Arbres de décision

Ghazi Bel Mufti

ghazi.belmufti@gmail.com

Plan

1. Introduction
2. Arbres de régression : Y quantitative
3. Arbres de classification : Y nominale à K modalités
4. Arbres sur r : package rpart
5. Conclusion

Introduction I

- Les arbres de décision sont couramment utilisés pour expliquer et/ou prédire les valeurs d'une variable à l'aide de variables explicatives qui peuvent être quantitatives ou qualitatives.
- Cette technique est l'une des plus intuitives et des plus populaires du data mining car elle fournit des règles explicites de classement.
- En présence d'une variable privilégiée Y "à expliquer" par les variables X^1, ..., X^p, il s'agit de sélectionner parmi les variables explicatives celles qui sont les plus discriminantes pour la variable Y
- Un arbre de décision est appelé *arbre de régression* (resp. *arbre de classification*) si la variable à expliquer est quantitative (resp. qualitative).

Introduction II

- Il s'agit de construire des règles de décision permettant d'affecter un nouvel individu à l'une des q classes (i.e arbre de discrimination) ou de lui affecter une valeur y (i.e. arbre régression).
- Les arbres de décision ont été introduits dans les années 60 (cf. Morgan et Sonquist, 1963), et sont devenus populaires avec la méthode CART (Classification And Regression Trees) proposée par Breiman *et al.* en 1984.
- Plusieurs variantes de CART existent, e.g. les méthodes ID3 (1986) et C4.5 (1993) proposées par Quinlan. Il existe aussi des propositions pour améliorer CART : Chipman *et al.* (1998), Loh (2002) et Su *et al.* (2004).

Introduction III

- Le modèle proposé par les arbres de décision peut être considéré comme complémentaire de celui de la régression linéaire dans le sens suivant :
  - La régression linéaire propose un modèle global, c'est-à-dire une unique formule globale de prédiction qui s'applique à l'espace tout entier des données.
  - Lorsque les données contiennent un grand nombre de variables explicatives qui interagissent de façon complexe, et notamment de façon non linéaire, il peut être difficile d'estimer un modèle global qui, de plus, est assez souvent difficile à interpréter dans le cas où l'ajustement est réussi.

Introduction IV

- L'approche des arbres consiste à diviser l'espace des données selon des régions où les interactions entre les variables explicatives sont plus faciles à modéliser.
- Les régions sont obtenues par subdivisions, selon un processus de partitionnement récursif, de façon à obtenir finalement des parties de l'espace des données suffisamment circonscrites pour qu'un modèle simple d'ajustement puisse s'appliquer.
- Le modèle proposé par un arbre de décision comprend deux aspects :
  1. le partitionnement récursif de l'espace des données qui est représenté par un arbre,
  2. le modèle d'ajustement simple pour chaque région, les régions étant définies par les noeuds terminaux (ou feuilles) de l'arbre.

- Dans le reste de ce chapitre, nous présentons la méthode CART qui est souvent appliquée pour construire un arbre de décision.

Principe de CART

L'idée de base est d'effectuer la division d'un noeud de telle sorte que les deux segments descendants soient plus homogènes que le noeud parent et les plus différents possible entre eux vis-à-vis de la variable Y.

1. La méthode consiste à rechercher d'abord la variable X^j qui explique le mieux la variable Y.

2. Cette variable définit une première division de l'échantillon en deux segments.

3. On réitère cette procédure à l'intérieur de chacun de ces segments en recherchant la deuxième variable, et ainsi de suite...

Arbre de régression

Exemple : considérons d'abord le cas élémentaire où l'on souhaite expliquer une variable continue Y à l'aide de deux variables continues X^1 et X^2 à valeurs dans $[0, 1]$. Deux partitions du carré $[0, 1] \times [0, 1]$ sont représentées dans la figure 1 ci-dessous.

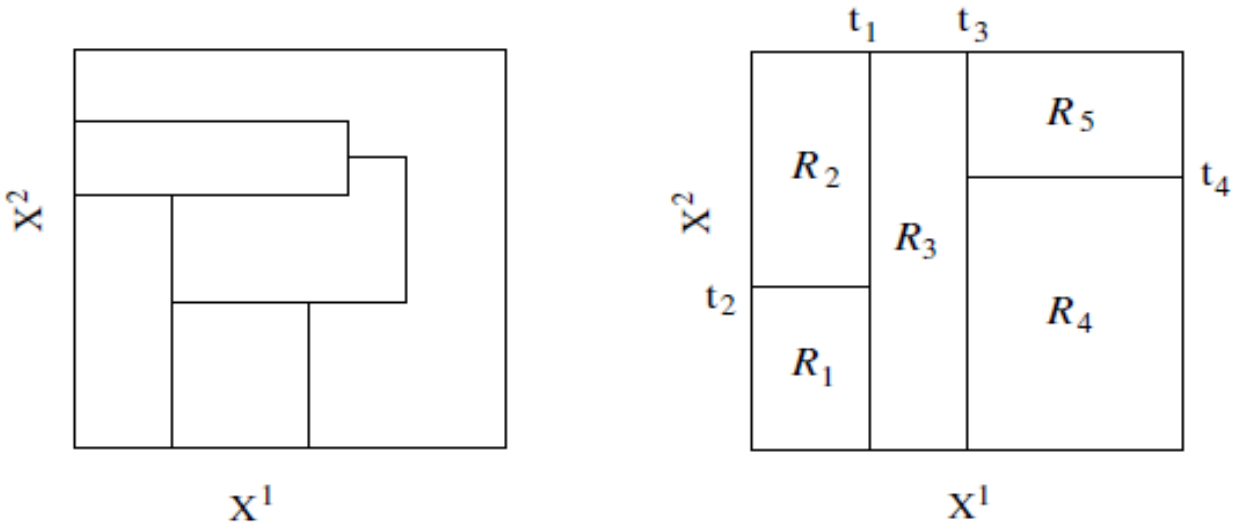

Figure – Deux partitions de $[0, 1] \times [0, 1]$

- Les éléments de ces deux partitions sont des régions de $\mathbb{R}^2$ dont les frontières sont parallèles aux axes.
- Pour chacune d'elle, on peut modéliser Y par la moyenne des y_i calculée uniquement pour les observations (x_i^1, x_i^2) de cette région.
- Pour la partition de gauche, la règle de prédiction de Y sera complexe car deux de ses régions ne sont pas rectangulaires et donc ne peuvent pas être décrites simplement à l'aide des variables X^1 et X^2.

