UP2 : Apprentissage Statistique et Analyse de données

Arbre de décision et Forêt aléatoire

ANIS S. HOAYEK

Octobre 2025

Plan du cours

1. Apprentissage statistique pour l'aide à la décision.
2. Méthodes de Classification : Principes généraux
3. Arbres de décision.
4. Validité : Sensibilité, Spécificité, ROC, AUC, LIFT, etc.
5. Forêts aléatoires.
6. Lien avec Bagging et Boosting.
7. Forêts aléatoires non supervisées (Isolation Forest).
8. Applications classiques TD (à la main) + TP (sous R et/ou Python).
9. Évaluation : Examen écrit (13 Novembre 2025) + TP noté (compte rendu par groupes selon la liste prédéfinie).

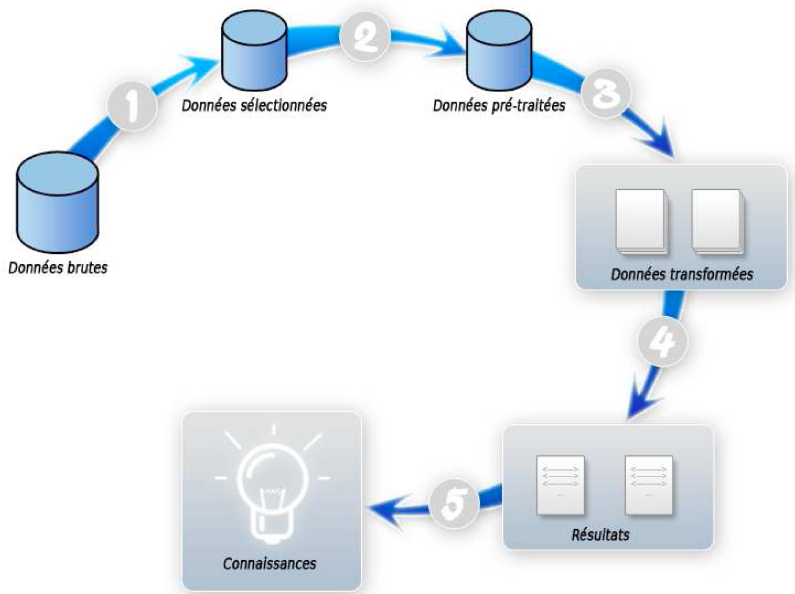
Apprentissage statistique pour l'aide à la décision

Introduction

1. La statistique est incontournable dans les sciences expérimentales : données dans le but spécifique de confirmer ou d'infirmer des hypothèses scientifiques.
2. Données existantes avant même qu'on se pose des questions spécifiques.
3. Apprendre de ces données les réponses à des questions $\Longrightarrow$ La statistique permet de faire cet apprentissage $\Longrightarrow$ prendre des décisions.
4. Méthodes de la statistique dans un contexte expérimental :
    1. Planification du recueil des données (Échantillonnage, Plan d'expérience, Data management)
    2. Description et présentation des données (Statistique descriptive)
    3. Induction de règles générales à partir de ce qui a été observé (Modélisation et Inférence statistique)

Définition.

Le data-mining, est un processus d'extraction des connaissances qui comprend plusieurs étapes :
Pré-traitement des données $\Longrightarrow$ Fouille des données pré-traitées $\Longrightarrow$ Apprentissage à partir de ce qui a été observé



Domaines d'application

Entreprises stockent dans des entrepôts de données (data warehouse) des des picas octets de données relatives à leurs activités à des fins de :

1. Gestion des stocks, des services, des ressources humaines, des clients, etc.
2. Contrôle de qualité : suivi en ligne des paramètres de production, traçabilité, etc.
3. Secteur bancaire : Identification du risque de crédit d'un client en fonction de sa probabilité de défaut de paiement.
4. Médecine : Identification des patients à risque et les tendances de la maladie.
5. Marketing : Identification du taux de désabonnement des clients.

Remarque.

Les données n'ont pas été recueillies à ces fins !

Étape 1 : Nettoyage et mise en forme des données

1. S'assurer de la fiabilité des données,
2. Éliminer les sujets atypiques ou non pertinents pour l'analyse
3. Obtenir un "feeling" du jeu de données :
    1. Analyses descriptives unidimensionnelles adaptées à la nature des variables.
    2. Analyses statistiques bivariées adaptées à la nature des couples de variables pour faire apparaître les liens entre elles.
    3. Analyses statistiques multivariées.

BUT : Repérer les variables ayant des distributions statistiques "bizarres". Le cas échéant, décider s'il convient d'éliminer ou de transformer les données.

Étape 2 : Analyse exploratoire des données

1. Réduire la dimension du vecteur de variables
2. Éliminer les variables redondantes

Via des techniques statistiques qui ne sont pas l'objet de ce cours : ACP, AFC, AFD, etc.

Étape 3 : Classification/Régression

1. Une variable d'intérêt (ou cible) Y qualitative/discrète ou quantitative, définissant des groupes de la population
2. Une ou plusieurs variables explicatives ou prédictives $X_1, \ldots, X_p$.
3. Sur un individu, on va observer uniquement la valeur $(x_1, \ldots, x_p)$ de $(X_1, \ldots, X_p)$.
4. On voudra lui assigner sa valeur y de Y (son groupe ou sa valeur) en minimisant les risques d'erreur.
5. Pour établir le classifieur qui fera ce travail, on dispose de données du type $(x_1, \ldots, x_p, y) \Longrightarrow$ On veut à partir de ces données établir la meilleure règle de classification/régression possible pour classer par la suite les éventuels prospects.

Étape 3 bis : Clustering

1. On dispose des valeurs observées $(x_1, \ldots, x_p)$ sur n individus des variables $(X_1, \ldots, X_p)$.
2. Regrouper, en K groupes les plus dissemblables entre eux possibles, les individus de façon à ce qu'à l'intérieur de chaque groupe, les données soient le plus homogène possible.
    1. K connu : Clustering supervisée
    2. K inconnu : Clustering non supervisée
3. Analyser les particularités de chacun des K groupes

Remarque.

Le clustering revient à "créer" une variable qualitative/discrète indiquant les groupes auxquels peuvent appartenir les individus.

Étape 3 bis : Modélisation

1. La modélisation a pour objectif d'étudier les liens plus fins entre :
   1. Une (ou plusieurs) variables d'intérêt (ou cible) Y
   2. Une ou plusieurs variables explicatives (ou prédictives) $X_1, \ldots, X_p$
2. Elle se fait notamment via les méthodes de :
   1. Régression linéaire simple, multiple, non-linéaire : Y et $X_1, \ldots, X_p$ quantitatives/continues
   2. ANOVA, ANCOVA : Y quantitative/continue, au moins une des $X_1, \ldots, X_p$ qualitative/discrète
   3. Régression logistique : Y qualitative/discrète et $X_1, \ldots, X_p$ quantitatives/continues
   4. Régression log-linéaire : Y et $X_1, \ldots, X_p$ qualitatives/discrètes
   5. Autres méthodes : séries chronologiques etc...

Remarque.

Dans ce cours, on se concentre sur les méthodes d'apprentissage que sont la classification, et la régression.

Stratégies d'une application classique du data-mining

1. Extraction des données de l'entrepôt, via éventuellement un sondage,
2. Nettoyage : analyse graphique, validation des codages etc..
3. Exploration : réduction de la dimension, élimination de variables redondantes,
4. Analyse (classification, régression, clustering, modélisation)
5. Exploitation du modèle et diffusion des résultats.

Pièges du data-mining

1. Attaque du problème sans but précis.
2. Data-snooping : “If you torture the data long enough, it will confess to anything” (Ronald Coase)
3. Incompréhension des algorithmes utilisés par les logiciels.
4. Confiance aveugle dans les sorties des logiciels informatiques de plus en plus ”tout intégré” et de plus en plus ”boîte noire”.
5. Absence d’interrogation et d’esprit critique dans le choix des ”valeurs par défaut” fait par les logiciels.

Méthodes de Classification : Principes généraux

Principes généraux

Problème : Déterminer à quel groupe $\mathcal{G}_1, ..., \mathcal{G}_K$ appartient un individu :

- L'appartenance à un groupe est déterminée par la valeur y d'une v.a. $Y \in \{1, 2, ..., K\}$.
- Si $y = k$, alors l'individu $\in \mathcal{G}_k$.

Exemple : Un client demande un prêt à une banque. La banque veut savoir si le client est en mesure de rembourser son prêt ($y = 1$) ou non ($y = 2$). Évidemment, l'observation de y est impossible, sauf à attendre l'issue du prêt.

Idée : Mesurer sur l'individu un certain nombre de variables $X = (X_1, \ldots, X_p)$, (e.g. : ancienneté, compte courant, compte sur livret, assurance-vie, etc...) dites discriminantes, classifiantes ou de "procuration", et déterminer à partir de la valeur observée $x = (x_1, \ldots, x_p)$ de X si l'individu fait (fera) parti de $\mathcal{G}_1$ ($y = 1$), $\mathcal{G}_2$ ($y = 2$), ..., $\mathcal{G}_K$ ($y = K$).

Postulat de base : X est un vecteur aléatoire. Il a donc une loi de probabilité (notée $\mathcal{L}_k$) qui diffère selon la valeur de Y (c-à-d : si X provient de $\mathcal{G}_k$, $X \sim \mathcal{L}_k$)

Définition.

Un classifieur est une fonction $d(\cdot) : \mathbb{R}^p \longrightarrow \{1, 2, ..., K\}$ qui, à chaque $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$, associe un et un seul $k \in \{1, 2, ..., K\}$.
La classification se fait comme suit : Si un individu a x pour valeur de X, on l'assigne à $\mathcal{G}_K$ si $d(x) = k$.

Comment déterminer $d(\cdot)$? On aura besoin :

1. D'une méthode pour construire $d(\cdot)$
2. D'information sur le contexte du problème ($\mathcal{L}_k$, probabilités a priori d'appartenance aux groupes)
   et/ou
3. D'un échantillon $\mathcal{E}$ composé de valeurs observées $(x_i, y_i), i = 1, \ldots, n$.

Difficulté : $\exists$ une infinité de façons de choisir $d(\cdot)$. Quels critères utiliser pour ce choix ?

Information disponible pour construire un classifieur

La construction d'un classifieur dépend de l'information dont on dispose.

