Arbres CART et Forêts aléatoires - Importance et sélection de variables

Robin Genuer (ISPED, Bordeaux)
Jean-Michel Poggi (Paris Descartes et LMO, Orsay)

Remerciements à S. Arlot, S. Gey, C. Tuleau-Malot et N. Villa-Vialaneix

"Apprentissage statistique et données massives"
JES de la SFdS

2-7 Octobre 2016

Plan

Leo Breiman

- De CART aux RF : 20 ans d'une trajectoire scientifique
- Olshen, Breiman (2001) et Cutler (2010)
- D'abord, en probabilités sous un angle très proche des mathématiques pures, il a ensuite marqué de son empreinte la statistique appliquée et l'apprentissage
- Série de papiers dans les *Annals of Statistics* et dans *Machine Learning*

Cadre

$\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ v.a. i.i.d. de même loi que (X, Y).

$X \in \mathbb{R}^p$ (variables explicatives) ; on peut aussi avoir $X \in \mathbb{R}^{p'} \otimes \mathcal{Q}$ mixte. $Y \in \mathcal{Y}$ (réponse) :

- $\mathcal{Y} = \mathbb{R}$: régression
- $\mathcal{Y} = \{1, \ldots, L\}$: classification

But : construire un prédicteur $\widehat{h} : \mathbb{R}^p \to \mathcal{Y}$

Arbres CART Breiman et al. (1984)

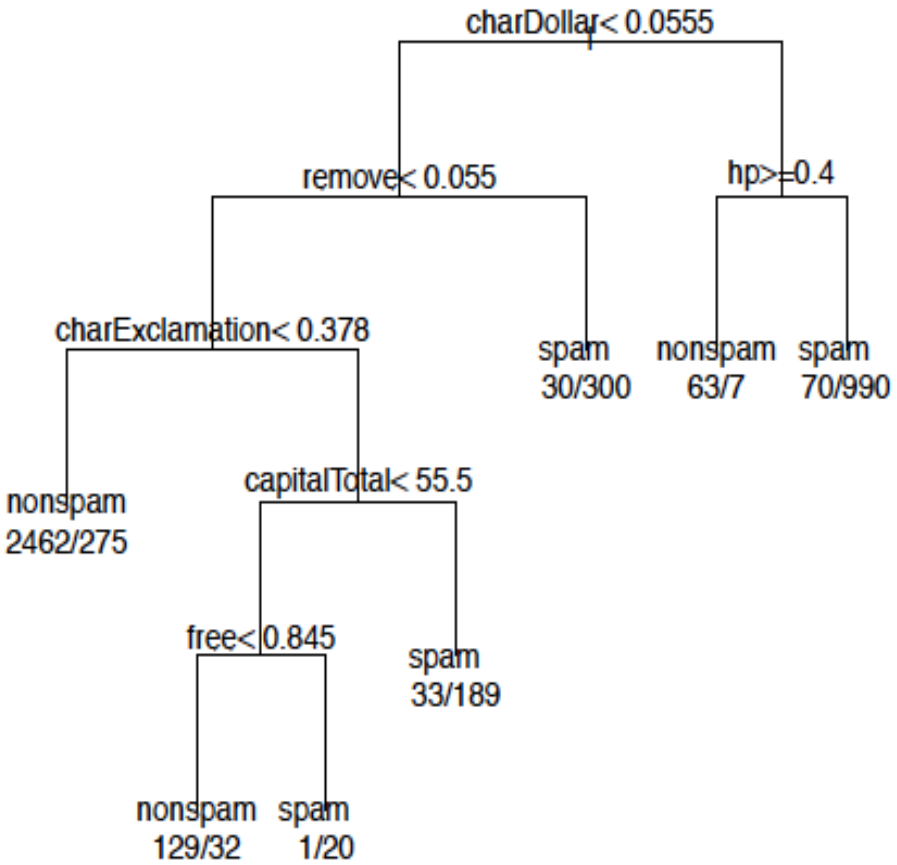
- famille des méthodes d'arbres de décision
- algorithme qui est la base de méthodes très efficaces

Forêts aléatoires Breiman (2001)

- famille des méthodes d'ensemble
- algorithme d'apprentissage statistique très performant, à la fois pour des problèmes de classification et de régression

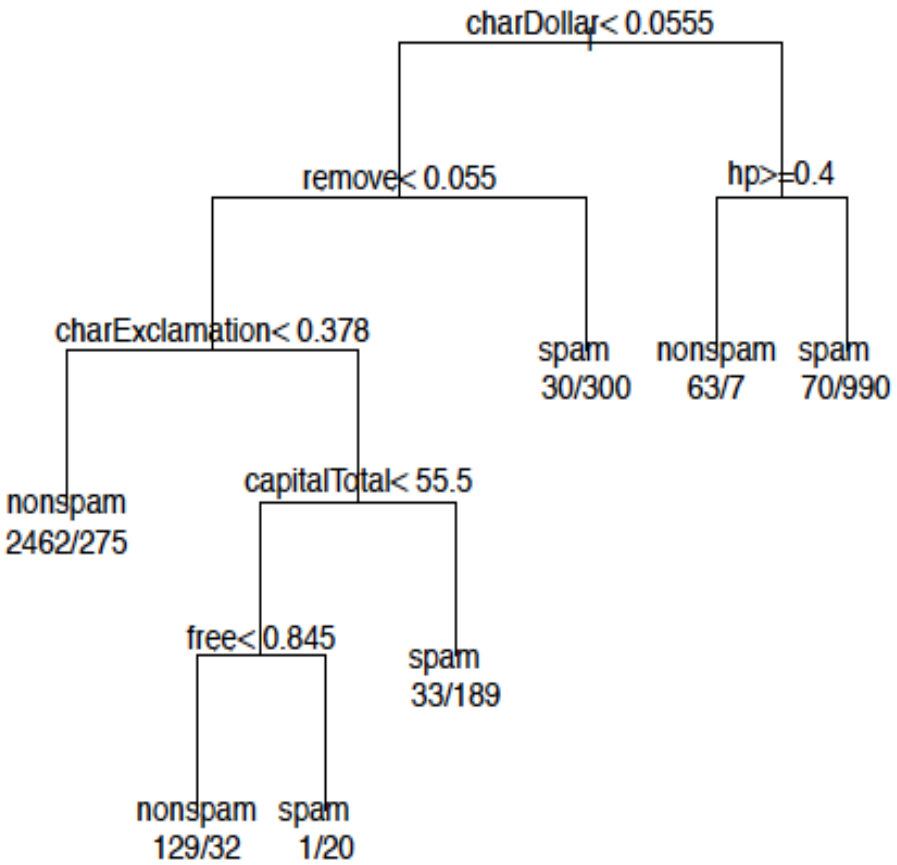
Plan

Le jeu de données *spam*



- Construire un détecteur automatique de spams et déterminer les variables importantes
- n=4601 emails (1813 spams, 40%)
- p=57 prédicteurs :
  - 54 sont des % de mots ou de caractères donnés comme "$", "!", "remove", "free"
  - 2 liées aux longueurs de suites de majuscules (moyenne, maximum) et enfin le nombre de majuscules

Un arbre CART pour les données *spam*



- Structure de l'arbre : 5 noeuds internes et 7 feuilles ; splits basés sur *charDollar*, *remove*, *hp*, *free*, *charExclamation*, et *capitalTotal*
- Prédiction par l'arbre : les feuilles donnent les prédictions de Y (*spam* ou *nonspam*) et sa distribution
- Interprétation : chemin racine - la feuille la plus à droite : si beaucoup de $ et peu de *hp* alors presque toujours spam

CART *vs* arbres de décision

- Parfois introduites avant CART, d'autres méthodes pour construire des arbres de décision sont disponibles :
  - CHAID par Kass (1980)
  - *C*4.5 par Quinlan (1993)
- La méthode des arbres de décision souffrait de fortes critiques justifiées et CART leur offre un cadre conceptuel de type sélection de modèles, qui leur confère ainsi à la fois une large applicabilité, une facilité d'interprétation et des garanties théoriques
- L'actualité des arbres de décision perceptible dans deux synthèses récentes :
  - Patil et Bichkar (2012) en informatique
  - Loh (2014) en statistique

Construction

Arbre : prédicteur constant par morceaux, obtenu par partitionnement récursif binaire de $\mathbb{R}^p$

Restriction : coupures parallèles aux axes

Classiquement, à chaque étape du partitionnement binaire, on vise à séparer "au mieux" les données du noeud courant, en recherchant la coupure qui conduit à la plus forte décroissance de l'hétérogénéité des deux noeuds fils

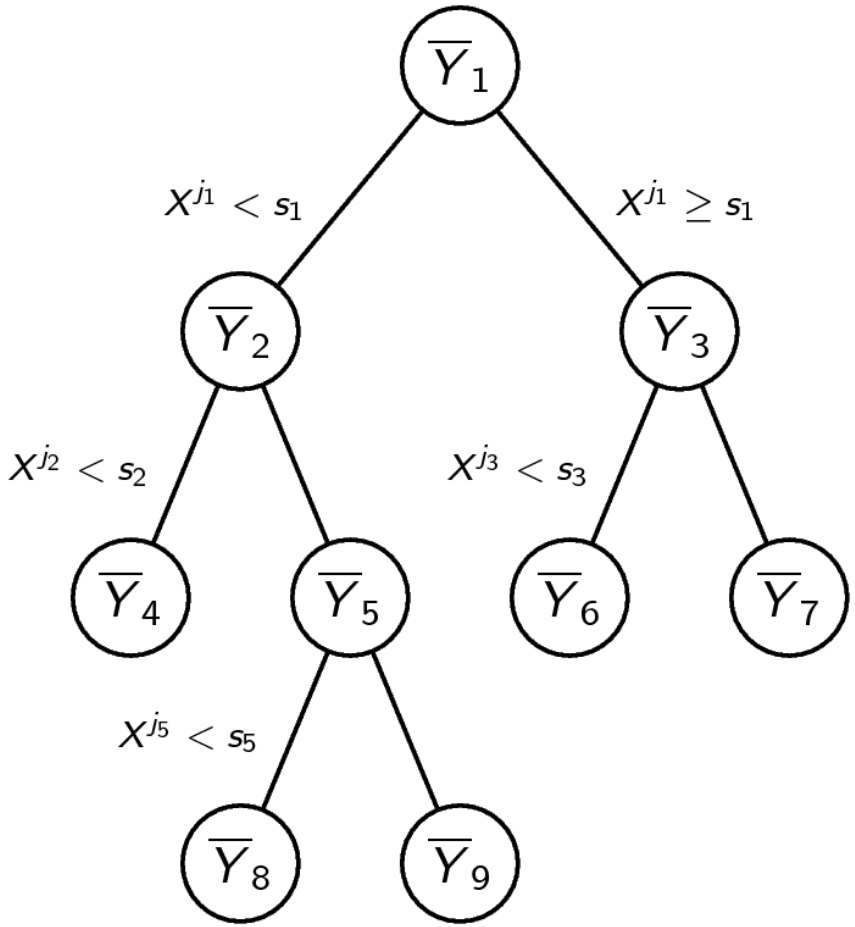
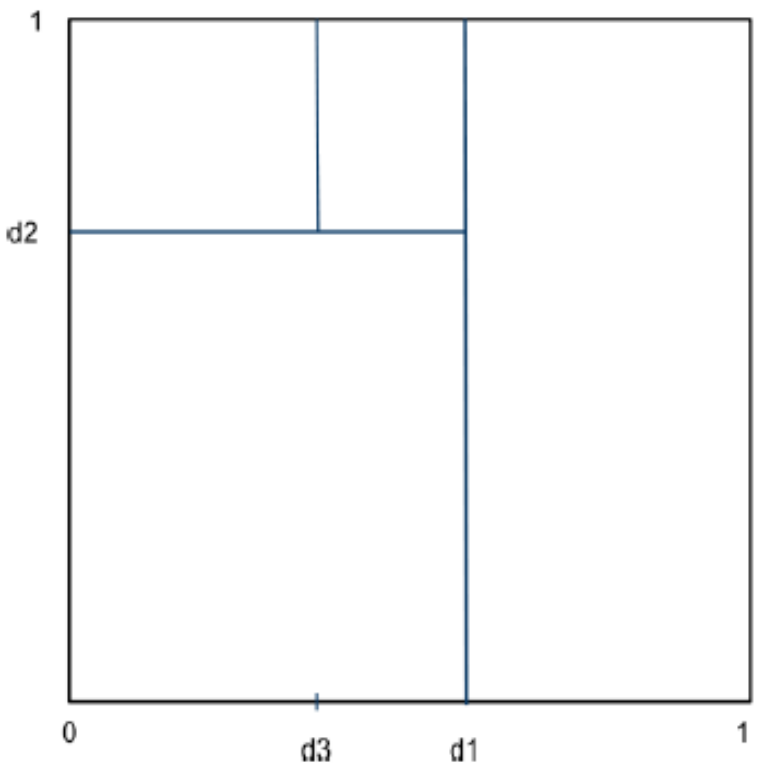
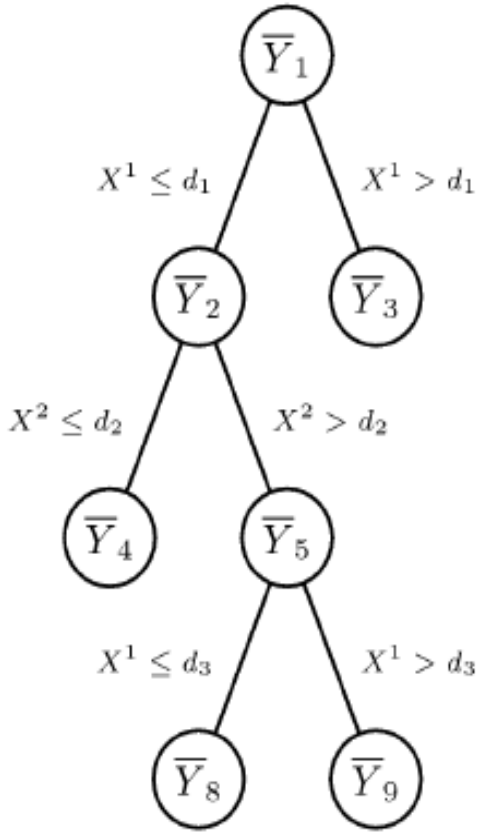


