

STA211 : Arbres de régression et de classification

Vincent Audigier, Ndèye Niang-Keita

15 avril, 2019

- 1 Contexte
- 2 Généralités
 - 2.1 Principe
 - 2.2 Représentations graphiques du partitionnement
- 3 Construction d'un arbre
 - 3.1 Arbres de régression
 - 3.1.1 Modèle
 - 3.1.2 Recherche d'une partition optimale
 - 3.1.3 Elagage
 - 3.2 Arbre de classification
 - 3.3 Cas des variables explicatives qualitatives
- 4 Evaluation des performances
- 5 Propriétés
- 6 Compléments
 - 6.1 Autres algorithmes
 - 6.1.1 C5.0
 - 6.1.2 CHAID
 - 6.1.3 AID
 - 6.2 Forêts aléatoires
- Références

1 Contexte

Les arbres font partie des méthodes supervisées, ils ont pour but d'expliquer et/ou prédire une variable réponse Y à partir d'un ensemble de variables explicatives (X_1, \dots, X_p). Par ailleurs, ce sont des méthodes non-paramétriques. En ce sens, la nature du lien entre X et Y n'est pas définie a priori et plus généralement, aucune hypothèse sur la distribution des données n'est nécessaire, bien que les performances de la méthode restent dépendantes des données considérées. Par ailleurs, les arbres présentent l'avantage de pouvoir être mis en oeuvre sur des jeux de nature et de dimensions variées, i.e. que les variables d'entrée (X_1, \dots, X_p), comme la variable de sortie Y , peuvent être de nature quantitative, ou qualitative, et le nombre d'individus n peut être supérieur ou inférieur au nombre de variables p . On parlera d'*arbre de régression* si Y est quantitative, et d'*arbre de classification* si Y est qualitative.

2 Généralités

2.1 Principe

Les arbres font partie des méthodes dites de segmentation. Le principe est de partitionner l'espace des

variables explicatives $X = (X_1, \dots, X_p)$, appelé l'espace des entrées et noté \mathcal{X} (on supposera \mathcal{X} inclu dans \mathbb{R}^p), en R régions et d'associer un modèle simple à chacune d'entre elle (généralement une constante), permettant ainsi de prédire une sortie Y appartenant à \mathcal{Y} , l'espace des sorties (par exemple \mathbb{R} dans un cadre de régression). Différentes méthodes de partitionnement existent (CART Breiman et al. (1984) ; C4.5 Quinlan (1993) ; etc), nous présentons ici la méthode CART (Classification and regression trees) qui consiste à partitionner de façon récursive chaque région en deux autres régions (on parle alors d'*arbre binaire*). Les régions de \mathcal{X} obtenues sont des parallélépipèdes rectangles (ou simplement des rectangles si l'espace des entrées est de dimensions $p = 2$). La valeur prédictive pour chaque région est alors la moyenne des valeurs de Y dans le cas d'arbre de régression ($\mathcal{Y} \subset \mathbb{R}$), ou la modalité la plus fréquente dans le cas d'une variable qualitative ($\mathcal{Y} = \{1, \dots, M\}$).

2.2 Représentations graphiques du partitionnement

La Figure 2.1 illustre les deux représentations de la partition obtenue via un arbre dans le cas où $p = 2$ et les variables d'entrée sont quantitatives.

Le graphique de gauche représente les 3 régions ($R = 3$) qui constituent la partition de l'espace des entrées. De façon équivalente, le graphique de droite permet de lire de façon séquentielle la règle d'affectation d'un individu à une région. Il se lit de haut en bas, de la *racine* aux *feuilles* en passant par les *noeuds intermédiaires* (les feuilles sont aussi appelées *noeuds terminaux*). La racine de l'arbre représente l'ensemble des individus, c'est le premier noeud. Celui-ci est ensuite divisé en deux nouveaux noeuds appelés *noeuds fils*. Sur cet exemple, la division est effectuée selon la variable X_1 : les individus dont la valeur est inférieure à t_1 sont affectés au noeud fils de gauche et ceux supérieurs ou égaux à t_1 sont affectés au noeud fils de droite. Le noeud fils de gauche est un *noeud terminal*, appelé également *feuille*. Il définit la région $\mathcal{R}_1 (X|X_1 < t_1)$. Le noeud de droite est un noeud intermédiaire. Ce dernier est ensuite subdivisé en deux selon la variable X_2 en fonction du seuil t_2 . Ses deux noeuds fils sont des feuilles qui définissent les régions \mathcal{R}_2 et \mathcal{R}_3 .

Ces représentations synthétiques et simples constituent un point fort des arbres.

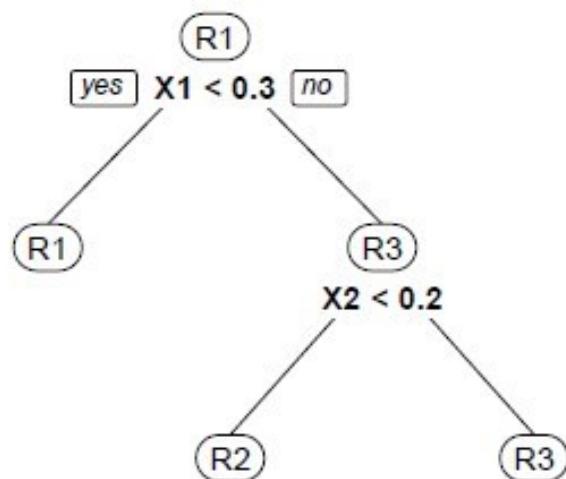
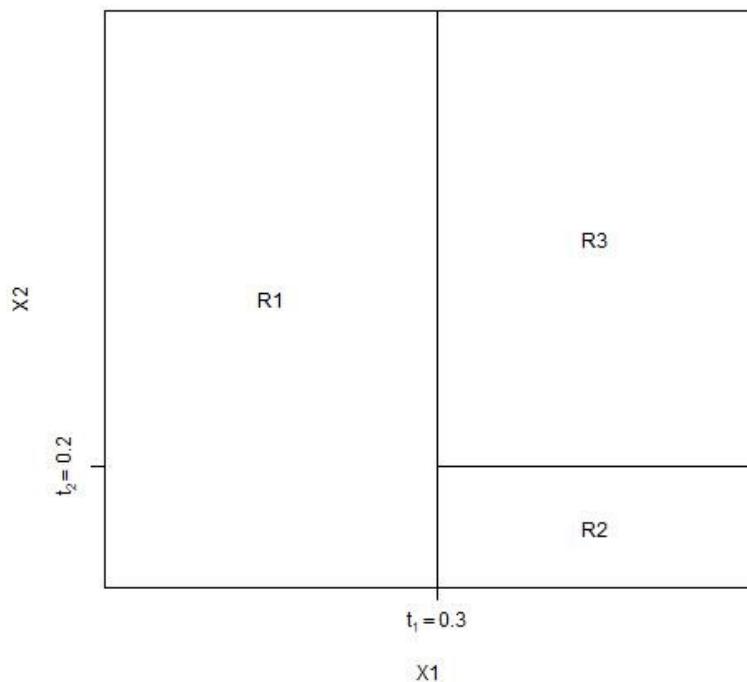


