Mémoire de diplôme de l'ENS

Année 2011-2012

Apprentissage et forêts aléatoires

Erwan Scornet

Sous la direction de Gérard Biau et de Jean-Philippe Vert

14 octobre 2012

Table des matières

Introduction

Tout au long du XXe siècle, la multiplication des appareils de mesure (capteurs atmosphériques, appareils de séquençage génomique...) dans des domaines très divers a permis de recueillir un nombre considérable de données. Ces données sont systématiquement entachées de bruit, qui peut être anecdotique comme il peut fausser toute analyse ultérieure des données. D'autre part, le trop grand nombre de données et l'absence de loi a priori régissant ces données n'incite plus à se servir des informations pour valider une loi (comme on peut encore le faire en physique) : on préfèrerait extraire directement la loi régissant les données, des données elles-même. C'est le but de l'apprentissage statistique.

Dans ce qui suit, on s'intéresse exclusivement aux problèmes de classification supervisée. On se donne une suite $(X_1, Y_1), ..., (X_n, Y_n)$ qui sera notre échantillon d'apprentissage. Les variables $X_i = (X_i^{(1)}, X_i^{(2)}, ..., X_i^{(d)})$ appartiennent à $\mathbb{R}^d$ et peuvent par exemple représenter la quantité de certaines molécules dans le sang pour un individu i donné (i allant de 1 à n). Les variables Y_i quant à elles sont à valeurs dans $\{0, 1, 2.., k\}$. Elles sont associées à l'individu i et permettent de catégoriser la population en k classes. Par exemple, si on cherche à déterminer la survenue ou non d'une maladie $Y \in \{0, 1\}$ où Y_i vaut 1 si l'individu i est malade, 0 sinon. On cherche alors, étant donné notre échantillon d'apprentissage, à savoir si un nouveau patient $n + 1$ va développer la maladie en question, sachant ses variables cliniques : on connaît X_{n+1} et on cherche à déterminer Y_{n+1}.

Pour répondre à cette problématique, une méthode appelée CART (Classification And Regression Tree) a été inventée par Breiman et Friedman en 1973. Elle a été formalisée dans ([Bre11]). Cette méthode s'est montrée très efficace et est maintenant couramment utilisée notamment dans le domaine médical. Dans la première partie, on définira ce que sont les arbres binaires de décision et la manière dont ils sont construits en utilisant l'algorithme CART. On présentera également quelques résultats concernant leurs capacités prédictives.

Bien que ces arbres possèdent certaines propriétés intéressantes, leur construction est fortement dépendante de l'échantillon initial ce qui peut provoquer des problèmes si certaines données sont fausses. Même si les données sont exactes, il est possible que l'ajout de quelques données à l'échantillon change complètement l'arbre ainsi construit. Le modèle n'est donc pas stable. Afin de pallier cette difficulté, plusieurs modèles ont été proposé par Breiman ([Bre96], [Bre01]) et par d'autres ([Die00], Ho 1998). Le point commun de ces modèles est d'introduire de l'aléatoire dans la construction des arbres de décision. De cette manière, on ne construit plus un seul arbre de décision mais M arbres de décision, chacun ayant ses propres caractéristiques dues à l'aléa introduit. Chacun des M arbres vote pour déterminer si un nouveau point possède une étiquette $Y_{n+1} = 1$ (malade) ou $Y_{n+1} = 0$ (sain). Les résultats des votes sont agrégés et la majorité l'emporte. Ce n'est donc plus l'arbre qui vote, mais la forêt (aléatoire) cachée derrière.

Cette méthode a prouvé en pratique son extrême efficacité. Certains résultats théoriques montre qu'elle est, à certains égards, meilleure que les arbres de décision seuls. Cependant, beaucoup reste à faire afin de déterminer pourquoi elle est si puissante. Dans la deuxième partie, on présente différents types de forêts aléatoires et on expose le modèle général. On donne également quelques résultats importants sur la convergence de ces forêts.

