Arbres

Les arbres de classification et de régression (*Classification And Regression Trees*, CART) font parties des méthodes d'apprentissage supervisée. Ils permettent de prédire une variable réponse à partir de plusieurs variables explicatives $X_1, \dots, X_p$. On dispose donc de n observations appartenant à $\mathbb{R}^p$. Un arbre de classification cherche à assigner une observation i à l'un des K groupes sur la base de $X_{i1}, \dots, X_{ip}$, les valeurs des variables pour cette observation i. Tandis qu'un arbre de régression cherche à prédire la valeur d'une variable numérique continue Y à partir des observations de $X_1, \dots, X_p$.

1 Objectif

L'objectif de l'algorithme CART est de partionner l'espace des observations, i.e. $\mathbb{R}^p$, en sous-ensembles homogènes à l'aide de découpages successifs. Chaque sous-ensemble est représenté par une **feuille** de l'arbre, à laquelle est associée une prédiction.

De façon plus précise, l'algorithme CART est un algorithme récursif de découpage binaire. Il procède de la façon suivante :

1. Le découpage de l'espace $\mathbb{R}^p$ en sous-régions rectangulaires (que l'on appelle hyper-rectangles), en coupant selon une variable $X_j, j = 1, \dots, p$ à une certaine valeur seuil s.
2. À chaque étape, on choisit la variable $X_j, j = 1, \dots, p$ et la valeur seuil s qui offrent la meilleure amélioration d'un **critère d'homogénéité** (cf. la section "Critères d'homogénéité").
3. On répète ce découpage tant qu'un **critère d'arrêt** n'est pas atteint (cf. la section "Choix de la complexité").
4. Chaque feuille contient des observations similaires et fournit une prédiction unique : la classe majoritaire pour la classification et la moyenne des Y pour la régression. Ainsi, toutes les observations dans une même feuille reçoivent la même prédiction.

Remarque

L'algorithme CART est un algortihme glouton : à chaque étape, cette algorithme choisit localement la meilleure coupure. Mais cela ne garantit pas l'optimum global !
Il serait théoriquement préférable de chercher l'arbre à K feuilles qui minimise une fonction de coût globale :

$$\min_{R_1,\dots,R_K} \sum_{k=1}^{K} c_k,$$

où c_k mesure l'erreur dans le région R_k. Cependant, il y aurait trop de combinaisons possibles pour résoudre ce problème. C'est un problème intraitable (*intractable*).

2 Algorithme

L'algorithme se déroule comme suit, pour chaque noeud :

1. Pour chaque variable $X_j, j = 1, \dots, p$ et chaque seuil (valeur de coupure) possible s, on calcule la réduction du critère d'homogénéité entre les deux sous-ensembles.
2. On sélectionne le couple (X_j, s) qui maximise la réduction du critère d'homogénéité, i.e. qui offre le gain maximal.
3. On divise le noeud en deux sous-noeuds (sous-ensembles) selon la règle $X_j \leq s$ vs. $X_j > s$.
4. On répète ce processus jusqu'à ce qu'un nombre maximal de feuilles soit atteint ou bien que le nombre d'observations dans chaque feuille soit suffisant.

Le noeud de départ consiste en l'ensemble des observations.

3 Critères d'homogénéité pour la classification

Soit $\hat{p}_{jk}$, la proportion d'observations de la région R_j de l'ensemble $\mathbb{R}^p$ qui appartiennent à la classe k. Trois critères sont habituellement utilisés pour déterminer les divisions optimales à chaque étape de la construction de l'arbre :

1. Le **Taux d'erreur de classification** correspond à la proportion d'observations mal classées dans une feuille. Il est défini comme

   $$E_j = 1 - \max_k \hat{p}_{jk}.$$

   Ce critère est facile à interpréter, mais il est peu sensible et donc il est rarement utilisé pour la construction de l'arbre. Il est plutôt emplyé pour évaluer la performance du modèle une fois l'arbre construit.

2. L'**Indice de Gini** est un des critères les plus utilisés. Il est défini comme

$$G_j = \sum_{k=1}^{K} \hat{p}_{jk}(1 - \hat{p}_{ij}).$$

   Ce critère est minimale et égale à 0 lorsque tous les individus d'une région appartiennent à une unique classe, ce qui reflète une homogénéité parfaite. On cherche donc à avoir un indice de Gini faible.