Figure 2.1: Deux représentations du partionnement obtenu via un arbre dans le cas d'un espace d'entrée de dimension deux

3 Construction d'un arbre

Dans cette partie nous supposons toujours dans un premier temps que les variables explicatives sont de nature quantitative. Dans ce cadre, nous présentons les arbres de régression, puis les arbres de classification. Par la suite nous traitons le cas où les variables explicatives sont de nature qualitative.

3.1 Arbres de régression

3.1.1 Modèle

Indexant par i ($1 \leq i \leq n$) les individus, par j ($1 \leq j \leq p$) les variables d'entrée, et par r ($1 \leq r \leq R$) les régions de l'espace des entrées, et notant x un point de cet espace, un arbre peut être modélisé par :

$$y = f(x) = \sum_{r=1}^R c_r I(x \in \mathcal{R}_r)$$

avec c_r une constante correspondant à la valeur de y pour la région \mathcal{R}_r et $I(x \in \mathcal{R}_r)$ la fonction indicatrice de l'événement $x \in \mathcal{R}_r$. Autrement dit, si x est un élément de la région \mathcal{R}_l , alors $I(x \in \mathcal{R}_r)$ vaut 1 si $r = l$ et 0 sinon, ainsi on a $f(x) = c_l$.

Un arbre est donc caractérisé par ses régions $(\mathcal{R}_1, \dots, \mathcal{R}_R)$ et les constantes (c_1, \dots, c_R) qu'il convient de définir. Dans un cadre de régression, on pourrait donc chercher l'arbre qui minimise le critère des moindres carrés :

$$\mathcal{C}_{MC} = \sum_{i=1}^n (y_i - f(x_i))^2$$

Pour une partition de l'espace des entrées donnée (autrement dit, pour $(\mathcal{R}_1, \dots, \mathcal{R}_R)$ connu), la valeur de c_r qui minimise ce critère est donnée par la moyenne des valeurs de Y pour les individus au sein de la région r : $\hat{c}_r = \text{moy}(y_i | x_i \in \mathcal{R}_r)$.

Malheureusement, la détermination des régions est beaucoup moins évidente dans la mesure où on ne sait pas minimiser simplement le critère des moindres carrés sur l'ensemble des partitions possibles. Le coût algorithmique étant trop important, on approche la solution du problème d'optimisation en procédant pas à pas. On commence par rechercher la meilleure division dyadique de \mathcal{X} au sens du critère \mathcal{C}_{MC} . Puis on divise à nouveau les deux régions obtenues en deux, de façon à minimiser ce même critère, et ainsi de suite.

3.1.2 Recherche d'une partition optimale

La division d'une région en deux nécessite d'identifier à la fois la variable j de l'espace d'entrée selon laquelle la division est effectuée, ainsi qu'un seuil s associé pour effectuer la division. Pour cela, on recherche j et s tels que la dispersion au sein des noeuds fils soit la plus petite possible, on dit que l'on cherche à minimiser l'hétérogénéité des noeuds fils (ou de façon équivalente à maximiser leur homogénéité). Formellement, on recherche

$$\arg \min_{s,j} \left[\min_{c_1} \sum_{x_i \in \mathcal{R}_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in \mathcal{R}_2(j,s)} (y_i - c_2)^2 \right]$$

avec $\mathcal{R}_1(j,s) = \{X | X_j \leq s\}$ et $\mathcal{R}_2(j,s) = \{X | X_j > s\}$.

La solution de ce problème est facile à identifier : il suffit dans un premier temps d'identifier le meilleur seuil pour chaque variable X_1, X_2, \dots, X_p . On obtient ainsi une valeur du critère des moindres carrés pour chacune. Il s'agit alors dans un deuxième temps de choisir parmi les p variables celle qui minimise le critère. Une fois les deux noeuds fils déterminés, la procédure est réitérée sur chacun d'entre eux. La procédure s'arrête si le noeud est homogène, i.e. que tous les individus du noeud prennent la même valeur, ou qu'il

n'existe plus de *divisions admissibles*, i.e. telles qu'aucun des noeuds fils ne soit vide, ou que le nombre d'individus par noeud terminal est inférieur à un seuil prédéfini (5 par exemple).

Ainsi, cet algorithme permet d'identifier les régions de l'espace et d'attribuer à chacune d'entre elles une valeur estimée de \hat{c}_r . Il devient alors facile de prédire une nouvelle valeur de Y pour un nouvel individu : on regarde dans quelle région se situe cet individu et on prédit alors sa sortie y par la valeur \hat{c}_r correspondante.

La Figure 3.1 représente un arbre de régression ajusté sur un jeu de données comportant 2000 individus et 100 variables quantitatives. Celui-ci possède 9 noeuds terminaux. Par exemple, un individu tel que $X_1 = X_2 = \dots = X_{100} = 0$ appartient à la région définie par $(X_{23} < 0.18, X_{75} > -1.3, X_{63} < 0.86, X_5 > -2.4)$. La moyenne des valeurs observées de Y pour cette région vaut -0.24 , on a $\hat{y} = -0.24$ pour cet individu.

