Agrégation de modèles

Philippe BESSE

Laboratoire de Statistique et Probabilités

UMR CNRS 5583

Université Paul Sabatier Toulouse III

besse@math.ups-tlse.fr

www.lsp.ups-tlse.fr/Besse

1 Introduction

1.1 Apprentissage

- Supervisé vs. non-supervisé
- Discrimination vs. régression
- Modélisation (explicative) vs. Apprentissage (prédictif)
- Statistique vs. *Data Mining*
- Choix de méthode et estimation de l'erreur
- Choix de modèle : équilibre biais-variance
- Choix de modèle : sélection vs. régularisation

1.2 Stratégie

1. Extraction avec ou sans échantillonnage
2. Exploration (valeurs atypiques, incohérences, transformations)
3. Partition de l'échantillon (apprentissage, validation, test)
4. Pour chacune des méthodes considérées : modèle linéaire général, discrimination paramétrique ou non paramétrique, k plus proches voisins, arbre, réseau de neurones, support vecteur machine, combinaison de modèles (bagging, boosting).
   - Estimer le modèle pour une valeur donnée d'un paramètre de complexité
   - Optimiser ce paramètre (échantillon de validation)
5. Comparaison des modèles optimaux obtenus (échantillon test)
6. Itération éventuelle (3 à 5)
7. Choix de la méthode

- *Enjeux :* rechercher un *modèle parcimonieux*.

2 Arbres binaires

2.1 Introduction

- Classification and regression trees (CART)
- X^j explicatives quantitatives ou qualitatives,
- Y quantitative : regression tree ;
- Y qualitative à m modalités $\{\mathcal{T}_\ell; \ell = 1 \ldots, m\}$: classification tree ;
- Objectif : construction d'un arbre de décision binaire simple à interpréter.
- Méthodes calculatoires : peu d'hypothèses mais beaucoup de données.

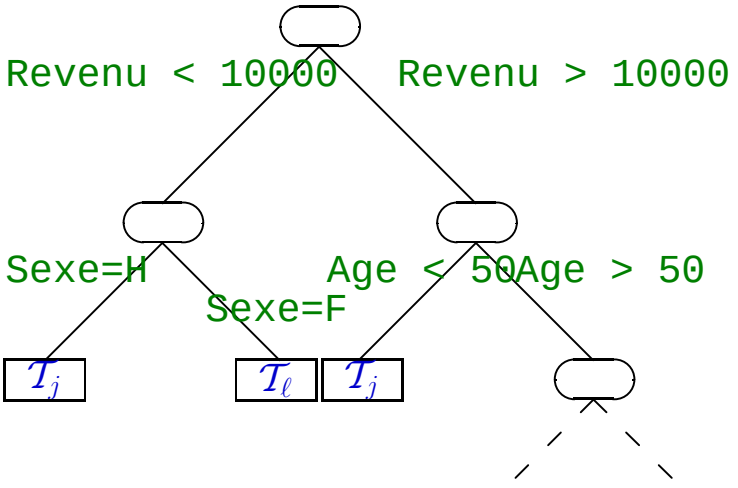
2.2 Construction d'un arbre binaire

2.2.1 Principe

Déterminer une séquence itérative de *nœuds*.

- *Racine* : nœud initial ou ensemble de l'échantillon.
- *Nœud* : choix d'une variable et d'une *division* ; sous-ensemble auquel est appliquée une dichotomie.
- *Division* : valeur seuil ou groupes des modalités.

Exemple élémentaire d'arbre de décision.



Choix nécessaires :

1. Critère de la "meilleure" division parmi celles *admissibles* ;
2. Règle de nœud terminal : *feuille* ;
3. Règle d'affectation à une classe $\mathcal{T}_\ell$ ou une valeur de Y.

Obtenir ensuite un modèle parcimonieux par *élagage* (pruning) de l'arbre.

2.2.2 Critère de division

- Division *admissible* : descendants $\neq \emptyset$.
- X^j réelle ou ordinale : $(c_j - 1)$ divisions possibles.
- X^j nominale : $2^{(c_j-1)} - 1$ divisions.
- Fonction d'hétérogénéité $D_{(k)}$d'un nœud

1. Nulle : une seule modalité de Y ou Y constante ;
2. Maximale : modalités de Y équiréparties ou grande variance.

Notations

- k : numéro d'un nœud.
- $(k+1)$ et $(k+2)$ les nœuds fils.

L'algorithme retient la division rendant minimales $D_{(k+1)} + D_{(k+2)}$.
Chaque étape k de construction de l'arbre :

$$\max_{\{\textit{divisions de} X^j; j=1,p\}} D_k - (D_{(k+1)} + D_{(k+2)})$$

2.2.3 Règle d'arrêt

Un nœud donné, est terminal ou appelé feuille, lorsqu'il est homogène :

- plus de partition admissible ou
- nombre d'observations inférieur à un seuil.