FIGURE : Arbre de régression

Arbre CART et fonction constante par morceaux





Arbre de régression *vs* de classification

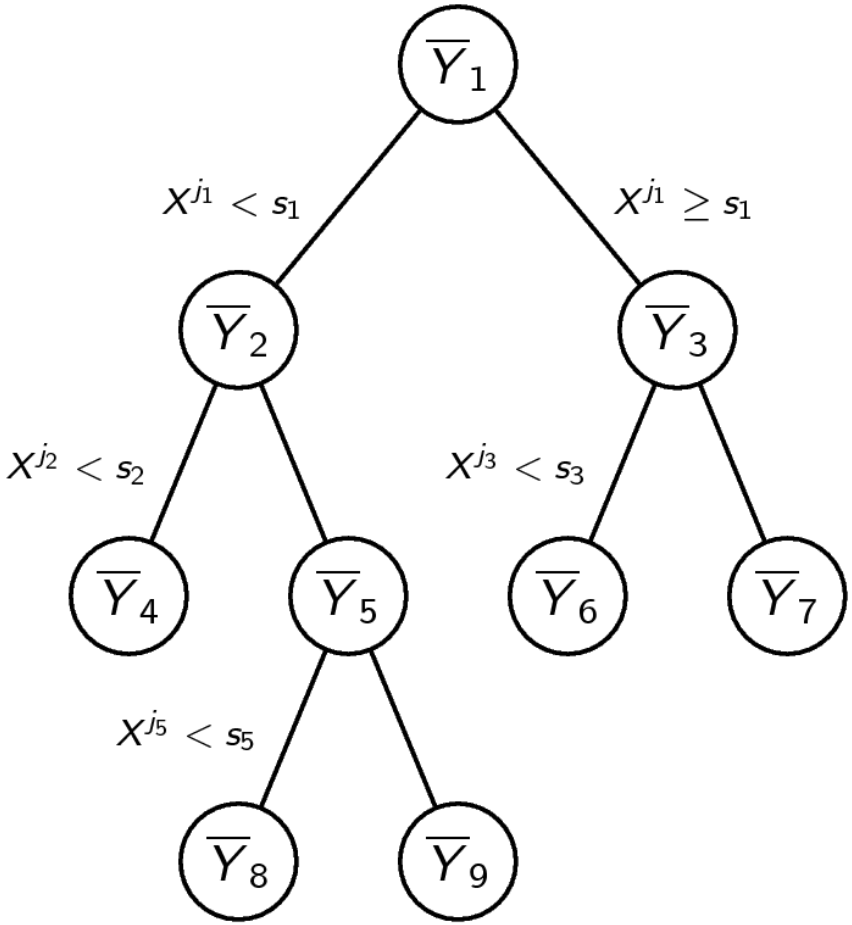


FIGURE : Arbre de régression

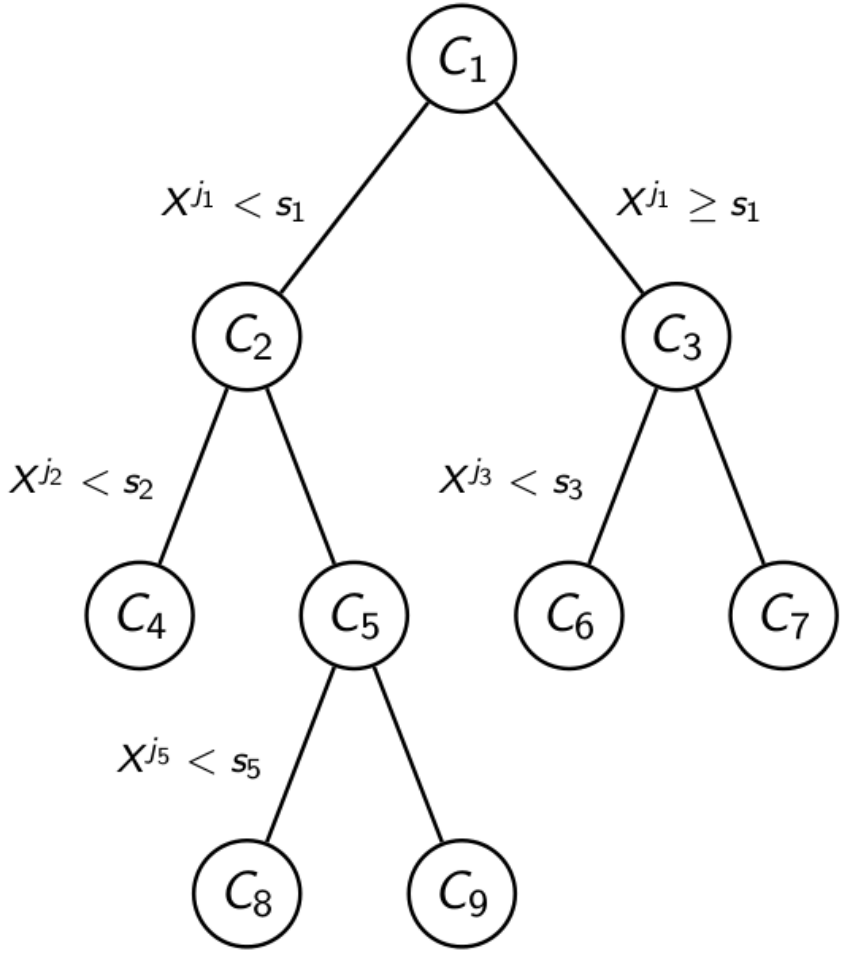


FIGURE : Arbre de classification

Construction

- Coupure (découpe ou split) :
  $\{X^j \leq d\} \cup \{X^j > d\}$ ou $\{X^j \in d\} \cup \{X^j \in \bar{d}\}$
- Régression. Si l'on note la variance d'un nœud t par $V(t) = \frac{1}{\#t} \sum_{i:x_i \in t} (y_i - \bar{y}_t)^2$, on minimise la variance intra-groupes après la découpe de t en 2 fils t_L et t_R, soit

$$\frac{\#t_L}{n} V(t_L) + \frac{\#t_R}{n} V(t_R)$$

- Classification. On définit l'impureté des nœuds fils, le plus souvent par le biais de l'indice de Gini. L'indice de Gini d'un nœud t : $\Phi(t) = \sum_{c=1}^{L} \hat{p}_t^c (1 - \hat{p}_t^c)$, où $\hat{p}_t^c$ est la proportion d'observations de classe c dans le nœud t. On maximise :

$$\Phi(t) - \left(\frac{\#t_L}{\#t} \Phi(t_L) + \frac{\#t_R}{\#t} \Phi(t_R)\right)$$

Arbre maximal et élagage

Arbre maximal, règle d'arrêt :

- Partitionnement récursif par maximisation locale de la décroissance de l'hétérogénéité
- Ne pas découper un noeud pur ou contenant trop peu de données

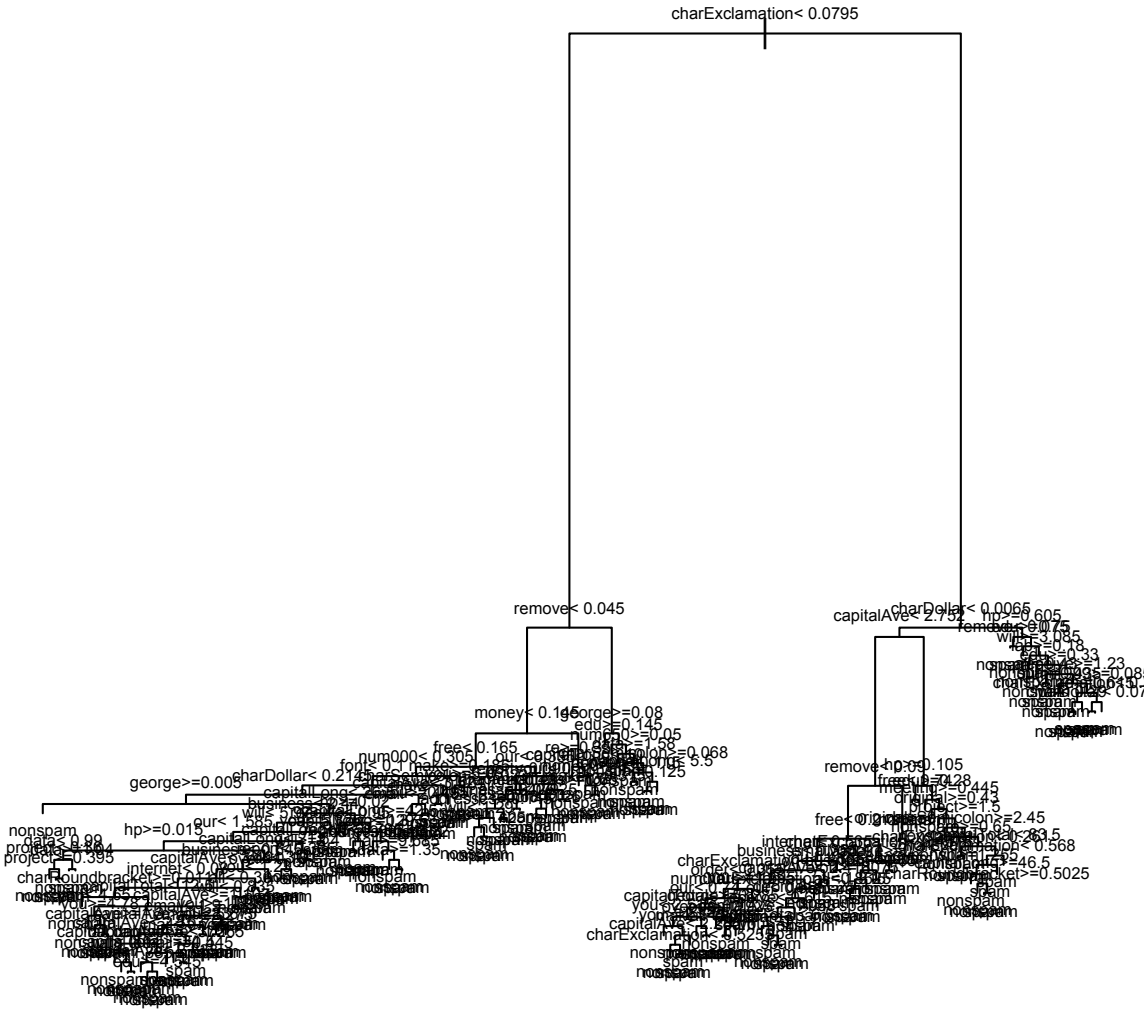
Elagage :

- L'arbre maximal est surajusté aux données
- L'arbre optimal est un sous-arbre élagué minimisant l'erreur de prédiction pénalisée par la complexité du modèle
- Critère pénalisé

$$crit_\alpha(T) = R_n(f, \hat{f}_{|T}, \mathcal{L}_n) + \alpha \frac{|\tilde{T}|}{n}$$

$R_n(f, \hat{f}_{|T}, \mathcal{L}_n)$ le terme d'erreur (MSE en régression ou taux de mauvaises classifications) et $|\tilde{T}|$ le nombre de feuilles de T

Données *spam* : arbre maximal



Algorithme d'élagage

Proposition

La suite de paramètres $(0 = \alpha_1; \ldots; \alpha_K)$ *est strictement croissante, et, pour tout* $1 \leq d \leq K$

$$\begin{aligned} \forall \alpha \in [\alpha_d, \alpha_{d+1}[\; T_d &= \textit{argmin}_{\{T \text{ sous-arbre de } T_{max}\}} \textit{crit}_\alpha(T) \\ &= \textit{argmin}_{\{T \text{ sous-arbre de } T_{max}\}} \textit{crit}_{\alpha_d}(T) \end{aligned}$$

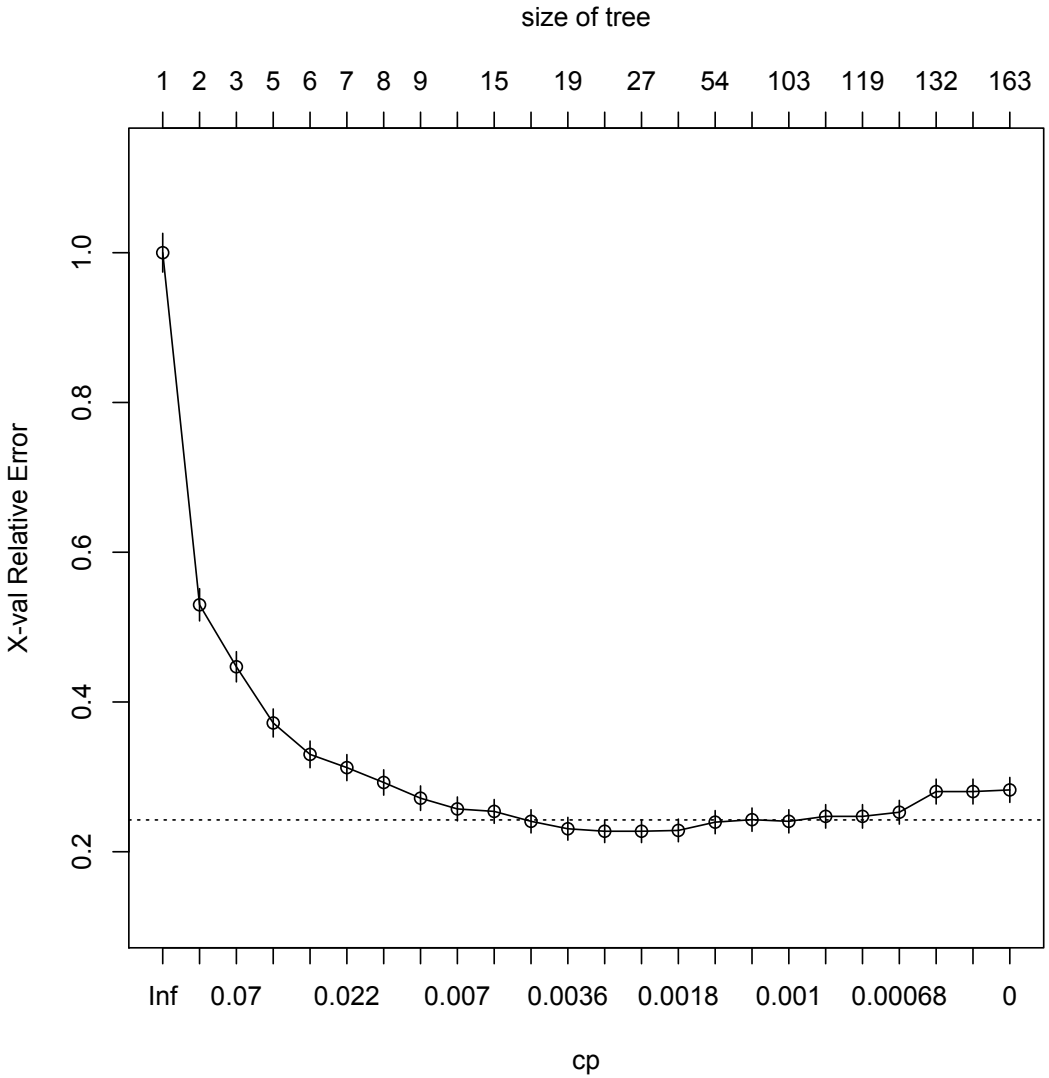
On a donc :

- la suite $T_1, \ldots, T_K$ contient toute l'information statistique
- pour tout $\alpha \geqslant 0$, le sous-arbre minimisant $\textit{crit}_\alpha$ est un sous-arbre de la suite
- algorithme d'élagage itératif : nécessite très peu d'opérations