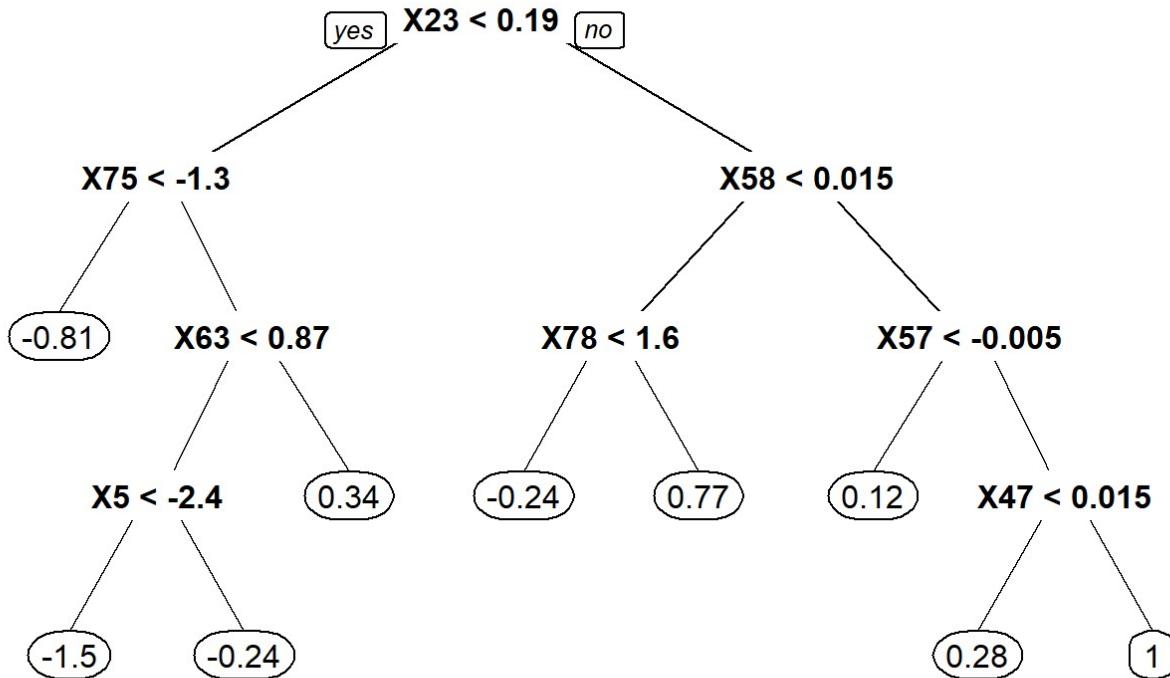


Figure 3.1: Exemple d'arbre de régression avec variables explicatives quantitatives

Remarque

Certains algorithmes (autres que CART) ont également un critère supplémentaire qui consiste à arrêter la procédure si le gain à effectuer une nouvelle division est inférieur à un certain seuil (seuil qu'il faut définir, ce qui constitue une difficulté supplémentaire). Les approches utilisant cette règle d'arrêt supplémentaire sont critiquables dans la mesure où ce n'est pas parce qu'une première division ne diminue pas de façon importante le critère de division, que les divisions successives, elles, ne le diminueront pas par la suite. L'approche utilisée dans CART consiste elle à développer l'arbre maximal puis à élaguer l'arbre maximal obtenu.

3.1.3 Élagage

Plus l'espace des entrées est partitionné en un grand nombre de régions, plus le critère des moindres carrés diminue et donc plus la partition obtenue est optimale au sens de ce critère. Néanmoins, il n'est pas souhaitable de construire des arbres avec un très grand nombre de noeuds (on parle d'*arbres profonds*). En effet, si celui-ci est trop important, on observe un problème de *sur-apprentissage*, c'est-à-dire que le nombre de paramètres estimés est trop important relativement au nombre de données disponibles, ce qui conduit à des arbres avec peu de biais, mais avec une grande variabilité de prédiction (mauvais compromis biais-variance (<https://par.moodle.lecnam.net/mod/resource/view.php?id=97911>)). Aussi, l'interprétation du lien entre la variable réponse Y et les variables explicatives devient trop complexe.

Pour ces raisons il est important d'*élaguer* l'arbre. Partant d'un arbre profond à R noeuds terminaux, noté A_R , l'élagage consiste à rechercher un *sous-arbre* A de A_R avec un meilleur compromis biais-variance. On appelle A *sous – arbre* de A_R , n'importe quel arbre qui peut être obtenu en fusionnant des noeuds intermédiaires de A_R .

Le problème est que le nombre de sous-arbres peut être extrêmement grand et qu'il n'est donc pas envisageable d'utiliser une approche classique de sélection de modèle afin d'identifier le meilleur d'entre eux. Fort heureusement, Breiman (2001) a montré qu'il n'est pas nécessaire de considérer tous les sous-arbres, une séquence bien choisie de sous-arbres emboîtés (i.e. élagués les uns dans les autres) suffit.

3.1.3.1 Définition d'une séquence de sous-arbres emboîtés

Il s'agit d'établir une suite d'arbres A_1, \dots, A_K tous élagués de A_R , telle que les arbres de cette suite minimisent un critère de moindres carrés pénalisé. Cette suite est obtenue de manière itérative en coupant des branches à chaque étape.

Le critère pénalisé utilisé est appelé *critère coût-complexité*. Il s'agit de trouver l'arbre qui minimise l'erreur d'ajustement en favorisant les arbres avec peu de noeuds. Ce critère s'écrit :

$$C_\alpha(A) = \underbrace{\sum_{r=1}^{|A|} \sum_{x_i \in \mathcal{R}_r} (y_i - \hat{c}_r)^2}_{\text{erreur d'ajustement}} + \underbrace{\alpha |A|}_{\text{pénalité}}$$

avec $|A|$ le nombre de noeuds terminaux de A .

Cette quantité dépend d'un paramètre $\alpha \geq 0$ qui règle la pénalité : si $\alpha = 0$ alors l'arbre qui minimise $C_\alpha(A)$ est A_R , car on ne pénalise pas les arbres avec un grand nombre de noeuds. Au contraire, plus α est grand, plus on recherche un arbre avec peu de noeuds.

Etant donné un arbre A_R , l'algorithme d'élagage est donné ci-dessous. Dans cet algorithme, pour tout noeud interne a d'un arbre A , on note A_a la branche de A issue du noeud a (contenant tous les descendants du noeud a) et l'erreur du noeud est donnée par $C(a) = \sum_{x_i \in a} (y_i - \bar{y}_a)^2$, i.e. la variance au sein du noeud a :

- Initialisation

- $\alpha_1 = 0, A_1 = \arg \min_{A \text{ élagué de } A_R} C(A)$

- $A \leftarrow A_1$
- $k \leftarrow 1$

- Itération

- Tant que $|A| > 1$

- Calculer $\alpha_{k+1} = \min_{a \text{ noeud interne de } A} \frac{C(a) - C(A_a)}{|A_a| - 1}$

- Élaguer toutes les branches A_a de A telles que $C(A_a) + \alpha_{k+1}|A_a| = C(a) + \alpha_{k+1}$
- Prendre A_{k+1} le sous-arbre élagué ainsi obtenu
- $A \leftarrow A_{k+1}$
- $k \leftarrow k + 1$

L'initialisation sert à définir le premier arbre de la séquence. Il peut être différent de A_R . En effet, par construction, la somme des variances de deux noeuds fils est toujours inférieure ou égale à celle du noeud père. En cas d'égalité, on n'a pas d'intérêt à considérer ces noeuds fils car ils augmentent la complexité de l'arbre sans en améliorer la qualité de l'ajustement. Le premier arbre de la séquence est donc le sous-arbre de A_R qui ajuste aussi bien les données que A_R avec un nombre minimal de feuilles.

