
- Spécificité : $1-\alpha = \mathbb{P}\left[S < s | \overline{M}\right]$.

Définition 5.

La courbe ROC représente l'évolution de $1-\beta$ (Se/puissance du test) en fonction de α (1− Sp) lorsque le seuil varie.

La courbe ROC (Receiver Operating Characteristic)

Exemple : répartition du pic de fièvre pour les vraies et fausses grippes.

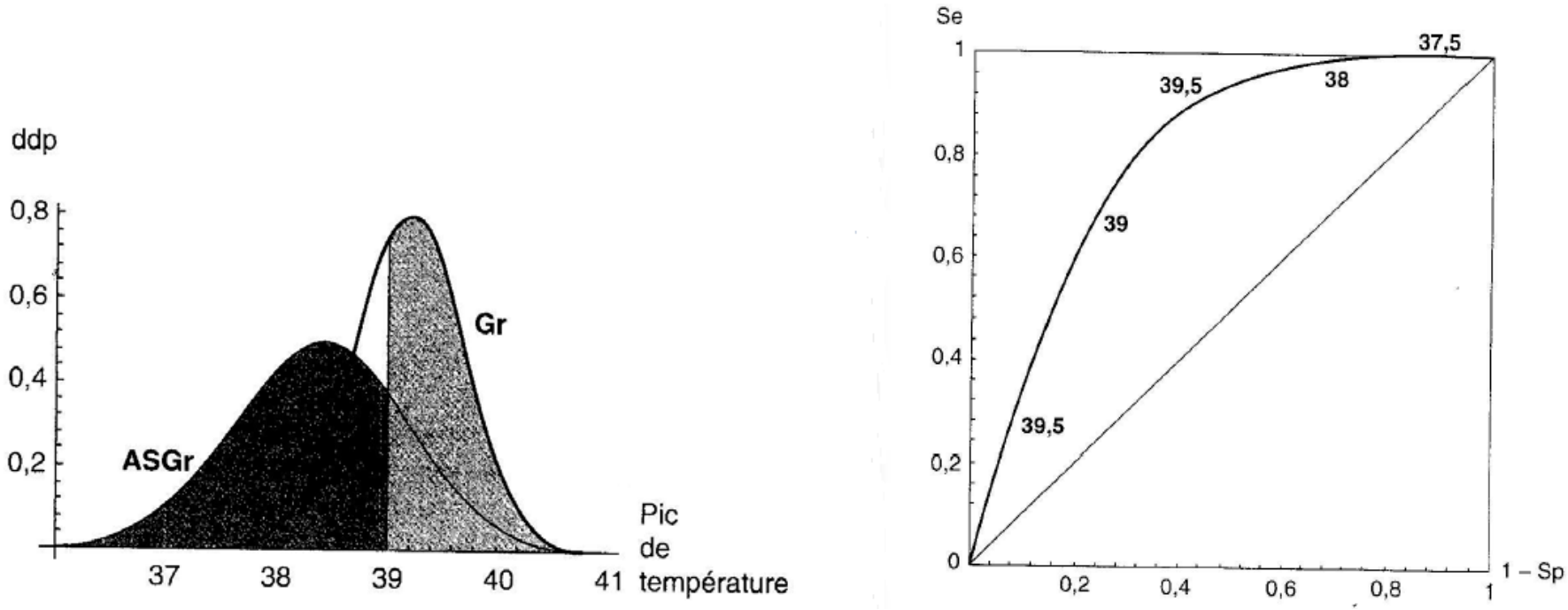


Figure 2 – A gauche : distribution de probabilité pic de fièvre chez des sujets ayant une vraie grippe (Gr) et une fausse grippe (ASGr) ; A droite : Courbe ROC pour l'exemple du diagnostic de la vraie grippe vs celui de la fausse grippe.

La courbe ROC (Receiver Operating Characteristic)

Choix du seuil optimal à l'aide de la courbe ROC :

- Le seuil correspondant au point le plus proche de l'idéal $(0; 1)$ (i.e. le point le plus loin de la diagonale).
- L'AUC ou surface sous la courbe ROC (Area Under Curve) : plus l'AUC est grand, meilleur est le test.
- L'AUC fournit un ordre partiel sur les tests.
- Courbe ROC (AUC) est une mesure intrinsèque de séparabilité, invariante pour toute transformation monotone croissante des différents seuils.