Algorithme d'élagage

Entrée	Arbre maximal T_{max}
Initialisation	$\alpha_1 = 0$, $T_1 = T_{\alpha_1} = \text{argmin}_{T \text{ élagué de } T_{max}} \overline{err}(T)$ `initialiser` $T = T_1$ et $k = 1$
Iteration	`Tant que` $\lvert T \rvert > 1$, `Calculer` $\alpha_{k+1} = \min_{\{t \text{ nœud interne de } T\}} \frac{\overline{err}(t) - \overline{err}(T_t)}{\lvert T_t \rvert - 1}$ `Elaguer` toutes les branches T_t de T telles que $\overline{err}(T_t) + \alpha_{k+1}\lvert T_t \rvert = \overline{err}(t) + \alpha_{k+1}$ `Prendre` T_{k+1} le sous-arbre élagué ainsi obtenu. `Boucler sur` $T = T_{k+1}$ et $k = k + 1$
Sortie	Arbres $T_1 \succ \ldots \succ T_K = \{t_1\}$ Paramètres $(0 = \alpha_1; \ldots; \alpha_K)$

Données *spam* : suite de sous-arbres élagués



Les meilleurs arbres à k feuilles

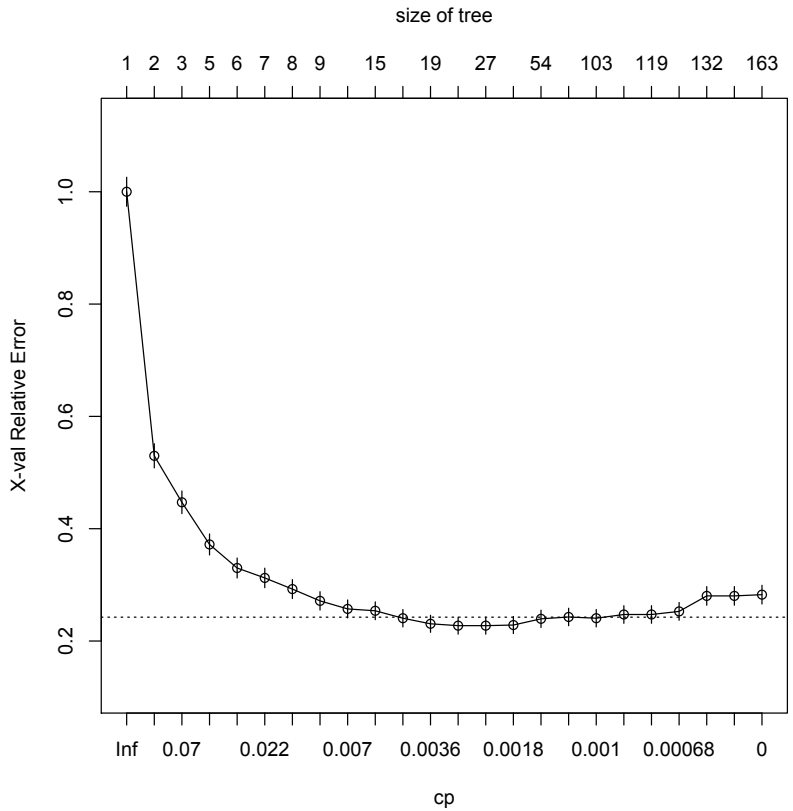


FIGURE : Si un arbre de cette suite comporte k feuilles, c'est le meilleur arbre à k feuilles mais la suite ne contient pas tous les meilleurs arbres à k feuilles pour $1 \leq k \leq |T_{max}|$

Un résultat théorique en classification binaire

- Le critère pénalisé : $\hat{R}_{pen}(T) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\hat{f}_T(X_i)\neq Y_i} + \alpha|T|$
- Lorsque T_{opt} est choisi par la méthode du Hold-out avec un échantillon $\mathcal{L}_1$ pour construire et élaguer T_{max}, un échantillon $\mathcal{L}_2$ pour choisir l'arbre minimisant l'erreur de prédiction

Théorème (Gey 2012)

Sous une condition sur la marge h, il existe C_1, C_2, C_3 telles que :

$$\mathrm{E}\left[l(f^*, \hat{f}_{T_{opt}})|\mathcal{L}_1\right] \leq C_1 \inf_{T \preceq T_{max}} \left[\inf_{f \in S_T} l(f^*, f) + h^{-1}\frac{|T|}{n_1}\right] + \frac{C_2}{n_1} + C_3\frac{\ln n_1}{n_2}$$

où S_T est l'ensemble des classifieurs définis sur la partition induite par l'ensemble des feuilles de T, et
$l(f^*, f) = \mathcal{P}(f(X) \neq Y) - \mathcal{P}(f^*(X) \neq Y)$

Un résultat théorique en régression

Théorème (Gey, Nedelec 2005)

Il existe C_1, C_2, C_3 des constantes positives telles que :

$$\mathrm{E}\left[\|\tilde{f} - f\|^2 | \mathcal{L}_1\right] \leq C_1 \inf_{T \preceq T_{max}} \left[\inf_{u \in S_T} \|u - f\|^2 + \sigma^2 \frac{|\tilde{T}|}{n_1}\right] + \frac{C_2}{n_1} + C_3 \frac{\ln n_1}{n_2}$$

où S_T est l'ensemble des fonctions constantes par morceaux sur la partition engendrée par les feuilles de T

- La performance de l'arbre sélectionné est, au premier ordre, du même ordre de grandeur que la performance du meilleur prédicteur augmentée de la pénalité, en justifiant ainsi la forme
- La qualité de la sélection de l'estimateur est appréciée conditionnellement à l'échantillon $\mathcal{L}_1$, la famille de modèles à l'intérieur de laquelle on fouille étant dépendante des données

CART en pratique

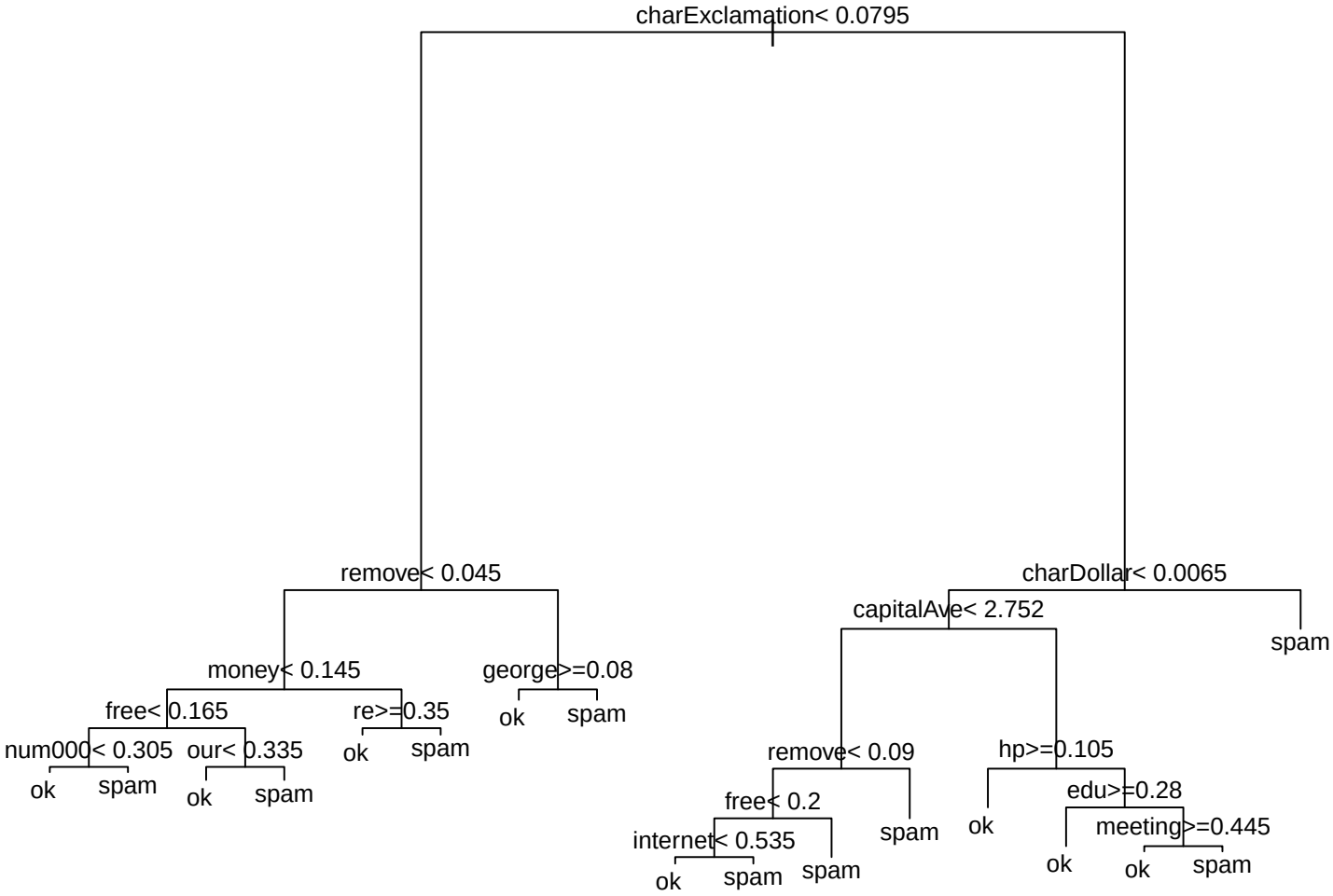
Les arbres CART sont ici obtenus grâce à :

- *R* package *rpart*, voir Therneau et al. (2015)
- avec paramètres par défaut (hétérogénéité de Gini pour la construction de l'arbre maximal et élagage par 10-fold CV)

On considère quatre arbres dans la suite :

- l'arbre ci-dessus obtenu avec les paramètres par défaut
- celui obtenu avec paramètres par défaut avec l'application de la règle du 1 SE de Breiman
- un bump (arbre à 2 feuilles) optimal
- l'arbre maximal

Données *spam* : arbre optimal à 1 SE près



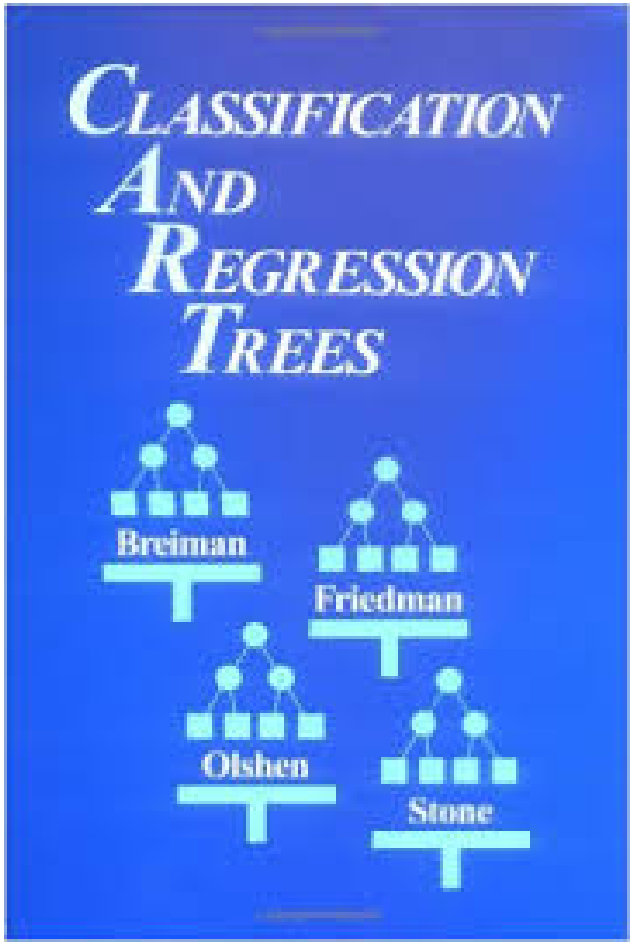
Données *spam* : arbre optimal à 1 SE près

- Le meilleur sous-arbre élagué de l'arbre maximal (à 1 SE près)
  - 17 feuilles
  - seules 14 variables (parmi les 57 initiales) figurent dans les découpes des 16 nœuds internes : `charExclamation`, `charDollar`, `remove`, `capitalAve`, `money`, `george`, `hp`, `free`, `re`, `num000`, `our`, `edu`, `internet` `meeting`
- Deux chemins interprétés :
  - de la racine à la feuille la plus à droite : un mail qui contient beaucoup de $ et de ! est presque toujours un spam
  - de la racine à la cinquième feuille la plus à droite : un mail contenant beaucoup de !, de lettres capitales et de `hp` mais peu de $ n'est presque jamais un spam

Arbre	2 feuilles	1 s.e.	maximal	optimal
Erreur empirique	0.208	0.073	0.000	0.062
Erreur test	0.209	0.096	0.096	0.086

TABLE : Erreurs (empirique et test) des 4 arbres

La référence



- CART Classification And Regression Trees, Breiman et al. (1984)
- Une introduction compacte et claire de la méthode CART en régression se trouve dans le chapitre 2 de la thèse de S. Gey (2002)
- voir Zhang, Singer (2010) et bien entendu le livre Hastie, Tibshirani, Friedman (2009)

Pour et ... contre

- Modele non paramétrique + partition des données
- Un cadre unique pour la régression et la classification binaire or multi-classes
- Modèles faciles à interpréter
- Predicteurs numériques mélangés à des catégoriels
- Découpes compétitives : développement manuel de l'arbre maximal
- Traitement élégant des valeurs manquantes en prédiction : découpes de substitution

- Principal inconvénient : manque de stabilité
- Prédicteur de base pour : bagging, boosting, random forests

Découpes compétitives, découpes de substitution

```
##
## Node number 1: 2300 observations
## predicted class=nonspam  expected loss=0.393913
##  left son=2 (1369 obs) right son=3 (931 obs)
##
##  Primary splits:
##      charExclamation < 0.0795 to the left,  improve=351.9304
##      charDollar < 0.0555 to the left,  improve=337.1138
##      free < 0.095 to the left,  improve=296.6714
##      remove  < 0.01 to the left,  improve=290.1446
##      your < 0.605  to the left,  improve=272.6889
##
##  Surrogate splits:
##      free < 0.135 to the left,  agree=0.710, adj=0.285
##      your < 0.755 to the left,  agree=0.703, adj=0.267
##      charDollar < 0.0555 to the left,  agree=0.702, adj=0.264
##      capitalLong < 53.5 to the left,  agree=0.694, adj=0.245
##      all < 0.325 to the left,  agree=0.685, adj=0.221
##
```

Extensions ou variantes

- Variantes
  - En régression, prédicteurs plus réguliers que les constants par morceaux, e.g. MARS introduit par Friedman (1991)
  - Ortho-CART Donoho et al. (1997), en traitement d'images, splits dyadiques + élagage par un algorithme classique de choix de la meilleure base de paquets d'ondelettes
  - Dyadic-CART, idées généralisées dans Blanchard et al. (2007)
- Extensions
  - L'une des extensions les plus utilisées : CART pour les données de survie, LeBlanc, Crowley (1993), Molinaro et al. (2004) et le récent article de synthèse Bou-Hamad et al. (2011)
  - Extension aux données spatiales avec des idées de type krigeage Bel et al. (2009)
  - Dans Zhang, Singer (2010) variantes pour les données longitudinales ou pour les données fonctionnelles
  - CART en chimiométrie dans Questier et al. (2005)