Ensuite, on calcule α_{k+1} , le paramètre coût-complexité minimum sur tous les noeuds interne de l'arbre courant. Supposons que l'arbre courant ait ℓ noeuds terminaux, alors il possède $\ell - 1$ noeuds intermédiaires (racine inclue). Pour chacun d'entre eux, on peut calculer $C(a)$ qui est la variance au sein du noeud a , $C(A_a)$ la variance dans les noeuds terminaux issus de la branche A_a et $|A_a|$ le nombre de noeuds terminaux issus de la branche A_a . α_{k+1} est alors la valeur minimale des paramètres coût-complexité calculés pour chacun des noeuds. On supprime alors toutes les branches qui proviennent d'un noeud intermédiaire ayant cette valeur de α_{k+1} pour paramètre coût complexité. On recommence ensuite l'opération à partir du nouvel arbre obtenu.

A l'issue de cet algorithme, on obtient une suite d'arbres emboîtés A_1, \dots, A_K et une suite de paramètres $\alpha_1, \dots, \alpha_K$

3.1.3.2 Choix du meilleur sous-arbre

Une fois la séquence définie, il reste à identifier le sous-arbre au meilleur compromis biais-variance parmi les K retenus.

Une première façon de faire est d'évaluer les performances prédictives de chacun des sous-arbres de la séquence à partir d'un échantillon de validation (voir cours précédent (<https://par.moodle.lecnam.net/mod/resource/view.php?id=97911>)). Ceci n'est envisageable que si l'échantillon d'apprentissage est suffisamment grand. La plupart des logiciels procèdent directement par validation croisée.

Dans le cas où un échantillon de validation est utilisé, il est naturel de retenir le sous-arbre d'erreur test minimale, mais il est aussi courant d'utiliser la *règle d'un écart type*. En effet, il est fréquent que l'erreur calculée sur l'échantillon test diminue avec la taille des sous-arbres, puis passe par un plateau puis remonte. Afin de ne pas retenir le sous-arbre au milieu du plateau (qui serait assez complexe sans vraiment diminuer l'erreur de test) on pourra retenir celui, moins complexe, dont l'erreur de test est un écart type inférieure à celle de l'arbre au plus faible taux d'erreur.

Notons que l'approche par validation-croisée est relativement subtile, nous en donnons l'idée générale et renvoyons vers Nakache and Confais (2003) pour plus de précisions (un extrait en ligne est disponible dans le cours (http://maths.cnam.fr/IMG/pdf/Segmentation_2013_cle0b5466.pdf) de P-L. Gonzalez). Elle consiste à procéder par $k - fold$. On note V le nombre de blocs partitionnant l'échantillon. Pour chacun des V blocs, on établit les séquences de sous-arbres, comme expliqué précédemment. On obtient donc V séquences d'arbres (notée S_1, \dots, S_V respectivement) en plus de la séquence préalablement obtenue à partir de l'ensemble des données (notée S). Par ailleurs, pour chacune d'entre elles, on peut calculer une erreur sur le bloc n'ayant pas été utilisé pour construire S_v ($1 \leq v \leq V$). Pour choisir, le sous-arbre de S au meilleur compromis, on identifie au sein de chaque séquence S_v , l'arbre dont le coût complexité est le plus proche de chacun des arbres de S . Ceci permet d'associer à chaque arbre de S , V erreurs. Il s'agit alors de moyenner ces erreurs et de retenir le sous-arbre de S dont l'erreur est la plus faible.

NB : Pour effectuer un élagage, certains logiciels demandent de spécifier le nombre de noeuds, tandis que d'autres demandent de spécifier le paramètre coût-complexité α . En fait, cela est équivalent car pour chaque sous-arbre, il existe un paramètre coût-complexité tel que ce sous-arbre minimise le critère coût-complexité.

La Figure 3.2 illustre la procédure de l'élagage. Le schéma de gauche représente un arbre de régression ajusté sur un jeu de données comportant 2000 individus et 100 variables quantitatives. Celui-ci possède 9 noeuds terminaux. Afin de trouver un meilleur compromis biais-variance, la sélection d'un sous-arbre est effectuée par validation croisée par une méthode de 10-fold. On voit que le paramètre coût-complexité optimal est obtenu pour $\alpha = 62$, ce qui correspond à un sous-arbre avec 4 noeuds terminaux (graphique de droite). Le schéma central représente alors l'arbre élagué de façon à obtenir un arbre avec seulement 4 noeuds terminaux.