Principe de la méthode CART

- CART considère uniquement des partitions avec des régions rectangulaires, et plus généralement des hyperrectangles lorsque le modèle comporte plus de deux variables explicatives.
- CART procède de façon récursive, en construisant des partitions binaires comme dans la figure de droite ci-dessus.
- CART commence par diviser l'espace tout entier en deux rectangles et modélise la variable Y par sa moyenne dans chacune des deux régions.
- Puis une ou deux de ces régions sont à nouveau divisées en deux régions, et ainsi de suite jusqu'à ce qu'un critère d'arrêt soit satisfait.

Suite de l'exemple

- Dans le cas de la figure de droite ci-dessus,
  1. l'espace est d'abord divisé en deux parties selon la frontière $X^1 = t_1$.
  2. la région $X^1 \leq t_1$ est divisée par la droite $X^2 = t_2$
  3. la région $X^1 > t_1$ est divisée par la droite $X^1 = t_3$.
  4. la région $X^1 > t_3$ est découpée en deux parties selon la frontière $X^2 = t_4$.
- Le résultat obtenu est un partitionnement en cinq régions $R_1, \ldots, R_5$ comme indiqué par la figure de droite.
- Le modèle de régression associé s'écrit donc de la façon suivante :

$$\hat{f}(X) = \sum_{m=1}^{5} c_m \mathbb{1}_{R_m}(X),$$

où $X = (X^1, X^2)$, $\mathbb{1}_A$ désigne la fonction indicatrice de la partie $A \subseteq \mathbb{R}^2$ et $c_m \in \mathbb{R}$ une constante associée à la région R_m.

- Ce modèle est graphiquement représenté par l'arbre binaire représenté dans la figure 2 ci-dessous.

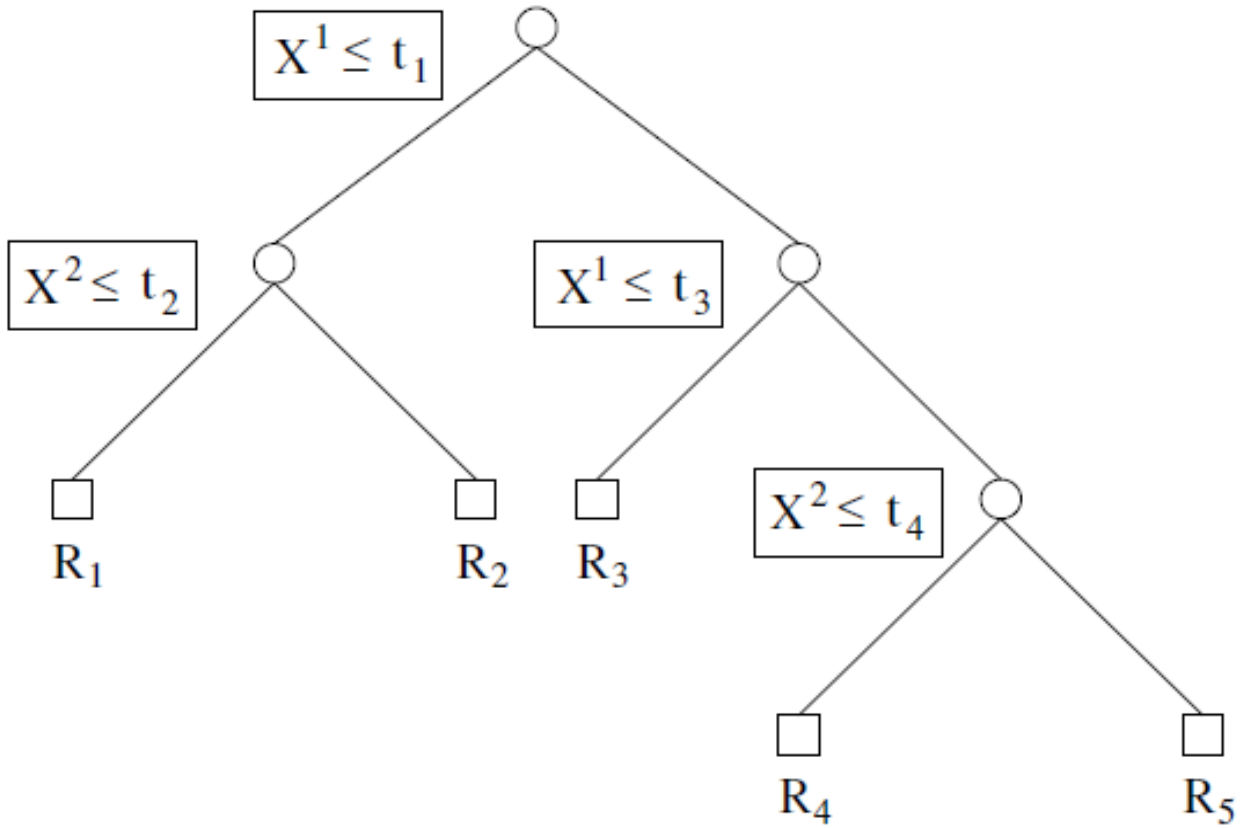

Figure – Exemple d'arbre de régression.

- Le sommet ou la racine de l'arbre représente l'ensemble tout entier.
- L'ensemble des individus représentés par un noeud (non terminal) est divisé en deux parties : les individus satisfaisant la condition du noeud sont affectés à la branche gauche et les autres à la branche droite.
- Les noeuds terminaux (ou feuilles) correspondent aux régions R_1, R_2, R_3, R_4 et R_5.
- Etant donné un arbre T, on définit le un sous-arbre $T' \subset T$ comme étant tout arbre obtenu en élaguant T, i.e. obtenu par suppression d'un nombre arbitraire de ses noeuds internes (c.-à-d. non terminaux).
- Dans un arbre T, élager un noeud interne m consiste à supprimer de T tous les descendants du noeud m et les branches qui les relient, tandis que le noeud m est conservé et devient un noeud terminal dans l'arbre ainsi élagué.