1 Arbres de décisions

Les arbres de décisions ne fournissent pas uniquement une solution au problème de classification (prédire la survenue d'une maladie en fonction des données biologiques de l'individu) mais permet également de retracer les questions à poser pour effectuer ce choix.

Les variables biologiques peuvent être quantitatives ou qualitatives. Les variables qualitatives possédant p modalités peuvent être décomposées en $p-1$ variables binaires. On peut donc se ramener à des variables quantitatives ou binaires. Afin de fixer les idées et de pouvoir facilement interpréter graphiquement l'arbre de décision, on suppose que les variables d'intérêt sont uniquement quantitatives et à valeurs dans $[0,1]$ quitte à les renormer. De cette manière, à chaque sous arbre de l'arbre final correspond une partition de $[0,1]^d$ où d est le nombre de variables.

Par exemple, on peut chercher à prédire une variable binaire (rouge ou bleue dans la figure 1) en fonction de deux variables quantitatives comprises dans $[0,1]$. Pour cela, on construit des partitions de plus en plus fines de l'ensemble $[0,1]^2$ en réalisant des coupures parallèles aux axes. L'arbre de décision associé à cette construction correspond à l'ordre des coupures ainsi réalisées. En voici un exemple :

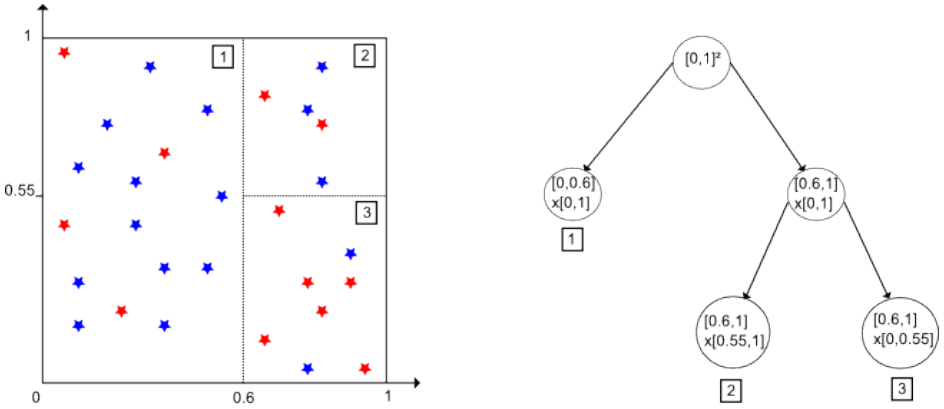

FIGURE 1 – Arbre de décision et partitions associées

Si l'on cherche à prédire la couleur d'un nouveau point, on lui associera la couleur majoritairement présente dans la cellule à laquelle il appartient (ici, on dispose de trois cellules $1, 2$ et 3). Le principal critère de construction d'un tel arbre est le choix de la coupure à effectuer à chaque étape pour affiner la partition. C'est ce choix que nous allons commencer par étudier.

1.1 Croissance de l'arbre

Dans cette partie, nous formalisons la construction de l'arbre de décision. On se donne un n-échantillon $(X_1, Y_1), ..., (X_n, Y_n)$ indépendant, identiquement distribué (i.i.d.) de même loi que (X, Y). On suppose dans tout ce qui suit que Y_i est à valeur dans $\{1, ..., j\}$ et que les X_i sont à valeurs dans $[0,1]^d$. A chaque noeud, on calcule, pour toutes les coupures possibles (il y en

a au plus n pour chaque variable puisqu'il y a n points dans l'échantillon) un critère qu'on définira ensuite, et on choisit la coupure qui minimise ce critère. On continue jusqu'à ce que chaque cellule de la partition finale ne possède qu'un seul point. On obtient ainsi l'arbre final qu'on notera T_{max}. De cette manière, on construit une partition de $[0,1]^d$ telle que chaque cellule contienne un élément de l'échantillon initial.