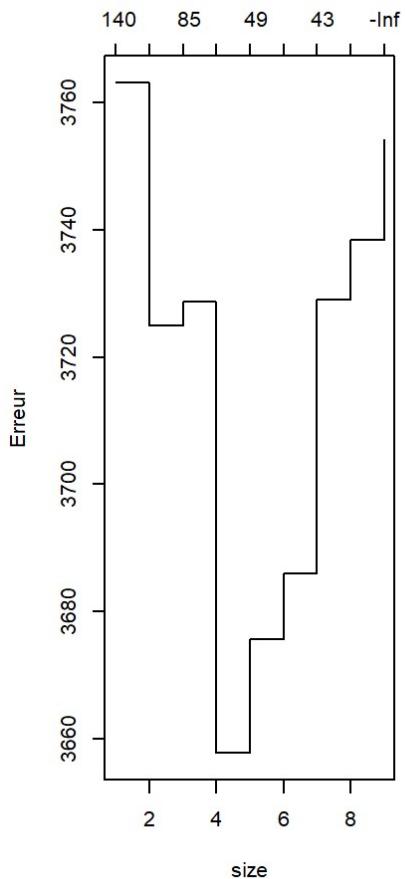
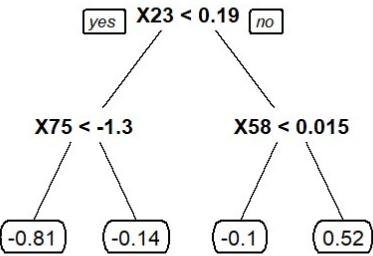
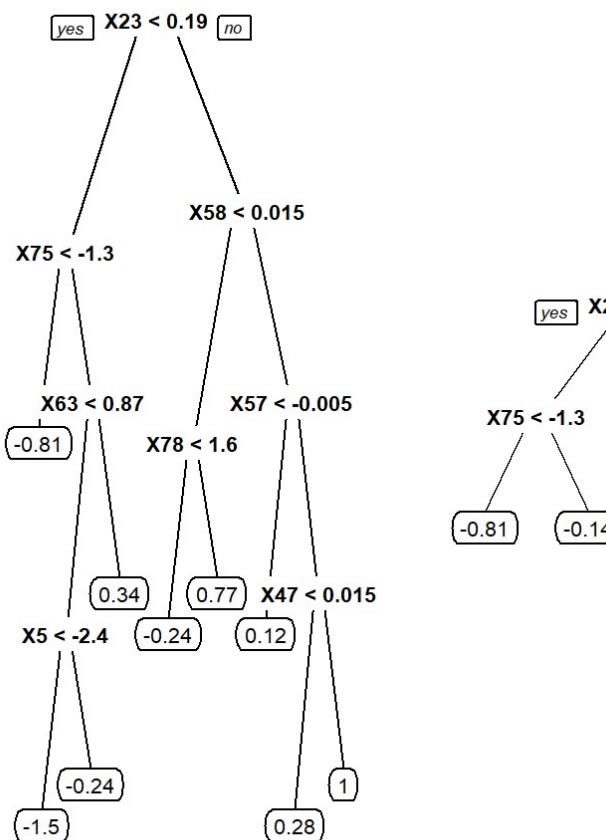


Figure 3.2: Arbre avant élagage, arbre élagué avec 4 noeuds, erreur de validation croisée en fonction du paramètre coût-complexité.

3.2 Arbre de classification

Quand la variable réponse Y est qualitative à M modalités ($M \geq 2$), la construction de l'arbre de classification s'effectue de façon similaire à celle d'un arbre de régression. Il convient néanmoins de modifier le critère mesurant la qualité d'ajustement de l'arbre.

Soit n_r , le nombre d'individus dans la région \mathcal{R}_r , et \hat{p}_{rm} la proportion d'individus prenant la modalité m ($1 \leq m \leq M$) au sein de la région r :

$$\hat{p}_{rm} = \frac{1}{n_r} \sum_{x_i \in \mathcal{R}_r} I(y_i = m)$$

Plusieurs critères mesurant l'hétérogénéité du noeud peuvent être envisagés :

- l'erreur de mauvais classement :

$$\frac{1}{n_r} \sum_{x_i \in \mathcal{R}_r} I(y_i \neq \hat{y}_i)$$

- l'indice de Gini :

$$\sum_{m \neq m'} \hat{p}_{rm} \hat{p}_{rm'} = \sum_{m=1}^M \hat{p}_{rm} (1 - \hat{p}_{rm})$$

- la déviance (ou entropie de Shannon)

$$-\sum_{m=1}^M \hat{p}_{rm} \log \hat{p}_{rm}$$

L'erreur de mauvais classement est simplement la proportion d'individus de la région \mathcal{R}_r pour lesquels la valeur y_i est différente de la valeur \hat{y}_i , donnée par l'arbre. L'indice de Gini lui peut être vu comme une mesure d'hétérogénéité au sein de la région. En effet, pour le cas $M = 2$, on reconnaît dans son expression l'estimation de la variance d'une loi de Bernoulli (au facteur 2 près). Enfin, le critère de déviance est assez proche de l'indice de Gini. Dans le cas $M = 2$, il correspond à la déviance d'une loi de Bernoulli (au coefficient n_r près), i.e. moins 2 fois le log de la vraisemblance pour cette loi. La déviance quantifiant l'écart entre la distribution observée au sein du noeud et celle qu'on aurait obtenue si les modalités étaient réparties au hasard, le critère fournit également une mesure de dispersion au sein du noeud.

La Figure 3.3 compare les différents critères d'impureté pour une variable Y à deux classes. On représente l'erreur en fonction de la proportion de la modalité 1 au sein du noeud (notée p_1). La mesure de la déviance a été normalisée pour passer par le point de coordonnées (0.5,0.5). Tous ces critères valent 0 si le noeud est homogène, et atteignent leur valeur maximale quand les modalités sont équi-réparties. Par ailleurs, on constate la proximité entre l'indice de Gini et la déviance.