Plan

Forêts aléatoires

- Introduites par Breiman (2001), elles font partie de la famille des méthodes d'ensemble, Dietterich (1999,2000), on peut citer *Bagging, Boosting, Randomizing Outputs, Random Subspace*
- Algorithme d'apprentissage statistique très performant, à la fois pour des problèmes de classification et de régression. De plus en plus utilisées pour traiter de nombreuses données réelles dans des domaines d'application variés :
  - biopuces Díaz-Uriarte et Alvarez De Andres (2006)
  - l'écologie Prasad et al. (2006)
  - la prévision de la pollution Ghattas (1999)
  - la génomique Goldstein et al. (2010) et Boulesteix et al. (2012)
  - et pour une revue plus large, voir Verikas et al. (2011)
- "Couronnées" dans Fernández-Delgado et al. (2014), elles étaient absentes de Wu et al. (2008) qui mentionne CART

Forêts aléatoires

$\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ v.a. i.i.d. de même loi que (X, Y).
$X \in \mathbb{R}^p$ (variables explicatives), $Y \in \mathcal{Y}$ (variable réponse) $\mathcal{Y} = \mathbb{R}$ en régression et $\mathcal{Y} = \{1, \ldots, L\}$ en classification
But : construire un prédicteur $\widehat{h} : \mathbb{R}^p \to \mathcal{Y}$

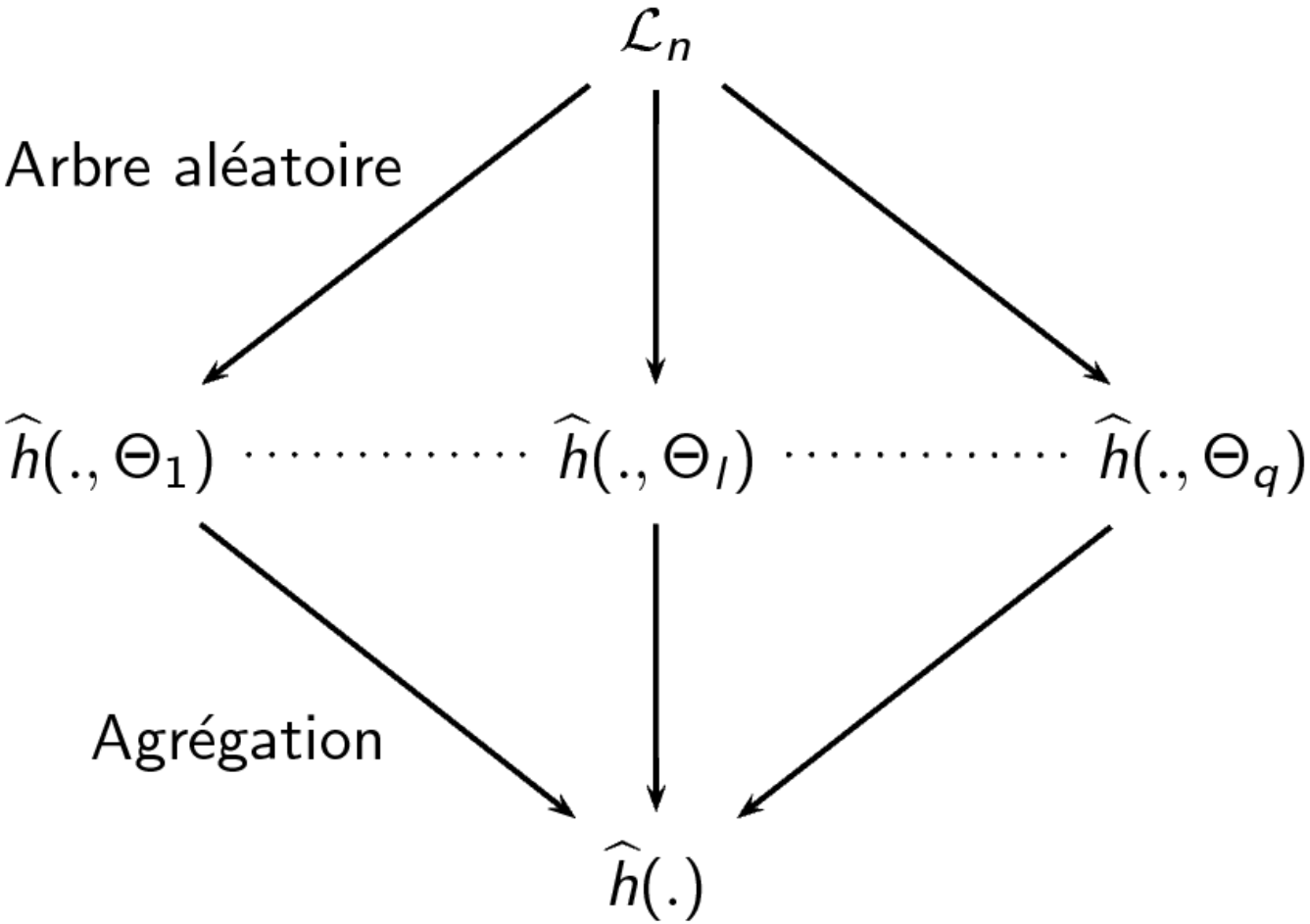
Définition : Forêts aléatoires (Breiman 2001)

$\left\{\widehat{h}(., \Theta_\ell), 1 \leq \ell \leq q\right\}$ collection de prédicteurs par arbre,
$(\Theta_\ell)_{1 \leq \ell \leq q}$ v.a. i.i.d. indépendantes de $\mathcal{L}_n$.
Prédicteur des forêts aléatoires $\widehat{h}$ obtenu en agrégeant la collection d'arbres.

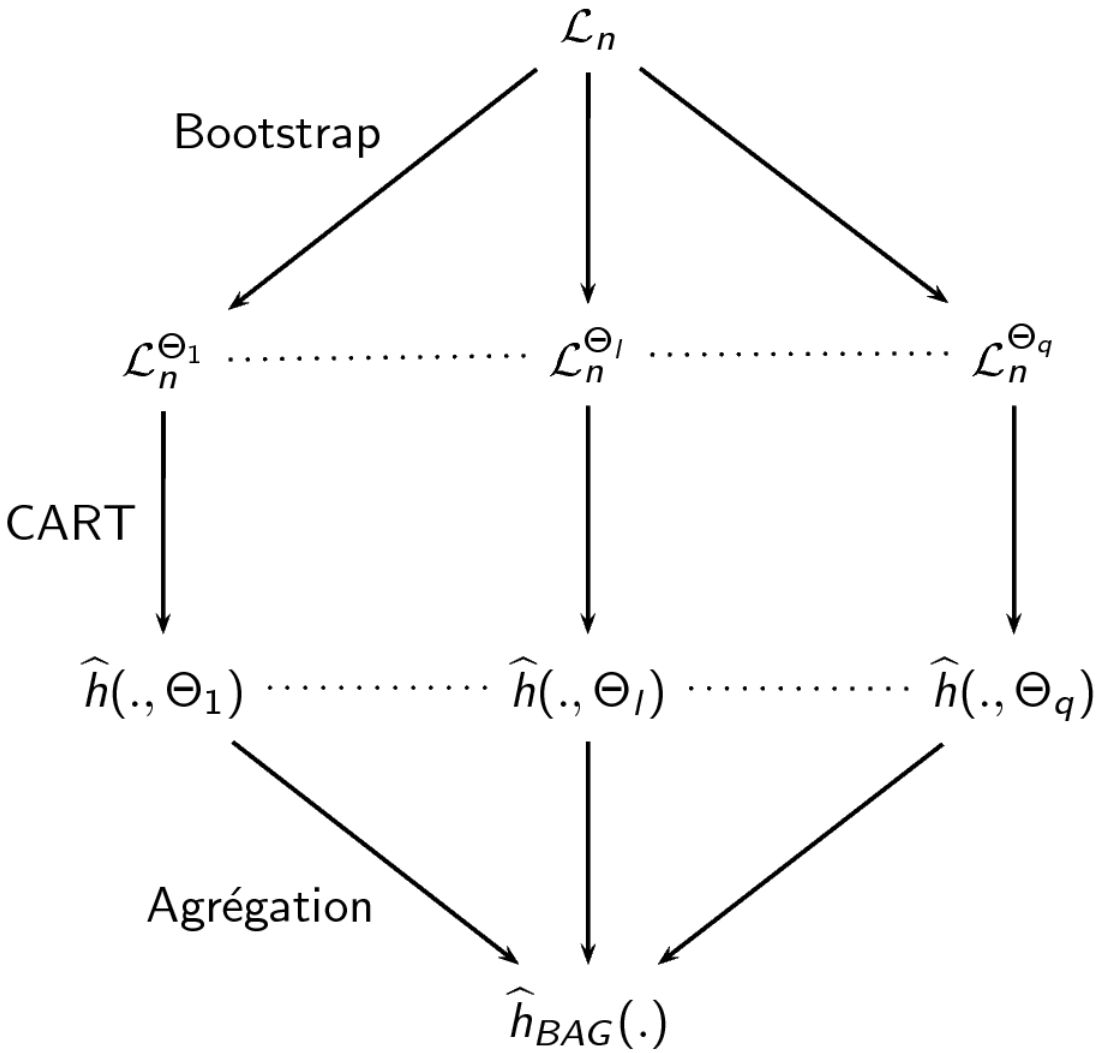
Agrégation :

- $\widehat{h}(x) = \dfrac{1}{q} \displaystyle\sum_{\ell=1}^{q} \widehat{h}(x, \Theta_\ell)$ en régression
- $\widehat{h}(x) = \underset{1 \leq c \leq L}{\operatorname{argmax}} \displaystyle\sum_{\ell=1}^{q} \mathbb{1}_{\widehat{h}(x, \Theta_\ell) = c}$ en classification

Forêts aléatoires



Bagging (Breiman 1996)



Instabilité de CART $\Rightarrow$ amélioration des performances

Random Forests-Random Inputs (Breiman 2001)

Définition : Arbre RI

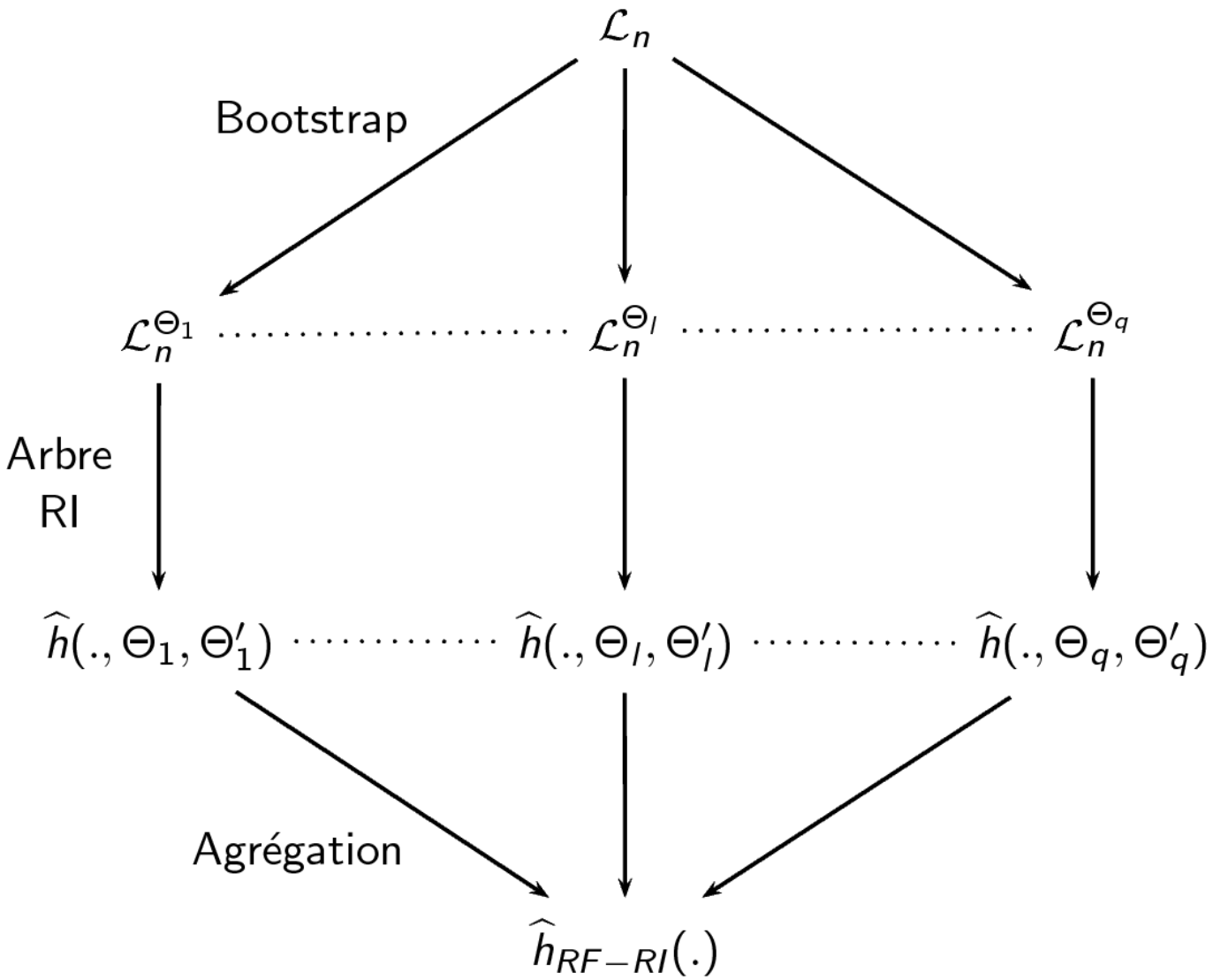
Un arbre RI consiste à tirer aléatoirement, à chaque noeud `mtry` variables, puis à chercher la meilleure coupure uniquement parmi les variables sélectionnées.

`mtry` est le même pour tous les noeuds de tous les arbres de la forêt mais, bien sûr, les variables considérées en chaque noeud pour le choix de la meilleure découpe changent aléatoirement

Définition : Random Forests-RI

Une forêt Random Forests-RI est obtenue en effectuant du Bagging avec des arbres RI.