La courbe ROC (Receiver Operating Characteristic)

Choix du seuil optimal selon la minimisation du coût d'erreur :
C_M : coût de déclarer malade à tort.
C_{NM} : coût de déclarer non malade à tort.
Espérance du coût global pour un seuil s :

$$\mathbb{E}\left[Coût\right] = C_{NM}\mathbb{P}\left[M \cap -\right] + C_M\mathbb{P}\left[\overline{M} \cap +\right].$$

On montre que le seuil avec le cout minimal vérifie l'équation suivante :

$$\frac{f_M(s)}{f_{\overline{M}}(s)} = \frac{C_M}{C_{NM}}\frac{1-p}{p},$$

avec, $f_M(s)$: la densité de probabilité des malades
$\left(f_M(s) = \frac{\mathrm{d}}{\mathrm{d}s}F_M(s) = \frac{\mathrm{d}}{\mathrm{d}s}\mathbb{P}\left[S \leq s|M\right]\right)$,
$f_{\overline{M}}(s)$: la densité de probabilité des non malades
$\left(f_{\overline{M}}(s) = \frac{\mathrm{d}}{\mathrm{d}s}F_{\overline{M}}(s) = \frac{\mathrm{d}}{\mathrm{d}s}\mathbb{P}\left[S \leq s|\overline{M}\right]\right)$.
Preuve ?
Pente de la tangente à la courbe ROC ?

Arbre de décision pour la régression

- On considère le cas d'une variable à expliquer quantitative (régression) e.g. $Y =$ salaire annuel (en milliers de dollars).
- En deux étapes :
  - Diviser l'espace prédicteur (les variables explicatives $X_1, X_2, ..., X_p$) en K régions exhaustives et non chevauchantes $R_1, R_2, ..., R_K$.
  - Pour chaque observation tombant dans la région R_k, la prédiction de Y est donnée par la moyenne des valeurs des Y dans R_k.
- Ainsi l'algorithme nécessite :
  - Sélectionner la meilleure division pour les différentes variables (fixer un critère).
  - Une règle d'arrêt (décider qu'un nœud est terminal).
  - Élagage pour éviter le sur-ajustement.

Arbre de décision pour la régression

- Pour la Meilleur division : Trouver les régions $R_1, R_2, ..., R_K$ qui minimisent la fonction de perte :

$$\sum_{k=1}^{K} \sum_{y_i \in R_k} \left(y_i - \overline{y}_{R_k}\right)^2,$$

avec $\overline{y}_{R_k} = \frac{1}{\text{Card}(R_k)} \sum_{y_i \in R_k} y_i$.

- Au niveau de chaque nœud on cherche la variable X_j et le seuil de division s, assurant la plus forte décroissance de l'hétérogénéité des nœuds enfants, qu'on les note par R_1 (celui de gauche) et R_2 (celui de droite). i.e.

$$R_1(j,s) = \{\text{observations} | X_j \leq s\} \text{ et } R_2(j,s) = \{\text{observations} | X_j > s\}.$$

- Ainsi l'objectif est de trouver j et s qui minimisent la fonction de perte :

$$\sum_{y_i \in R_1(j,s)} \left(y_i - \overline{y}_{R_1(j,s)}\right)^2 + \sum_{y_i \in R_2(j,s)} \left(y_i - \overline{y}_{R_2(j,s)}\right)^2.$$

Arbre de décision pour la régression

- Règle d'arrêt : utiliser les paramètres "minsplit" et "minbucket" en rpart.
- Élagage : Construction d'une séquence d'arbres emboîtés en se basant sur une pénalisation de la complexité de l'arbre. i.e. pour une valeur du paramètre de complexité α, $\exists$ un sous arbre $T \subset T_{max}$ qui minimise :

$$\sum_{m=1}^{|T|} \sum_{y_i \in R_m} \left(y_i - \overline{y}_{R_m}\right)^2 + \alpha \left|T\right|,$$

α est estime selon la validation croisée par V-ensembles.

Application 3

- Application sous R.
- En utilisant les données ci dessous construisez un arbre de régression (sans l'élagage) en considérant les paramètres suivants : minsplit = 6 et minbucket = 2.