3. L'**Entropie croisée** est un autre critère couramment utilisé. Il est basé sur la théorie de l'information de Shannon et est défini comme

$$D_j = -\sum_{k=1}^{K} \hat{p}_{jk} \log(\hat{p}_{jk}).$$

   Ce critère atteint aussi sa valeur minimale lorsque la classe est parfaitement prédite. On cherche donc aussi à avoir une entropie croisée faible.

Lors de la construction de l'arbre, on évalue pour chaque division possible la réduction d'hétérogénéité qu'elle entraîne, appelée **gain d'information**. Par exemple, pour l'indice de Gini, le gain d'information s'écrit comme la différence entre l'indice avant la division et une moyenne pondérée des indices dans les sous-groupes obtenus :

$$\Delta G = G_{\text{avant la division}} - \left(\frac{n_1}{n} G_1 + \frac{n_2}{n} G_2 \right),$$

où n_1 et n_2 représente le nombre d'observations dans les deux sous-ensembles et $n = n_1 + n_2$. On choisit alors la division qui maximise ce gain.

4 Choix de la complexité : élagage de l'arbre

Plus l'arbre devient profond, i.e plus le nombre de feuilles augmente, plus il s'adapte fidèlement aux données d'apprentissage. Toutefois, cette grande adaptation conduit au phénomène de surajustement (cf. Section sur l'évaluation de modèles). Ainsi, dans ce cas, le modèle ne sera plus capable de généraliser à des données inconnues. Cependant, si l'arbre n'a pas assez de feuilles, il y a un manque d'ajustement aux données car la classification est trop simpliste.

Pour éviter ce phénomène, on recourt à une stratégie d'élagage (*purning*). Dans un premier temps, on fait croître l'arbre jusqu'à un développement complet, i.e. jusqu'à ce que toutes les feuilles soient pures (ne contiennent qu'une seule observation) ou qu'aucune division ne soit pertinente. Dans un second temps, on coupe l'arbre en se débarassant des branches qui n'apportent que peu d'amélioration en termes de performance.

L'élagage peut être réalisé à l'aide d'un critère appelé coût-complexité. On cherche alors à minimiser une fonction combinant l'erreur de prédiction et la complexité de l'arbre. Cette fonction prend la forme suivante :

$$\sum_{j=1}^{|T|} c_j + \alpha |T|,$$

où $|T|$ est le nombre de feuille de l'arbre, c_j le coût de classification ou de régression dans la feuille j et α un paramètre de régularisation contrôlant la pénalité associée à la taille de l'arbre. Plus le paramètre α est élevé, plus on favorise les arbres simples. Le choix optimal de α peut se faire par validation croisée, en comparant les performances sur des jeux de validation pour différentes valeurs de ce paramètre.

5 Quelques commentaires

Les arbres présentent quelques avantages intéressants. Ils sont d'abord très interprétables. En effet, chaque prédiction peut être expliquée à l'aide d'un chemin dans l'arbre, ce qui les rend particulièrement adaptés à une utilisation dans un contexte appliqué. Ils sont également robustes aux valeurs extrêmes et aux transformations monotones des variables explicatives. De plus, ils gèrent naturellement les interactions entre les variables sans qu'il soit nécessaire de les expliciter. Ils peuvent être utilisés à la fois avec des variables quantitatives et quantitatives, et ne reposent sur aucune hypothèse de distribution des données. Ils effectuent automatiquement une sélection des variables au cours de leur construction. Les variables importantes auront tendance à être choisies en premières pour la découpe des noeuds. Enfin, ils constituent la base de nombreuses méthodes modernes d'apprentissage statistique, comme les forêts aléatoires ou bien le *gradient boosting* (cf. Section "Méthodes ensemblistes").

Malgré leurs nombreux avantages, les arbres présentent aussi certaines limites. Ils sont instables. En effet, de légères modifications dans les données peuvent entraîner des arbres très différents. Ainsi, lorsqu'ils sont utilisés seuls, ils peuvent être moins performants que d'autres méthodes, notemment lorsque les données sont bruitées. La forme des divisions (en rectangles alignés sur les axes) peut être inadaptée si les frontières entre les classes sont non-linéaires. Par ailleurs, sans élagage, les arbres ont tendance à surajuster les données. Ils sont également sensibles au déséquilibre des classes, i.e. lorsque certaines classes sont fortement sous-représentées dans les données. Ainsi, on leur préférera généralement une forme agrégée (e.g. forêts aléatoires, *boosting*) pour améliorer leur stabilité et leur performance.