2.2.4 Affectation

- Y quantitative, la valeur est la moyenne des observations.
- Y qualitative, chaque feuille est affectée à une classe $\mathcal{T}_\ell$ de Y en considérant le mode conditionnel :
  - la classe la mieux représentée dans le nœud ;
  - la classe a posteriori la plus probable si des a priori sont connus ;
  - la classe la moins coûteuse si des coûts de mauvais classement sont donnés.

2.3 Critères d'homogénéité

2.3.1 Y quantitative

"Variance inter classe" ou "désordre" des barycentres :

$$\Delta = n_1 n_2 (\mu_{.1} - \mu_{.2})^2$$

Objectif à chaque étape : maximiser Δ.

- Chercher la division rendant le test de Fisher le plus significatif possible.
- Critère équivalent à la déviance d'un modèle gaussien.

2.3.2 Y qualitative

- Fonction d'hétérogénéité : entropie, critère de concentration de Gini ou statistique du test du χ^2.
- L'entropie est le terme de déviance d'un modèle multinomial.

Y qualitative à m modalités ou catégories $\mathcal{T}$ numérotées $\ell = 1, \ldots, m$.
L'arbre induit une partition où n_{+k} est l'effectif du $k$ème nœud.
Probabilité qu'un élément du $k$ème nœud appartienne à la ℓème classe de Y.

$$p_{\ell k} = P[\mathcal{T}_\ell \mid k] \quad \text{avec} \quad \sum_{\ell=1}^{m} p_{\ell k} = 1.$$

Désordre du $k$ème nœud, ou entropie, (convention $0 \log(0) = 0$). :

$$D_k = -2 \sum_{\ell=1}^{m} n_{+k} p_{\ell k} \log(p_{\ell k}).$$

Hétérogénéité ou désordre de la partition :

$$D = \sum_{k=1}^{K} D_k = -2 \sum_{k=1}^{K} \sum_{\ell=1}^{m} n_{+k} p_{\ell k} \log(p_{\ell k}).$$

Quantité positive ou nulle, nulle ssi les probabilités $p_{\ell k}$ sont toutes nulles sauf une égale à 1.

- $n_{\ell k}$ effectif observé de la ℓème classe dans le $k$ème nœud.
- Un nœud k est un sous-ensemble de l'échantillon d'effectif $n_{+k} = \sum_{\ell=1}^{m} n_{\ell}$

$$\mathcal{D} = -2 \sum_{k=1}^{K} \sum_{\ell=1}^{m} n_{\ell k} \log \frac{n_{\ell k}}{n_{+k}} = \widehat{D}.$$

- Les probabilités conditionnelles sont définies par la règle de Bayes lorsque les probabilités *a priori* π_ℓ sont connues.
- Sinon, les probabilités de chaque classe sont estimées sur l'échantillon et donc les probabilités conditionnelles s'estiment par des rapports d'effectifs :

$$p_{\ell k} \quad \text{est estimée par} \quad n_{\ell k}/n_{+k}.$$

- Des coûts de mauvais classement connus conduisent à la minimisation d'un risque bayésien.

2.4 Élagage

Recherche d'un modèle parcimonieux.

2.4.1 Construction de la séquence d'arbres

Complexité d'un arbre A :
$H(A) =$ nombre de feuilles de A.
Qualité de *discrimination* de A :

$$D(A) = \sum_{h=1}^{H} D_h(A).$$

où $D_h(A)$: nombre de mal classés ou déviance ou le coût de mauvais classement de la feuille h.

Critère de qualité pénalisé par la complexité :

$$C(A) = D(A) + \gamma H.$$

Pour $\gamma = 0$: $A_{\max} = A_H$ minimise $C(A)$.
Lorsque γ croît, la division de A_H, dont l'amélioration de D est inférieure à γ, est annulée ; ainsi,

- deux feuilles sont regroupées (élaguées),
- le nœud père devient terminal,
- A_H devient A_{H-1}.

Après itération du procédé :

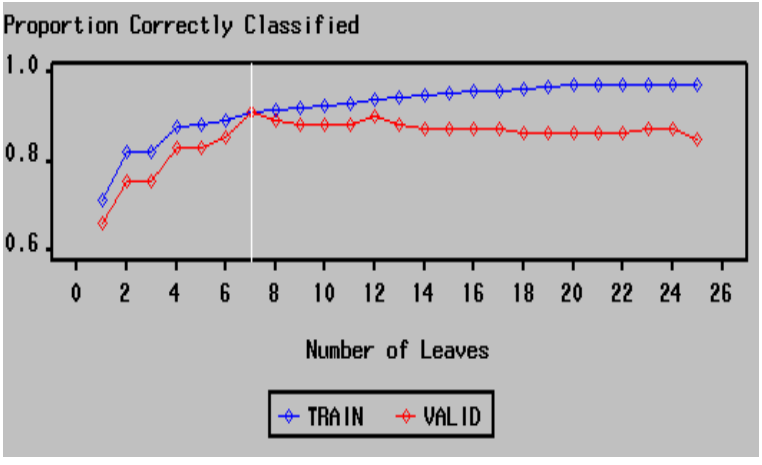
$$A_{\max} = A_H \supset A_{H-1} \supset \cdots A_1.$$