- La partition de l'espace des variables explicatives, est entièrement déterminée par l'arbre de décision. Dans le cas où il existe plus de deux variables explicatives, il est clair que les partitions sont plus difficiles à représenter visuellement que dans le cas de l'exemple de la figure ci-dessus, alors que la représentation via un arbre binaire reste toujours valable et facilement interprétable.

Construction de l'arbre

Départ : un seul segment contenant l'ensemble des individus.

1. Etablir pour ce premier noeud l'ensemble des divisions admissibles en déterminant pour chacune des variables $X^j, j = 1..., p$ la meilleure condition de séparation d_j^* (au sens d'un critère donné).
2. Sélectionner, en utlisant un critère, la "meilleure" division d'un noeud : on retiendra parmi les p divisions précédente, celle, notée d^*, qui fournit les deux segments les plus typés, vis-à-vis de Y.
3. Appliquer la même procédure à chacun des segments descendants obtenus.(Les var. explicatives peuvent être différentes selon les segments), etc...

Arrêt : La procédure s'arrête lorsque tous les segments sont déclarés terminaux (i.e. leur taille est inférieure à un seuil fixé).

Condition de séparation : var. explicative continue

- Pour une variable continue X_j ayant n valeurs distinctes, il y'a $n-1$ conditions de séparation possibles.

- Une fois les valeurs $x_1^j, ..., x_n^j$ de X^j triées dans l'ordre les conditions de séparation s'expriment sous la forme

$$X^j \leq \ moyenne\,(x_k^j, ..., x_{k+1}^j) = d_j^k$$

De toutes les conditions de séparation d_j^k possibles, la procédure sélectionne la "meilleure" d_j^* au sens d'un critère de division à préciser.

Condition de séparation : var. explicative nominale

- une variable nominale à deux modalités ne peut fournir qu'une seule division,
- une variable à q modalités ordonnées fournit $q - 1$ divisions,
- une variable à q modalités non ordonnées fournit $2^{q-1} - 1$ divisions ; toutes les divisions correspondant aux différents sous ensembles de modalités sont examinés (cas de l'arbre CART).

Critère à optimiser : variance résiduelle minimale

La variance de Y dans les segments descendants doit être plus faible que la variance de Y dans le noeud parent.

1. Pour toute division d_j^k d'un noeud m en m_g et m_d par une variable X^j, on calcule la variance résiduelle du noeud m :

$$var(d_j^k, m) = (\frac{n_g}{n_m} s_g^2) + (\frac{n_d}{n_m} s_d^2)$$

où n_g, n_d et n_m sont resp. les effectifs des segments m_g, m_d et m et s_g^2, s_d^2 sont les variances de Y à l'intérieure des segments m_g et m_d.

2. On retient la meilleure division d_j^* realisée par X^j :

$$var(d_j^*, m) = \min_k \{var(d_j^k, m)\}$$

3. Parmi toutes les divisions d_j^*, la meilleure division est effectuée à l'aide de la var. qui assure :

$$var(d^*, m) = \min_{j=1,\ldots,p} \{var(d_j^*, m)\}$$

Erreur Apparente de Prévision

Si certaines variances des segments sont encore importantes, on peut continuer les divisions dans le but de réduire davantage les variances des segments terminaux.

- On associe à chaque segment terminal m de l'arbre T l'erreur ER_m suivante

$$ER_m = \frac{n_m}{n} s_m^2$$

où n est le nombre total d'ind., n_m est le nombre d'individu du segment m et s_m la variance à l'intérieur de m.

- L'Erreur Apparente de Prévision (EAP) associée à l'arbre T :

$$EAP(T) = \sum_{m \in \widetilde{T}} ER_m$$

où $\widetilde{T}$ désigne l'ensemble des noeuds terminaux de l'arbre T.

Règle d'affectation

- Affecter une valeur à Y pour chaque noeud terminal.
- Un nouvel individu qui descend dans l'arbre et arrive dans un segment terminal prendra comme valeur la moyenne de ce segment.

Arbres de regréssion avec rpart

- **La commande rpart**

```
fr <- data.frame(x = runif(1000, 0, 3),
 y = runif(1000, 2, 5))
fr$z <- ifelse(fr$x < 2, 2 * fr$x, 3 * fr$y)
fit <- rpart(z  ~ x + y, fr, method = "anova")
```

- **Les sorties Impression de l'arbre sous forme textuelle pour un arbre de régression :**

```
n= 1000
node), split, n, deviance, yval
* denotes terminal node
1) root 1000 18975.56000 4.807525
     2) x< 2.003762 668 889.05200 1.996973
          4) x< 1.0084 341 115.48120 1.021080 *
          5) x>=1.0084 327 110.15180 3.014648 *
     3) x>=2.003762 332 2192.93300 10.462490
```

Interprétation

A chaque noeud, on a :

- le numéro du noeud : 2)
- le critère de split (ou root pour la racine) : $x < 2.003762$
- le nombre total d'instances pour le noeud : 668
- la déviance, c'est à dire la somme des carrés des écarts à la valeur prédite pour les valeurs de toutes les instances du noeud : 889.05200
- la valeur prédite de la variable à prédire : 1.996973
- une '*' si c'est un noeud terminal.

Arbre optimal

- La taille de l'arbre est ici considérée comme un paramètre que l'on règle de façon adaptative en fonction des données étudiées ("tuning" de paramètre).
- Une première approche, a priori naturelle, est d'arréter le découpage si, pour chaque noeud, la décroissance de la somme des carrés des erreurs n'excède pas un certain seuil fixé à l'avance. Cependant, cette stratégie est à courte vue, car un découpage inutile peut très bien être suivi d'un découpage pertinent.
- L'approche préférée consiste à construire un arbre de grande taille, noté T_0, qui est obtenu en stoppant le processus de découpage uniquement lorsque la taille de chaque noeud n'excède pas une taille limite, par exemple 5. Puis on applique à cet arbre T_0 un élagage dit de "coût-complexité" qui est détaillé dans la suite.