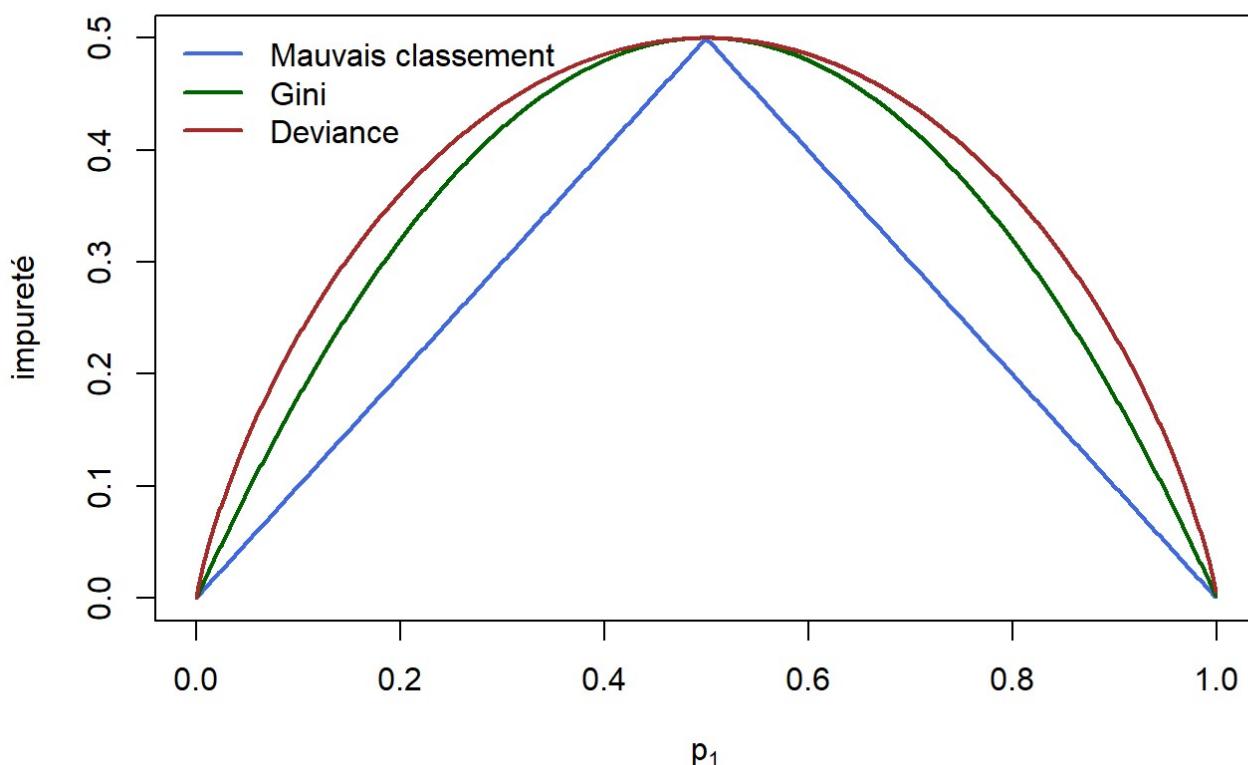


Figure 3.3: Valeurs des critères d'impureté pour une variable Y à deux classes, en fonction de la proportion d'une des modalités au sein du noeud. La mesure de la déviance a été normalisée pour passer par le point de coordonnées (0.5,0.5)

L'erreur de mauvais classement correspond à ce que l'on pourrait attendre de l'arbre : classer le mieux possible les individus dans la classe qui leur correspond. Mais construire les divisions successives selon ce critère rend l'arbre instable et augmente ainsi la variabilité de prédiction de l'arbre. C'est l'une des raisons pour lesquelles on lui préfère le critère de Gini ou le critère de déviance qui contrôlent mieux la variabilité autour de la prédiction donnée par les noeuds fils.

Une fois l'arbre construit, l'élagage peut se faire selon l'un des 3 critères. Généralement, on utilise cette fois l'erreur de mauvais classement.

Remarque

Dans certains contextes, on pourra préférer pondérer les erreurs de façon à affecter un poids plus fort à certaines erreurs plutôt qu'à d'autres. Dans ce cas, on introduit une fonction de coût $\gamma(m, m')$ dans les critères d'impureté indiquant le coût à attribuer la classe m' à un individu de la classe m . Quand la classification est correcte, on choisira généralement un coût nul ce qui implique $\gamma(m, m) = 0$ pour tous m dans $\{1, \dots, M\}$. Ainsi, l'indice de Gini se réécrit :

$$\sum_{m \neq m'} \gamma(m, m') \hat{p}_{rm} \hat{p}_{rm'}$$

et la règle d'affectation d'un individu d'un noeud terminal à une classe n'est plus la modalité m la plus fréquente au sein du noeud, mais celle qui minimise le coût moyen :

$$\sum_{m=1}^M \gamma(m, m') \hat{p}_{rm}$$

On notera que dans le cas où l'on choisit des coûts identiques ($\gamma(m, m') = 1$ pour tous $m \neq m'$), c'est bien la modalité la plus fréquente qui minimise ce coût moyen.

Il est important de noter que cette pondération n'a aucun effet sur la construction des noeuds si la variable Y est binaire. En effet, le critère de Gini pondéré se réécrit

$$\underbrace{(\gamma(m, m') + \gamma(m', m)) \hat{p}_{rm} \hat{p}_{rm'}}_{\text{constante}}$$

et ne fait donc que multiplier le critère par une constante, ne changeant en rien la règle de division qui minimise l'indice.

3.3 Cas des variables explicatives qualitatives

Les arbres permettent aussi de prendre en compte des variables explicatives qualitatives. Ceci complexifie l'identification de la variable utilisée pour diviser le noeud ainsi que la règle de division pour minimiser l'impureté des noeuds fils (la notion de "seuil" n'a notamment plus de sens si la variable n'est plus ordonnée). En effet, soit n_r le nombre d'individus d'un noeud, alors pour une variable explicative quantitative, le nombre de divisions dyadiques possibles du noeud est $n_r - 1$, mais pour une variable explicative à M modalités, ce nombre de divisions est $2^{M-1} - 1$ ce qui très grand quand M est grand. En toute généralité, il faut alors calculer le critère d'impureté pour chaque découpage et retenir celui qui minimise l'impureté des noeuds fils.

Ainsi, les variables qualitatives non-ordonnées offrent beaucoup plus de possibilités de découpage du noeud que les autres types de variables (quantitatives ou qualitatives ordonnées), c'est pourquoi elles ont mécaniquement tendance à être privilégiées pour découper le noeud, favorisant un important sur-ajustement.

NB : L'utilisation des critères de Gini ou de déviance offrent des astuces calculatoires qui permettent de déterminer plus efficacement la division optimale pour le cas où Y est binaire (à valeurs dans $\{0, 1\}$ par exemple). En effet, dans ce cas la division optimale du noeud s'obtient en ordonnant les M modalités de la variable explicative selon la proportion d'individus prenant la modalité 1, puis en découplant l'échantillon selon un seuil. Le même procédé est utilisé pour une variable qualitative ordonnée, quel que soit le nombre de modalités de la variable Y .

4 Evaluation des performances

Une fois l'arbre construit, il convient d'en évaluer les performances prédictives. Le principe est de prédire les sorties pour un nouvel échantillon n'ayant pas servi à construire l'arbre. On compare alors les valeurs prédites et les valeurs observées sur cet échantillon. Les critères utilisés pour la comparaison seront naturellement les mêmes que ceux utilisés pour l'élagage via la validation croisée, à savoir la variance de prédiction pour un arbre de régression ou le taux de mauvais classement pour arbre de classification. D'autres critères peuvent néanmoins être analysés (e.g. l'AUC pour un arbre de classification pour une réponse binaire), même s'ils ne sont pas de premier intérêt pour l'analyse effectuée (sinon, il convient de revoir les critères utilisés pour l'élagage).

Pour un arbre de classification, on pourra notamment s'intéresser à la table de contingence croisant les

valeurs prédites et les valeurs observées, appelée *matrice de confusion*. Dans un cas de variable à expliquer binaire prenant ses valeurs dans $\{0, 1\}$, on utilise généralement la terminologie “négatif” et “positif” pour parler des modalités 0 et 1 respectivement. La matrice de confusion prend alors la forme suivante :

Table 4.1: Matrice de confusion pour une variable binaire (0,1)

	valeur observée positive	valeur observée négative
valeur prédite positive	VP	FP
valeur prédite négative	FN	VN

où VP signifie vrai positif, VN vrai négatif, FP faux positif et FN faux négatif.