2.4.2 Recherche de l'arbre optimal

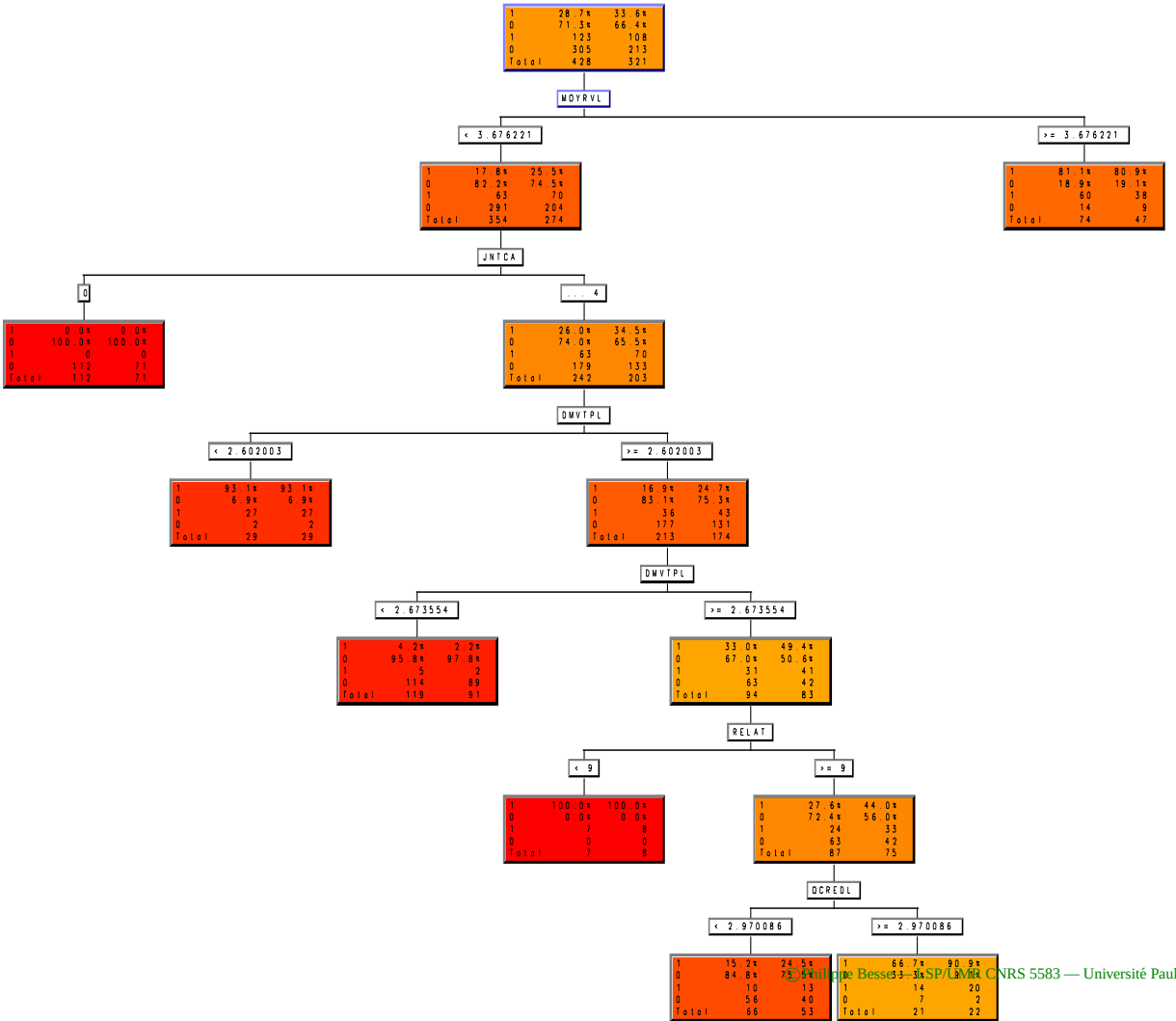
- Construction de l'arbre maximal $A_{\max}$.
- Construction de la séquence $A_K \dots A_1$ d'arbres emboîtés.
- Estimation sans biais (échantillon de validation ou validation croisée) des déviances $D(A_K), \dots, D(A_1)$.
- Représentation de $D(A_k)$ en fonction de k ou de γ.
- Choix de k rendant $D(A_k)$ minimum.