1. Si on connaît parfaitement la loi $\mathcal{L}_k$ de X (de densité $f_k(\cdot)$) quand $y = k$, on peut utiliser cette loi pour construire $d(\cdot)$ (cas d'école car il n'utilise pas $\mathcal{E}$)
2. Si on connaît imparfaitement cette densité (i.e. à des paramètres près), on peut utiliser $\mathcal{E}$ pour estimer ces paramètres, puis fonctionner comme en 1) (classifieur du Maximum de Vraisemblance avec paramètres estimés)
3. Si on n'a aucune information sur les différentes lois $\mathcal{L}_k$, on doit utiliser $\mathcal{E}$ pour construire un classifieur (e.g. arbre de décision). C'est le cas le plus réaliste.

Information disponible pour construire un classifieur

À ces informations peut s'ajouter :

- Une information a priori sur les "chances" qu'un individu $\in \mathcal{G}_k$. Cette information prend la forme de probabilités a priori π_k, $k = 1, \ldots, K$ avec $0 < \pi_k < 1$ et $\pi_1 + \ldots + \pi_K = 1$
- Une information sur les coûts d'une mauvaise classification $C(k', k)$ (par. ex coût de consentir un prêt à un client mauvais payeur)

Remarque.

Nous allons nous concentrer sur le cas 3) qui est celui rencontré en pratique.

Évaluation d'un Classifieur : Probabilités d'assignation correcte et incorrecte

- L'individu sera assigné à $\mathcal{G}_k$ selon la valeur observée x de X.
- Cette assignation est correcte si l'individu provient bien de $\mathcal{G}_k$. Sinon, l'assignation est incorrecte.
- Comme cette assignation se fait à partir de la valeur observée x de X, qui est aléatoire, il existe donc des probabilités de bonne et mauvaise assignation.

Définition.

La probabilité d'assigner un individu à $\mathcal{G}_{k'}$ alors qu'il est issu du groupe $\mathcal{G}_k$ est définie par

$$\rho_{k'k} = \mathbb{P}\left[X \longmapsto \mathcal{G}_{k'} \mid X \in \mathcal{G}_k \text{ ou } X \in \mathcal{L}_k\right] = \int_{\mathcal{G}_{k'}} f_k(x)\,\mathrm{d}x,$$

où $f_k(\cdot)$ est la densité de X (donc de $\mathcal{L}_k$) quand l'individu est issu de $\mathcal{G}_k$ (note : premier indice = là où il est envoyé ; deuxième indice, d'où il provient)

Évaluation d'un Classifieur : Probabilités d'assignation correcte et incorrecte

- Si les densités $f_k(\cdot)$ des lois $\mathcal{L}_k$ sont connues, on peut en principe calculer les $\rho_{k'k}$ (mais ce n'est pas nécessairement facile).
- ρ_{kk} $(k = 1, \ldots, K)$ sont les probabilités d'assignation correcte d'un individu provenant de $\mathcal{G}_k$.
- $1 - \rho_{kk}$ $(k = 1, \ldots, K)$ est la probabilité d'assigner de façon incorrecte un individu provenant de $\mathcal{G}_k$.
- Ces quantités peuvent être regroupées dans une matrice dite matrice de confusion dont la somme de chaque colonne $= 1$.

Mat. Confusion		vérité			
		$\mathcal{G}_1$	$\mathcal{G}_2$	$\cdots$	$\mathcal{G}_K$
prédiction	$\mathcal{G}_1$	ρ_{11}	ρ_{12}	$\cdots$	ρ_{1K}
	$\mathcal{G}_2$	ρ_{21}	ρ_{22}	$\cdots$	ρ_{2K}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	$\mathcal{G}_K$	ρ_{K1}	ρ_{K2}	$\cdots$	ρ_{KK}
	total	1	1	...	1

Comparaison de deux classifieurs : Admissibilité et Probabilité globale d'assignation incorrecte

Définition.

Soit d un classifieur avec matrice de confusion $\{\rho_{k'k}, k', k = 1, ..., K\}$ et $\widetilde{d}$ un second classifieur avec matrice de confusion $\{\widetilde{\rho}_{k'k}, k', k = 1, ..., K\}$.
On dit que d est aussi bon que $\widetilde{d}$ si $\rho_{kk} \geq \widetilde{\rho}_{kk} \, \forall k$. On dit que d est meilleur que $\widetilde{d}$ si $\rho_{kk} > \widetilde{\rho}_{kk}$ pour au moins un k.
Si d est un classifieur pour lequel il n'existe pas de meilleur classifieur, alors d est dit admissible.

Comparaison de deux classifieurs : Admissibilité et Probabilité globale d'assignation incorrecte

- Dans la définition précédente, les classifieurs sont jugés en fonction des éléments diagonaux de la matrice de confusion. Ce n'est pas la seule possibilité.
- Supposons que l'on dispose de l'information supplémentaire suivante :
  - On sait qu'a priori l'individu a une probabilité π_k ($\in]0,1[$) d'appartenir à $\mathcal{G}_k$.
  - La probabilité globale (ou a posteriori) d'assignation correcte d'un classifieur d avec matrice de confusion $\{\rho_{k'k}\}$ est définie par :

$$\begin{aligned}\mathbb{P}\left[\text{Individu} \longmapsto \text{à son groupe}\right] &= \sum_{k=1}^{K} \mathbb{P}\left[\text{Individu} \longmapsto \mathcal{G}_k \mid \text{Individu} \in \mathcal{G}_k\right] \mathbb{P}\left[\text{Individu} \in \mathcal{G}_k\right], \\ &= \sum_{k=1}^{K} \rho_{kk}\pi_k .\end{aligned}$$

Comparaison de deux classifieurs : Admissibilité et Probabilité globale d'assignation incorrecte

La probabilité globale d'assignation incorrecte de d est ainsi définie par :

$$\rho(d) = 1 - \sum_{k=1}^{K} \rho_{kk}\pi_k = \sum_{k=1}^{K}\sum_{k' \neq k} \rho_{k'k}\pi_k$$

Définition.

Supposons données les probabilités a priori π_k ($\in]0,1[$) d'appartenir à $\mathcal{G}_k$. Soit d un classifieur avec matrice de confusion $\{\rho_{k'k}, k', k = 1, ..., K\}$ et probabilité globale d'assignation incorrecte $\rho(d)$. Soit $\widetilde{d}$ un second classifieur avec matrice de confusion $\{\widetilde{\rho}_{k'k}, k', k = 1, ..., K\}$ et probabilité globale d'assignation incorrecte $\rho\left(\widetilde{d}\right)$. On dit que d est aussi bon que $\widetilde{d}$ si $\rho(d) = \rho\left(\widetilde{d}\right)$. On dit que d est meilleur que $\widetilde{d}$ si $\rho(d) < \rho\left(\widetilde{d}\right)$.

Note : En pratique, quand $\rho(d) < 0.2$, on considère que le classifieur est "bon".

Estimation des probabilités de bonne et mauvaise assignation

De façon générale, avec K groupes on peut estimer les $\rho_{k'k}$ par

$$p_{k'k} = \frac{n_{k'k}}{n_k}$$

où $n_{k'k}$ = nombre d'observations issues de $\mathcal{G}_k$ et assignées à $\mathcal{G}_{k'}$ et n_k = nombre d'observations de $\mathcal{E}$ provenant de $\mathcal{G}_k$. L'estimateur ainsi obtenu est appelé l'estimateur par resubstitution. Le tableau des $n_{k'k}$ est appelé la matrice d'incidence par resubstitution.

Mat. Incidence		vérité $\mathcal{G}_1$	$\mathcal{G}_2$	...	$\mathcal{G}_K$	total
prédiction	$\mathcal{G}_1$	n_{11}	n_{12}	...	n_{1K}	n_{1+}
	$\mathcal{G}_2$	n_{21}	n_{22}	...	n_{2K}	n_{2+}
	⋮	⋮	⋮	⋱	⋮	⋮
	$\mathcal{G}_K$	n_{K1}	n_{K2}	...	n_{KK}	n_{K+}
	total	n_1	n_2	...	n_K	n

Estimation des probabilités de bonne et mauvaise assignation

Le tableau des $p_{k'k}$ est la matrice de confusion estimée par resubstitution :

Mat. Conf. Est.		vérité			
		$\mathcal{G}_1$	$\mathcal{G}_2$	...	$\mathcal{G}_K$
prédiction	$\mathcal{G}_1$	p_{11}	p_{12}	...	p_{1K}
	$\mathcal{G}_2$	p_{21}	p_{22}	...	p_{2K}
	⋮	⋮	⋮	⋱	⋮
	$\mathcal{G}_K$	p_{K1}	p_{K2}	...	p_{KK}
	total	1	1	...	1

Ces estimateurs sont en général trop optimistes car **les mêmes données servent à la fois à construire le classifieur et à estimer ces probabilités.**

Estimation des probabilités de bonne et mauvaise assignation

Il existe plusieurs façons de contourner ce problème.
Nous en évoquons une ici rapidement. Elle consiste à partitionner l'échantillon $\mathcal{E}$ en deux sous-échantillons : $\mathcal{E}_{\text{app}}$ et $\mathcal{E}_{\text{est}}$
$\mathcal{E}_{\text{app}}$ = échantillon d'apprentissage (avec n_{app} données) sur lequel on construit le classifieur d_{app}.
$\mathcal{E}_{\text{est}}$ = échantillon d'estimation (avec n_{est} données) qu'on "passe" dans le classifieur pour obtenir les estimateurs :

- $p_{k'k}^{\text{est}}$ de $\rho_{k'k}$ pour le classifieur d
- $R^{\text{est}}(d)$, l'estimateur de $\rho(d) = 1 - \sum_{k=1}^{K} \rho_{kk}\pi_k$ (si les π_k sont disponibles).