Random Forests-RI



Aléa supplémentaire $\Rightarrow$ amélioration des performances

Random Forests-RI en pratique

Paquet R `randomForest` :

- basé sur le code de Breiman, Cutler (2000)
- décrit dans Liaw, Wiener (2002)

Principaux paramètres de l'algorithme `randomForest` :

- `ntree` : nombre d'arbres dans la forêt (par défaut 500)
- `mtry` : le nombre de variables tirées aléatoirement à chaque noeud. C'est le paramètre le plus important :
  - par défaut : $\sqrt{p}$ en classification, $p/3$ en régression
  - l'étude empirique Genuer et al. (2008) précise :
    - en régression, hors du temps de calcul, pas d'amélioration drastique par rapport au Bagging non élagué ($mtry = p$)
    - en classification standard, la valeur par défaut est bonne
    - mais pour des problèmes de classification de grande dimension, des valeurs plus grandes pour *mtry* donnent parfois des résultats bien meilleurs

Données *spam* : arbre optimal, Bagging, RF

Prédicteur	arbre optimal	bagging	forêt aléatoire
Erreur test	0.086	0.060	0.052

TABLE : Erreurs test du bagging et des forêts aléatoires, comparées à celles de l'arbre optimal pour les données `spam`

- Bagging en utilisant aussi le package `randomForest` et en construisant un prédicteur Bagging avec comme règle de base un arbre CART non-élagué (le package ne permet pas d'élaguer les arbres d'une forêt)
- Forêt aléatoire construite à l'aide du package `randomForest` avec les paramètres par défaut

Famille de forêts aléatoires

Exemples d'aléas supplémentaires :

- rééchantillonnage préalable à la construction de l'arbre,
- choix aléatoire de la variable de coupure à chaque noeud,
- choix aléatoire du point de coupure à chaque noeud.

Deux grandes familles de forêts aléatoires :

- Classiques : partition optimisée sur les données d'apprentissage $\mathcal{L}_n$
- Purement aléatoires : partition tirée aléatoirement, indépendamment de $\mathcal{L}_n$

Des variantes et ... des résultats théoriques

Définition : Forêts purement aléatoires (PRF)

Une PRF est une agrégation d'arbres purement aléatoires, si la partition associée à chacun de ces arbres est tirée aléatoirement indépendamment de $\mathcal{L}_n$

- PRF en théorie :
  - Breiman (2000, Biau et al. (2008), Zhu et al. (2015), Ishwaran, Kogalur (2010), Denil et al. (2014) : résultats de consistance
  - Genuer (2012) : résultat de réduction de variance et vitesse de convergence en dim. 1 puis Arlot, Genuer (2014) en dim. d
  - Biau (2012) : résultat de réduction de variance et de biais dans un contexte de réduction de dimension
  - Mentch, Hooker (2014), Wager (2014) : normalité asympt.
  - Scornet, Biau, Vert (2015) : consistance pour les RF de Breiman, pour les modèles additifs
- PRF en pratique :
  - Cutler, Zhao (2001), Geurts et al. (2006), Duroux et al. (2016)

Des variantes et ... des résultats théoriques

- Récent papier de revue Biau, Scornet (2016) : excellente synthèse des travaux théoriques + discussion

- Dans celle-ci, Arlot, Genuer (2016) étudient l'apport des ingrédients des RF, théoriquement pour une variante simple de RF et par simulation pour une variante proche des RF-RI

  - c'est la randomisation des partitions (qu'elle soit obtenue grâce au bootstrap, au tirage des m variables à chaque nœud ou au tirage du point de coupure) qui serait la plus cruciale

  - Voici pourquoi le Bagging (qui ne randomise pas sur la recherche de la coupure) et Extra-Trees de Geurts et al. (2006) (qui n'utilise pas de bootstrap) donnent des résultats très satisfaisants en pratique alors bien que très différentes dans le choix de l'aléa supplémentaire Θ

Extensions et variantes

- Extensions pour des objectifs variés :
  - Problèmes de classement Clemençon et al. (2013),
  - Analyse des données de survie Hothorn et al. (2006) et Ishwaran et al. (2008)
  - La régression quantile Meinshausen (2006)
  - Cluster forests, Yan et al. (2013), Afanador et al. (2016))
- Variantes :
  - LOFB-DRF vise l'amélioration de la diversité des arbres d'une RF, Fawagreh et al. (2015) utilisent Local Outlier Factor (LOF) pour identifier les arbres divers et sélectionner ceux dont les scores de LOF sont les plus élevés
  - Pondérer *a posteriori* les arbres pour améliorer la performance prédictive, Winham et al. (2013)
  - Random Forests-RC (RC pour "random combination"), coupures non parallèles aux axes, déjà dans Breiman (2001), plus récemment Blaser, Frizlewicz (2015), Menze et al. (2011)
  - Une dernière variante neuronale des RF due à Biau et al. 2016

Erreur OOB et Estimation de l'erreur de prédiction

Erreur OOB, Out Of Bag (≈ "En dehors du Bootstrap")

Pour prédire Y_i, on agrège uniquement les prédicteurs $\widehat{h}(.,\Theta_\ell)$ construits sur des échantillons bootstrap ne contenant pas (X_i, Y_i)

- Erreur OOB $= \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$ en régression
- Erreur OOB $= \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{Y_i \neq \widehat{Y}_i}$ en classification

- Estimation semblable aux estimateurs classiques de l'erreur de généralisation (par échantillon test ou par validation croisée)
- Pas de découpage de l'échantillon d'apprentissage, inclus dans la génération des échantillons bootstrap
- Mais attention : c'est bien une sous-forêt différente (en général) qui est utilisée pour calculer chaque $\widehat{Y}_i$

Importance des variables

- Au delà des performances et du caractère automatique des RF, l'un des aspects les plus importants sur le plan appliqué est la quantification de l'importance des variables
- Azen et Budescu (2003) : discussion générale sur cette notion
- Notion relativement peu examinée par les statisticiens et principalement dans le cadre des modèles linéaires, Grömping (2015) ou la récente thèse de Wallard (2015)

- Les RF offrent un cadre idéal alliant
  - une méthode non-paramétrique, ne prescrivant pas de forme particulière à la relation entre Y et les composantes de X
  - le rééchantillonnage bootstrap

  pour disposer d'une définition à la fois efficace et commode de tels indices

Importance des variables

Breiman (2001), Strobl *et al.* (2007, 2008), Ishwaran (2007), Archer *et al.* (2008), Genuer et al. (2010), Gregorutti et al. (2013, 2015), Louppe et al. (2013)

Importance des variables

Soit $j \in \{1, \ldots, p\}$. Pour chaque échantillon OOB, on permute aléatoirement les valeurs de la j-ième variable des données

Importance de la j-ième variable = augmentation moyenne de l'erreur d'un arbre après permutation

Plus l'augmentation d'erreur est forte,
plus la variable est importante

Importance des variables par permutation

$L_k, \bar{L}_k^j,\ k = 1, \ldots, ntree$ k-ème échantillons OOB et OOB permuté (obtenu par permutation aléatoire des valeurs de la j-ème variable)

$$I(X^j) = \frac{1}{ntree} \sum_{k=1}^{ntree} \left[\hat{R}(\hat{f}_k, \bar{L}_k^j) - \hat{R}(\hat{f}_k, \bar{L}_k) \right]$$

X_1	...	X_j	...	X_p	Y
$x_{1,1}$		$x_{\pi_j(1),j}$		$x_{1,p}$	y_1
$\vdots$		$\vdots$		$\vdots$	$\vdots$
$x_{i,1}$		$x_{\pi_j(i),j}$		$x_{i,p}$	y_i
$\vdots$		$\vdots$		$\vdots$	$\vdots$
$x_{n,1}$		$x_{\pi_j(n),j}$		$x_{n,p}$	y_n

"Toys data", Weston *et al.* (2003)

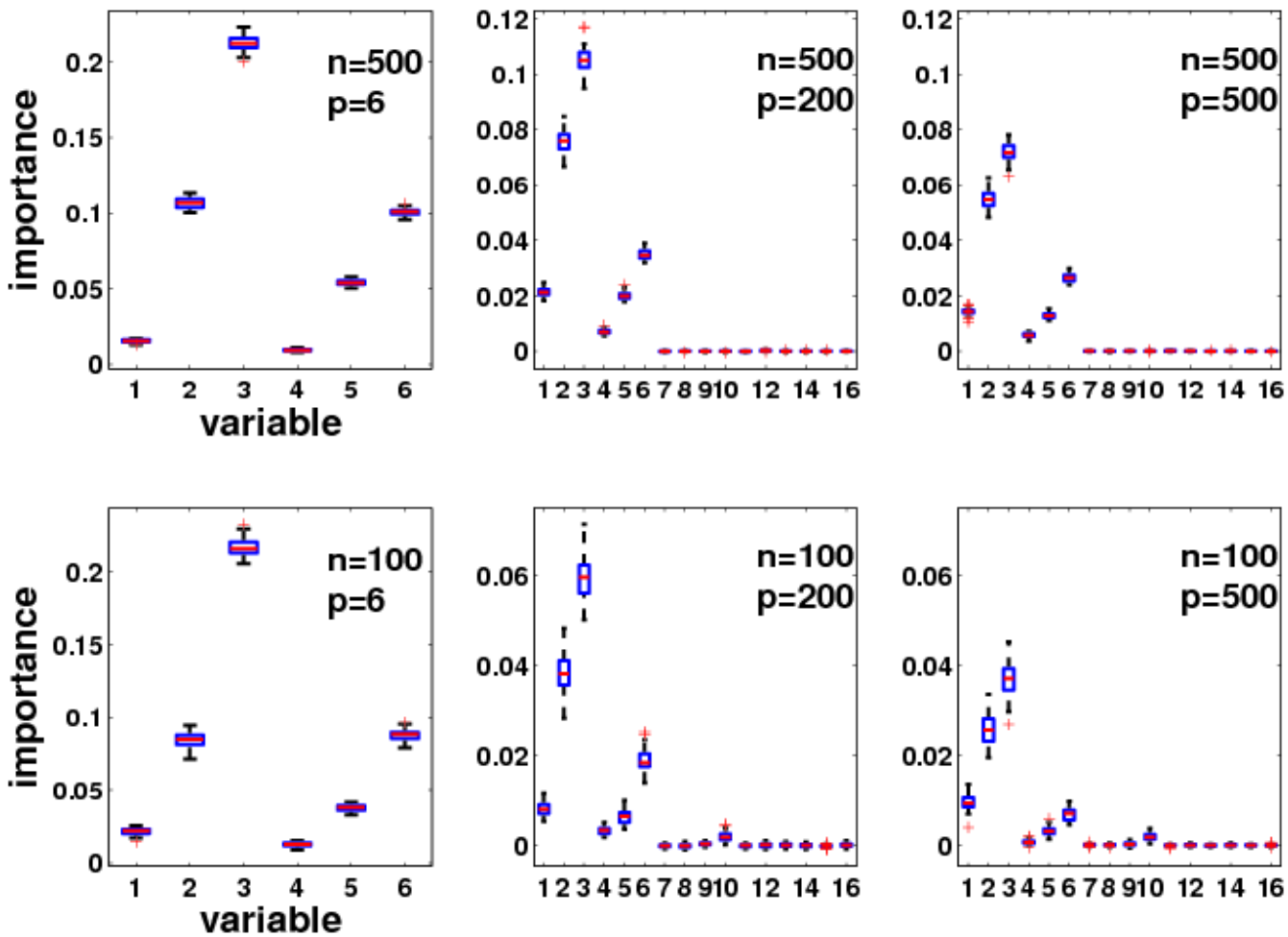
Problème de classification binaire, $Y \in \{-1, 1\}$, avec 6 vraies variables, les autres étant non informatives :

- 2 groupes presque indépendants de 3 variables influentes (fortement, moyennement et faiblement corrélées avec la réponse Y)
- un groupe additionnel de variables qui sont non informatives : des bruits indépendants de Y

Modèle défini par les distributions des X^i conditionnellement à $Y = y$:

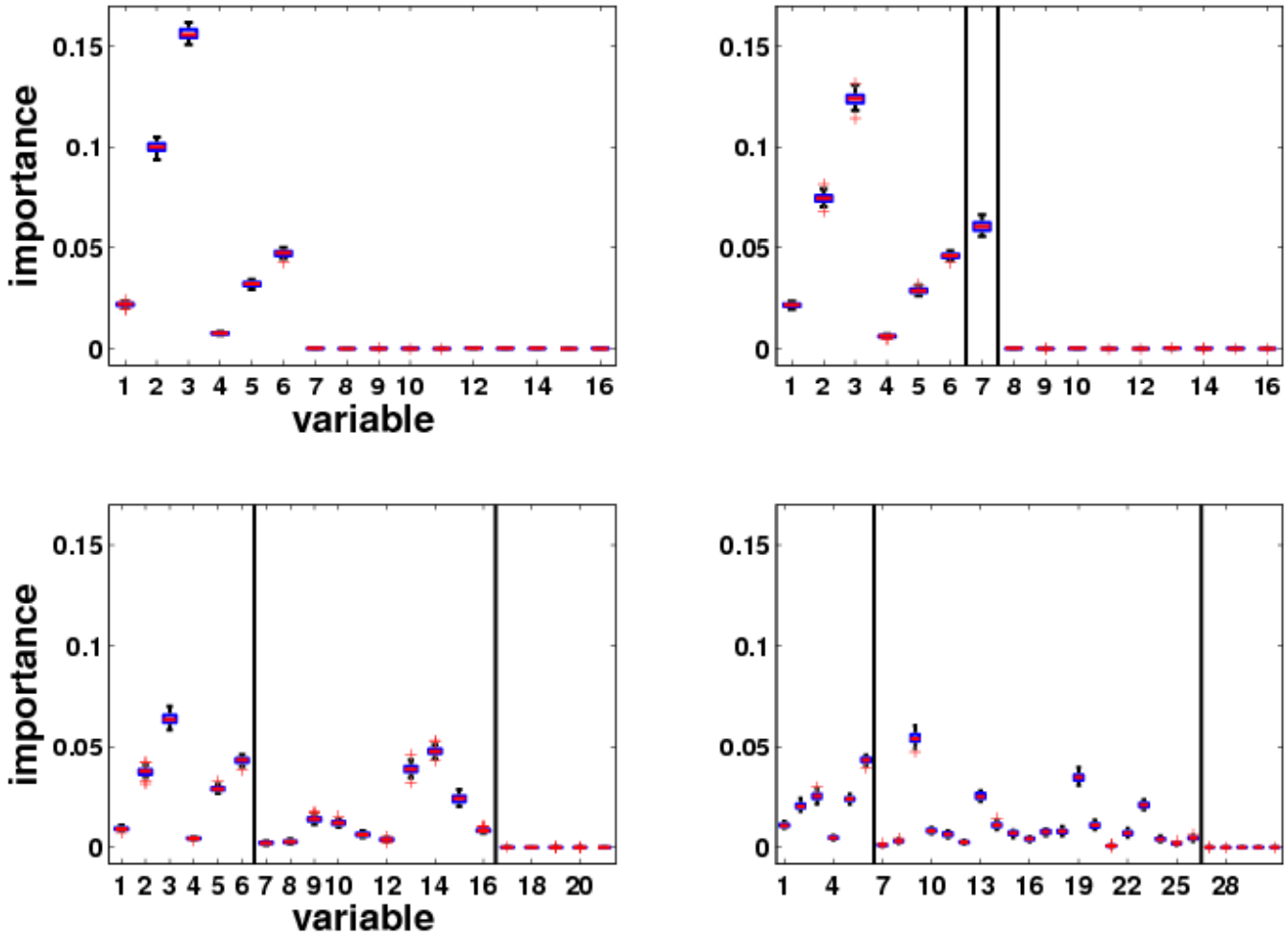
- pour 70% des données, $X^i \sim y\mathcal{N}(i, 1)$ pour $i = 1, 2, 3$ et $X^i \sim y\mathcal{N}(0, 1)$ pour $i = 4, 5, 6$
- pour les 30% restants, $X^i \sim y\mathcal{N}(0, 1)$ pour $i = 1, 2, 3$ et $X^i \sim y\mathcal{N}(i - 3, 1)$ pour $i = 4, 5, 6$
- les autres variables sont du bruit, $X^i \sim \mathcal{N}(0, 1)$ pour $i = 7, \ldots, p$