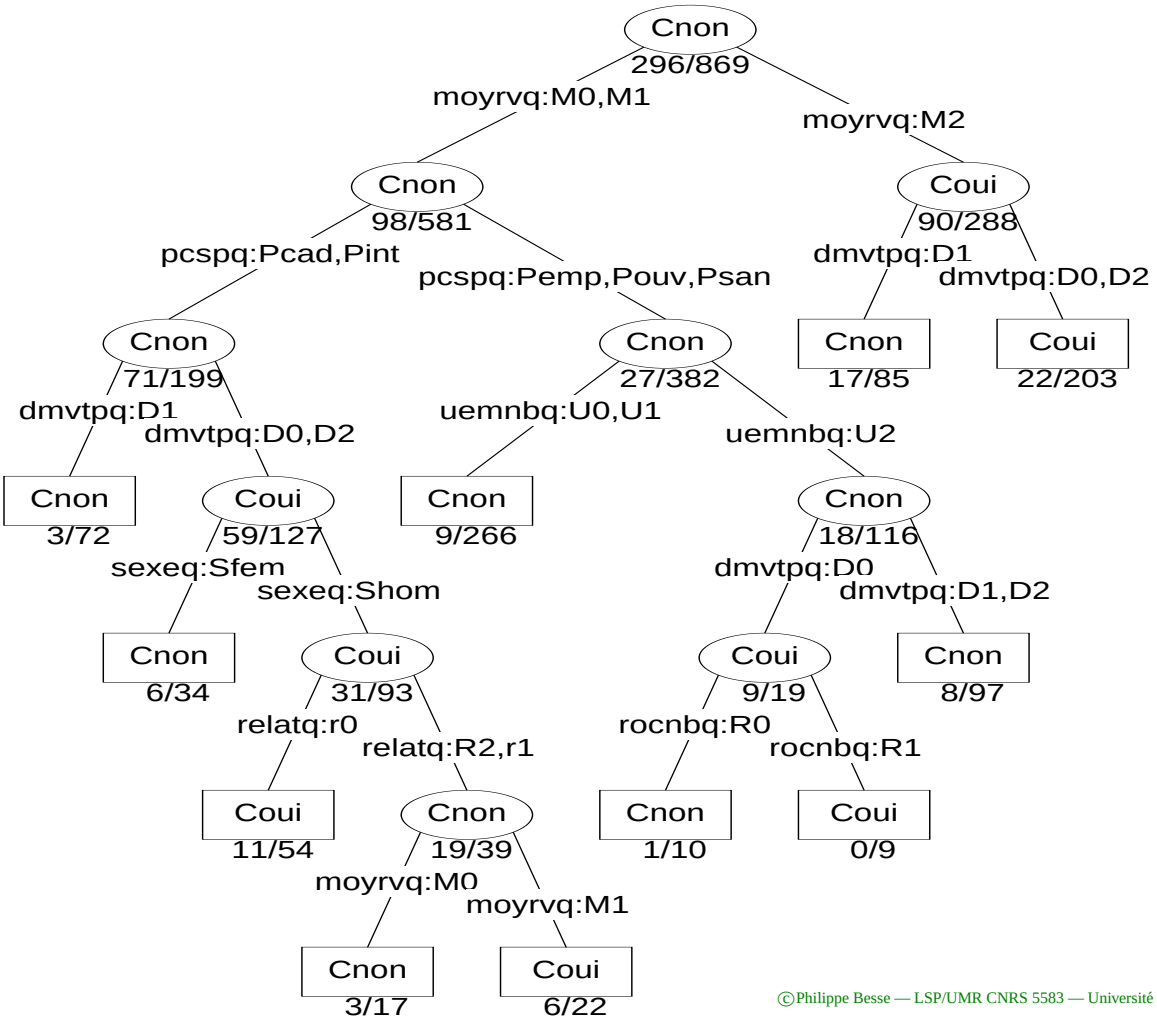
2.4.3 Exemple : marketing bancaire

Identif.	Libellé	Identif.	Libellé
matric	Matricule (identifiant client)	qcred	Moyenne des mouvements créditeurs en Kf
sexec	Sexe (qualitatif)	dmvtp	Age du dernier mouvement (en jours)
ager	Age en années	boppn	Nombre d'opérations à M-1
famil	Situation familiale	facan	Montant facturé dans l'année en francs
	(Fmar : marié, Fcel : célibataire, Fdiv :divorcé,	lgagt	Engagement long terme
	Fuli :union libre, Fsep : séparé de corps, Fveu :veuf)	vienb	Nombre de produits contrats vie
relat	Ancienneté de relation en mois	viemt	Montant des produits contrats vie en francs
prcsp	Catégorie socio-professionnelle (code num)	uemnb	Nombre de produits épargne monétaire
opgnb	Nombre d'opérations par guichet dans le mois	xlgnb	Nombre de produits d'épargne logement
moyrv	Moyenne des mouvements nets créditeurs	xlgmt	Montant des produits d'épargne logement en francs
	des 3 mois en Kf	ylvnb	Nombre de comptes sur livret
tavep	Total des avoirs épargne monétaire en francs	ylvmt	Montant des comptes sur livret en francs
endet	Taux d'endettement	rocnb	Nombre de paiements par carte bancaire à M-1
gaget	Total des engagements en francs	jntca	Nombre total de cartes
gagec	Total des engagements court terme en francs	nptag	Nombre de cartes point argent
gagem	Total des engagements moyen terme en francs	itavc	Total des avoirs sur tous les comptes
kvunb	Nombre de comptes à vue	havef	Total des avoirs épargne financière en francs
qsmoy	Moyenne des soldes moyens sur 3 mois	dnbjd	Nombre de jours à débit à M
		carvp	**Possession de la carte VISA Premier**



Carte Visa : choix du nombre de feuilles par échantillon de validation (SEM, 2001).