La prédiction qu'on aurait envie de faire est d'associer à tout point tombant dans une cellule l'étiquette du point de l'échantillon initial. Cependant, cette classification est très fortement dépendante des données : il suffit de rajouter dans l'échantillon initial deux étiquettes opposées à celle déjà présente dans une cellule pour que la classification de la cellule entière change. On cherche donc à avoir des partitions qui contiennent plus qu'un élément. Pour cela, on va construire, à partir de l'arbre final, une suite décroissante d'arbres et on choisira parmi ceux-ci, celui qui minimise un critère combinant précision de la prédiction et complexité de l'arbre.

Le critère permettant de sélectionner la meilleure coupure provient d'une fonction d'impureté qui doit vérifier les propriétés suivantes.

Fonction d'impureté Une fonction d'impureté est une fonction ϕ définie sur l'ensemble

$$\left\{(p_1,...,p_j), \forall i \in \{1,...,j\}, p_i \geq 0, \sum_{i=1}^{j} p_i = 1\right\}$$

vérifiant

- ϕ admet un unique maximum en $(1/j,...,1/j)$
- ϕ admet j minima en chaque e_i (où e_i a toutes ses coordonnées nulles sauf la i-ème)
- ϕ est une fonction symétrique de $p_1,...,p_n$

À partir d'une fonction vérifiant ces propriétés, on peut définir l'impureté de chaque noeud t.

Impureté d'un noeud On appelle $i(t)$ l'impureté du noeud t définie par la relation

$$i(t) = \phi(p(1|t), p(2|t), ..., p(j|t))$$

où $p(k|t)$ est la probabilité d'avoir l'étiquette k sachant qu'on se trouve dans la cellule correspondant au noeud t.

On peut également définir l'impureté du noeud t dans l'arbre total.

Impureté d'un noeud dans l'arbre L'impureté d'un noeud t dans l'arbre total est notée $I(t)$ et est définie par

$$I(t) = p(t)i(t)$$

où $p(t)$ est la probabilité qu'une donnée tombe dans la cellule t.

Diminution d'impureté entre un noeud et ses deux fils On peut enfin définir la quantité qui nous intéresse, à savoir la différence d'impureté entre un noeud t et ses deux fils t_L et t_R, obtenus par la coupure s, par la relation :

$$\Delta i(s,t) = i(t) - p_R i(t_R) - p_L i(t_L)$$

où p_L (resp. p_R) est la probabilité qu'une donnée appartienne à la cellule t_L (resp. t_R) sachant qu'elle se trouvait dans la cellule t.

Impureté d'un arbre Notons $\tilde{T}$ l'ensemble des feuilles de l'arbre total T. L'impureté de l'arbre total T est alors définie par

$$I(T) = \sum_{t \in \tilde{T}} I(t)$$

On peut montrer que maximiser la différence d'impuretés $\Delta i(s,t)$ pour un noeud t et une coupure s est équivalent à maximiser la différence d'impureté suivante :

$$\Delta I(s,t) = I(t) - I(t_L) - I(t_R)$$

Un critère assez naturel à considérer est l'erreur commise en donnant la mauvaise étiquette à certains points. Considérons un noeud non terminal de l'arbre T. Si la construction de l'arbre s'arrêtait à ce noeud, on affecterait à tous les points de la cellule correspondant à ce noeud l'étiquette suivante

$$j^* = argmax_j p(j|t)$$

Pour un point dont on sait qu'il tombe dans ce noeud, on commettrait une erreur de

$$r(t) = 1 - p(j^*|t)$$

Donc pour un point quelconque, la probabilité de se tromper au noeud t est $R(t) = p(t)r(t)$.

De la même manière que précédemment, on définit l'erreur de classification totale de l'arbre T par

$$R(T) = \sum_{t \in \tilde{T}} R(t)$$

On cherche à utiliser l'erreur de classification comme critère pour choisir les noeuds de l'arbre. Étant donné le parallèle entre $r(t)$ et $i(t)$, on cherche donc à étudier un arbre dont le choix des noeuds serait donné par la maximisation de

$$r(t) - p_L r(t_L) - p_R r(t_R)$$

ou de manière équivalent

$$R(t) - R(t_L) - R(t_R)$$

Dans ce qui suit, on se place dans le cas où $i(t) = r(t)$. La fonction d'impureté correspondant au choix de $r(t) = i(t)$ est

$$\phi(p_1, ..., p_j) = 1 - \max_j p_j$$

et vérifie bien les propriétés de la définition correspondante.