Sensibilité de VI à n et p



Variabilité de VI plus grande pour les vraies variables que pour les variables inutiles

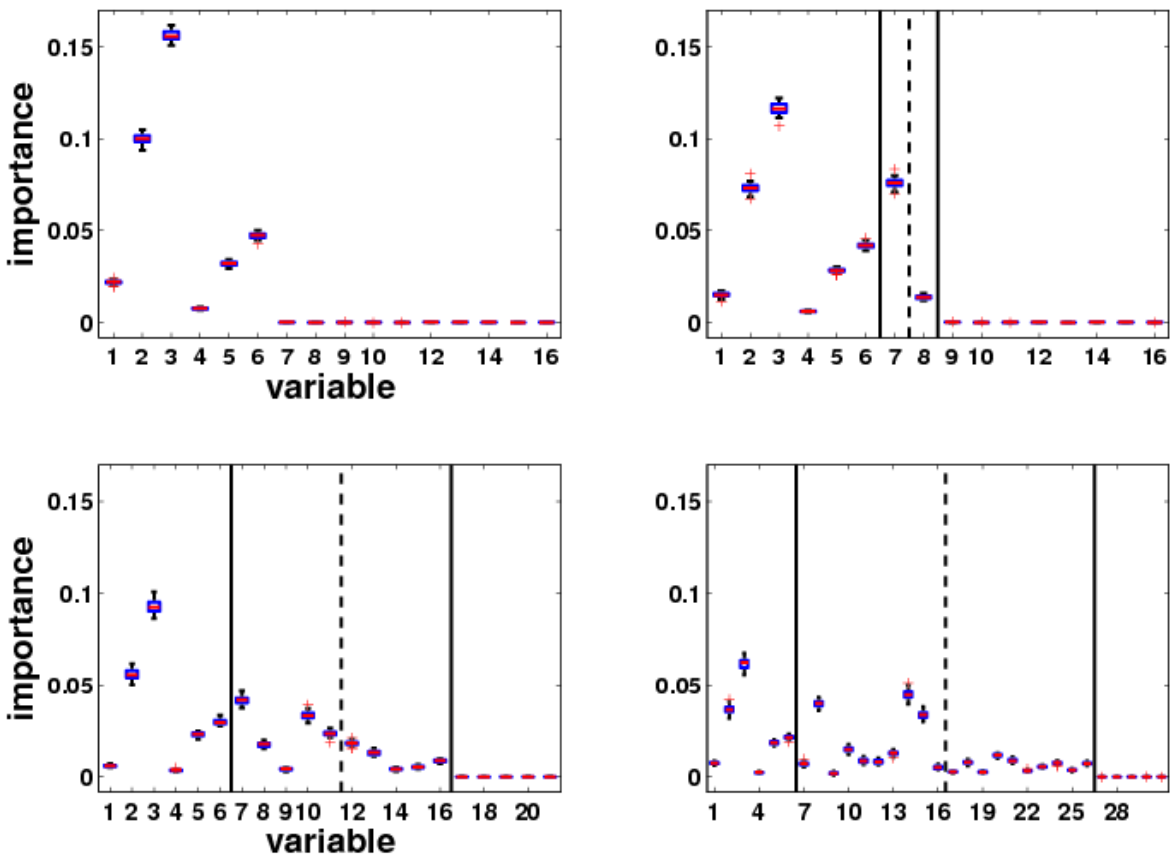
VI d'un groupe de variables corrélées



$\{1, 2, 3\}$ diminuent avec le nombre de réplications de 3, $\{4, 5, 6\}$ inchangées

VI n'est pas divisé par le nombre de réplications

VI de deux groupes de variables corrélées



Les deux groupes décroissent lorsque l'on ajoute plus de réplications de 3 et 6

Importance relative entre les deux groupes préservée

Données *spam* : importance des variables

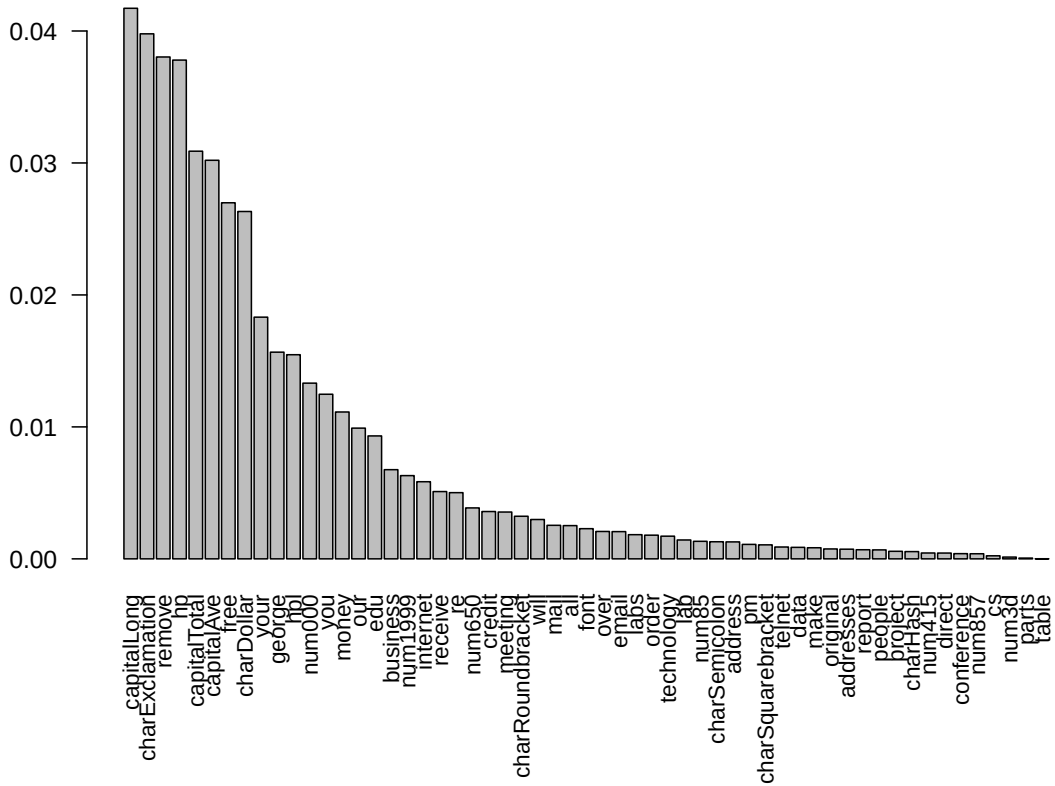


FIGURE : Les 8 plus importantes : les proportions d'occurrences des mots ou caractères *remove*, *hp*, *$*, *!*, *free* ainsi que les 3 variables liées aux longueurs des suites de lettres majuscules

Données *spam* : importance des variables

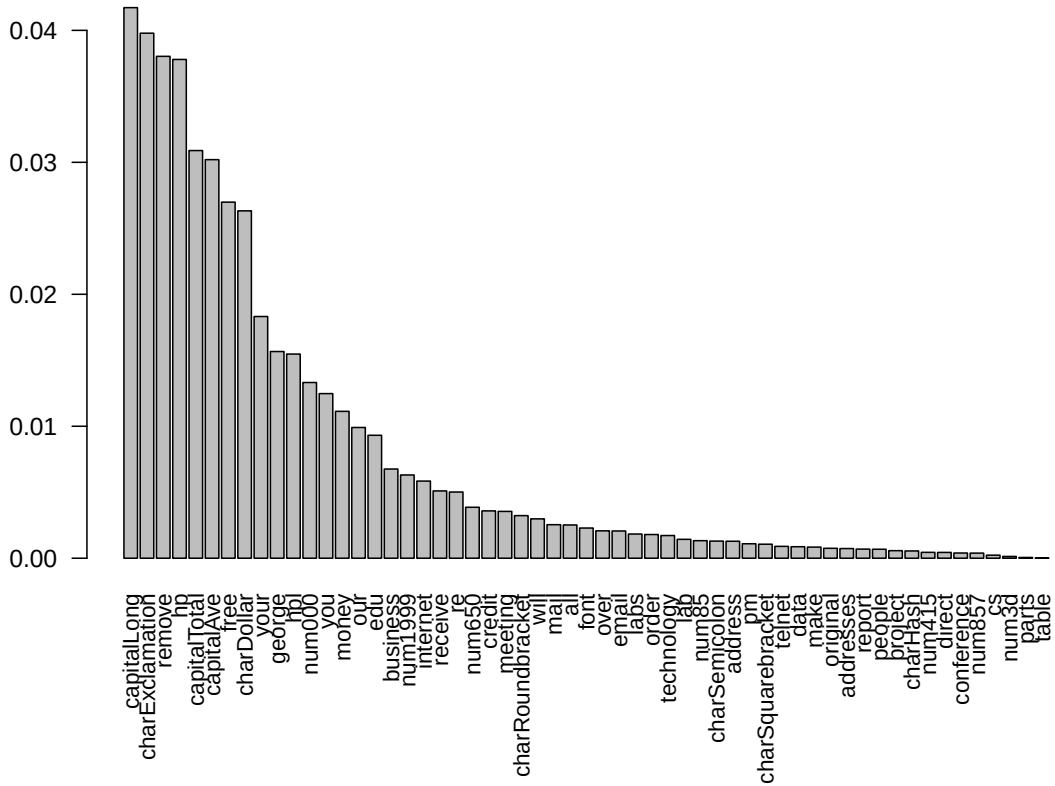


FIGURE : Les variables des 1ères découpes de l'arbre CART optimal ne sont pas en tête et la plus importante : *capitalLong* n'y figure pas

Procédure(s) de sélection

Genuer, Poggi, Tuleau (2010), PRL et (2015), R Journal

Deux objectifs différents de sélection de variables :

1. sélectionner toutes les variables importantes, même si elles sont redondantes, dans un but d'interprétation
2. trouver un ensemble parcimonieux de variables importantes suffisant pour la prédiction

Notre but est de proposer une procédure automatique qui vise ces deux objectifs

Citons simplement deux travaux antérieurs qui ont inspiré notre proposition :

- Díaz-Uriarte, Alvarez de Andrés (2006)
- Ben Ishak, Ghattas (2008)

Sélection(s) sur *toys*

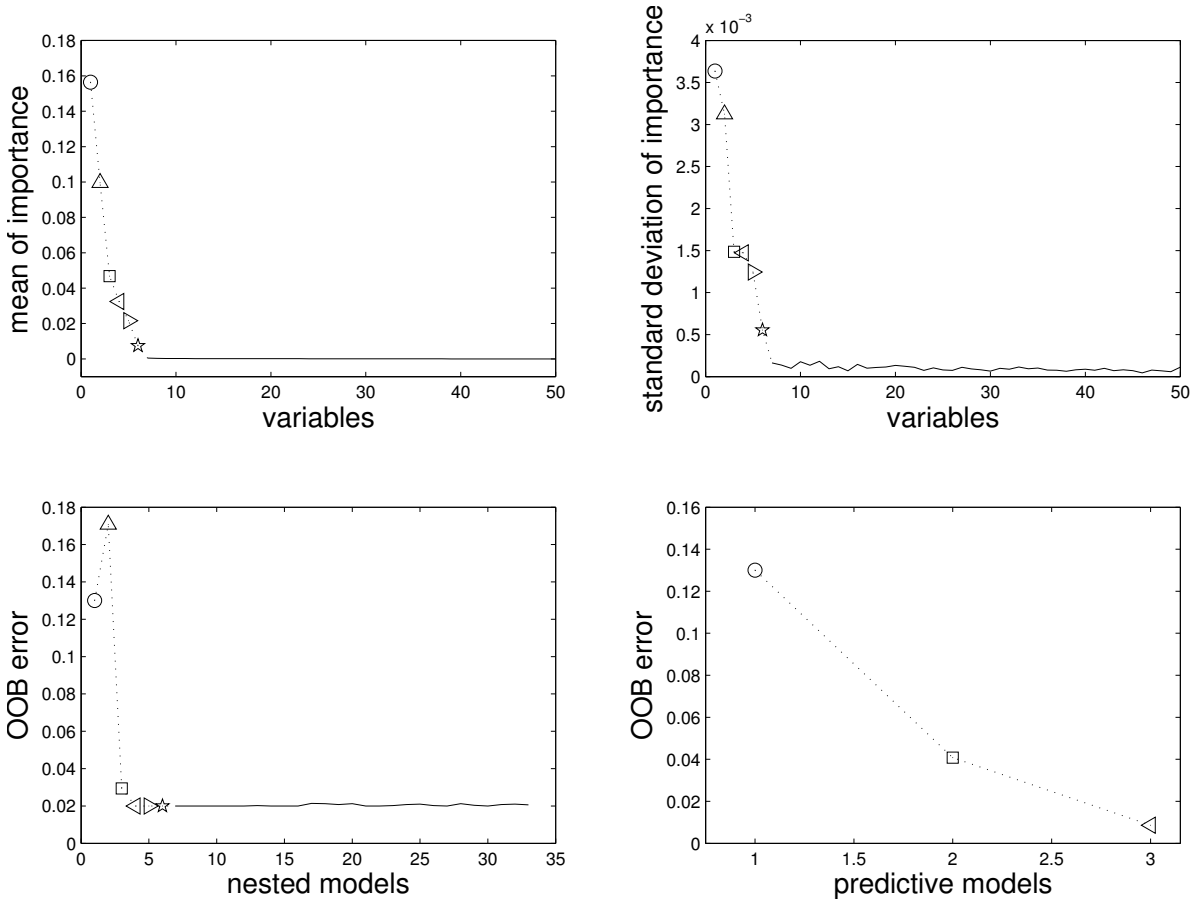


FIGURE : toys data $n = 100$, $p = 200$

- Vraies variables (1 à 6) représentées par $(\triangleright, \triangle, \circ, \star, \triangleleft, \square)$
- VI basées sur 50 forêts avec *ntree* $= 2000$, et *mtry* $= 100$