3 Agrégation de modèles

3.1 Introduction

- Stratégies adaptatives (boosting) ou aléatoires (bagging).
- Combinaison ou agrégation de modèles sans sur-ajustement.
- Apprentissage machine (machine learning) et Statistique.
- Comparatifs heuristiques et propriétés théoriques.

- Bagging pour bootstrap aggregating (Breiman, 1996),
- forêts aléatoires (random forests) (Breiman, 2001),
- boosting (Freund et Shapiro,1996) déterministe et adaptatif.

- Toute méthode de modélisation non linéaire.

3.2 Famille de modèles aléatoires

3.2.1 Bagging

Principe et algorithme

Soit Y une variable à expliquer quantitative ou qualitative,
$X^1, \ldots, X^p$ les variables explicatives,
$\phi(\mathbf{x})$ un modèle fonction de $\mathbf{x} = \{x^1, \ldots, x^p\} \in \mathbb{R}^p$.
$\mathbf{z} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ échantillon de loi F et de taille n.

- $\phi(.) = E_F(\widehat{\phi}_{\mathbf{z}})$ estimateur sans biais de variance nulle.
- B échantillons indépendants $\{\mathbf{z}_b\}_{b=1,B}$
  - Y quantitative : $\widehat{\phi}_B(.) = \frac{1}{B}\sum_{b=1}^{B} \widehat{\phi}_{\mathbf{z}_b}(.)$ (moyenne),
  - Y qualitative : $\widehat{\phi}_B(.) = \arg\max_j \text{card}\left\{b \mid \widehat{\phi}_{\mathbf{z}_b}(.) = j\right\}$ (vote).

Principe : Moyenner des prédictions indépendantes pour réduire la variance.

- B échantillons indépendants remplacés par B réplications bootstrap.

ALGORITHME **1** : **Bagging**

- *Soit* $\mathbf{x}_0$ *à prévoir et*
- $\mathbf{z} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ *un échantillon*
- `Pour` $b = 1$ `à` B `Faire`
  - *Tirer un échantillon bootstrap* $\mathbf{z}_b^*$.
  - *Estimer* $\widehat{\phi}_{\mathbf{z}_b}(\mathbf{x}_0)$ *sur l'échantillon bootstrap.*
- `Fin Pour`

  Calculer l'estimation moyenne $\widehat{\phi}_B(\mathbf{x}_0) = \frac{1}{B}\sum_{b=1}^{B} \widehat{\phi}_{\mathbf{z}_b}(\mathbf{x}_0)$ *ou le résultat du vote.*

Utilisation

- Estimation bootstrap out-of-bag de l'erreur de prédiction : contrôle de la qualité et du surajustement.
- CART pour construire une famille d'arbres binaires.
- Trois stratégies d'élagage sont alors possibles :
  - garder un arbre complet pour chacun des échantillons,
  - arbre d'au plus q feuilles,
  - arbre complet élagué par validation croisée.

Première stratégie compromis entre calculs et qualité de prédiction : faible biais de chaque arbre et variance réduite par agrégation.

Problèmes :

- temps de calcul et contrôle de l'erreur,
- stockage de tous les modèles de la combinaison,
- modèle boîte noire.

3.2.2 Forêts aléatoires

Algorithme

- Amélioration du bagging des Modèles CART (arbres binaires),
- ajout d'une randomisation pour rendre les arbres plus indépendants,
- choix raléatoire des variables.

Intérêt : situations hautement multidimensionnelles.

ALGORITHME **2** : **Forêts aléatoires**

- *Soit* $\mathbf{x}_0$ *à prévoir et*
- $\mathbf{z} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ *un échantillon*
- `Pour` $b = 1$ `à` B `Faire`
  - *Tirer un échantillon bootstrap* $\mathbf{z}_b^*$
  - *Estimer un arbre avec randomisation des variables :*
    1. `Si` p *grand, la recherche de chaque nœud optimal est précédé d'un tirage aléatoire d'un sous-ensemble de* q *prédicteurs.*
    2. `Sinon`, *tirer* $q_1 \approx 3$ *variables explicatives puis construire* q_2 *"prédicteu par combinaisons linéaires avec des coefficients uniformes sur* $[0, 1]$.
- `Fin Pour`

  Calculer l'estimation moyenne $\widehat{\phi}_B(\mathbf{x}_0) = \frac{1}{B} \sum_{b=1}^{B} \widehat{\phi}_{\mathbf{z}_b}(\mathbf{x}_0)$ *ou le vote.*

Élagage

- Arbres de taille q réduite voire triviale : $q = 2$ (stump).
- La sélection aléatoire des prédicteurs ($q = \sqrt{p}$) accroît la variabilité.
- Chaque modèle de base est moins performant mais l'agrégation est performante.
- Évaluation itérative de l'erreur out-of-bag.