	Years	Hits	Salary
-Rey Quinones	1	68	70
-Barry Bonds	1	92	100
-Pete Incaviglia	1	135	172
-Dan Gladden	4	97	210
-Juan Samuel	4	157	640
-Joe Carter	4	200	250
-Tim Wallach	7	112	750
-Rafael Ramirez	7	119	875
-Harold Baines	7	169	950

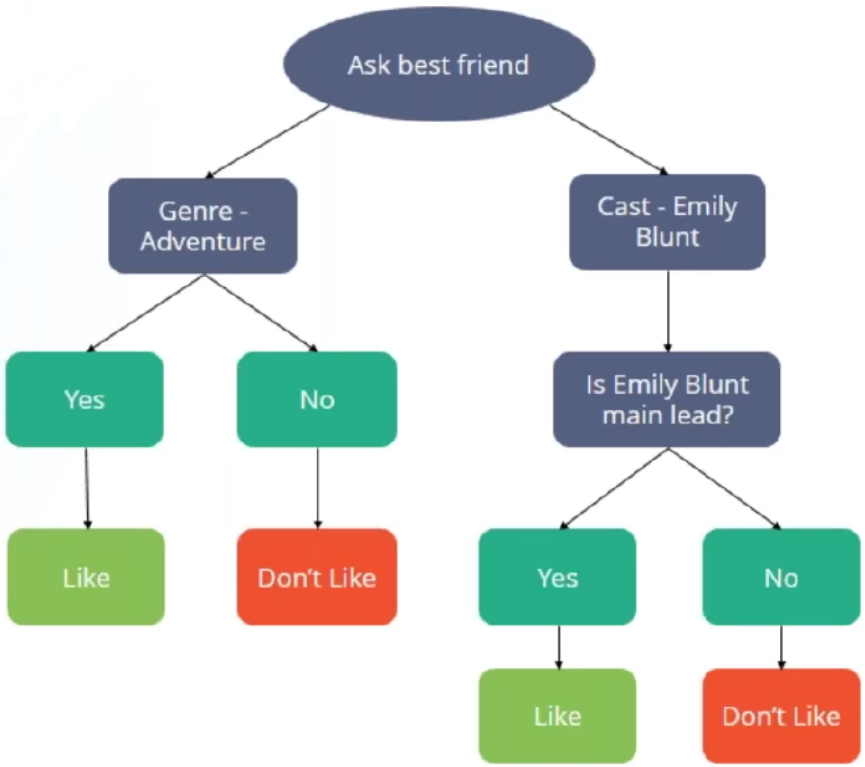
Conclusion : Arbre de décision

- Résultats simples à interpréter.
- Pas d'hypothèse de linéarité ou de distribution.
- Traitement des interactions complexes.
- Traitement des données manquantes.
- Données de grande dimension.
- Modèles instables, sensibles à des fluctuations de l'échantillon ($\Longrightarrow$ forêts aléatoires).