Sélection de variables : Classement

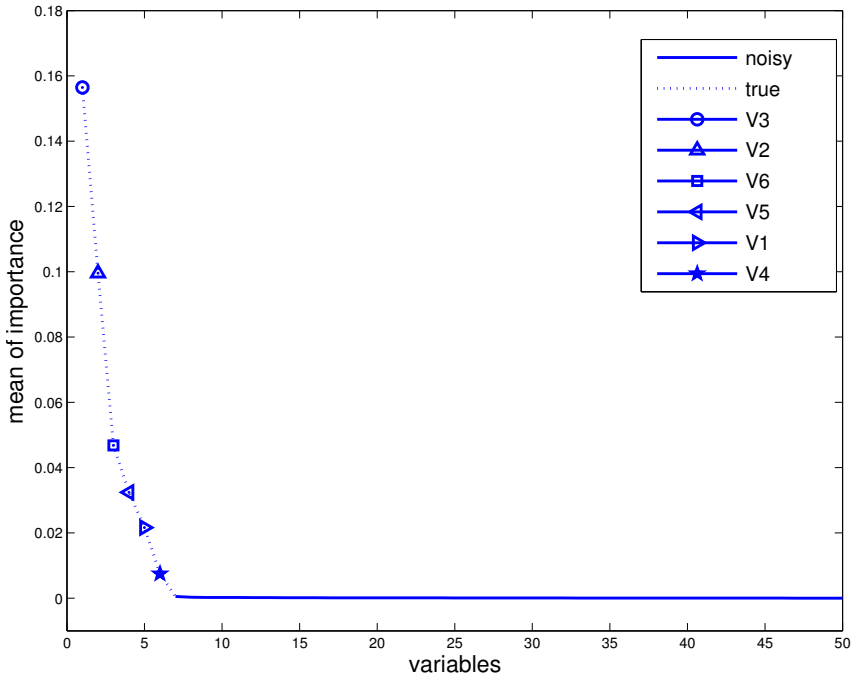


FIGURE : *Classement par VI décroissantes*

- Graphe des 50 variables les plus importantes (les autres ayant une importance quasi nulle)
- Vraies variables significativement plus importantes que les autres

Sélection de variables : Elimination

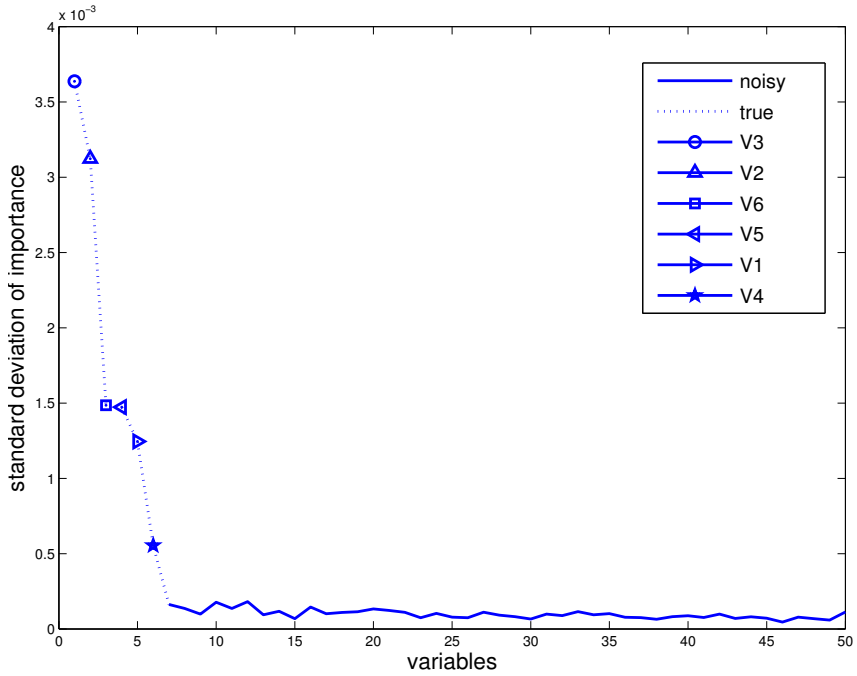


FIGURE : *On considère les écart-types de VI pour estimer un seuil et conserver les variables d'importance dépassant le seuil*

- Seuil = argmin de la valeur prédite par un modèle CART ajustant cette courbe (conservatif en général)
- Les vraies variables ont des VI plus dispersées que les autres
- Le seuil retenu conduit à conserver 33 variables

Sélection de variables pour l'interprétation

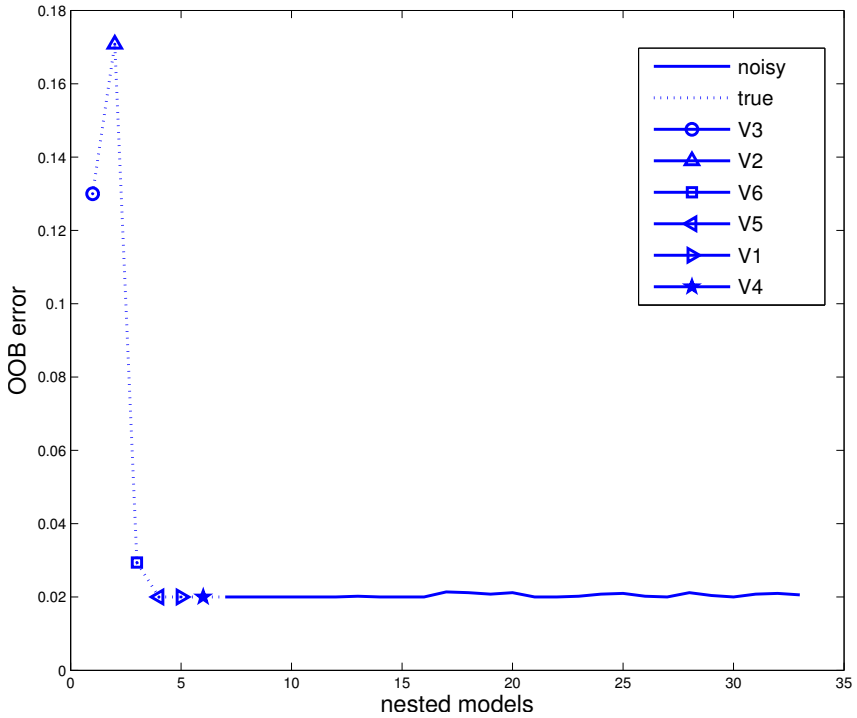


FIGURE : *On considère les erreurs OOB des modèles RF emboîtés et on sélectionne les variables du modèle associé à l'erreur la plus faible*

- L'erreur décroît rapidement et atteint son minimum lorsque les 4 premières variables sont dans le modèle, puis il est *quasiment* constant
- Le modèle contenant 4 des 6 vraies variables est sélectionné. En fait, le minimum et atteint pour 24 variables mais l'on utilise une règle semblable à la 1 SE rule de Breiman *et al.* (1984)

Sélection de variables pour la prédiction

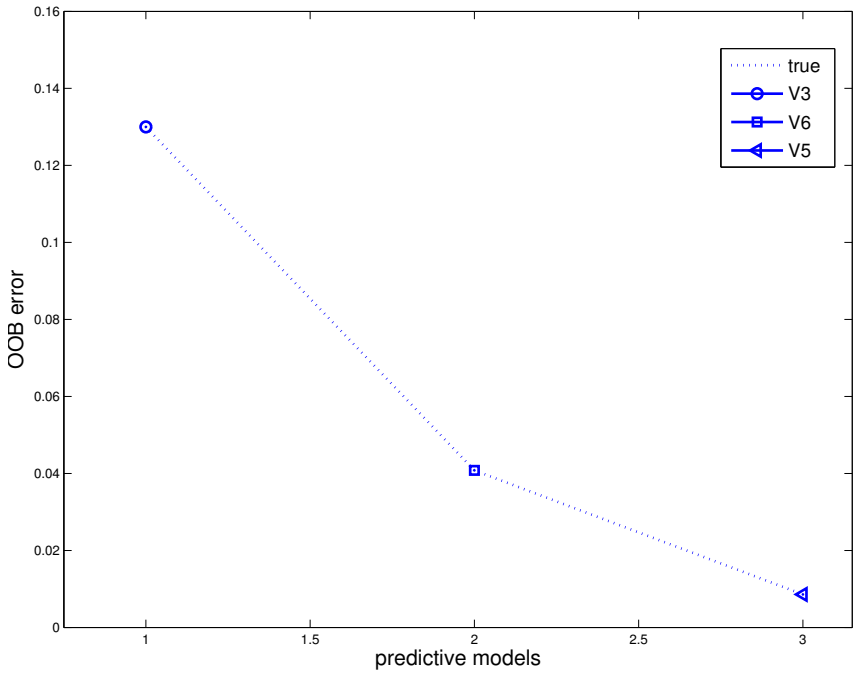
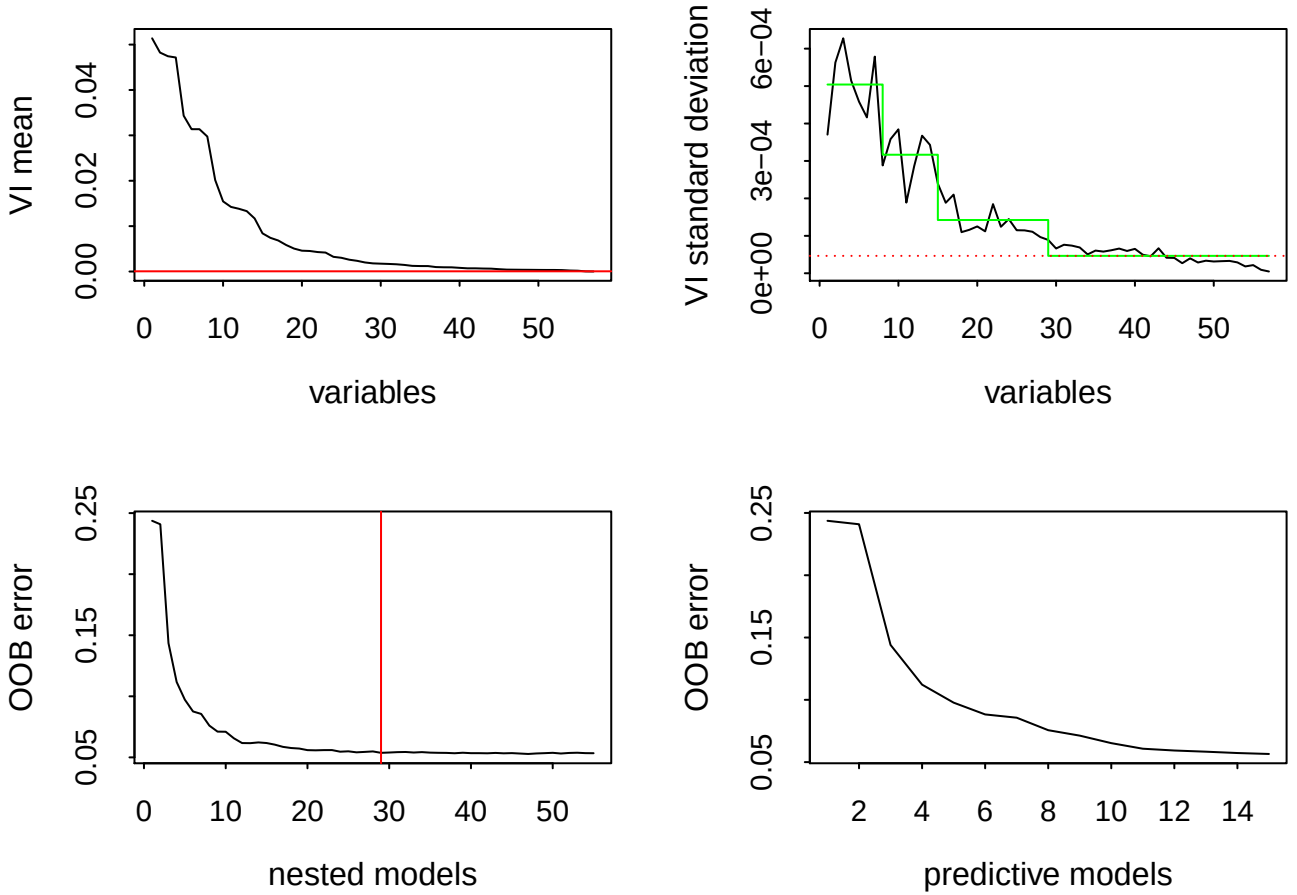


FIGURE : *Introduction séquentielle des variables avec test*

- Une variable est introduite seulement si la réduction de l'erreur est plus grande qu'un seuil : la réduction de l'erreur doit être significativement plus grande que la variabilité moyenne obtenue en ajoutant des variables de bruit
- Le modèle final pour la prédiction : seulement les variables 3, 6 et 5

VSURF appliqué aux données spam



Forêt	initiale	interprétation	prédiction
Erreur test	0.052	0.056	0.060

Une application

Genuer, Michel, Eger, Thirion (2010)

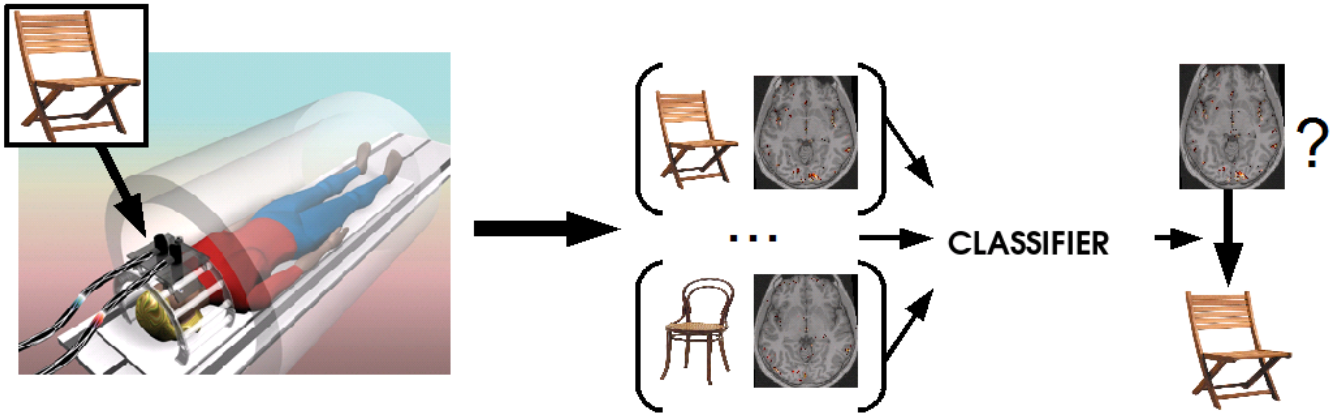


FIGURE : Expérience, IRMf

12 sujets : 4 types de chaises (4 classes), 100 000 voxels, 72 observations.
Etape préliminaire : réduction à 1000 parcelles (et donc 1000 variables) par un algorithme de Ward.