Approche "coût-compléxité" I

1. On commence par construire un arbre de grande taille, noté T_0, qui est obtenu en stoppant le processus de découpage uniquement lorsque la taille de chaque noeud n'excède pas une taille limite, par exemple 5.
2. Etant donné T un sous-arbre de T_0. En notant $\widetilde{T}$ l'ensemble des noeuds terminaux de l'arbre T, $|\widetilde{T}|$ désigne le nombre de ces noeuds terminaux. De plus, pour tout $\alpha \geq 0$, on note :

$$C_\alpha(T) = \sum_{m \in \widetilde{T}} ER_m + \alpha |\widetilde{T}|. \quad (1)$$

3. Pour chaque valeur de $\alpha \geq 0$, on cherche un sous-arbre $T_\alpha \subseteq T_0$ qui minimise $C_\alpha(T)$.
   - Le paramètre de tuning $\alpha \geq 0$ règle le compromis entre la taille de l'arbre et l'adéquation de l'ajustement aux données.

Approche "coût-compléxité" II

- Intuitivement, de grandes valeurs de α déterminent des arbres de petite taille, et inversement pour de petites valeurs de α. Pour $\alpha = 0$, la solution est l'arbre entier T_0.

4. Le réglage ("tuning") de la valeur de α se fait de la manière suivante.
   - Pour chaque α, on peut montrer qu'il existe un unique plus petit sous-arbre T_α qui minimise $C_\alpha(T)$.
   - Pour trouver T_α, on utilise un élagage dit du plus faible lien : on élimine successivement le noeud interne qui produit le plus petit accroissement dans $\sum_{m \in \widetilde{T}} ER_m$ (inférieur à α), et on continue jusqu'à produire un arbre avec un seul noeud interne (i.e. la racine).
   - Il en résulte une suite décroissante de sous-arbres de T_0, notée de la façon suivante :

$$T_0 \supseteq T_1 \supset T_2 \supset \ldots \supset \{r\},$$

où r est le noeud à la racine de l'arbre.

Approche "coût-compléxité" III

- On peut montrer que cette suite d'arbres contient T_α pour tout $\alpha \geq 0$, et que de plus, T_k est le plus petit sous-arbre qui minimise (1) pour $\alpha \in [\alpha_k, \alpha_{k+1}[$.

5. L'estimation de la meilleure valeur de α s'effectue ensuite à l'aide d'une validation croisée à k plis (k-fold cross validation) ;
   - En général on choisit $k = 5$ ou 10. On détermine ainsi la valeur $\widehat{\alpha}$ qui minimise la somme des carrés des erreurs mesurée par validation croisée.
   - L'arbre produit par CART est l'arbre $T_{\widehat{\alpha}}$.

Arbres de classification

- Un arbre de classification a pour but d'expliquer une variable qualitative par des variables qui peuvent être qualitatives ou non.
- Par exemple, considérons le cas d'une variable qualitative qui indique le risque pris (bon / mauvais) lorsque l'on accorde un prêt à un individu. Les variables explicatives concernent ici les caractéristiques de l'individu candidat au prêt, i.e. l'âge, le statut marital, si le candidat est propriétaire, le revenu et le sexe.

Exemple I

Le tableau suivant présente les données qui ont été observées pour ces variables sur un échantillon de 10 individus :

Individu	age	marié	propriétaire	revenu	sexe	classe (risque)
1	22	non	non	28000	masculin	mauvais
2	46	non	oui	32000	féminin	mauvais
3	24	oui	oui	24000	masculin	mauvais
4	25	non	non	27000	masculin	mauvais
5	29	oui	oui	32000	féminin	mauvais
6	45	oui	oui	30000	féminin	bon
7	63	oui	oui	58000	masculin	bon
8	63	oui	non	52000	masculin	bon
9	23	non	oui	40000	féminin	bon
10	23	oui	oui	28000	féminin	bon

Table – Données de crédit bancaire

Exemple II

La figure 3 ci-dessous présente un arbre de classification construit à partir de ces données.

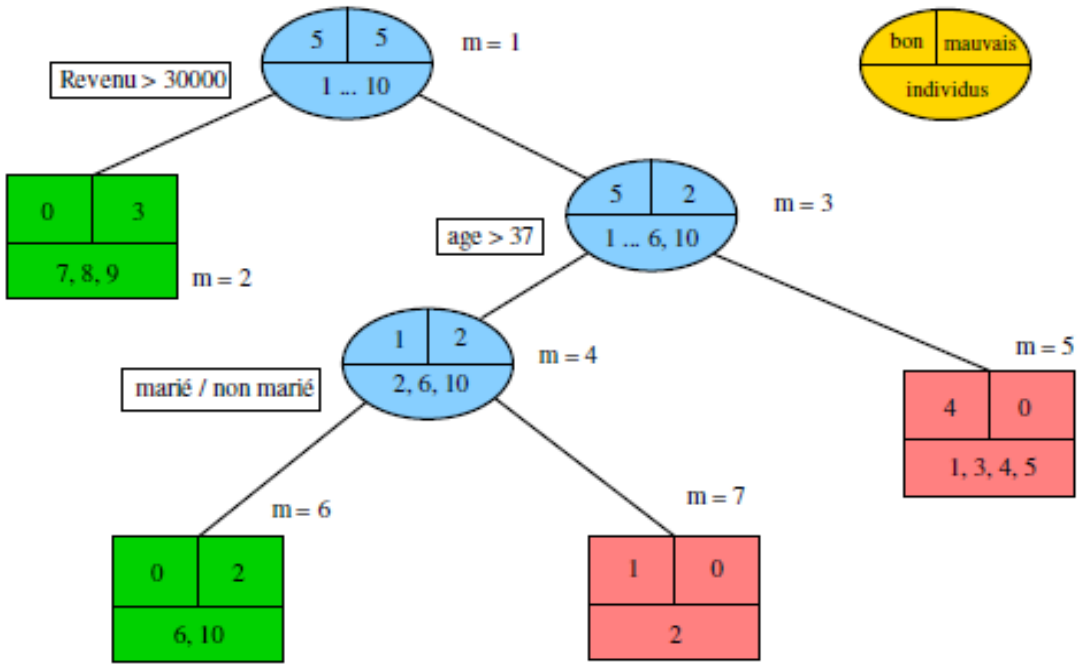

Figure – Exemple d'arbre de classification

Exemple III

- Supposons qu'un nouvel individu candidat à l'attribution d'un prêt, ait les caractéristiques suivantes : *age* = 42 ; *marié* = non ; *propriétaire* = oui ; *revenu* = 30000 ; *sexe* = masculin.
- Pour déterminer la classe de cet individu, on commence par l'affecter à la racine de l'arbre, puis on l'affecte à l'un des deux noeuds fils (droit vs gauche) de cette racine, en effectuant le test associé à la racine.
- On réitère le processus d'affectation du candidat pour chaque noeud non terminal où il est affecté. On obtient alors un chemin dans l'arbre qui suit les affectations de l'individu :

$$m : 1 \to 3 \to 4 \to 7$$

- On voit que ce chemin aboutit au noeud terminal $m = 7$. Comme ce noeud est pur, i.e. constitué uniquement de candidats ayant un risque élevé : le prêt n'est donc pas accordé au candidat.