Interprétation : graphe d'un indice proportionnel à l'importance de chaque variable.

- Moyenne sur toutes les observations de la décroissance de leur marge lorsque la variable est aléatoirement perturbée.
- Marge d'une observation : proportion de votes pour la vraie classe moins le maximum des proportions des votes pour les autres classes.

3.3 Famille de modèles adaptatifs

3.3.1 Principes du *Boosting*

- Améliorer les compétences d'un faible classifieur (Schapire, 1990 ; Freund et Schapire, 1996).
- AdaBoost (Adaptative boosting) prédiction d'une variable binaire.
- k classes, problèmes de régression, intérêt pratique.
- Réduire la variance mais aussi le biais de prédiction.
- Meilleure méthode "off-the-shelf".
- Agrégation d'une famille de modèles récurents.

Chaque modèle est une version adaptative du précédent en donnant plus de poids, lors de l'estimation suivante, aux observations mal ajustées.

- Variantes : type de la variable à prédire (binaire, k clases, réelles), fonction perte (robustesse).

3.3.2 Algorithme de base

δ : fonction de discrimination $\{-1, 1\}$.

ALGORITHME **3** : **AdaBoost** *(adaptative boosting) discret*

- *Soit* $\mathbf{x}_0$ *à prévoir et*
- $\mathbf{z} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ *un échantillon*
- `Initialiser` *les poids* $\boldsymbol{w} = \{w_i = 1/n \; ; \; i = 1, \ldots, n\}$.
- `Pour` $m = 1$ `à` M `Faire`
  - *Estimer* δ_m *sur l'échantillon pondéré par* $\boldsymbol{w}$.
  - *Calculer le taux d'erreur apparent :* $\widehat{\mathcal{E}}_p = \frac{\sum_{i=1}^n w_i \mathbf{1}\{\delta_m(\boldsymbol{x}_i) \neq y_i\}}{\sum_{i=1}^n w_i}$.
  - *Calculer les logit :* $c_m = \log((1 - \widehat{\mathcal{E}}_p)/\widehat{\mathcal{E}}_p)$.
  - *Nouvelles pondérations :* $w_i \leftarrow w_i . \exp\left[c_m \mathbf{1}\{\delta_m(\boldsymbol{x}_i) \neq y_i\}\right] ; i = 1, \ldots, n$.
- `Fin Pour`

  Résultat du vote : $\widehat{\phi}_M(\boldsymbol{x}_0) = \textit{signe}\left[\sum_{m=1}^M c_m \delta_m(\boldsymbol{x}_0)\right]$.

- Arbre comme modèle de base.
  – Avec $q = 2$, AdaBoost mieux qu'un arbre sophistiqué.
  – Recommandation : q entre 4 et 8.
- Variantes : Adaboost M1, M2, MH ou encore MR. Schapire (2002).
- Composante aléatoire : Arcing (Breiman, 1998)

3.3.3 Pour la régression

- Algorithme de Drucker (1997), présentation de Gey et Poggi (2002)
- Freund et Schapire (1996) ont proposé *Adaboost.R*
- Friedman (2002) propose MART.

ALGORITHME : **Boosting pour la régression**

- Soit $\mathbf{x}_0$ à prévoir et $\mathbf{z} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ un échantillon
- `Initialiser` $\boldsymbol{p}$ distribution uniforme $\mathbf{p} = \{p_i = 1/n \ ; \ i = 1, \ldots, n\}$.
- `Pour` $m = 1$ `à` M `Faire`
  - Tirer avec remise dans $\mathbf{z}$ un échantillon $\mathbf{z}_m^*$ suivant $\mathbf{p}$.
  - Estimer $\widehat{\phi}_m$ sur l'échantillon $\mathbf{z}_m^*$.
  - Calculer à partir de l'échantillon initial $\mathbf{z}$:

$$l_m(i) \ = \ Q\left(y_i, \widehat{\phi}_m(\mathbf{x}_i)\right) \quad i = 1, \ldots, n; \quad (Q : \text{fonction perte})$$

$$\widehat{\mathcal{E}_m} \ = \ \sum_{i=1}^{n} p_i l_m(i); \quad w_i = g(l_m(i)) p_i.$$

  - Calculer les nouvelles probabilités : $p_i \leftarrow \frac{w_i}{\sum_{i=1}^n w_i}$.
- `Fin Pour`
- Calculer $\widehat{\phi}(\mathbf{x}_0)$ moyenne ou médiane des prévisions $\widehat{\phi}_m(\mathbf{x}_0)$ pondérées par des coefficients $\log(\frac{1}{\beta_m})$.