Estimation des probabilités de bonne et mauvaise assignation

- Ces dernières quantités estiment en fait les $\rho_{k'k}$ de d_{app} et $\rho(d_{\text{app}})$ et non pas $\rho_{k'k}$ de d et $\rho(d)$
- Si n est grand, n_{app} et n_{est} le seront aussi. Ainsi $d \simeq d_{\text{app}} \Longrightarrow p^{\text{est}}_{k'k} \approx \rho_{k'k}$ de d et $R^{\text{est}}(d) \approx \rho(d)$.
- Ainsi, cette méthode est bien adaptée au cas où n est grand. On appelle cette méthode la validation externe ou encore "**out-of sample validation**". Elle dépend cependant du choix (aléatoire) de $\mathcal{E}_{\text{est}}$. Si n est grand, les $p^{\text{est}}_{k'k}$ est et $R^{\text{est}}(d)$ ne devraient pas trop varier d'un $\mathcal{E}_{\text{est}}$ à l'autre.

Estimation des probabilités de bonne et mauvaise assignation

- Si n est petit, une variante est la validation croisée ou interne (leave one out cross validation)
- L'idée consiste à prendre $\mathcal{E}_{\text{est}} = \{x_1\}$ et $\mathcal{E}_{\text{app}} = \mathcal{E} \setminus \{x_1\}$. On calcule le classifieur sur $\mathcal{E}_{\text{app}}$; celui-ci ne va pas différer de beaucoup de celui obtenu de $\mathcal{E}$.
- On regarde ensuite si l'individu de $\mathcal{E}_{\text{est}}$ est bien classé.
- On répète successivement cette procédure avec chacune des données. Au final, on estime les $\rho_{k'k}$ par le nombre de fois qu'une observation de $\mathcal{G}_k$ est classée en $\mathcal{G}_{k'}$.
- Des variantes utilisent des $\mathcal{E}_{\text{est}}$ de taille > 1 (**V-fold cross validation**).
- Le prix est un temps de calcul plus long.

Arbres de Décision

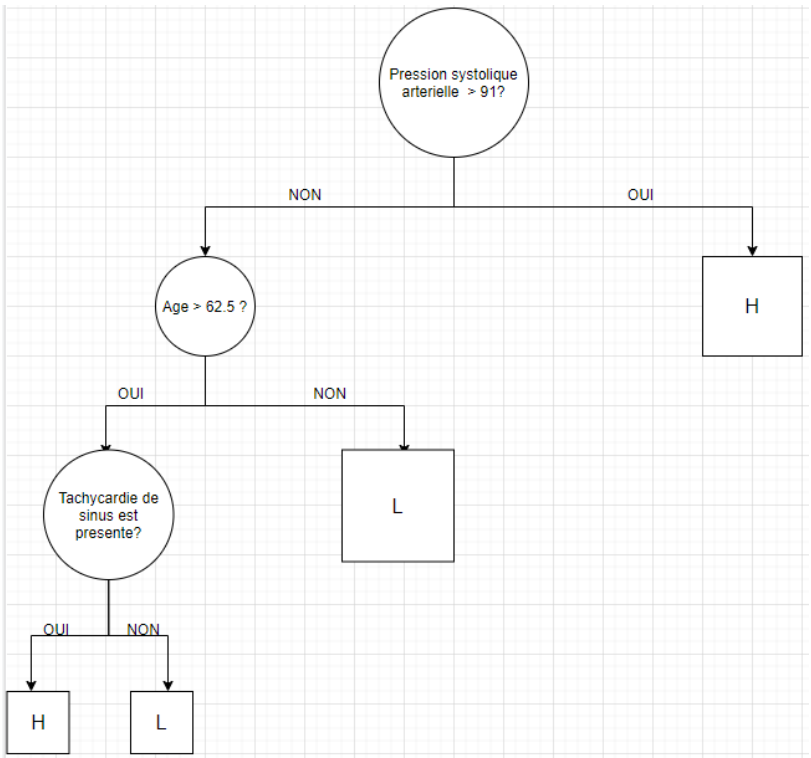
Fait.

Référence "classique" : Breiman, L. Friedman, J. H. Olshen, R.A., Stone, C. (1984) : Classification and regression trees. Wadsworth International Group. Belmont, California.

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

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).



- 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 :
  - Un arbre de classification se construit par segmentations récursives de l'échantillon $\mathcal{E} = \{(x_i, y_i), i = 1, ..., n\}$.
  - 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 (convertir le nœud en feuille) ou continuer de le scinder à nouveau. (L'idée principale est de choisir chaque split de façon à ce que les deux nœuds descendants soient chacun plus **"purs"** que le nœud parent).
- 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.

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 :

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

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

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

Remarque.

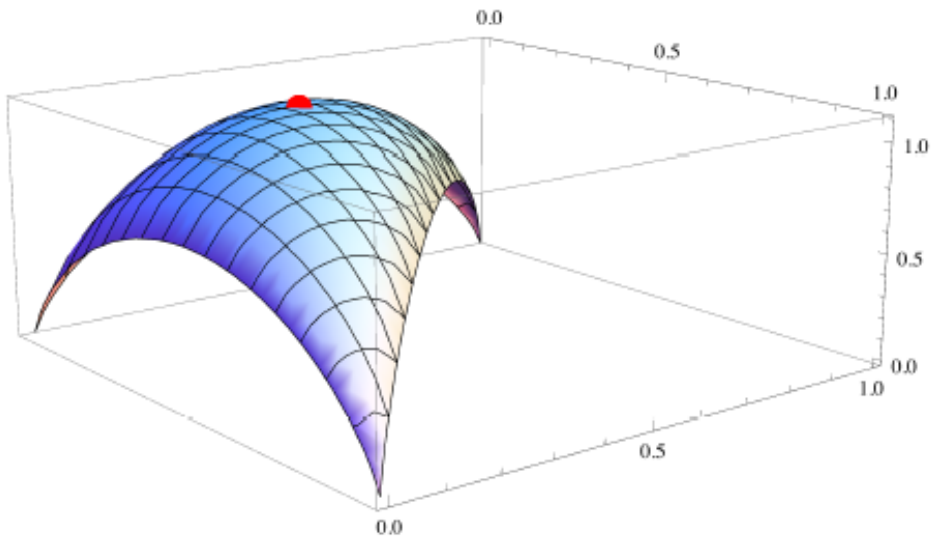
$Imp(t)$ est maximale quand toutes les classes sont mélangées avec des "parts égales" (Distribution uniforme/équiprobable) et est minimale quand le nœud ne contient qu'une seule classe (certitude totale).

Remarque.

Comme $p(1|t) + p(2|t) + \ldots + p(K|t) = 1$, on a

$$\begin{aligned} Imp(t) &= \phi(p(t)), \\ &= \phi(p(1|t), p(2|t), \ldots, p(K-1|t), \\ &\quad 1 - p(1|t) - p(2|t) - \ldots - p(K-1|t)) \end{aligned}$$

Ce qui montre que $Imp(t)$ est en fait une fonction de $K-1$ variables. Quand $K = 3$, on peut représenter une telle fonction et un exemple est : (le • indique le maximum de la fonction, atteint en $(1/3, 1/3)$) :



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 :

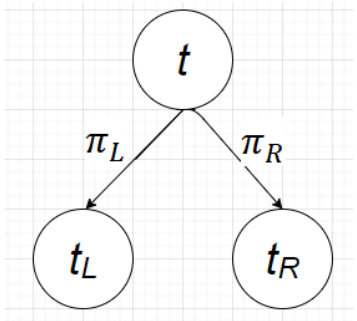
$$Imp(t) = \mathcal{G}(t) = 1 - \sum_{k=1}^{K} p^2(k|t).$$

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 $\Phi(\mathcal{S}, t)$. On a :
$\Phi(\mathcal{S}, t) \in [0, Imp(t)]$,

- Si $\Phi(\mathcal{S}, t) = 0$, le split n'a pas diminué l'impureté.
- Si $\Phi(\mathcal{S}, t) = Imp(t)$, alors $Imp(t_L) = Imp(t_R) = 0$ et les 2 nœuds descendants sont purs : le split a parfaitement séparé les groupes de t.

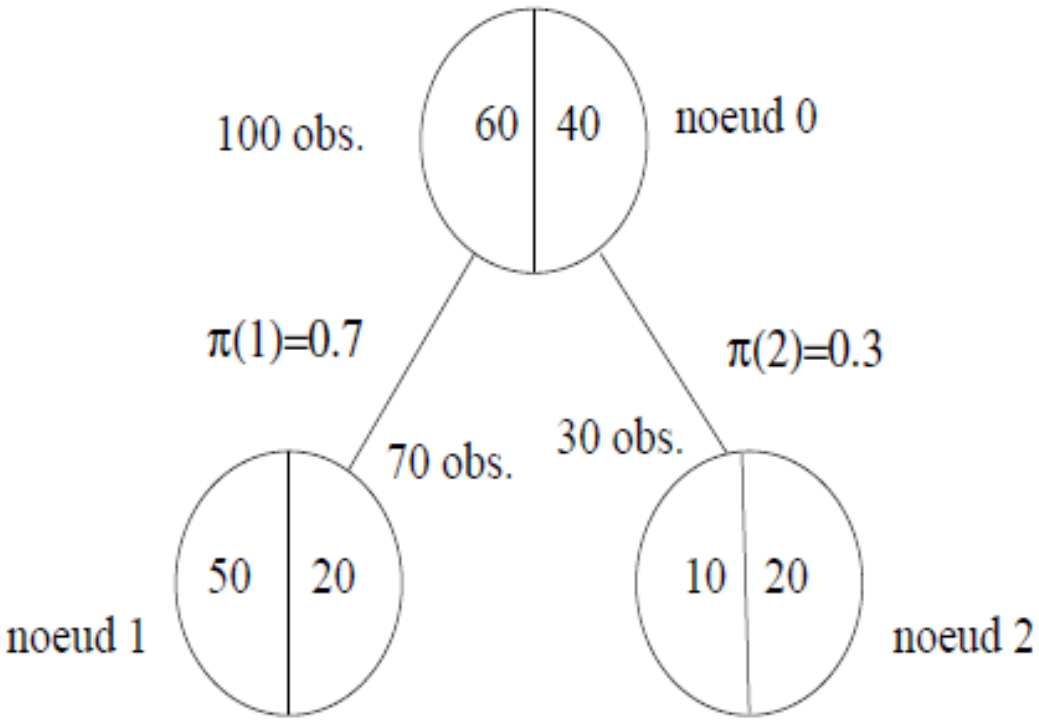
Définition 1.

