

## ILLUSTRATION DES CONCEPTS RELATIFS À L'UTILISATION DES GRADIENT BOOSTING MACHINE

Les librairies R utiles pour implémenter des techniques de Gradient Boosting sont les suivantes (pas nécessaires dans le cadre de ce TP introductif) :

*gbm*, *mboost* et *xgboost*

L'objectif de ce TP est de mettre en pratique les principales notions vues en cours sur les Gradient Boosting Machine, et plus précisément les Gradient Tree Boosting. Nous essaierons de comprendre comment manipuler les principaux paramètres des fonctions R utilisées, dans l'optique d'éviter le phénomène de sur-apprentissage notamment.

### Exercice d'introduction : compréhension du boosting pas-à-pas.

Le but de cet exercice d'introduction est d'illustrer le phénomène de sur-apprentissage en pratique. Pour cela, nous allons tenter de prédire l'âge des individus à partir de leur hobbies. Au fur et à mesure, nous montrerons que le boosting permet de traiter partiellement cette question de par son approche.

Créer le jeu de données suivant :

PersonID	Age	LikesGardening	PlaysVideoGames	LikesHats
1	13	FALSE	TRUE	TRUE
2	14	FALSE	TRUE	FALSE
3	15	FALSE	TRUE	FALSE
4	25	TRUE	TRUE	TRUE
5	35	FALSE	TRUE	TRUE
6	49	TRUE	FALSE	FALSE
7	68	TRUE	TRUE	TRUE
8	71	TRUE	FALSE	FALSE
9	73	TRUE	FALSE	TRUE

Sur ce jeu de données, faire les étapes suivantes.

- (1) Intuitivement, quelles variables devraient jouer selon vous pour expliquer l'âge ?
- (2) Vérifier vos intuitions de manière très sommaire en regardant la distribution des âges en fonction des différentes variables explicatives à disposition.
- (3) Charger la librairie **rpart**. On vous propose de construire un arbre ayant au minimum 3 observations dans chacune des feuilles. L'afficher. Combien cet arbre possède-t-il de feuilles ?
- (4) Sur quelle(s) variable(s) cet arbre segmente-t-il la population ? Cela vous paraît-il cohérent ?
- (5) Une variable a priori segmentante semble ne pas apparaître. On se propose donc de construire maintenant un deuxième arbre avec au minimum 2 individus par feuille. Afficher l'arbre. La variable qui semblait manquer apparaît-elle maintenant ? Quelles autres observations pouvez-vous faire et quel phénomène êtes-vous en train d'illustrer ? Comparer à l'arbre maximal.
- (6) Calculer les erreurs individuelles de prévision (résidus : observé - prédit) issues de la première modélisation. Nous voyons le problème de l'utilisation d'un seul arbre CART pour construire des prévisions robustes...

- (7) Construire un arbre CART sur ces résidus en utilisant les mêmes variables explicatives que précédemment, avec la même contrainte sur le nombre minimum d'observations par feuille (3 obs.). Qu'observez-vous sur cet arbre ? Quelle variable segmentante apparaît alors ? Comment se fait-il que les résultats soient plus cohérents ici ? Vous commencez à toucher du doigt la stratégie du boosting.
- (8) Ajouter aux prévisions d'âge effectuées au départ par le premier modèle ces prévisions de résidus par ce nouvel arbre CART. Recalculer les résidus, et comparer avec les résidus précédents. Calculer l'erreur quadratique du tout premier modèle, et celle du dernier modèle.

Cet enchainement correspond au “Gradient Boosting”, ici appliqué avec des *weak learners* (tree-based, comprenant un minimum d'observations par feuille). Il s'agit donc

- i) d'estimer un modèle simple sur des données, noté  $F_1(x)$  ;
- ii) d'estimer un modèle simple sur les résidus du 1<sup>er</sup> modèle, noté  $G_1(x) = y - F_1(x)$  ;
- iii) de créer un nouveau modèle, basé sur ces 2 modèles :  $F_2(x) = F_1(x) + G_1(x)$  ;
- iv) on réitère cette idée un nombre  $M$  de fois... Pour déterminer  $M$ , on procède par validation croisée (car c'est un hyper-paramètre).

### Exercice 2 : Et si la fonction de coût diffère ?

Vous cherchez maintenant à estimer l'âge médian en fonction des caractéristiques. La fonction de perte à minimiser correspondante est donc

$$F_0(x) = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma) = \arg \min_{\gamma} \sum_{i=1}^n |y_i - \gamma|$$

- (1) Quel est l'âge médian de la population ? Calculer les résidus correspondants.
- (2) Considérer les 1<sup>er</sup> et 4<sup>e</sup> résidus. Supposez maintenant que vous seriez capable d'améliorer votre prévision d'une unité pour ces 2 individus...
  - Quel serait alors le gain en réduction d'erreur quadratique pour ces 2 cas ?
  - Si vous aviez considéré une erreur en valeur absolue, quel serait ce gain ?
 Sachant que le gradient boosting se concentre sur les observations mal prédites, on voit que la fonction de coût a un impact important sur le modèle final estimé.
- (3) Illustrons ceci en généralisant avec le concept de descente du gradient : notons  $L$  la fonction à minimiser. Notre point de départ est le modèle  $F_0(x)$ . pour la première itération, on calcule le gradient de  $L$  par rapport à  $F_0$ , puis on estime un *weak learner* sur les composantes du gradient.
  - Ds le cas arbres de rég., cela produit une moyenne de gradient dans chq feuille ;
  - Pour chq feuille, on avance en direction du gradient moyen, résultant en  $F_1(x)$  ;
  - Puis on répète le processus  $M$  fois.
 On peut donc généraliser notre algorithme à n'importe quelle fn. dérivable comme suit : on initialise le modèle avec une valeur constante. Puis pour  $m = 1$  à  $M$  :
  - on calcule les pseudo-résidus  $r_{im} = - \left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$ , pour  $i = 1, \dots, n$  ;
  - on estime sur ces pseudo-résidus un *base learner*, noté  $h_m(x)$  ;
  - on calcule le multiplicateur (step magnitude)  $\gamma_m$ , facteur de régularisation (potentiellement différent pour chaque feuille) ;
  - On met à jour  $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$ .
 Comment devrait se comporter l'algorithme en fonction de  $\gamma_m$  ?
- (4) Mettre en pratique cet algorithme sur un exemple simple : considérons la perte

$$L(x, y) = 0,5 (x - 15)^2 + 0,5 (y - 25)^2$$

Trouver la solution du problème par la fonction `optim`. Puis implémenter l'algorithme avec les paramètres suivants :  $(x_0, y_0) = (0, 0)$ ,  $M = 35$ ,  $\gamma = 1$ , déplacements possibles vers la solution :  $x \leftarrow x + 1$  ou  $y \leftarrow y + 1$