- Q peut être exponentielle, quadratique ou la valeur absolue.
- $L_m = \sup_{i=1,\dots,n} l_m(i)$ maximum de l'erreur observée par le modèle $\widehat{\phi}_m$ sur l'échantillon initial.

$$g(l_m(i)) = \beta_m^{1-l_m(i)/L_m} \quad (1)$$

$$\text{avec} \quad \beta_m = \frac{\widehat{\mathcal{E}_m}}{L_m - \widehat{\mathcal{E}_m}}. \quad (2)$$

- Algorithme arrêté ou réinitiallisé à des poids uniformes si l'erreur se dégrade trop : si $\widehat{\mathcal{E}_m} < 0.5 L_m$.

3.3.4 Modèle additif pas à pas

Approximation de ϕ par un modèle additif pas à pas (Hastie et col., 2001).

$$\widehat{\phi}(\boldsymbol{x}) = \sum_{m=1}^{M} c_m \delta(\boldsymbol{x}; \gamma_m)$$

c_m est un paramètre,
δ le classifieur de base fonction de $\boldsymbol{x}$ et dépendant d'un paramètre γ_m,
Q une fonction perte.

Problème :

$$(c_m, \gamma_m) = \arg\min_{(c,\gamma)} \sum_{i=1}^{n} Q(y_i, \widehat{\phi}_{m-1}(\boldsymbol{x}_i) + c\delta(\boldsymbol{x}_i; \gamma));$$

$\widehat{\phi}_m(\boldsymbol{x}) = \widehat{\phi}_{m-1}(\boldsymbol{x}) + c_m\delta(\boldsymbol{x}; \gamma_m)$ améliore l'ajustement précédent.
ϕ binaire, $Q(y, \phi(\boldsymbol{x})) = \exp[-y\phi(\boldsymbol{x})]$.

$$\begin{aligned} \text{Résoudre} \quad (c_m, \gamma_m) &= \arg\min_{(c,\gamma)} \sum_{i=1}^{n} \exp\left[-y_i\widehat{\phi}_{m-1}(\boldsymbol{x}_i) + c\delta(\boldsymbol{x}_i; \gamma)\right]; \\ &= \arg\min_{(c,\gamma)} \sum_{i=1}^{n} w_i^m \exp\left[-cy_i\delta(\boldsymbol{x}_i; \gamma)\right] \\ \text{avec} \quad w_i &= \exp[-y_i\widehat{\phi}_{m-1}(\boldsymbol{x}_i)]; \end{aligned}$$

w_i ne dépendant ni de c ni de γ : "poids" fonction de la qualité de l'ajustement précédent.

Solution du problème de minimisation en deux étapes :
Recherche du classifieur optimal puis optimisation du paramètre γ.

$$\begin{aligned} \gamma_m &= \arg\min_{\gamma} \sum_{i=1}^{n} \mathbf{1}\{y_i \neq \delta(\boldsymbol{x}_i;\gamma)\}, \\ c_m &= \frac{1}{2}\log\frac{1-\widehat{\mathcal{E}}_p}{\mathcal{E}_p} \end{aligned}$$

avec $\widehat{\mathcal{E}}_p$ erreur apparente de prédiction ; les w_i sont mis à jour avec :

$$w_i^{(m)} = w_i^{(m-1)} \exp[-c_m].$$

Adaboost approche ϕ pas à pas par un modèle additif en utilisant une fonction perte exponentielle.
D'autres fonctions perte (robustesse) mais algorithmes plus compliqués.

3.3.5 MART

(multiple additive regression trees) (Friedman, 2002)
Modèles : arbres de régression avec fonction perte différentiable.
Principe :

- Construire une séquence de modèles de sorte qu'à chaque étape, chaque modèle ajouté à la combinaison, apparaisse comme un pas vers une meilleure solution.
- Ce pas est franchi dans la direction du gradient, approché par un arbre de régression, de la fonction perte.

ALGORITHME **4** : **MART** *(Multiple additive regression trees)*

- *Soit* $\mathbf{x}_0$ *à prévoir*
- `Initialiser` $\widehat{\phi}_0 = \boldsymbol{arg}\min_\gamma \sum_{i=1}^n Q(y_i, \gamma)$
- `Pour` $m = 1$ `à` M `Faire`
  - *Calculer* $r_{im} = -\left[\frac{\delta Q(y_i, \phi(\boldsymbol{x}_i))}{\delta \phi(\boldsymbol{x}_i)}\right]_{\phi=\phi_{m-1}}$,
  - *Ajuster un arbre de régression aux* $r_m i$ *donnant les feuilles ou régions terminales* $R_{jm}; j = 1, \ldots, J_m$.
  - `Pour` $m = 1$ `à` M `Faire`
    - *Calculer* $\gamma_{jm} = \boldsymbol{arg}\min_\gamma \sum_{\boldsymbol{x}_i \in R_{jm}} Q(y_i, \widehat{\phi}_{m-1} + \gamma)$.
  - `Fin Pour`
  - *Mise à jour :* $\widehat{\phi}_m(\boldsymbol{x}) = \widehat{\phi}_m(\boldsymbol{x}) \sum_{j=1}^{J_m} \gamma_{jm} \mathbf{1}\{\boldsymbol{x} \in R_{jm}\}$.
- `Fin Pour`
- *Résultat* : $\widehat{\phi}_M(\boldsymbol{x}_0)$.