Exemple IV

- Le principe général de construction d'un arbre de classification est le suivant : à chaque noeud terminal de l'arbre en construction, on sélectionne le test qui génère le plus d'information sur les classes à expliquer/prédire dans chaque noeud créé, i.e. celui qui génère des noeuds qui sont les moins impurs.
- Par exemple, si l'on considère le test effectué à la racine de l'arbre, le test effectué sur le revenu est préférable à celui effectué sur le sexe, comme le montre la figure suivante.

Exemple V

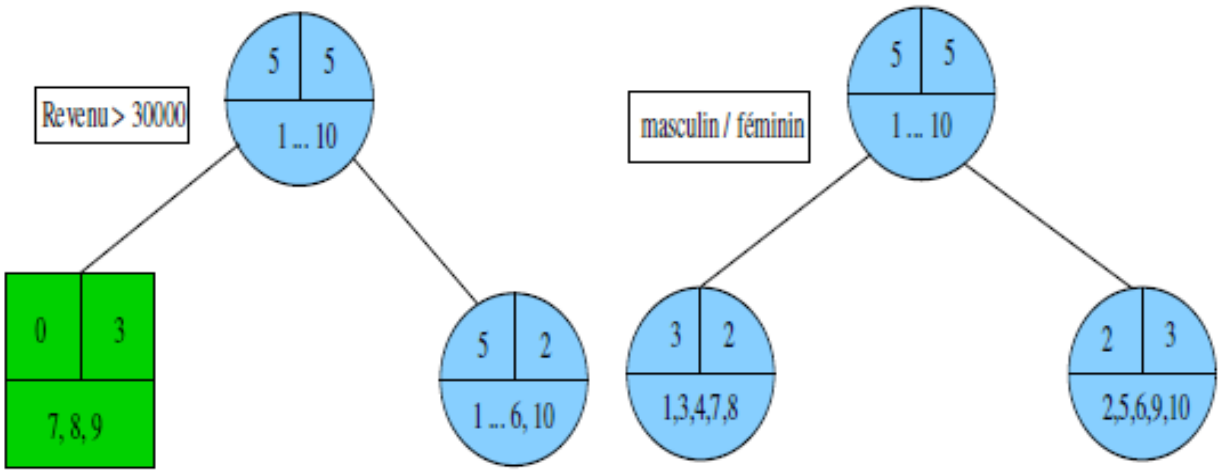

Figure – Comparaison de deux coupures possibles

Critère à optimiser : impureté I

- Envisageons à présent le cas général, i.e. on souhaite expliquer une variable qualitative Y à K modalités, par des variables qui peuvent être quantitatives ou qualitatives.
- Dans ce qui suit, on note X la variable vectorielle ayant pour coordonnées les variables explicatives. De plus, on note $1, 2, \ldots, K$ les modalités de la variable Y à expliquer.
- Les modalités définissent donc K classes d'individus. La différence principale avec un arbre de régression porte sur la définition du critère de découpage d'un noeud.
- Rappelons tout d'abord que dans un arbre de régression, la qualité d'un noeud m est mesurée par l'erreur ER_m. Cette mesure de qualité n'est pas appropriée ici car la variable à expliquer est qualitative.

Critère à optimiser : impureté II

- Afin de définir la qualité d'un noeud m, nous définissons d'abord une mesure de son impureté, qui est notée $i(m)$.
- A chaque noeud m est associée la région R_m qui contient les individus qui satisfont les conditions de tous les noeuds situés sur l'unique chemin reliant la racine au noeud m.
- Chaque région R_m contient N_m individus, et on note $\hat{p}_{mk}$ la quantité définie par :

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbb{1}_{(y_i = k)},$$

qui est la proportion d'individus appartenant à la classe k dans la région R_m.

Critère à optimiser : impureté III

1. Une mesure d'impureté $i(m)$ du noeud m est fonction de ces proportions :
$$i(m) = \Phi(\hat{p}_{mk}, \ldots, \hat{p}_{mK}).$$

Il est raisonnable également d'imposer à la mesure $i(m)$ de vérifier les propriétés suivantes :

2. L'impureté $i(m)$ est maximale quand les individus du noeud m se répartissent uniformément dans les classes, i.e. pour $\hat{p}_{mk} = 1/K$ quel que soit $k \in \{1, \ldots, K\}$.
3. L'impureté $i(m)$ est minimale quand les individus du noeud m appartiennent à une seule classe, i.e. lorsqu'il existe $k \leq K$ tel que $\hat{p}_{mk} = 1$ et $\hat{p}_{mk'} = 0$ si $k' \neq k$.
4. La fonction Φ est symmétrique.

- Néanmoins, de nombreuses fonctions Φ satisfont ces propriétés.
- Quel que soit le choix de Φ, il est naturel de mesurer la qualité de la coupure d'un noeud m comme étant la réduction de l'impureté réalisée par cette coupure :

$$\Delta i(m) = i(m) - \frac{N_{m_G}}{N_m} i(m_G) - \frac{N_{m_D}}{N_m} i(m_D),$$

où m_G et m_D désignent respectivement les noeuds fils gauche et droit du noeud m.