Il sera alors utile de s'intéresser aux rapports

- $\frac{VP}{VP+FN}$, appelé *taux de vrais positifs* et correspondant à la probabilité de classer comme positive une observation dont la valeur est 1
- $\frac{FP}{FP+VN}$, appelé *taux de faux positifs* et correspondant à la probabilité de classer comme positive une observation dont la valeur est 0.

En effet, on recherchera un arbre avec un taux de vrais positifs proche de 1 et un taux de faux positifs proche de 0.

Il sera également utile de représenter la courbe ROC et de calculer l'AUC. Pour construire celle-ci, il faut tout d'abord définir un score par individu à partir de l'arbre de classification. Pour un arbre, le score d'un individu est défini comme la proportion de succès dans la feuille (ou région) à laquelle cet individu est affecté. Une fois ce score défini pour chaque individu, on peut construire la courbe ROC en modifiant la règle d'affectation en fonction d'un seuil que l'on fait varier (voir ici (<https://par.moodle.lecnam.net/mod/resource/view.php?id=97911>)).

5 Propriétés

Les arbres présentent différents avantages pour la fouille de données. En premier lieu leur fonctionnement est facile à expliquer à des non-initiés, plus que le modèle linéaire par exemple. De plus, les résultats peuvent être interprétés sous la forme d'un graphique, et ceci de façon simple quand il y a un nombre raisonnable de noeuds terminaux. Par ailleurs, la méthode est sensible aux effets d'interaction par sa structure hiérarchique. La sélection de variables est également inhérente à la méthode. Aussi, les arbres ne nécessitent pas d'inversion de matrice de covariance, et ainsi ils pourront être appliqués directement sur un jeu de données où les liaisons entre variables sont fortes (contrairement au modèle de régression linéaire par exemple). Un autre avantage des arbres est qu'ils permettent de considérer des variables quantitatives comme qualitatives, que ce soit pour les variables explicatives, comme pour la variable à expliquer. Enfin, ces méthodes algorithmiques ne sont pas coûteuses, rendant possible leur mise en oeuvre sur de grands jeux de données.

Cependant, les arbres possèdent aussi des défauts substantiels. En particulier, ce sont des méthodes instables. En effet, une petite variation dans les données risque de modifier de façon importante les divisions successives, et donc l'interprétation que l'on fait de l'arbre. Ceci est dû principalement à la structure hiérarchique : la moindre modification sur une division à des conséquences directes sur les divisions successives. Par ailleurs, cette instabilité rend ces méthodes assez peu performantes pour la prédiction par rapport à d'autres. De plus, un nombre d'individus assez important est nécessaire pour pouvoir les mettre en oeuvre (disons plusieurs centaines pour fixer les idées). En particulier, en présence d'une variable réponse de nature quantitative, une valeur prédite sera toujours comprise entre les valeurs minimales et maximales de

l'échantillon d'apprentissage, d'où la nécessité de disposer d'un large échantillon. Enfin, le partitionnement sous forme de parallélépipèdes ne permet pas de considérer des frontières autres que rectilignes. En effet, l'utilisation d'arbres binaires impose de diviser les noeuds selon une seule variable, et pas selon une combinaison de variables. La Figure 5.1 illustre un cas où une variable Y , binaire, s'explique à partir de la somme de deux autres variables quantitatives X_1 et X_2 (graphique de gauche), la frontière est ici oblique. La partition obtenue sur ces données, via l'ajustement d'un arbre, est assez complexe (schéma de droite) alors que le lien entre les variables s'exprime simplement selon $Y = \begin{cases} 0 & \text{si } X_1 + X_2 > 0 \\ 1 & \text{sinon} \end{cases}$.

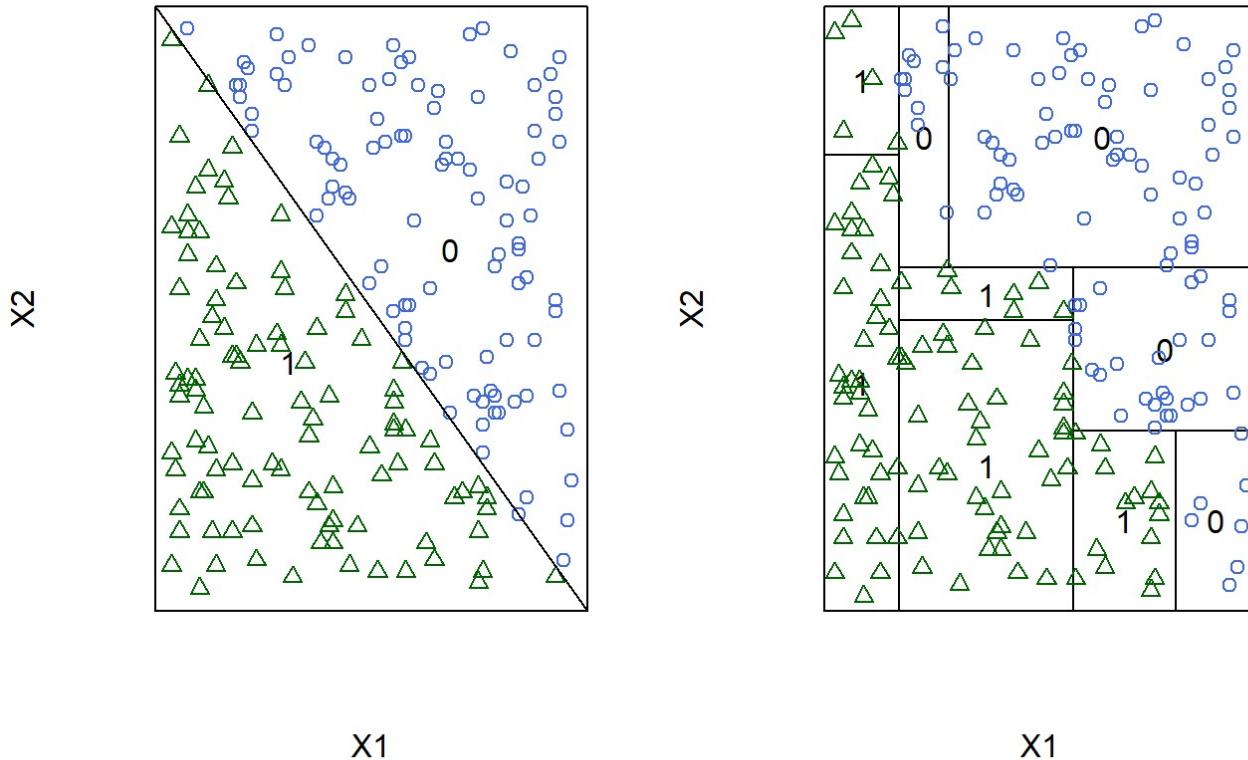


Figure 5.1: Illustration des performances d'un arbre en présence de frontière oblique. A gauche, le jeu de données où les deux classes sont représentées par les triangles verts et les points bleus. La bissectrice représente la frontière entre les deux régions ; à droite, la partition donnée par l'arbre.