Forêts aléatoires

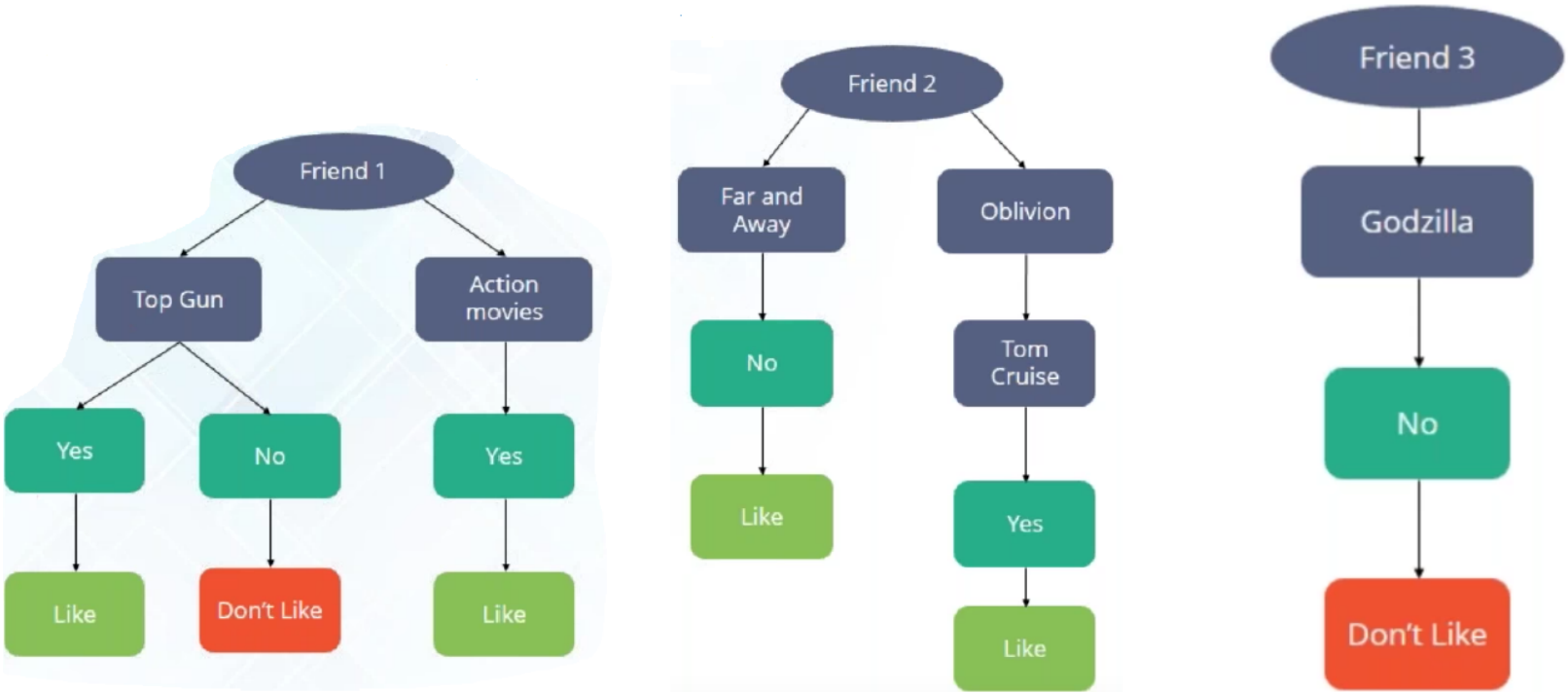
Introduction

- Vous voulez décider si vous regardez “Edge of Tomorrow” ou pas. Vous pouvez décider en demandant votre meilleur ami ou un groupe d’amis.
- Pour savoir si vous aimerez “Edge of Tomorrow” votre ami considèrera deux critères : 1) Aimez-vous les films d’aventure ? 2)Aimez-vous Emily Blunt ?
- Ainsi, un arbre de décision est créé.
- Mais si vous n’êtes pas convaincu ? ? ? ?



Introduction

- Afin d'obtenir des recommandations plus précises, vous demandez à plus d'amis.
- Vous décidez à la majorité des voix $\Longrightarrow$ Vous construisez une forêt aléatoire.



Introduction

- Random Forest, Leo Breiman (2001) : hyper-performant pour les problèmes de classification ou de régression.
- Une forêt aléatoire est l'agrégation d'une collection d'arbres de décision choisis indépendamment.
- Deux niveaux de hasard : Bootstrapping avec remise au niveau de l'échantillon d'apprentissage + choix des variables qui interviennent dans le modèle (tirage aléatoire de m variables explicatives parmi les p).

Algorithme

- Tirer aléatoirement q échantillons $\mathcal{L}^1, \ldots, \mathcal{L}^q$, aléatoire avec remise de la partie apprentissage (e.g. qui fait $2/3$ de la donnée globale).
- Estimer un arbre de décision sur chaque échantillon de $\mathcal{L}^l, l = 1, \ldots, q$, avec randomisation des variables selon l'algorithme suivant :
  - à chaque fois qu'une coupe doit être faite : 1) on tire au hasard m variables parmi les p variables explicatives 2) on choisit le meilleur découpage dans ce sous ensemble.
- Après avoir obtenu les q prédicteurs à partir des arbres construits :
  - Si la variable dépendante Y est quantitative $\Longrightarrow$ on prédit Y en agrégeant les différentes décisions par la moyenne.
  - Si la variable dépendante Y est qualitative $\Longrightarrow$ on prédit Y en agrégeant les différentes décisions trouvées par la décision majoritaire (le mode).

Algorithme

- En général on se limite à des arbres pas très profonds (contrairement à d'autres techniques où il faut des arbres profonds pour réduire leur corrélation). E.g. contrôler le nombre minimal d'observations par division/coupe.
- Même si chaque arbre (de petite taille) risque d'être moins performant, mais l'agrégation compense ce manque.
- La sélection aléatoire d'un nombre de variables explicatives à chaque étape augmente significativement la variabilité (en mettant en avant nécessairement d'autres variables) ainsi que le pouvoir prédictif du modèle.
- Compromis Biais/Variance ?

Algorithme

- Le nombre m de variables tirées aléatoirement est un paramètre sensible. En général la valeur de m dépend de la nature de la variable cible ($m_{\text{quantitative}} \geq m_{qualitative}$).
- Contrôler le nombre q d'arbres de la forêt.
- En pratique ces paramètres dépendent de la base des données, il faut mieux les estimer par validation croisée.
- Avantage : Obtenir les variables les plus importantes pour le modèle agrégé c.à.d celles qui contribuent à la construction des arbres.
- La forêt aléatoire est une boite noire $\Longrightarrow$ on ne peut pas faire une interprétation simple et directe.