- La mesure d'impureté $i(m)$ la plus directe est le taux d'erreur (ou erreur de resubstitution), notée $i_r(m)$, qui mesure le pourcentage d'individus classés incorrectement lorsque les individus d'un noeud sont affectés à sa classe majoritaire, i.e. lorsque les individus du noeud m sont affectés à la classe $k(m)$ définie par :

$$k(m) = \arg\max_k \hat{p}_{mk}.$$

- La mesure $i_r(m)$ admet plusieurs expressions :

$$i_r(m) = 1 - \max_k(\hat{p}_{mk}) = 1 - \hat{p}_{mk(m)} = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbb{1}_{\left[y_i \neq k(m)\right]}.$$

Aprés simplification, on obtient :

$$\Delta i_r(m) = \frac{N_{m_G}}{N_m} \max_k \left(\hat{p}_{m_G k}\right) + \frac{N_{m_D}}{N_m} \max_k \left(\hat{p}_{m_D k}\right) - \max_k(\hat{p}_{mk}).$$

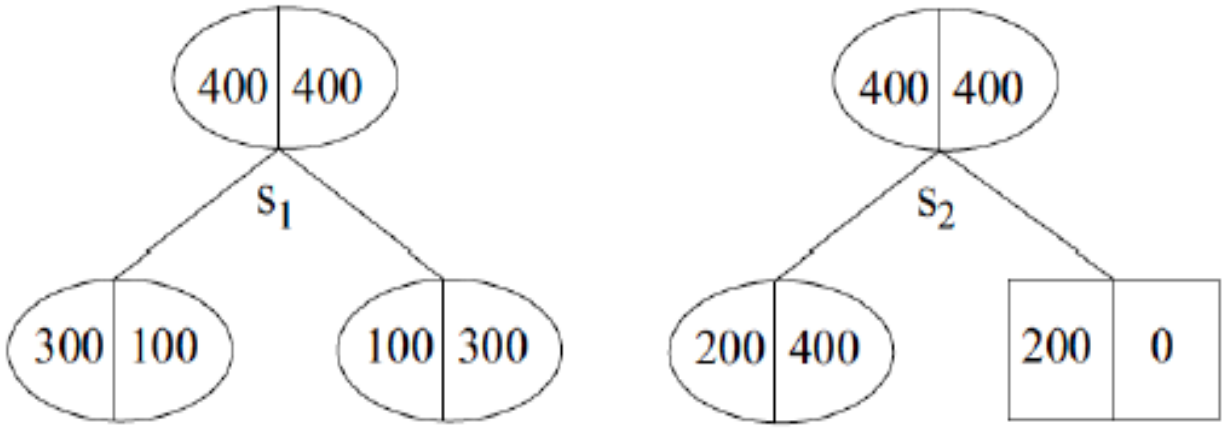

Figure – Exemple ”impureté” : cas où le taux d'erreur ne permet pas d'identifier la bonne coupure.

- Néanmoins le taux d'erreur n'est pas toujours approprié pour identifier la meilleure coupure d'un noeud comme le montre l'exemple présenté dans la figure ci-dessous.
- En effet, dans cet exemple, la mesure de la réduction du taux d'erreur est égale à 1/4 pour chacune des deux coupures, i.e. $\Delta i_r(s_1) = \Delta i_r(s_2) = 1/4$, alors que la coupure de droite s_2 est préférable puisqu'elle génère un noeud fils qui est pur.

Indice de Gini et mesure de l'entropie I

- Ainsi lors de la construction d'un arbre de classification, on préfère utiliser une autre mesure d'impureté, comme l'indice de Gini ou la mesure de l'entropie :

Indice de Gini :
$$i_g(m) = \sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right)$$

Entropie :
$$i_h(m) = -\sum_{k=1}^{K} \hat{p}_{mk} \log\left(\hat{p}_{mk}\right)$$

- Dans le cas de deux classes, i.e. lorsque $K = 2$, les expressions des mesures d'impureté se simplifient.

Indice de Gini et mesure de l'entropie II

- Si l'on note p la proportion d'une des classes dans le noeud m, les trois mesures d'impureté introduites ont pour expression :

$$\begin{aligned} i_r(m) &= 1 - \max(p, 1-p), \\ i_g(m) &= 2p(1-p), \\ i_e(m) &= -p\log(p) - (1-p)\log(1-p). \end{aligned}$$

- Le graphe de la figure suivante représente ces mesures en fonction de $p(0)$ où $p(0)$ désigne la proportion de l'une des deux classes, notée 0, dans le noeud considéré. Dans ce graphique, l'échelle des graphes de ces mesures a été modifiée de telle sorte que le maximum commun des trois mesures d'impureté, qui est atteint pour $p(0) = 0.5$, soit normalisé à 1.

Indice de Gini et mesure de l'entropie III

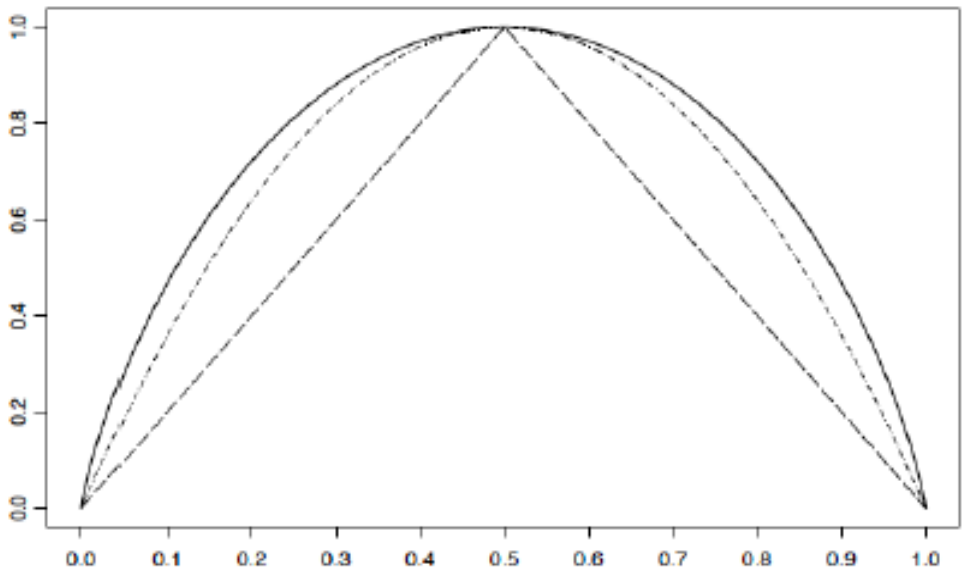

Figure – Graphe de l'entropie (trait plein), de l'indice de Gini (trait pointillé) et de l'indice du taux d'erreur (trait tireté) dans le cas de deux classes.