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

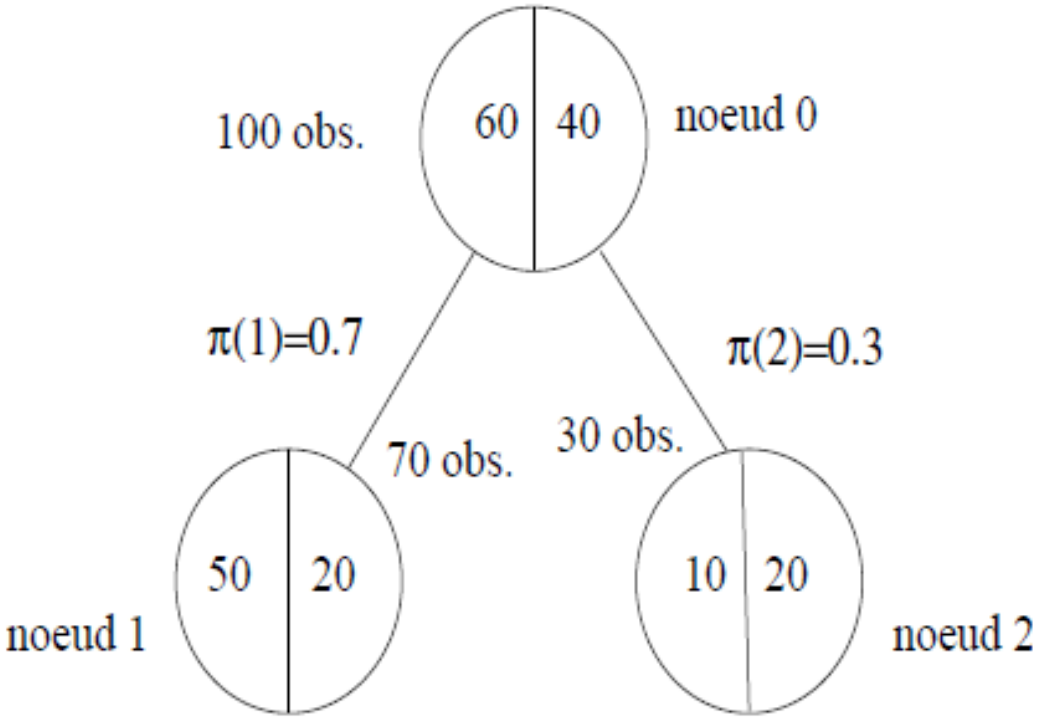
$$Imp(T) = \sum_{t \in \widetilde{T}} \pi(t)\, Imp(t),$$

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

Exemple Numérique



Exemple Numérique



$$Imp\left(t_0\right) = -{}^{60}/_{100}\log_2\left({}^{60}/_{100}\right) - {}^{40}/_{100}\log_2\left({}^{40}/_{100}\right) = 0.971,$$
$$Imp\left(t_1\right) = -{}^{50}/_{70}\log_2\left({}^{50}/_{70}\right) - {}^{20}/_{70}\log_2\left({}^{20}/_{70}\right) = 0.863,$$
$$Imp\left(t_2\right) = -{}^{10}/_{30}\log_2\left({}^{10}/_{30}\right) - {}^{20}/_{30}\log_2\left({}^{20}/_{30}\right) = 0.918,$$
$$\Delta Imp\left(t_0\right) = Imp\left(t_0\right) - 0.7\,Imp\left(t_1\right) - 0.3\,Imp\left(t_2\right) = 0.091.$$

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 :

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

- 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. (i.e. Quand $\Phi(\mathcal{S}^*, t)$ <seuil (on n'a quasiment rien gagné en splittant le nœud t).
- Profondeur de l'arbre est supérieure à une valeur prédéfinie.
- Nombre des observations est inférieur à une valeur prédéfinie.
- On rappelle que la construction de l'arbre se termine lorsque tous les nœuds terminaux ont été attribués à une classe.

Règles d'arrêt et d'attribution

On attribue la classe :

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

à un nœud terminal.
Supposons que $Y(t) = k$

- Le nombre d'individus appartenant au groupe k dans la feuille t assignée à $\mathcal{G}_{k'}$ $(k' \neq k)$ sera noté $n_{k'k}(t)$.
- Le nombre total d'individus appartenant au groupe k assignée à $\mathcal{G}_{k'}$ $(k' \neq k)$ par l'arbre sera noté $n_{k'k}$.

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 au lieu de règles d'arrêt.

Définition 2.

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 3.

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 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 :

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

Remarque.

En général, $R(T)$ est un mauvais estimateur de $\rho(T)$ car trop optimiste. Pour le voir, imaginons qu'on laisse pousser l'arbre jusqu'à ce que chaque feuille ne contienne d'une seule observation. Alors $n_{kk} = 1$ pour toutes les feuilles, de sorte que $R(T) = 0$.

Élagage et coût de complexité

Il existe plusieurs façons de contourner ce problème. Nous en évoquons une ici rapidement. Elle consiste à partitionner l'échantillon $\mathcal{E}$ en deux : $\mathcal{E}_{app}$ et $\mathcal{E}_{est}$.

- $\mathcal{E}_{app}$ = échantillon d'apprentissage (avec n_{app} données) sur lequel on construit l'arbre T_{app}.
- $\mathcal{E}_{est}$ = échantillon d'estimation (avec n_{est} données) qu'on passe dans l'arbre pour obtenir les estimateurs $p_{k'k}^{est}$ est et $R^{est}(T)$.

Noter que ces dernières quantités estiment les $\rho_{k'k}$ de T_{app} et $\rho(T_{app})$ et non pas $\rho_{k'k}$ de T et $\rho(T)$.
Cependant, si n est grand, n_{app} et n_{est} le seront aussi. Ainsi $T \simeq T_{app} \Longrightarrow p_{k'k}^{est} \approx \rho_{k'k}$ de T et $R^{est}(T) \approx \rho(T)$.
A tout le moins, le problème évoqué plus haut ($R(T) = 0$) disparaît. On appelle cette méthode la validation externe ou encore out-of sample.

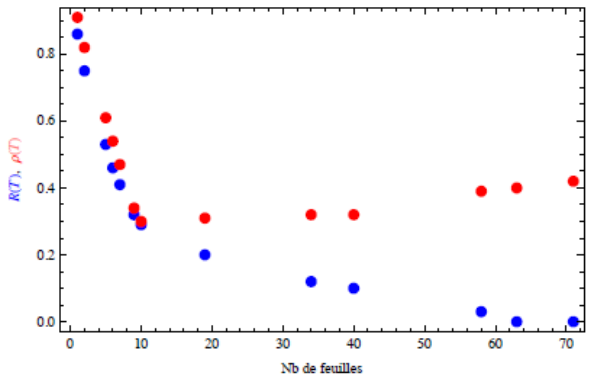
Élagage et coût de complexité

Dans la suite on va appeler T_{max} l'arbre maximal et T_0 l'arbre minimal formée juste de la racine :

- Entre T_0 et T_{max}, existe-il des arbres T qui donnent d'aussi bons résultats, voire même de meilleurs (en terme de $\rho(T) =$ probabilité globale d'assignation incorrecte) que T_{max} tout en étant plus simples (moins de niveau) ?
- Si oui, l'utilisation de cet arbre s'imposerait tant pour sa taille que son $\rho(T)$.
- On veut donc déterminer si, entre T_0 et T_{max}, il existe un arbre T tel que $\rho(T) \leq \rho(T_{max})$.

Élagage et coût de complexité

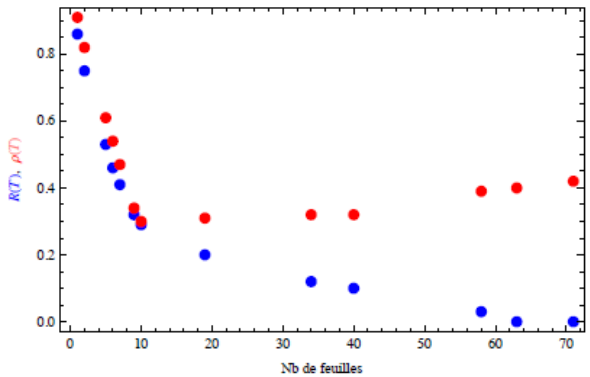
L'exemple (artificiel) suivant montre la situation typique dans le cas d'un ensemble d'arbres se situant "entre" T_0 et T_{max} de nombre différent de feuilles :



- Il y a bien un arbre qui minimise les valeurs de $\rho(T)$ et cet arbre, appelons le T_{opt}, est beaucoup plus petit que T_{max}.
- Quand T est "proche" de T_{max}, $R(T) \simeq 0$ et on a faussement l'impression que l'arbre est excellent en terme de probabilité globale d'assignation incorrecte (i.e. excellent pouvoir de classification).
- Quand T est "proche" de T_0, $R(T)$ est proche de $\rho(T)$.

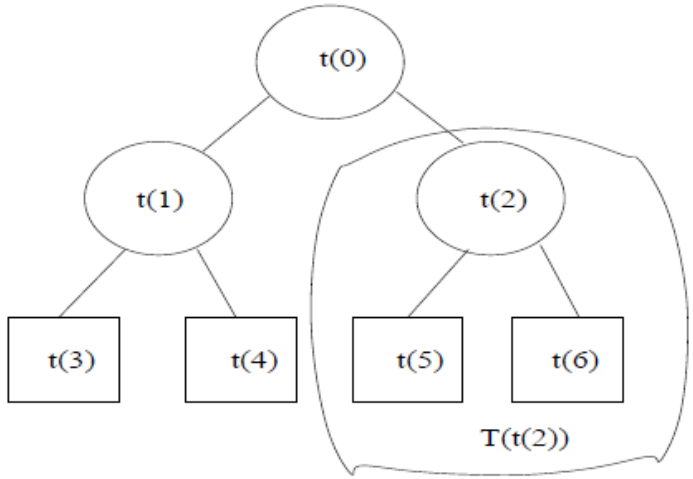
Élagage et coût de complexité

L'exemple (artificiel) suivant montre la situation typique dans le cas d'un ensemble d'arbres se situant "entre" T_0 et T_{max} de nombre différent de feuilles :



- Si on connaissait les $\rho(T)$, on pourrait déterminer ce T_{opt}. Mais mettre la main sur les $\rho(T)$ est une cause perdue.
- En solution, construire un estimateur "honnête" (c-à-d sans biais) de $\rho(T)$, notons-le $R^{hon}(T)$,
- Les points $R^{hon}(T)$ forment un graphique semblable à celui des $\rho(T)$,
- On pourrait utiliser ces $R^{hon}(T)$ pour déterminer un estimateur $\widehat{T}_{opt}$ de T_{opt}.