Cependant, considérer l'erreur de classification comme critère de coupure n'est pas une bonne idée car elle ne permet pas, dans de nombreux cas, de faire un choix.

Proposition Pour toute séparation du noeud t en t_L et t_R,

$$R(t) \geq R(t_L) + R(t_R)$$

avec égalité si $j^*(t) = j^*(t_L) = j^*(t_R)$.

Dans le cas où il n'y a que deux modalités, la fonction d'impureté vaut, avec $i(t) = r(t)$,

$$\begin{aligned}\phi(p_1, p_2) &= 1 - \max(p_1, p_2) \\ &= \min(p_1, p_2) \\ &= \min(p_1, 1 - p_1)\end{aligned}$$

Les auteurs de [Bre11] ont eu l'intuition que les noeuds purs, c'est-à-dire les noeuds contenant des étiquettes de même valeurs n'étaient pas assez récompensés par le critère d'erreur de classification (décroissance linéaire de la fonction d'impureté). Ils ont donc imposé à cette fonction d'être plus décroissante qu'une fonction linéaire en tout point, c'est-à-dire d'être concave (strictement). On cherche donc des fonctions d'impuretés ϕ appartenant à la classe $\mathcal{F}$ dont les éléments vérifient

1. $\phi(0) = \phi(1) = 0$
2. $\phi(p_1) = \phi(1 - p_1), \forall 0 \leq p_1 \leq 1$
3. ϕ doit être C^2 et vérifier $\phi''(p_1) < 0, \forall 0 \leq p_1 \leq 1$

Cette classe de fonction permet d'avoir un critère de choix de noeuds utilisables, au moins dans le cas où il n'y a que deux modalités.

Proposition Pour deux modalités seulement, pour tout noeud t et pour toute séparation s,

$$I(t) - I(t_L) - I(t_R) \geq 0$$

avec égalité si et seulement si $p(j|t_L) = p(j|t_R) = p(j|t)$ pour $j = 1, 2$. Plusieurs fonctions remplissent ces conditions et sont présentes dans la littérature. On se contente de donner celle utilisée dans l'algorithme CART, le critère de Gini.

Critère de Gini Le critère de Gini correspond au formalisme précédent dans lequel on a pris

$$i(t) = \sum_{j \neq i} p(j|t)p(i|t)$$

On peut remarquer que cet indice vérifie les propriétés précédentes puisque la fonction $\phi(p_1, ..., p_j)$ est une fonction quadratique en les p_i avec des coefficients positifs et est donc concave. On peut montrer qu'elle l'est strictement ce qui assure qu'elle appartient à $\mathcal{F}$

1.2 Élagage de l'arbre

Après avoir construit un arbre T de taille maximale, on cherche à l'élaguer, c'est-à-dire à supprimer des sous-arbres de cet arbre afin de le rendre moins complexe tout en essayant de conserver sa capacité prédictive. Pour cela, on cherche à minimiser la quantité suivante, sur tous les sous arbres T de l'arbre final T_{max}.

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|$$

où $\tilde{T}$ est le nombre de feuille de l'arbre et α est un paramètre réel. Cette expression est composée de deux termes : le premier est le risque d'erreur associé à l'arbre T, le deuxième est un terme rendant compte de la complexité de l'arbre.

- Pour $\alpha = 0$ on cherche comme précédemment à minimiser l'erreur de classification
- Plus α est grand, plus la pénalisation liée à la complexité de l'arbre est importante
- Pour $\alpha = +\infty$, l'arbre est vide.

On définit de la manière suivante les solutions de la minimisation précédente.