6 Compléments

6.1 Autres algorithmes

Nous nous sommes concentrés sur l'algorithme CART, qui est probablement le plus couramment utilisé. Nous mentionnons ici d'autres algorithmes populaires.

6.1.1 C5.0

Cet algorithme est assez proche de l'algorithme CART. Il consiste essentiellement, une fois l'arbre construit, à

simplifier les règles d'affectation d'un individu à un noeud terminal. Les groupes d'individus au sein des noeuds terminaux restent les mêmes, seules les règles d'affectation changent. On perd alors la structure hiérarchique de l'arbre, mais cela facilite l'interprétation des règles d'affectation. Cet algorithme est une version améliorée de l'algorithme C4.5, lui-même extension de l'algorithme ID3.

6.1.2 CHAID

L'algorithme CHAID (CHi-squared Automatic Interaction Detector) publié dans Kass (1980) permet de construire des arbres de classification. Pour un prédicteur donné X_j , qualitatif à M modalités, on recherche le découpage en deux classes qui maximise la statistique du khi-deux avec la variable Y . Celle-ci quantifie la force de liaison entre la nouvelle variable obtenue par ce découpage et la variable Y . On fait cela ensuite pour un découpage à 3 classes, et ainsi de suite jusqu'à M classes. On détermine ainsi le découpage de X_j qui maximise le lien avec la variable Y . Il s'agit d'effectuer l'opération pour tous les prédicteurs X_j pour les valeurs de j entre 1 et M et de retenir la plus significative. Si la p-valeur est inférieure à un seuil (généralement 5%), alors on continue à diviser les noeuds fils, sinon on s'arrête. Notons qu'une correction de Bonferroni (cf Wikistat (2016)) est apportée sur les p-valeurs de façon à tenir compte du nombre de modalités de chaque variable explicative.

6.1.3 AID

L'algorithme AID (Automatic Interaction Detection) proposé par Morgan and Sonquist (1963) permet quant à lui de construire des arbres de régression. La différence avec l'algorithme CART est qu'il stipule une règle d'arrêt de façon similaire à l'algorithme CHAID, en basant cette fois ci la règle sur le test de student.

6.2 Forêts aléatoires

La stabilité des arbres peut être grandement améliorée via l'utilisation des *Random Forests with Random Inputs* (Breiman (2001)) appelées par abus de langage *Random Forests* (ou *Forêts aléatoires* en français). Au lieu d'essayer d'optimiser l'arbre sur l'ensemble de données "en un coup", les forêts consistent à générer plusieurs arbres différents en utilisant des échantillons indépendants issus des données, et à agréger ensuite les différentes prédictions fournies par ces différents arbres. L'idée sous-jacente des forêts aléatoires, est qu'en générant beaucoup d'arbres, on explore mieux l'ensemble des règles de décision, et qu'en agrégeant toutes les prédictions fournies par ces règles, on récupère un prédicteur qui rend compte de toute cette exploration.

Pour construire une forêt de B arbres à partir d'un échantillon d'apprentissage, on procède de la façon suivante :

- on constitue B échantillons par tirage aléatoire avec remise des n individus de l'échantillon d'apprentissage (échantillons bootstrap)
- pour chacun des B échantillons, on construit un arbre tel que chaque division de l'arbre est obtenue non pas à partir de l'ensemble des variables comme dans un arbre classique, mais en considérant à chaque noeud un sous-ensemble aléatoire de L variables ($L < p$). Ce sous-ensemble est différent pour chaque noeud.

On obtient ainsi un ensemble de B arbres qui constitue la forêt. Pour effectuer une prédiction à partir de ces B arbres, on agrège les différentes prédictions : par la moyenne dans le cas d'un arbre de régression, ou par le vote majoritaire en classification.

Il est important de noter que les arbres constituant la forêt sont des arbres non-élagués. Cependant, c'est

l'étape d'agrégation qui permet de se dispenser d'élaguer les arbres individuels. Rappelons que les arbres CART utilisés de façon individuelle eux doivent être élagués.

Les forêts aléatoires constituent un cas particulier des méthodes dites d'agrégation, qui feront l'objet d'une étude particulière dans un prochain cours. Plus précisément, elles peuvent être vue comme une variante du bagging où la différence intervient dans la construction des arbres individuels. Contrairement aux arbres, les forêts ont, elles, de très bonnes performances en termes de qualité de prédiction. En revanche, elles deviennent trop complexes pour pouvoir interpréter le lien entre les entrées et la sortie.

Références

- Breiman, L. 2001. "Random Forests." *Machine Learning* 45 (1). Springer: 5–32.
- Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and Regression Trees*. Monterey, CA: Wadsworth; Brooks.
- Genuer, Robin, and Jean-Michel Poggi. 2017. "Arbres CART et Forêts aléatoires,Importance et sélection de variables." <https://hal.archives-ouvertes.fr/hal-01387654> (<https://hal.archives-ouvertes.fr/hal-01387654>).
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc. <https://web.stanford.edu/~hastie/ElemStatLearn/download.html> (<https://web.stanford.edu/~hastie/ElemStatLearn/download.html>).
- Kass, G. V. 1980. "An Exploratory Technique for Investigating Large Quantities of Categorical Data." *Applied Statistics* 29 (2). JSTOR: 119–27.
- Morgan, J. N., and J. A. Sonquist. 1963. "Problems in the Analysis of Survey Data, and a Proposal." *Journal of the American Statistical Association* 58 (302). Taylor & Francis Group: 415–34.
- Nakache, J.P., and J. Confais. 2003. *Statistique Explicative Appliquée: Analyse Discriminante, Modèle Logistique, Segmentation Par Arbre*. Technip.
- Quinlan, J. Ross. 1993. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Wikistat. 2016. "A Propos de La Méthode de Bonferroni — Wikistat." <https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-modmixt8-bonfer.pdf> (<https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-modmixt8-bonfer.pdf>).