3.3.6 Compléments

Sur-ajustement

- Nombre d'itérations contrôlé par un échantillon de validation.
- Coefficient de rétrécissement ou taux d"'apprentissage".

Interprétation

- Problème : interprétabilité
- Critère d'importance relative des variables.
- Pour chaque variable j à partir des valeurs $D_j^2(l,b)$, calculées pour chaque nœud l de chaque arbre b.
- Décroissance optimale de déviance produite par la segmentation associée à ce nœud par le choix de la variable j.
- Valeurs sont sommées par arbre sur l'ensemble des nœuds puis moyennées sur l'ensemble des arbres.

Propriétés

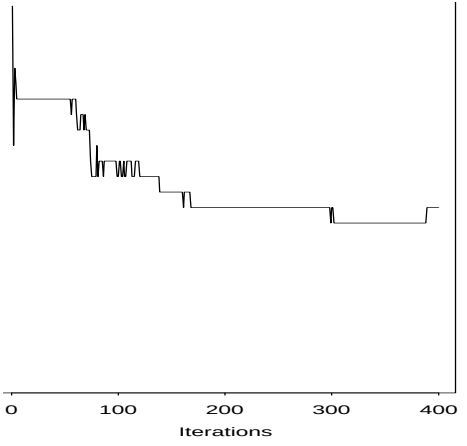
- Ce type d'algorithme, fait mieux que l'asymptotique.
- Empiriquement, l'erreur de prédiction continue à décroître après que l'erreur d'ajustement se soit annuler.
- Approche "stochastique" : même déterministe, l'algorithme simule une dynamique markovienne (Blanchard, 2001).
- Procédure d'optimisation globale par une méthode de gradient (Friedman, 2001).
- La probabilité d'erreur du boosting converge avec n vers celle du classifieur bayésien (Lugosi et Vayatis, 2001).

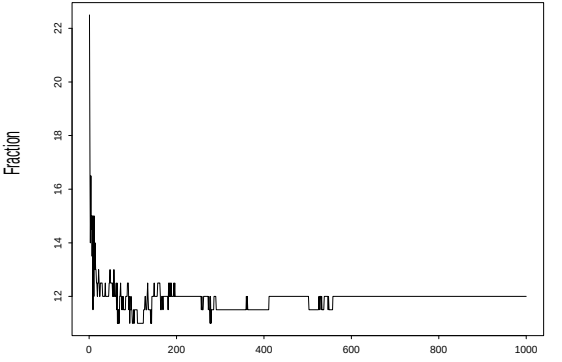
3.4 Application

3.4.1 Logiciels

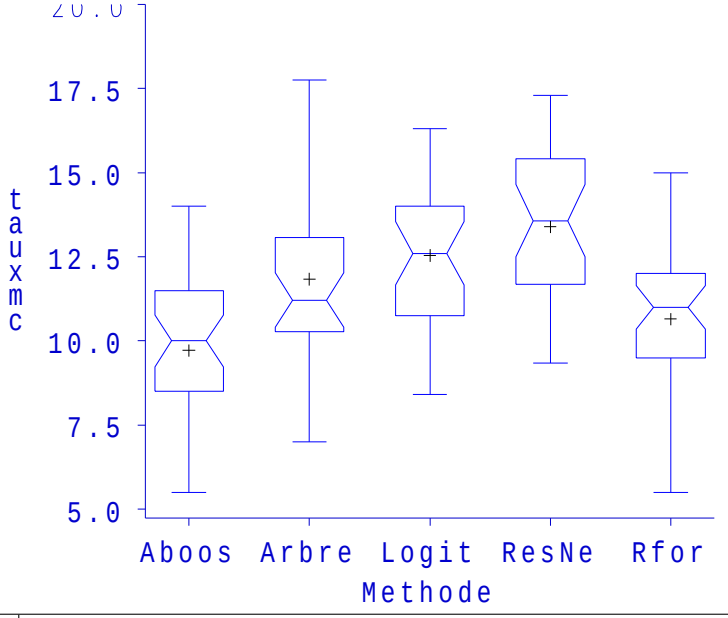
`www-stat.stanford.edu/~jhf/MART.html`
`www.research.att.com/~schapire`
`www.stat.Berkeley.edu/users/breiman/`
ou avec R : `www.r-project.org`.

3.4.2 Résultats comparatifs





Évolution du taux de mal classés en fonction du nombre d'arbres dans la combinaison de modèles.



Méthode	Adaboost	Arbre	Régression	Perceptron	Forêt
Moyenne	9.7	11.8	12.5	13.4	10.6
Écart-type	2.0	2.3	2.0	2.3	2.2

Table des matières