Définition des arbres α-minimaux Le sous-arbre T le plus petit minimisant le risque R_α est défini par

- $R_\alpha(T(\alpha)) = \min_{T \prec T_{max}} R_\alpha(T)$
- Si $R_\alpha(T) = R_\alpha(T(\alpha))$ alors $T(\alpha) \prec T$

La proposition suivante assure que l'existence de tels éléments

Proposition Pour tout α, il existe un plus petit arbre minimal au sens de la définition précédente.

On cherche ici à construire une suite d'arbre décroissante partant de l'arbre final T_{max}, $T_{max} \prec T_1 \prec T_2 \prec ... \prec T_p$, de sorte qu'il ne reste plus qu'à choisir quel arbre dans cette suite possède le meilleur compromis complexité/prédiction.

Afin de construire cette suite, on procède de la manière suivante :

- On part de l'arbre final T_{max}. On remonte progressivement, en supprimant deux feuilles d'un même noeud si celles-ci contiennent toutes les deux les mêmes étiquettes. On obtient un arbre possédant le même risque d'erreur de classification que le précédent mais avec moins de feuilles.
- On construit ensuite l'arbre T_2 de la manière suivante.

Pour tout noeud t, notons T_t le sous arbre ayant pour racine t et $\{t\}$ l'arbre formé du noeud t. On a,

$$R_\alpha(\{t\}) = R(t) + \alpha$$
$$R_\alpha(T_t) = R(T_t) + \alpha|\tilde{T}_t|$$

Tant que $\alpha < \alpha_c$, la branche T_t a un meilleur rapport coût-complexité que $\{t\}$. Déterminons α_c. Pour $\alpha < \alpha_c$,on a :

$$R(T_t) + \alpha|\tilde{T}_t| < R(t) + \alpha$$

Soit

$$\alpha < \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$$

Posons, pour tout $t \in T_1$,

$$g_1(t) = \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1} \text{ si } t \notin \tilde{T}_1$$
$$= +\infty \text{ si } t \in \tilde{T}_1$$

Le noeud $\bar{t_1}$ qui minimise $g_1(t)$ est le premier noeud à être supprimé quand α croît. Plus précisément, posons

$$\bar{t_1} = argmin_{t \in T_1} g_1(t)$$
$$\alpha_2 = g(\bar{t_1})$$

Alors, lorsque $\alpha \geq \alpha_2$, la branche $\{t\}$ devient préférable à la branche T_t. On pose alors $T_2 = T_1 - T_{\bar{t_1}}$ qui est l'arbre T_1 privé de $\bar{t_1}$ et de ses fils. On itère le procédé jusqu'à ce que $T_k = \{t_1\}$ où $\{t_1\}$ est la racine de l'arbre initial.

Il reste alors à trouver un critère permettant de sélectionner le meilleur arbre de cette suite, dans un sens à déterminer. Pour cela, on sépare l'échantillon initial en deux :

- le premier échantillon permet de construire l'arbre T_{max} puis de l'élaguer et d'obtenir la suite $T_1, ..., T_k$.
- le deuxième échantillon nous permet d'évaluer l'erreur de classification afin de choisir le meilleur des k arbres précédents.

Remarque : Précisons ici que l'erreur de classification $R(t)$ peut être écrite en différenciant les coûts de mauvaise classification. De manière générale, on considère une fonction $C(i|j)$ qui est le coût correspondant au fait de donner l'étiquette i à un point de classe j. On peut alors définir de la même manière, l'erreur de classification $R(t)$.

1.3 Consistance des arbres

Soit $(X_1, Y_1), ..., (X_N, Y_N)$ un n-échantillon i.i.d. de loi (X, Y). Pour tout noeud t, notons $p_N(t)$ la probabilité empirique qu'une donnée tombe dans $\{t\}$ (on confond ici noeud et cellule de la partition associée). Soit $\tilde{T_N}$ une certaine partition. On pose

$$\tilde{\tau_N}(x) = \{t \in \tilde{T_N}, x \in t\}$$

et $\delta(t)$ le diamètre de la cellule $\{t\}$. On pose également $D_N(x) = \delta(\tilde{\tau_N}(x))$ le diamètre de la cellule contenant x.