Élagage et coût de complexité



- On note par $T(t)$ le sous-arbre avec la racine en t.
- Élaguer d'un arbre T sa branche $T(t)$ consiste à remplacer la branche $T(t)$ par un nœud terminal à t. L'arbre élagué (qui peut-être noté $T - T(t)$) est appelé un sous-arbre de T.
- Il est naturel d'imposer que la suite d'arbres qu'on cherche à créer soit une suite de sous-arbres les uns des autres. Plus précisément, on se restreint au cas où $T_0 << T_1 << T_2 << \cdots << T_{max}$.
- À la fin du processus, on pourra repasser dans la suite de sous-arbres pour en déterminer le T_{opt} (ou plutôt $\widehat{T}_{opt}$)

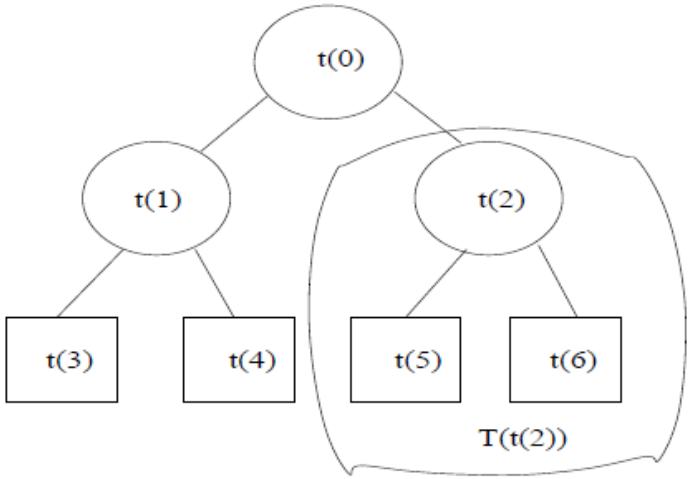
Remarque.

Soit $T_k << T_{k+1}$ alors $R(T_{k+1}) \leq R(T_k)$.
$\Longrightarrow$ L'utilisation de $R(T)$ pour déterminer T_{opt} ne peut fonctionner : plus on splitte, mieux on croit, à tort, bien faire. Il faut donc "corriger" la dérive de $R(T)$ vers 0. Une façon classique est de lui ajouter un terme pénalisant la complexité (ç-à-d. le nombre de feuilles/nœuds terminaux de l'arbre).

Si on dispose d'un estimateur "honnête" (e.g. sans biais), disons $R^{hon}(T)$ de $\rho(T)$, alors il est raisonnable d'estimer le meilleur arbre $\widehat{T}_{opt}$ par

$$\widehat{T}_{opt} = \underset{T_j}{\operatorname{argmin}} R^{hon}(T_j).$$

Élagage et coût de complexité



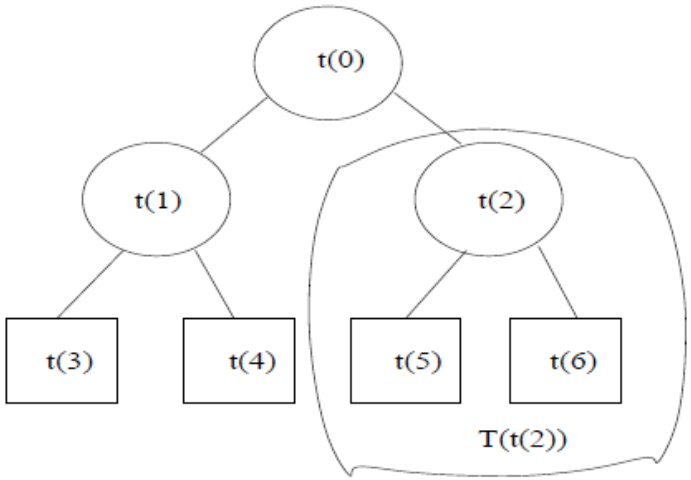
- L'erreur du sous-arbre $T(t_2)$ et l'erreur du nœud t_2 sont respectivement :

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

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 coût de complexité



L'élagage en vaut la peine si :

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

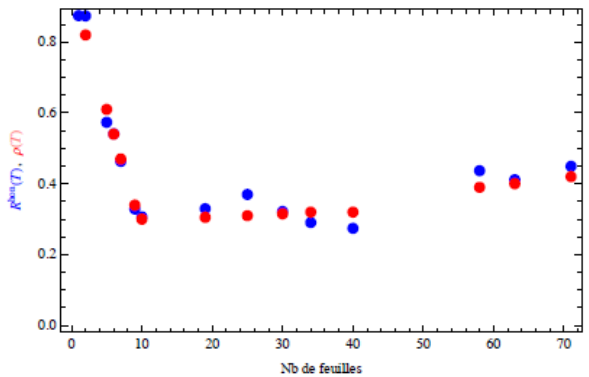
La fonction $g(t, T)$ peut être calculée pour chaque nœud interne de l'arbre.

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. (Out-of sample estimation/Cross Validation)

Élagage et coût de complexité

Typiquement, le graphe de $R^{hon}(T_j)$ en fonction de la taille de T_j prend la forme suivante :



La position du minimum est instable car $R^{hon}(T)$ est une v.a. et pourrait varier beaucoup selon le choix des sous- échantillons (de la validation croisée). Il importe donc de trouver une façon de stabiliser le choix de $\widehat{T}_{opt}$.

Élagage et coût de complexité

D'où la règle d'un écart-type qui :

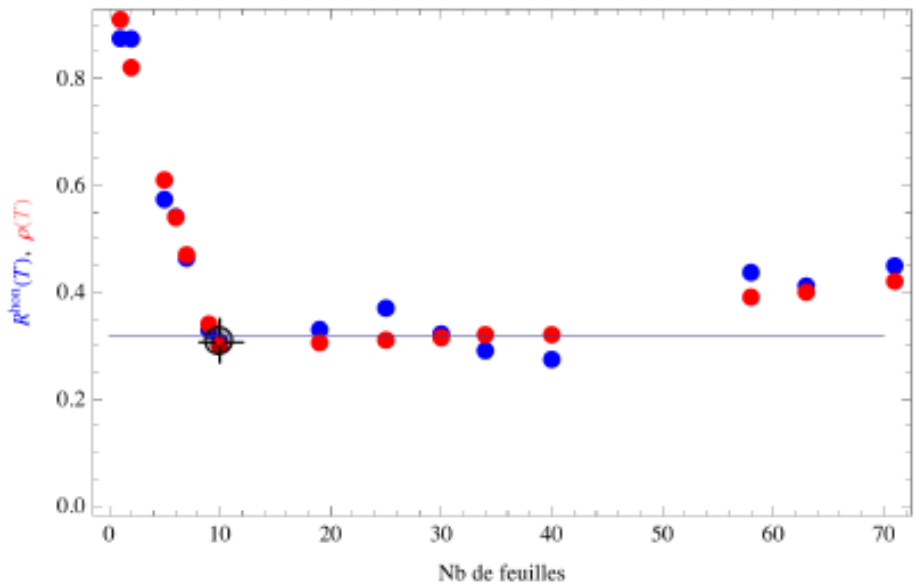
1. Cherche à réduire l'instabilité notée plus haut,
2. Cherche à sélectionner l'arbre le plus simple dont le taux de mauvaise classification est comparable à celui de $\widehat{T}_{opt}$.
3. Ici le terme comparable réfère à des quantités aléatoires pour lesquelles on ne peut pas dire qu'elles sont statistiquement différentes.
4. Cette règle mène à choisir comme meilleur arbre le plus petit arbre $\widehat{T}^*_{opt}$ (en nombre de feuilles) tel que

$$R^{hon}\left(\widehat{T}^*_{opt}\right) \leq R^{hon}\left(\widehat{T}_{opt}\right) + \sqrt{\frac{R^{hon}\left(\widehat{T}_{opt}\right)\left(1 - R^{hon}\left(\widehat{T}_{opt}\right)\right)}{n_{est}}}$$

où n_{est} est la taille de l'échantillon d'estimation.

Élagage et coût de complexité

Ainsi en appliquant cette approche à notre cas l'arbre $\widehat{T}^*_{opt}$ est celui a 10 feuilles.



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	**0.247**
Température	0.029
Humidité	0.152
Vent	0.048

- 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

Application 1

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

Variable	Variation d'impureté
Température	0.571
Humidité	**0.971**
Vent	0.02

- 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

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

Variable	Variation d'impureté
Température	0.571
Humidité	**0.971**
Vent	0.02

- 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	0.02
Humidité	0.02
Vent	**0.971**

Application 1

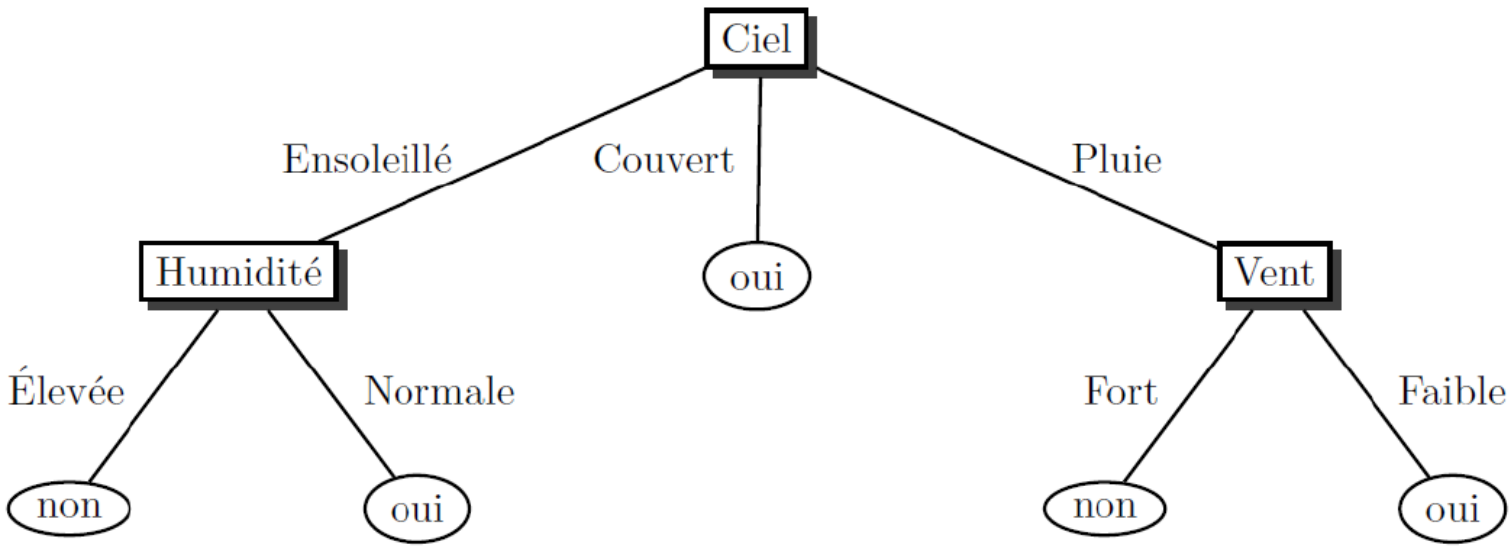


Figure 1 – Arbre de décision obtenu pour l'exemple "Jouer ou ne pas Jouer ?"