Indice de Gini et mesure de l'entropie IV

- Considérons à nouveau l'exemple "impureté". Lorsque l'on utilise les indices de Gini et de l'entropie pour calculer les mesures de réduction d'impureté relatives aux coupures s_1 et s_2, on obtient les résultats suivants :

$$\begin{aligned}
\Delta i_g(s_1) &= 2 \times \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} \times \left(2 \times \frac{1}{4} \times \frac{3}{4}\right) - \frac{1}{2} \times \left(2 \times \frac{1}{4} \times \frac{3}{4}\right) = \frac{1}{8} \\
\Delta i_g(s_2) &= 2 \times \frac{1}{2} \times \frac{1}{2} - \frac{6}{8} \times \left(2 \times \frac{1}{3} \times \frac{2}{3}\right) - \frac{1}{2} \times \left(2 \times 0 \times 1\right) = \frac{1}{6} \\
\Delta i_e(s_1) &= -\frac{2}{2} \log \frac{1}{2} + \frac{1}{4} \log \frac{1}{4} + \frac{3}{4} \log \frac{3}{4} = \frac{3}{4} \log 3 - \log 2 \approx 0.131 \\
\Delta i_e(s_2) &= -\frac{2}{2} \log \left(\frac{1}{2}\right) - \frac{6}{8}\left(-\frac{1}{3} \log \frac{1}{3}\right) - \frac{6}{8}\left(-\frac{2}{3} \log \frac{2}{3}\right) - \frac{2}{8}(-\log 1) - \frac{2}{8}(-0^+ \log 0^+) \\
\Delta i_e(s_2) &= \frac{2}{3} \log 2 - \frac{3}{4} \log 3 \approx 0.216
\end{aligned}$$

Indice de Gini et mesure de l'entropie V

- Ces calculs montrent que $\Delta i_g(s_1) < \Delta i_g(s_2)$ et que $\Delta i_e(s_1) < \Delta i_e(s_2)$. Par conséquent, dans cet exemple, chacun des indices de Gini et de l'entropie identifie bien la meilleure des deux coupures, i.e. s_2, contrairement à la mesure du taux d'erreur.

Règle d'affectation tenant compte des coûts

- $C_{kk'}$: coût de mauvaise affectation d'un individu de la classe k' dans la classe k.
- Le coût d'affectation d'une feuille m à une classe k est estimé par :

$$C_k = \sum_{k'=1}^{K} C_{kk'} \hat{p}_{mk'}$$

- Une feuille est affectée à la classe pour laquelle ce coût d'affectation est minimal : $c(m) = \min_k C_k$

Taux d'Erreur Apparente de classement

- A tout segment terminal m de l'arbre T associé à une classe k correspond une erreur de classement de la forme :

$$ER(k|m) = \sum_{k'=1}^{K} \hat{p}_{mk'}$$

avec $k \neq k'$ où $\hat{p}_{mk'}$ est la prop. d'ind. du segment m qui appartiennent à la classe k'.

- Taux d'Erreur Apparente de classement (TEA) associé à l'arbre vaut :

$$TEA(T) = \sum_{m \in T} \frac{n_m}{n} ER(k|m)$$

Les commandes I

Nous utilisons le fichier heart.txt. L'objectif est de prédire l'occurrence des maladies cardio-vasculaires (COEUR) : 12 variables prédictives et 270 observations.

```
>donnees <-
read.table(file="heart.txt",dec=".",header=TRUE)
>donnees.cnt <- rpart.control (minsplit = 1)
```

donnees.cnt stocke les paramètres de l'algo.
minsplit : nombre minimum d'ind. nécessaires à la création d'un noeud (valeur par défaut 20).

Les commandes II

- Construire et imprimer l'arbre calculé sur la totalité des individus :

```
>arbre.full <- rpart(coeur ~ ., data = donnees,
method = "class", control=donnees.cnt)
>print(arbre.full)
```

- Impression de l'arbre sous forme graphique :

```
>plot(arbre.full, uniform = TRUE, branch = 0.5,
margin = 0.1)
>text(arbre.full, all = FALSE, use.n = TRUE)
```

Les sorties

L'objet retourné est de la classe rpart. L'argument method="class" est optionnel (automatique par défaut car la variable prédite est de type facteur).
Impression de l'arbre sous forme textuelle pour un arbre de discrimination :

```
n= 270
node), split, n, loss, yval, (yprob)
* denotes terminal node
1) root 270 120 absence (0.55555556 0.44444444)
  2) type_douleur=A,B,C 141 29 absence (0.794 0.205)
    4) depression < 19.5 125 19 absence (0.848
0.152)
      8) age< 55.5 76 5 absence (0.934 0.065) *
      9) age≥ 55.5 49 14 absence (0.714 0.285)
```

Interprétation

A chaque noeud, on a :

- Le numéro du noeud : 2) (noeuds de gauche et droite numérotés 2x et 2x + 1 si père numéroté x).
- le critère de split (ou root pour la racine) : `type_douleur=A,B,C`
- le nombre total d'instances pour le noeud : 141
- le nombre d'instances mal classées (0 si toutes les instances sont bien prédites) : 29
- la valeur prédite (donc majoritaire) de la variable à prédire : `absence`
- entre parenthèses, les proportions d'instances bien et mal prédites : (0.794 0.205)
- une '*' si c'est un noeud terminal.

On peut avoir tout le détail en faisant : `summary(arbre.full)`.