Erreur OOB (Out Of Bag Error)

- Soit n le nombre d'observations dans l'ensemble d'apprentissage. Pour $i = 1, ...n$ on considère l'observation(X_i, Y_i).
- On considère l'ensemble des échantillons bootstrap ne contenant pas (X_i, Y_i) (i.e. (X_i, Y_i) est en dehors du bootstrap) ainsi que tous les arbres associés à ces échantillons.
- on prédit $\widehat{Y}_i$ en fonction de X_i en agrégeant uniquement les arbres de l'étape précédente.
- Ainsi le taux d'erreur OOB est le suivant :

$$\text{OOB}_{\text{Classification}} = \frac{1}{n}\sum_{i=1}^{n} 1_{\widehat{Y}_i \neq Y_i},$$

$$\text{OOB}_{\text{Régression}} = \frac{1}{n}\sum_{i=1}^{n} \left(\widehat{Y}_i - Y_i\right)^2.$$

Importance des variables

Afin de mesurer l'importance de chaque variable explicative, deux critères sont proposés :

- Indice de Gini ou mesure de l'entropie :
  - Étudier si la variable diminue l'impureté durant la construction au niveau de tous les arbres de la forêt (Principe d'agrégation).
  - L'addition des variations de Gini/Entropie pour chaque variable individuelle sur tous les arbres de la forêt peut être considérée comme une mesure de l'importance de la variable.
- Erreur OOB (Out Of Bag Error) :
  - On calcule l'erreur que chaque arbre commet sur son échantillon, c.à.d l'OOB associé à l'échantillon $\mathcal{L}^l$ de taille n_l (pour ce faire on considère les observations qui ne sont pas dans $\mathcal{L}^l$). On note cette erreur par $\mathcal{L}^l_{\text{OBB}}$.
  - Soit X^j , $j = 1, \ldots, p$ la j^{ieme} variable, dans tous les échantillons $\mathcal{L}^l$, $l = 1, \ldots, q$. On permute aléatoirement les valeurs de X^j selon une permutation Φ.
  - On recalcule $\mathcal{L}^l_{\text{OBB}}$ que chaque arbre commet sur son échantillon $\mathcal{L}^l$ permuté.
  - L'importance de X^j est définie comme l'augmentation moyenne de l'erreur d'un arbre après permutation de X^j.

Conclusions : Forêts aléatoires

Avantages :

- Bons résultats en grande dimension.
- Sur-apprentissage limité suite à l'utilisation de l'erreur OOB.
- Simples à mettre en œuvre (R/Python).
- Peu de paramètres.

Inconvénients :

- C'est une boite noire : difficilement interprétable.
- L'entrainement est plus lent car à chaque arbre crée on cherche les variables qui donnent la meilleure segmentation.

Pourquoi deux étapes de randomnisation ?

Étude de la variance

- L'estimation par la moyenne à partir des q échantillons $\mathcal{L}^1, \ldots, \mathcal{L}^q$ (par arbre de décision classique sans randomnisation au niveau du choix de m variables indépendantes parmi p), aide à réduire la variance :
  En effet, si $Y_1, Y_2, \ldots, Y_q$ sont q *v.a. i.i.d.* de moyenne μ et de variance σ^2, alors :

$$\frac{1}{q}(Y_1 + Y_2 + \ldots + Y_q) \text{ est de variance } \frac{\sigma^2}{q}.$$

- Les estimateurs ne sont pas en réalité indépendants.
- Les échantillons sont pratiquement corrélés.

Étude de la variance

- En effet, si $Y_1, Y_2, \ldots, Y_q$ sont q *v.a. i.d.* (mais pas indépendantes) de moyenne μ et de variance σ^2 et corrélation $\rho = \text{Corr}\,(Y_i, Y_j)$ pour $i \neq j$, alors :

$$\mathbb{V}\left[\frac{1}{q}\left(Y_1 + Y_2 + \ldots + Y_q\right)\right] = \rho\sigma^2 + \frac{1-\rho}{q}\sigma^2.$$

- Quand q est grand le second terme est négligeable mais le premier non.
- L'idée des forêts aléatoires est de baisser la corrélation entre les estimateurs à l'aide d'une étape supplémentaire de randomisation.
- **BAGGING$\Longrightarrow$RANDOM FOREST**.

Contrôler le choix des échantillons et des modèles

- Chaque modèle cherche à corriger les faiblesses du précédent.
- Utiliser un ou plusieurs modèles de classification.
- Le classifieur final est une combinaison linéaire des classifieurs construits au fur et à mesure.
- Réduction du biais.
- **RANDOM FOREST** $\Longrightarrow$ **BOOSTING**.