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 :

$$\begin{aligned}\Phi_{\text{Temperature}}(s = 17.25, t_0) = \Delta Imp(t_0) &= Imp(t_0) - \pi_L Imp(t_L) - \pi_R Imp(t_R), \\ &= \left[-\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)\right] - \frac{2}{14} \times 0 \\ &\quad - \frac{12}{14} \times \left[-\frac{7}{12}\log_2\left(\frac{7}{12}\right) - -\frac{5}{12}\log_2\left(\frac{5}{12}\right)\right], \\ &= 0.1.\end{aligned}$$

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

Seuil $= s$	$\Phi_{\text{Temperature}}(s, t_0)$
17.25	0.1
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.
$\Longrightarrow$ On applique cette méthode pour chaque attribut numérique et on détermine pour chacun un seuil produisant un gain d'information maximal.
$\Longrightarrow$ Le gain d'information associé à chacun des attributs numériques est celui pour lequel le seuil entraîne un maximum.
$\Longrightarrow$ 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 $R(T)/R(T_0)$.
- xerror : l'erreur calculée par validation croisée.
- xstd : l'écart-type de l'erreur calculée par validation croisée.

Généralités

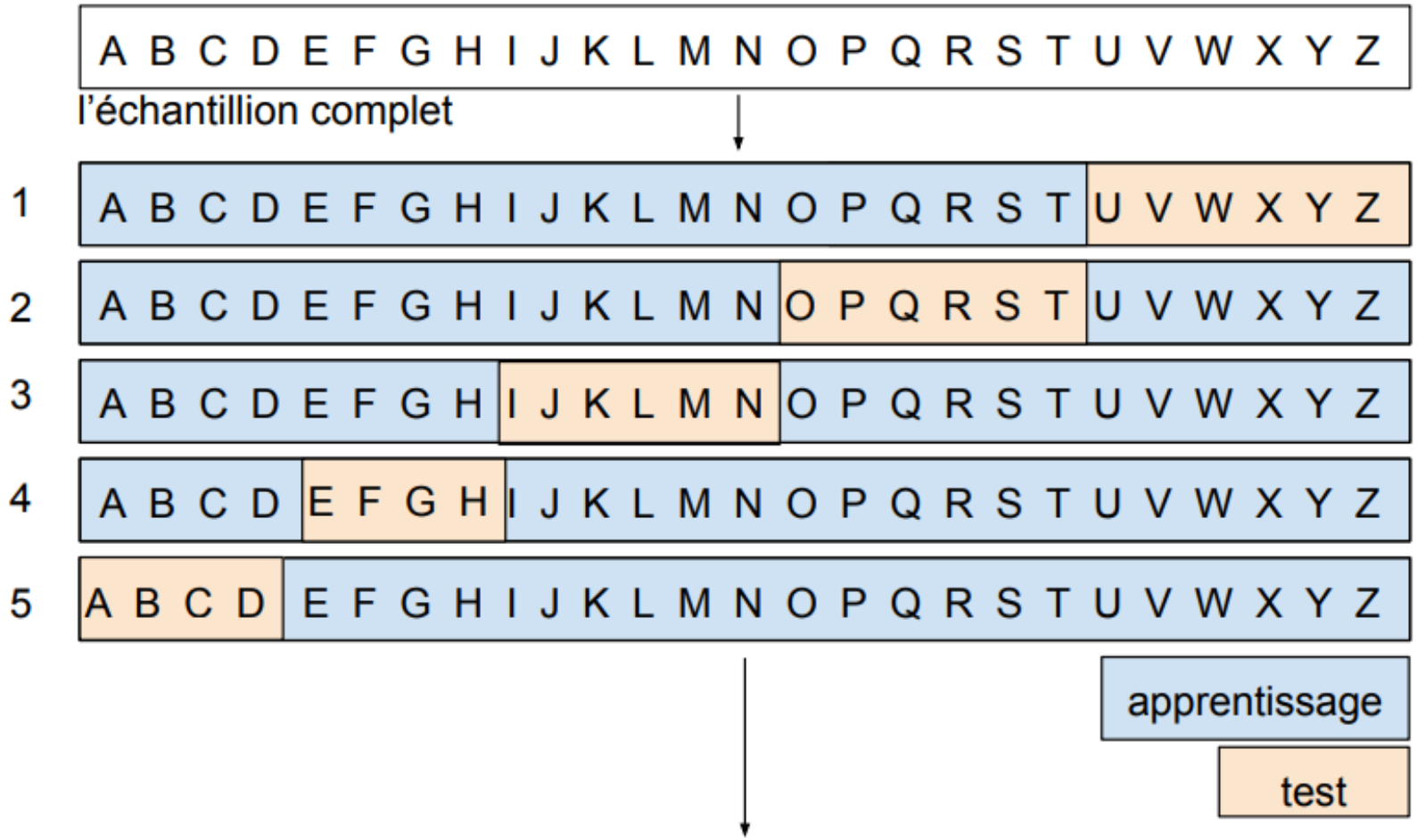
- Les différents algorithmes d'arbre de décision vont différer par les trois opérations suivantes :
  - Décider si un nœud est terminal (tous les individus sont dans la même classe ou il y a moins d'un certain nombre d'erreurs).
  - Affecter une classe à une feuille (nœud terminal) - la classe majoritaire !
  - Sélectionner un test à associer à un nœud (En utilisant des critères statistiques).
- 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

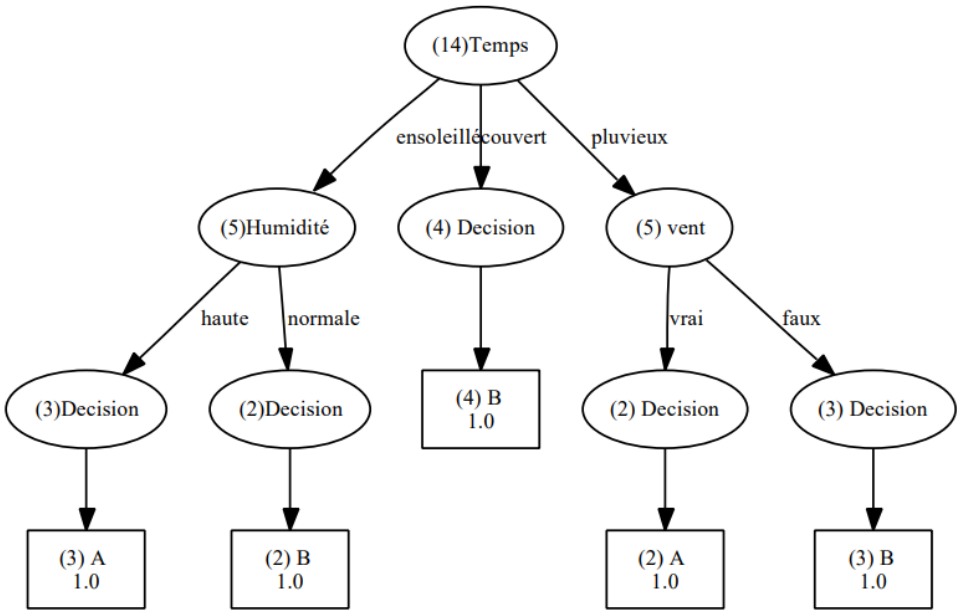


$$CV(\alpha) = \frac{1}{5} \sum_{v=1}^{5} MSE_v^{(test)}$$

- 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.

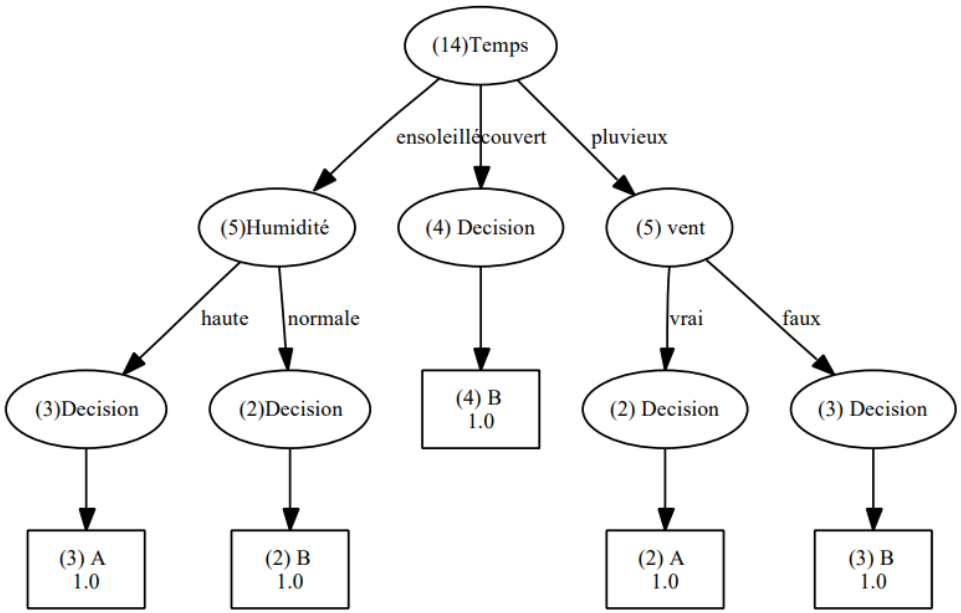
Exemple : Valeurs manquantes

Temps	Température	Humidité	Vent	Classe
Ensoleillé	basse	?	Faux	?