Soit $d_N(x)$ un classifieur. On appelle classifieur de Bayes et on le note $d_b(x)$, le classifieur minimisant l'erreur de classification pour x

$$\sum_j C(d(x)|j)P(j|x)$$

parmi l'ensemble des classifieurs.

Un classifieur naturel $d_N(x)$ est donc construit en minimisant l'erreur de classification empirique pour x

$$\sum_j C(d(x)|j)P_N(j|x)$$

On peut montrer que cela revient à minimiser l'expression suivante :

$$\sum_{n, X_n \in \tilde{\tau_N}(x)} C(d(x)|Y_n)$$

ce qui est exactement la manière dont on étiquetait les cellules après avoir construit l'arbre. Le classifieur associé à un arbre de décision binaire est consistant sous certaines hypothèses.

Théorème : Consistance des arbres binaires Soit $k_N > 0$ tel que

$$p_N(t) \geq k_N \frac{\log N}{N}.$$

Supposons de plus que

1. $\lim_N k_N = +\infty$
2. $\lim_N D_N(X) = 0$ en probabilité

Alors $d_N(x)$ est un classifieur consistant, c'est-à-dire que l'erreur moyenne due à ce classifieur tend vers la meilleure erreur moyenne envisageable (en probabilité).

2 Forêts aléatoires

2.1 Cadre général et différents exemples

Les forêts aléatoires ont été inventées par Breiman en 2001 ([Bre01]). Elles sont en général plus efficaces que les simples arbres de décision mais possède l'inconvénient d'être plus difficilement interprétables.

Forêts aléatoires

Une forêts aléatoire est un ensemble d'arbres de décision binaire dans lequel a été introduit de l'aléatoire. Autrement dit, soit $\theta_1, ...\theta_M$ des variables i.i.d. , une forêt aléatoire est un ensemble de classifieurs $\{d_i(\mathbf{x}, \theta_i), i = 1..M\}$ où les classifieurs d_i sont construits sur le même modèle que les arbres binaires.

Le nouveau classifieur correspondant à la forêt aléatoire est calculé en prenant la majorité des votes de chacun des classifieurs $d_i, i = 1..n$.

De nombreux modèle de forêts aléatoires ont été créés qui correspondent à autant de manière d'incorporer de l'aléatoire dans les arbres. À titre d'exemple, on peut citer :

- le Tree Bagging ([Bre96]) introduit de l'aléatoire dans l'échantillon initial sélectionnant certains points plutôt que d'autres et laisse grandir l'arbre jusqu'à ce que chaque noeud comporte un unique élément
- le random subspace (Ho, 1998) consiste à sélectionner à chaque noeud K variables de manière aléatoire et, parmi celles-ci, à choisir celle qui minimise un certain critère.
- la Random Forest ([Bre01]) qui consiste pour à mélanger le CART, le bagging et le random subspace : pour chaque arbre, on tire un échantillon à partir de l'échantillon initial ; à chaque noeud, on choisit aléatoire K variables et on prend, parmi celles-ci, celle qui minimise le critère de l'algorithme CART. On laisse grandir l'arbre jusqu'à ce qu'il n'y ait plus qu'un seul élément dans chaque noeud.
- le Random Select Split ([Die00]) qui sélectionne les K meilleures séparation et qui en choisit une parmi celles-ci de manière aléatoire. La position de la coupure est également calculée de manière aléatoire.

2.2 Convergence des forêts aléatoires

On s'intéresse ici à la forêt aléatoire définie par Breiman en 2001. Ce qui suit est directement tiré de ([GB08]).