Prédiction et Matrice de Confusion

```
>pred <- predict(arbre.full, newdata = donnees,
type = "class")
>mc <- table(donnees$coeur,pred) # matrice de confusion
>print(mc)
>err.resub <- 1.0 - (mc[1,1]+mc[2,2])/sum(mc)
>print(err.resub) #erreur de resubstitution
```

Arbre optimal I

- De façon similaire au cas des arbres de régression, la méthode CART commence par construire un arbre de grande taille, T_0.
- Ici l'arbre T_0 est un arbre obtenu par divisions successives des noeuds créés et en choisissant chaque nouvelle coupure de telle sorte que la réduction de l'impureté soit maximale, l'impureté étant mesurée soit par l'indice de Gini soit par l'entropie.
- L'arbre T_0 est obtenu par ce processus de partitionnement récursif qui s'arrête dès que tous les noeuds terminaux de l'arbre construit sont des noeuds purs (ou de taille inférieure à un seuil fixé).
- Puis, CART applique un élagage de type "coût-complexité" analogue à celui déjà présenté pour les arbres de régression, le critère (1) étant ici remplacé par le critère $C_\alpha(T)$, défini ci-après, qui est adapté à la nature qualitative de la variable Y à expliquer.

Arbre optimal II

- Pour tout $\alpha \geq 0$, on pose :

$$C_\alpha(T) = \sum_{m \in T} \frac{n_m}{n} ER(k|m) + \alpha |\, \widetilde{T}\, |,$$

où $R(T)$ désigne le pourcentage d'individus (dans l'échantillon d'apprentissage) qui sont mal classés par l'arbre T.

- Pour sélectionner l'arbre optimal noté $T_{\widehat{\alpha}}$, la méthode CART procède comme pour les arbres de régression, i.e. en appliquant la validation croisée pour évaluer la performance réelle d'un arbre élagué T_α. La valeur $\widehat{\alpha}$ sélectionnée est celle qui optimise cette performance.

Arbre optimal III

Algorithm 1 Sélection d'arbre ou élagage par validation croisée

Construction de l'arbre maximal $A_{\max}$
Construction de la séquence $A_K \dots A_1$ d'arbres emboîtés associée à une Séquence de valeurs de pénalisation γ_κ
for $v = 1, \dots, V$ **do**
Pour chaque échantillon, estimation de la séquence d'arbres associée la séquence des pénalisations γ_κ
Estimation de l'erreur sur la partie restante de validation de l'échantillon
end for
Calcul de la séquence des moyennes de ces erreurs
L'erreur minimale désigne la pénalisation γ_{Opt} optimale
Retenir l'arbre associé à γ_{Opt} dans la séquence $A_K \dots A_1$

Elagage et determination de l'arbre optimal sur r I

- **Elagage :** Il est possible d'élaguer automatiquement un arbre de décision avec la fonction `prune()` (qui veut dire élaguer, en anglais).

- **Détermination de l'abre optimal :** Par défaut, rpart() effectue un élagage de l'arbre et une validation croisée à 10 plis sur chaque arbre élagué. Les mesures effectuées au long de cette procédure sont stockées dans une table dénommée la cptable.

- La commande `arbre.full$cptable` ou `printcp(arbre.full)` affiche cette table qui va nous permettre de répondre à la question précédente (quel est l'arbre optimal ?) :

Elagage et determination de l'arbre optimal sur r II

```
> arbre.full$cptable
         CP  nsplit rel error    xerror       xstd
1 0.44166667      0 1.0000000 1.0000000 0.06804138
2 0.06666667      1 0.5583333 0.6416667 0.06182454
3 0.03333333      3 0.4250000 0.4916667 0.05658536
4 0.02777778      5 0.3583333 0.4916667 0.05658536
5 0.01000000      8 0.2750000 0.4500000 0.05477226
```

- `cp` paramètre de complexité
- `rel error` mesure l'erreur apparente (erreur d'entraînement).
- `xerror` mesure le taux d'erreur dans la validation croisée à 10 plis que l'on considère comme un estimateur correct de l'erreur réelle.
- `xstd` est l'écart-type de l'erreur de validation croisée.

Elagage et determination de l'arbre optimal sur r III

- L'arbre qui nous intéresse est celui qui minimise `xerror + xstd` (l'erreur moyenne estimée + 1 écart-type). Si plusieurs arbres minimise cette valeur, on prend toujours le plus petit (à performances équivalentes, on choisit le plus petit modèle).
- L'arbre que nous obtenons par la commande rpart() correspond à la dernière ligne de la cptable : c'est le plus grand. Les lignes précédentes correspondent aux différents élagages de cet arbre qui produisent des arbres de plus en plus petits, jusqu'à obtenir une racine-feuille.

Elagage et determination de l'arbre optimal sur r IV

- Les autres arbres peuvent aussi être obtenus, pour cela il faut indiquer le cp dans la fonction prune() :
  - l'arbre le plus élagué (ligne 1 de la cptable) s'obtient pour cp dans l'intervalle]0.44166 ; 1] ;
  - le deuxième (un peu moins élagué) (ligne 2 de la cptable) s'obtient pour cp dans l'intervalle]0.06666 ; 0.44166] ;
  - le troisième (encore un peu moins élagué) (ligne 3 de la cptable) s'obtient pour cp dans l'intervalle]0.03333 ; 0.06666] ;
  - etc.
    Notez bien que la borne inférieure de l'intervalle est exclus.

Pour conclure... I

Le succés des arbres de décision réside en grande partie dans ses caractéristiques :

- **Lisibilité du modèle de prédiction.** L'arbre de décision fournit des règles explicites de classement. Cette caractéristique est très importante, car le travail de l'analyste consiste aussi à faire comprendre ses résultats afin d'emporter l'adhésion des décideurs.
- **Sélection automatique des variables.** Le modèle sélectionne automatiquement les variables discriminantes dans un fichier de données contenant un très grand nombre de variables potentiellement intéressantes. En ce sens, un arbre de décision constitue une technique exploratoire privilégiée pour appréhender de gros fichiers de données.

Pour conclure... II

- **Robustesse du modèle.** Les résultats ne sont pas trés affectés par les données erronées ou les valeurs aberrantes. La méthode permet de gérer les données manquantes aussi bien dans la construction de l'arbre que dans l'application de la règle à un nouveau sujet.

- **Simplicité du modèle**. Même principe, même méthode, même algorithme qui sont mis en oeuvre pour analyser une variable nominale (**discrimination**) et une variable continue (**régression**).

Principal inconvénient :

Pour conclure... III

- **Sensibilité aux perturbations.** Les règles d'affectation peuvent être assez sensibles à de légères perturbations des données (individu ayant légèrement dépassé le seuil pour une variable surtout quand celle-ci est placée près du sommet...)