Exemple : Valeurs manquantes

Temps	Température	Humidité	Vent	Classe
Ensoleillé	basse	?	Faux	?



$$\mathbb{P}[A] = \mathbb{P}[A|\text{haute}]\,\mathbb{P}[\text{haute}] + \mathbb{P}[A|\text{normale}]\,\mathbb{P}[\text{normale}] = 1 \times \frac{3}{5} = 0.6,$$

$$\mathbb{P}[B] = \mathbb{P}[B|\text{haute}]\,\mathbb{P}[\text{haute}] + \mathbb{P}[B|\text{normale}]\,\mathbb{P}[\text{normale}] = 1 \times \frac{2}{5} = 0.4.$$

Généralités : Incorporation des coûts de mauvaise classification

- Supposons que le problème soit de classifier un individu comme étant sidatique ou non.
- Si une règle de classification le classe à tort sidatique, l'individu sera dépisté pendant un certain temps.
  $\Longrightarrow$ Sera suivi par le système de santé et tôt ou tard, des médecins détecteront éventuellement l'erreur.
  $\Longrightarrow$ Le "coût" de l'erreur est certes non négligeable pour l'individu mais pour la société, il est minime (le prix du suivi + des tests complémentaires).
- Par contre, s'il est classé à tort comme étant sain, il pourra continuer les activités qui l'ont amené à être "à risque" et contaminera d'autres personnes qui, à leur tour, pourront en contaminer d'autres et ainsi de suite.
  $\Longrightarrow$ Le coût pour la société peut être beaucoup plus important.

Généralités : Incorporation des coûts de mauvaise classification

Soit $C(k' \mid k)$ le coût de classer un individu dans $\mathcal{G}_{k'}$ alors qu'il appartient à $\mathcal{G}_k$. On suppose que : $C(k' \mid k) = 0$ si $k' = k$ et $\neq 0$ sinon. Il y a plusieurs façons d'incorporer une matrice de coût à la procédure arbre de décision. Celle qu'on va utiliser s'appelle la méthode des "priors modifiés" :

Définition.

Soit $\pi_1, ..., \pi_K$, les probabilités a priori d'appartenance au groupe et soit

$$C(+ \mid k) = \sum_{k=1}^{K} C(k' \mid k).$$

Les "priors modifiés" sont $\widetilde{\pi}_k = \frac{\pi_k C(+|k)}{\sum_{k'=1}^{K} \pi_{k'} C(+|k')}$. Si les π_k sont inconnus ils peuvent être remplacés par n_k/n.

Généralités : Incorporation des coûts de mauvaise classification

1. Si $C(k' \mid k) = 1$ pour tout $k' \neq k$, alors $C(+ \mid k) = K - 1$ et $\widetilde{\pi}_k = \pi_k$
2. Les priors modifiés sont uniquement utilisés pour faire un meilleur choix de split. On les injecte dans la formule

$$\widetilde{p}(k \mid t) = \frac{n_t(k) / n_k \widetilde{\pi}_k}{\sum_{k'=1}^{K} n_t(k') / n_{k'} \widetilde{\pi}_{k'}},$$

que l'on injecte à son tour dans l'expression de la mesure d'impureté choisie, et on sélectionne le meilleur split de la façon habituelle et ainsi de suite.
Noter que : $n_t(k)$: nombre d'observations dans le noeud t de la classe $k \in \{1, 2, \ldots, K\}$.

Généralités : Splits suppléants (surrogate splits)

- On a vu que dans l'ensemble $\mathcal{S} = \{\text{splits possibles au noeud } t\}$, le meilleur split s^* est celui qui satisfait :

$$\Phi(s^*, t) = \max_{s \in \mathcal{S}} \Phi(s, t).$$

- Supposons que s^* envoie les observations de t dans t_G avec probabilité (estimée) $p_G = \frac{n(t_G)}{n(t)}$ et de même dans t_D avec probabilité (estimée) $p_D = \frac{n(t_D)}{n(t)}$ (avec $p_G + p_D = 1$ et connues évidemment puisque T existe).
- Soit X_m, une des variables explicatives. Soit $\mathcal{S}_m$ l'ensemble des splits possibles sur cette variable au nœud t. Pour n'importe quel split $s_m \in \mathcal{S}_m$, soit $n_k(GG) =$ nombre d'observations de $n(t)$ qui sont envoyés par s^* **ET** par s_m dans le nœud descendant de gauche t_G et qui appartiennent au groupe $\mathcal{G}_k$.

Généralités : Splits suppléants (surrogate splits)

- Plus précisément, s^* splitte $\mathcal{E}_t$ en $\mathcal{E}_{t_G}$ et en $\mathcal{E}_{t_D}$ alors que s_m splitte $\mathcal{E}_t$ en $\mathcal{E}'_{t_G}$ et en $\mathcal{E}'_{t_D}$ (sur x_m). Alors, $n_k(GG)$ = nombre d'individus dans $\mathcal{E}_{t_G} \cap \mathcal{E}'_{t_G}$ qui sont dans $\mathcal{G}_k$.
- La probabilité qu'un individu se trouve dans $\mathcal{E}_{t_G} \cap \mathcal{E}'_{t_G}$ peut alors être estimée par :

$$\begin{aligned}\mathbb{P}\left[\mathcal{E}_{t_G} \cap \mathcal{E}'_{t_G}\right] &= \sum_{k=1}^{K} \pi_k \frac{n_k(GG)}{n_k}, \\ &= \sum_{k=1}^{K} \frac{n_k(GG)}{n}, \text{si } \pi_k \text{ inconnues.}\end{aligned}$$

- On peut aussi estimer la probabilité que simultanément s^* et s_m envoie un cas de t dans t_G par :

$$p_{GG}(s^*, s_m) = \frac{\mathbb{P}\left[\mathcal{E}_{t_G} \cap \mathcal{E}'_{t_G}\right]}{p(t)} = \frac{\sum_{k=1}^{K} n_k(GG)}{n(t)},$$

de même,

$$p_{DD}(s^*, s_m) = \frac{\mathbb{P}\left[\mathcal{E}_{t_D} \cap \mathcal{E}'_{t_D}\right]}{p(t)}.$$

Généralités : Splits suppléants (surrogate splits)

La probabilité que s_m imite correctement l'action de s^* est estimée par :

$$p\left(s^*, s_m\right) = p_{GG}\left(s^*, s_m\right) + p_{DD}\left(s^*, s_m\right).$$

Définition.

Un splits $\widetilde{s}_m \in \mathcal{S}_m$ est dit le meilleur suppléant de s^* sur la variable x_m si

$$p\left(s^*, \widetilde{s}_m\right) = \max_{s_m \in \mathcal{S}_m} p\left(s^*, s_m\right).$$

On peut interpréter le split suppléant comme étant celui qui, sur la variable x_m, imite le mieux l'action de s^* au nœud t.

Maintenant, on peut déterminer parmi toutes les variables explicatives le meilleur split suppléant au split s^* au nœud t, noté par $\widetilde{s}_{(1)}$.
Pour identifier cette variable on applique :

$$\underset{m \in \{1,\ldots,p\}}{\operatorname{argmax}} \; p\left(s^*, \widetilde{s}_m\right).$$

De la même façon, on peut définir le deuxième meilleur split suppléants $\widetilde{s}_{(2)}$ au nœud t comme étant celui qui donne le deuxième plus grand $p\left(s^*, \widetilde{s}_m\right)$ et ainsi de suite ...

Généralités : Splits suppléants (surrogate splits)

Remarque.

Les splits suppléants servent à trois fins :
a) La gestion de données manquantes,
b) La mesure de l'importance des variables
c) La détection de variables "masquées".

L'idée fondamentale de cette approche est que :

- Le meilleur split suppléant imite le mieux possible l'action de s^*.
- Il y a un maximum de chance que l'individu se retrouve dans le nœud descendant dans lequel il se serait retrouvé si s^* avait pu s'accomplir.
- Il a donc un maximum de chance de se retrouver dans la feuille (et donc le groupe) où il aurait normalement abouti.

Validité : Sensibilité, Spécificité, ROC, AUC, LIFT

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 connaître 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. $\left(prop. \pm z_{1-\alpha} \sqrt{\frac{prop.(1-prop.)}{n}} \right)$.