On rappelle brièvement le formalisme ci-dessus. On considère un n échantillon

$$\mathcal{D}_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$$

i.i.d. à valeurs dans $[0,1]^d \times \{0,1\}$. Le but est d'estimer la fonction de régression $r(x) = \mathbb{E}(Y|X = x)$. On dira qu'une estimateur de la fonction de régression est consistant si

$$\lim_n \mathbb{E}(r_n(X) - r(X))^2 = 0$$

On dispose d'un ensemble d'estimateur des fonctions de régression $\{r_n(x, \theta_i, \mathcal{D}_n), i = 1...M\}$. On associe ces différents estimateurs de la façon suivante :

$$\bar{r}_n(X, \mathcal{D}_n) = \mathbb{E}_\Theta(r_n(X, \Theta, \mathcal{D}_n))$$

On détermine ensuite l'étiquette de X selon la règle suivante :

$$d_n(x) = \mathbb{1}_{\bar{r}_n(X) \geq 1/2}$$

On se donne une suite de probabilité p_{nj} telle que $\forall n \in \mathbb{N}$,

$$\sum_{j=1}^{d} p_{nj} = 1$$

Afin d'approcher la forêt aléatoire de Breiman, on considère le modèle suivant : à chaque étape, on tire au hasard une des d coordonnées (la j-ème coordonnée est tirée avec probabilité p_{nj}). On effectue la coupure selon la coordonnée j au milieu du segment concerné. Cette procédure est répétée $\lceil \log_2 k_n \rceil$ fois où k_n est un paramètre fixé au préalable.

On a

$$r_n(X, \theta) = \frac{\sum_{i=1}^{n} Y_i \mathbb{1}_{X_i \in A_n(X,\theta)}}{\sum_{i=1}^{n} \mathbb{1}_{X_i \in A_n(X,\theta)}} \mathbb{1}_{\epsilon_n(X,\theta)}$$

où $A_n(X, \theta)$ est la cellule contenant X et où

$$\epsilon_n(X, \theta) = \left[\sum_{i=1}^{n} \mathbb{1}_{X_i \in A_n(X,\theta)} \neq 0 \right]$$

correspond à l'événement où la cellule $A_n(X, \theta)$ est non vide. On obtient ainsi l'expression de $\bar{r}_n(X)$.

Le nombre de points dans une cellule est donné par

$$N_n(X, \theta) = \sum_{i=1}^{n} \mathbb{1}_{X_i \in A_n(X,\theta)}$$

Théorème : consistance de l'estimateur de la fonction de régression Supposons que la distribution de X est à support dans $[0, 1]$. Alors l'estimateur des forêts aléatoires $\bar{r}_n$ est consistant dès que

1. $\lim_n p_{nj} \log k_n = +\infty$ pour tout $j = 1..d$
2. $\lim_n k_n/n = 0$

Preuve Pour cela, il suffit de montrer que

1. $\lim_n diam(A_n(X, \theta)) =_P 0$
2. $\lim_n N_n(X, \theta) =_P +\infty$

Montrons d'abord que $N_n(X, \theta) \to +\infty$. Soit $M \geq 0$,

$$\begin{aligned}
\mathbb{P}(N_n(X, \theta) < M) &= \mathbb{E}\left[\mathbb{P}(N_n(X, \theta) < M | \mathcal{C}, \theta)\right] \\
&= \mathbb{E}\left[\sum_{l=1..2^{\lceil \log_2 k_n \rceil}, N_l < M} \frac{N_l}{n+1} \right] \\
&\leq \frac{M 2^{\lceil \log_2 k_n \rceil}}{n+1} \\
&\leq \frac{2 M k_n}{n+1} \to 0
\end{aligned}$$