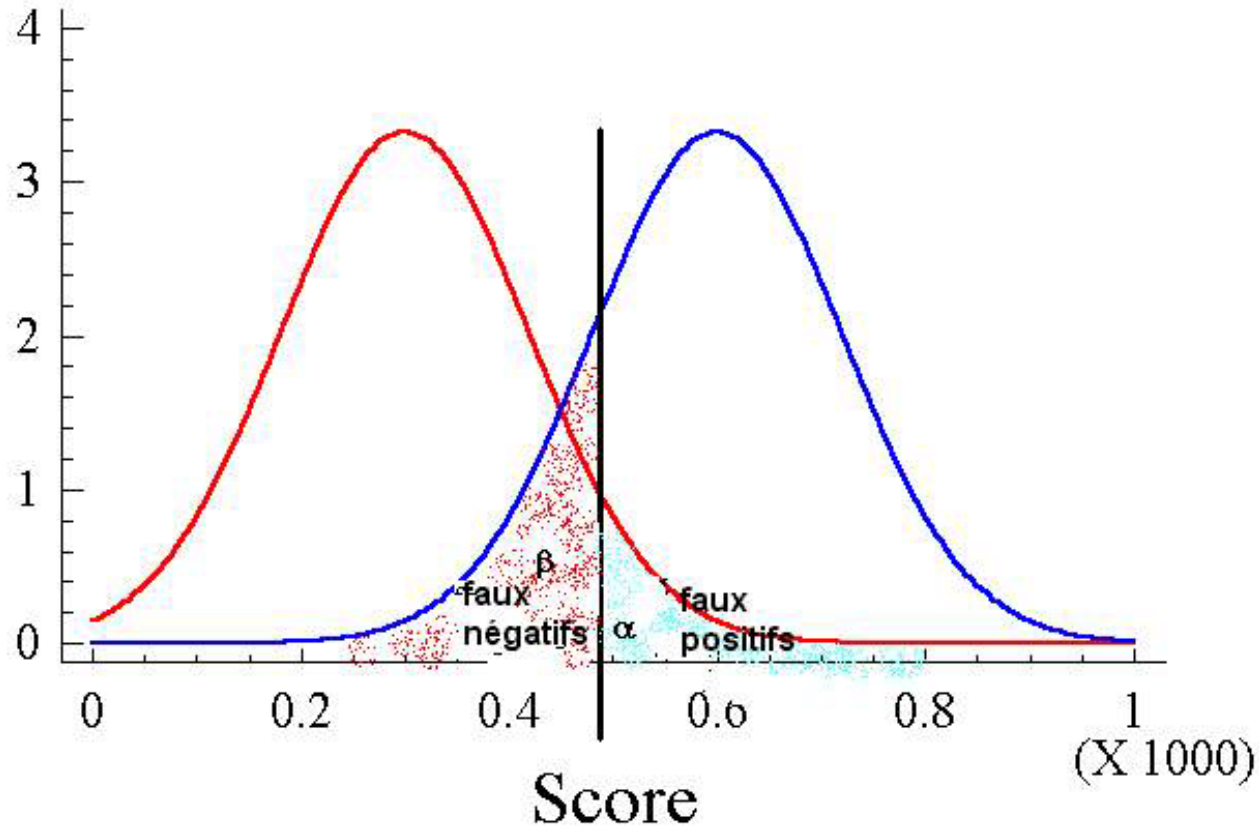
Remarque.

- 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.
- Jusqu'à maintenant on a utilisé les arbre pour classifier les individus. Mais les **scores** (**i.e.** $p(k \mid t)$) peuvent être exploités par les utilisateurs du classifieur pour affiner son utilisation (Courbes ROC et LIFT)

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)

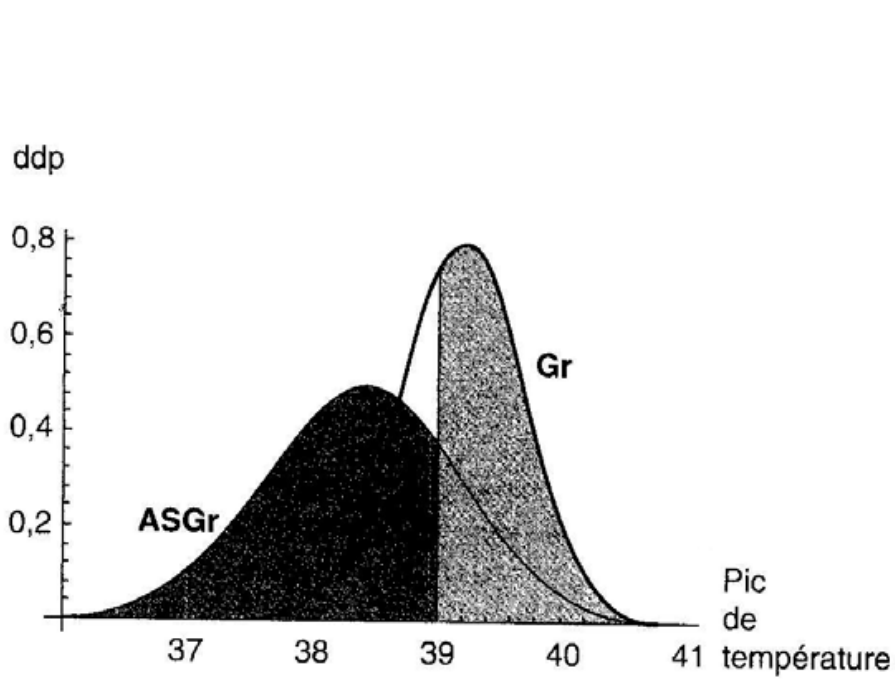
- Sensibilité : $1-\beta = \mathbb{P}\left[S > s | M\right]$ (puissance du test).
- Spécificité : $1-\alpha = \mathbb{P}\left[S < s | \overline{M}\right]$.

Définition 4.

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.



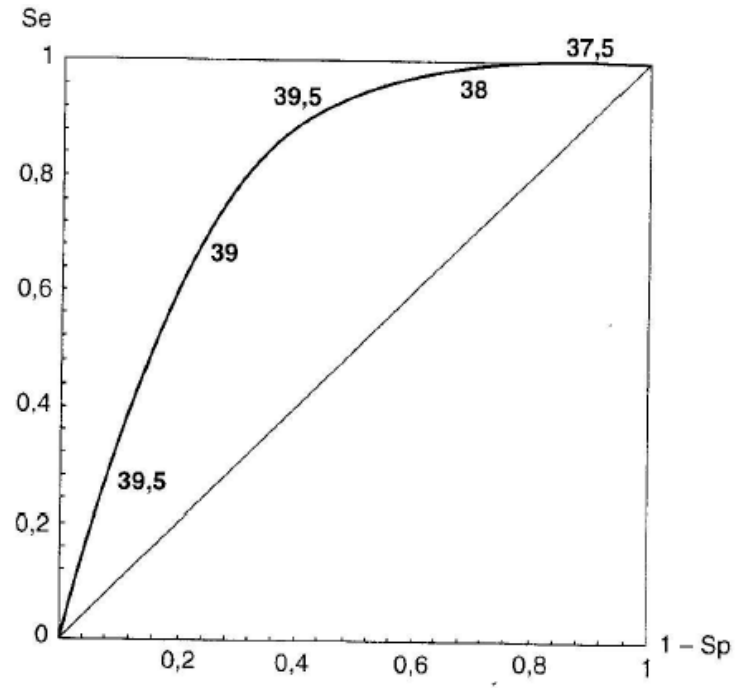


Figure 2 – A gauche : distribution de probabilité pic de fièvre chez des sujets ayant une vraie grippe (Gr) et une fausse grippe (ASGr) ; A droite : Courbe ROC pour l'exemple du diagnostic de la vraie grippe vs celui de la fausse grippe.

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 coût 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 ?

La courbe LIFT

Exercice.

Lorsqu'un client demande un prêt, la banque veut s'assurer que le client la remboursera. Un tel client est appelé un "bon client" alors que ceux qui posent problèmes sont de "mauvais clients". Pour discriminer entre les deux groupes, on utilise un classifieur qui fournit un score.

Ici l'intérêt principal est d'identifier les "bon clients' qui constituent disons le $\mathcal{G}_2$.

Si le score d'un client est supérieur à un seuil s, on estime alors qu'il a de fortes chances d'être un bon client et le prêt lui sera accordé.

Sinon, le client sera réputé mauvais et le prêt pourrait lui être refusé.

La difficulté est donc de trouver le seuil s qui "filtrera" au mieux les mauvais clients.

La courbe LIFT

- Supposons que l'on sache à l'avance que a des n clients de $\mathcal{E}$ sont $\mathcal{G}_2$. Imaginons que le budget alloué à la campagne marketing ne permette d'envoyer l'offre qu'à $M \in \{1, ..., n\}$ clients de $\mathcal{E}$. En

  envoyant l'offre à M clients choisis au hasard de $\mathcal{E}$, on peut espérer toucher $M \times \frac{a}{n}$ des $\mathcal{G}_2$. Mais cette démarche n'utilise pas le classifieur.
- Soit maintenant $\left\{s_{(1)} \geq s_{(2)} \geq \ldots \geq s_{(n)}\right\}$, les scores (obtenus via le classifieur) des individus de $\mathcal{E}$ et considérons les M individus ayant les scores les plus élevés. Comme ces scores sont connus, ceci

  revient au même que de fixer un seuil $s \in [0, 1]$ et de considérer les scores $\left\{s_{(1)} \geq s_{(2)} \geq \ldots \geq s_{(M)} = s\right\}$.
- Supposons que parmi les M individus ayant $s_{(i)} \geq s$, il y ait A clients $\mathcal{G}_2$. Alors en envoyant l'offre à ces M individus seulement, on peut espérer toucher A des $\mathcal{G}_2$ (les VP).

La courbe LIFT

- Le LIFT apporté par le classifieur au seuil s (ou à la taille M) est le quotient :

$$\text{LIFT}(s) = \frac{A/M}{a/n}.$$

- On parle généralement de LIFT à $M/n\%$ plutôt que de LIFT au seuil s. Mais il s'agit de la même chose car M détermine s et vice-versa.

La courbe LIFT

Ce LIFT s'interprète de la façon suivante :

- Supposons que le budget alloué pour une campagne de crédit soit de 1000 euros et qu'un envoi coûte 1 euro. On ne peut donc faire que 1000 envois. La banque dispose d'une base de données de $n = 10000$ clients dont, disons, 20% (soit $a = 2000$) sont bons. Il convient de choisir les 1000 clients à qui on enverra l'offre.

- Si on choisit au hasard $M = 1000$ clients de cette base, on peut espérer toucher $1000 \times \frac{2000}{10000} = 200$ bons. Il en coûterait donc, en moyenne, $1000/200 = 5$ euros pour joindre un bon.

- Imaginons que, pour un classifieur donné, le LIFT à 10% $(M/n\%) = \frac{A/M}{a/n} = \frac{A/1000}{2000/10000} = 3$, ce qui revient à dire qu'avec ce classifieur, on peut s'attendre à toucher $A = 600$ bons lors de la campagne marketing.

La courbe LIFT

Ce LIFT s'interprète de la façon suivante :

- Autrement dit, 600 (parmi les 2000) bons se retrouveraient parmi les 1000 clients à qui on enverrait l'offre.
- Ainsi le coût pour toucher un bon ne serait plus que de $1000/600 = 1,66$ euros. Notre classifieur a donc permis une sérieuse économie !

Remarque.

1) Il reste quand même 1400 bons clients qui n'ont pas été rejoints. On peut se poser la question : si le budget était augmenté à 2000 euros, quel serait le nombre de bons clients supplémentaires rejoints ?

2) ATTENTION, on ne peut pas répondre à cette question par une règle de trois. En effet, le lien entre les scores et le nombre de bons clients rejoints n'est pas nécessairement linéaire.