D'autre part, montrons que $diam(A_n(X, \theta)) \to_P 0$. Posons $V_n j(X, \theta)$ la taille de la j-ième dimension du rectangle contenant X. Posons, par définition,

$$V_{nj}(X, \theta) = 2^{-K_{nj}(X,\theta)}$$

où K_{nj} représente le nombre de fois où la j-ième coordonnées a été découpé pour obtenir la cellule A_n. Donc, conditionnellement à X, $K_{nj}(X,\theta)$ suit une loi binomiale $\mathcal{B}(\lceil \log_2 k_n \rceil, p_{nj})$. D'où

$$\begin{aligned}\mathbb{E}(V_{nj}(X,\theta)) &= \mathbb{E}(2^{-K_{nj}(X,\theta)}) \\ &= \mathbb{E}\left[\mathbb{E}(2^{-K_{nj}(X,\theta)}|X)\right] \\ &= (1-\frac{p_{nj}}{2})^{\lceil \log_2 k_n \rceil}\end{aligned}$$

2.3 Autres propriétés intéressantes des forêts aléatoires

Il a également été montré que la vitesse de convergence des forêts aléatoires était plus élevée que celle des estimateurs ordinaires. Voici un théorème, dont on pourra trouver la démonstration dans ([Bia10]).

On se place dans le cadre du modèle précédent. On suppose que seules certaines variables sont pertinentes pour la classification, c'est-à-dire que l'étiquette d'une donnée ne dépend que des variables fortes de l'ensemble $\mathcal{S}$. On note $\text{Card}(\mathcal{S}) = \mathcal{S}$. La fonction de régression peut se mettre sous la forme

$$r(X) = \mathbb{E}\left[Y|X_S\right]$$

ce qui nous invite à considérer la fonction r^* telle que

$$r(x) = r^*(x_S)$$

où $x_S = p_{|S}(x)$. Bien entendu l'ensemble $\mathcal{S}$ n'est pas connu a priori. Si tel était le cas, on poserait

$$\begin{aligned}p_{nj} &= 1/S \text{ si} j \in \mathcal{S} \\ p_{nj} &= 0 \text{ sinon}\end{aligned}$$

afin qu'à chaque noeud, on ne sélectionne les variables que parmi les variables pertinentes (i.e. appartenant à $\mathcal{S}$). Cependant, cette hypothèse est trop forte et il est préférable de choisir une hypothèse du type

$$\begin{aligned}p_{nj} &= \frac{1}{S}(1+\xi_{nj}) \text{ si} j \in \mathcal{S} \\ p_{nj} &= \xi_{nj} \text{ sinon}\end{aligned}$$

Théorème : vitesse de convergence des forêts aléatoires Supposons que X soit uniformément distribué selon $[0,1]^d$, $r* = r_{|S}$ soit L-Lipschitz sur $[0,1]^S$ et que

$$p_{nj} = \frac{1+\xi_n j}{S} \text{ pour } j \in \mathcal{S}.$$

Posons $\gamma_n = \min_{j \in \mathcal{S}} \xi_{nj}$. Alors, il existe Γ_n et C telles que,

$$\mathbb{E}\left[\bar{r}_n(X) - r(X)\right]^2 \leq \Gamma_n \frac{k_n}{n} + \frac{2SL^2}{k_n^{\frac{0.75(1+\gamma_n)}{S\log 2}}}$$

Ce résultat nous permet d'affirmer que les forêts aléatoires convergent plus vite que les arbres binaires classiques. En effet, leur vitesse de convergence est de $k_n^{-\frac{0.75(1+\gamma_n)}{S\log 2}}$ qu'il faut comparer à celle des estimateurs classiques de partitionnement $k_n^{-2/d}$. Dès que $S \leq 0.5d$, cette dernière vitesse est plus faible que celle des forêts aléatoires. Comme, en pratique, beaucoup de variables sont non pertinentes, l'hypothèse $S \leq 0.5d$ paraît tout à fait réaliste. Les forêts aléatoires sont donc une bonne alternative aux méthodes classiques.

Références

[Bia10] Gérard Biau. Analysis of a random forests model. 2010.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24(2), 1996.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45, 2001.

[Bre11] Olshen Stone Breiman, Friedman. Classification and regression tree. 2011.

[Die00] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees : bagging, boosting and randomization. *Machine Learning*, 40(2), 2000.

[GB08] Gàbor Lugosi Gérard Biau, Luc Devroye. Consistency of random forests and other averaging classifiers. 2008.