Classification $n = 72$ $p = 1000$ $L = 4$

Procédure de sélection pour un sujet

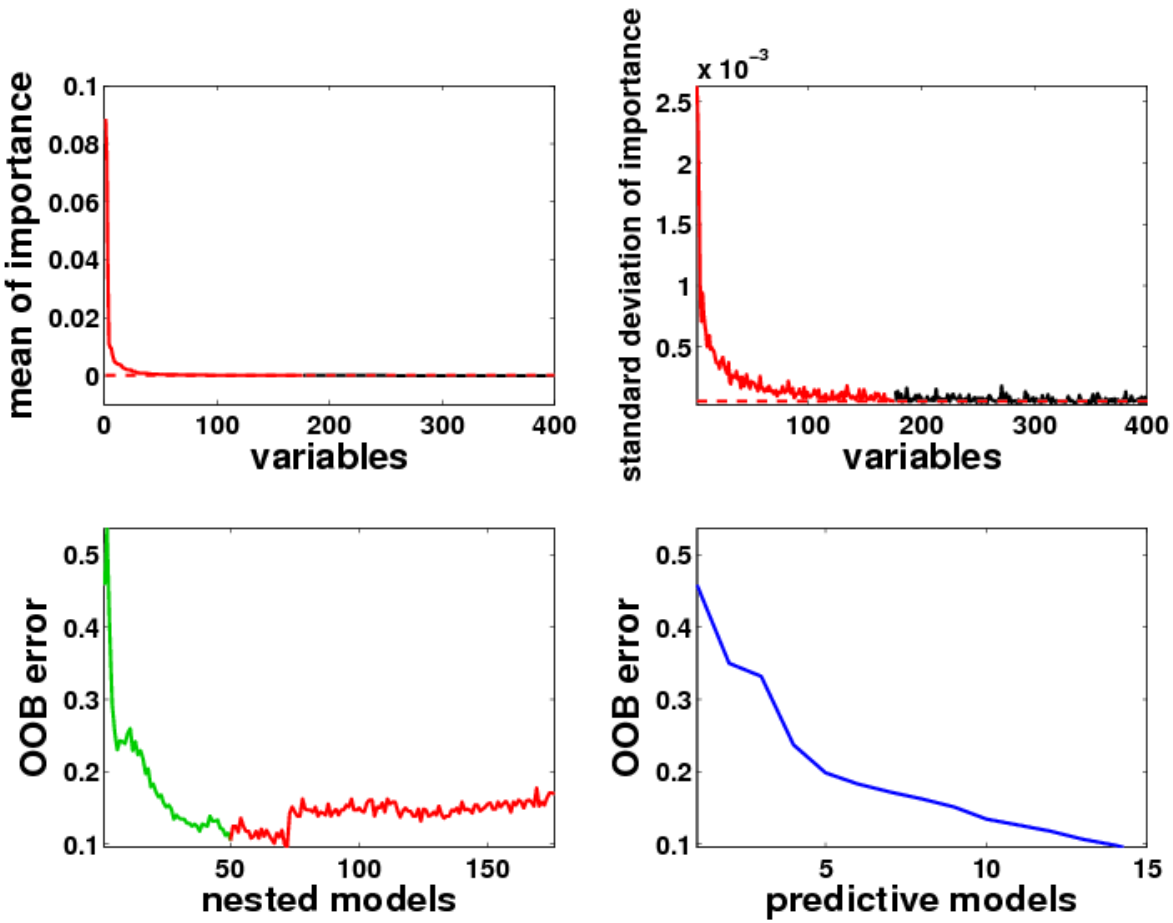


FIGURE : Procédure de sélection de variables pour un sujet ($ntree = 2000$, $mtry = p/3$)
Elimination : 176 variables, Interprétation : 50, Prédiction : 15

Les résultats : vue globale

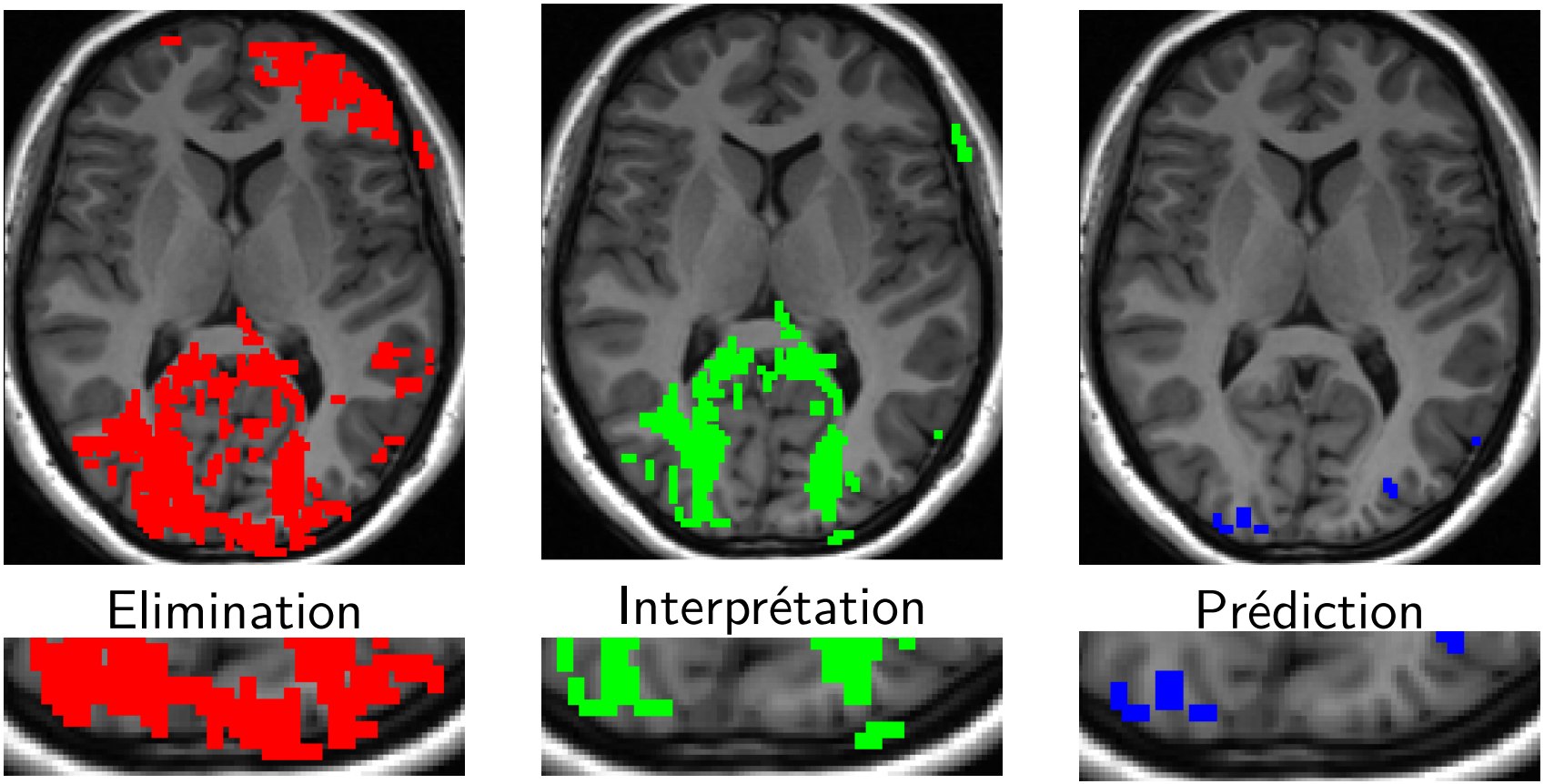
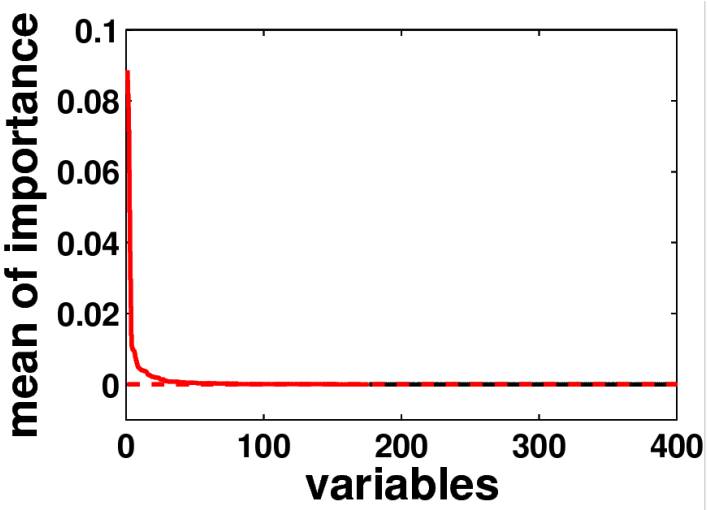
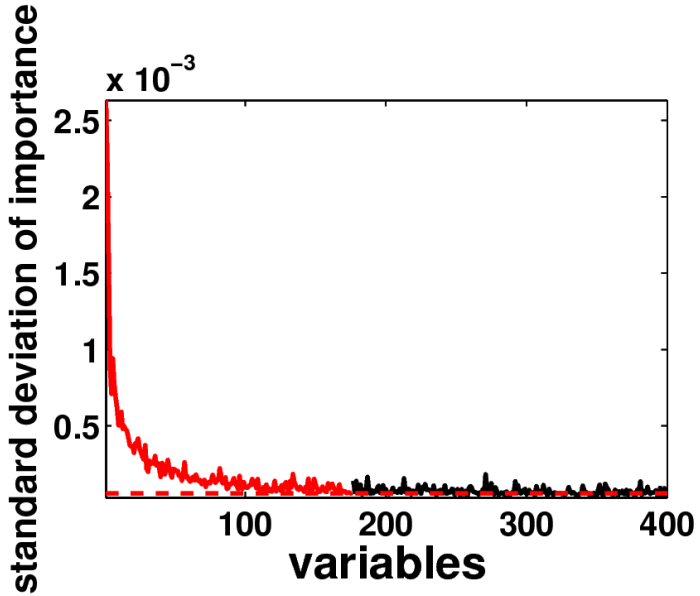
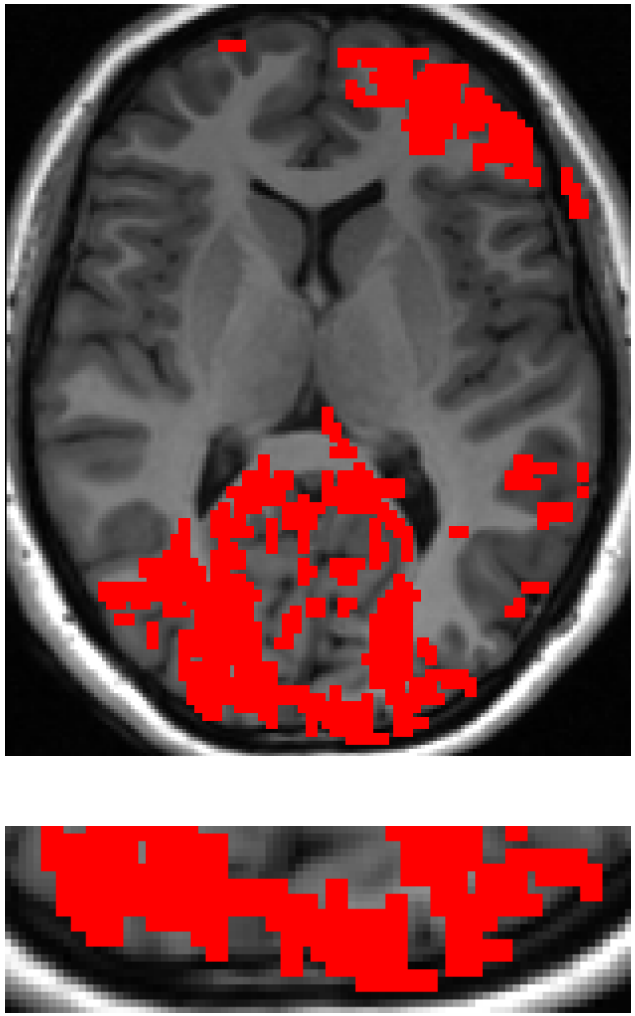


FIGURE : Variables sélectionnées aux différentes étapes de la procédure

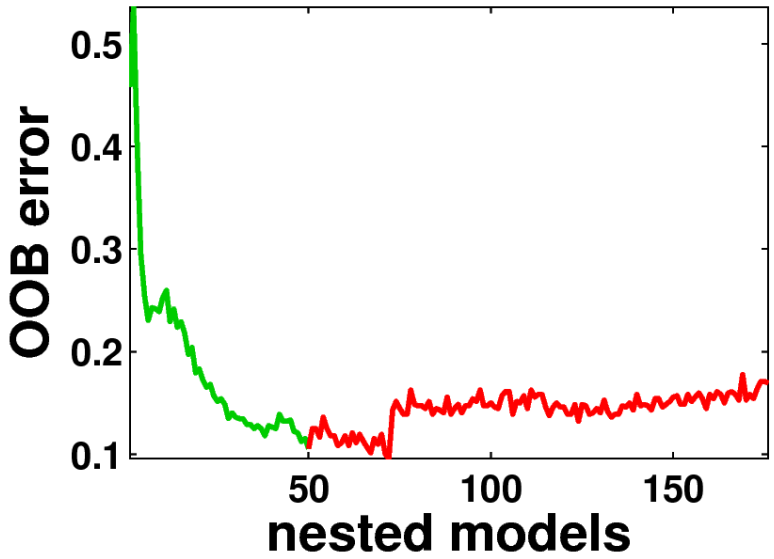
Etape d'élimination $\rightarrow$ 176 variables

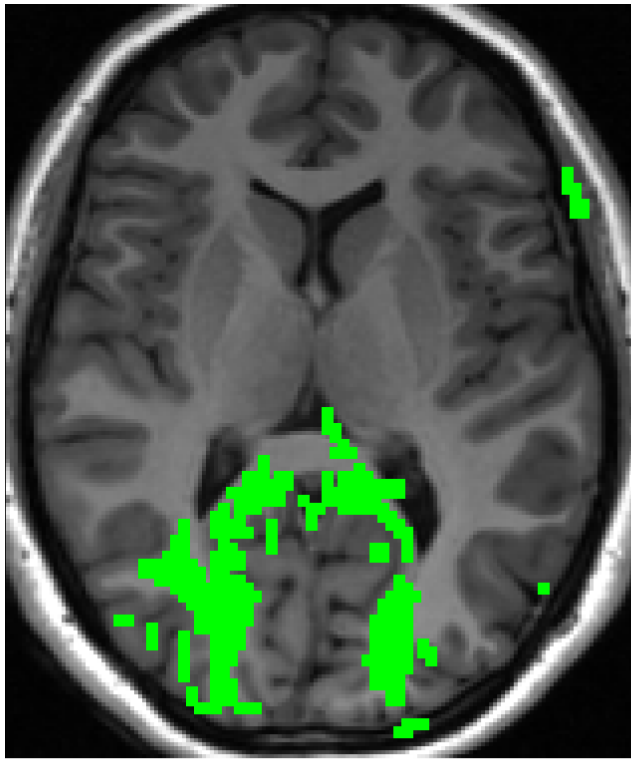


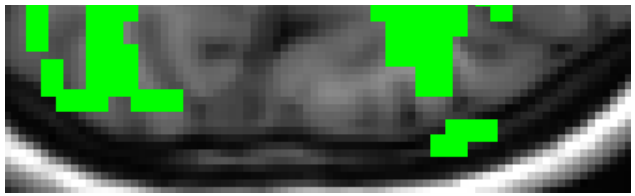




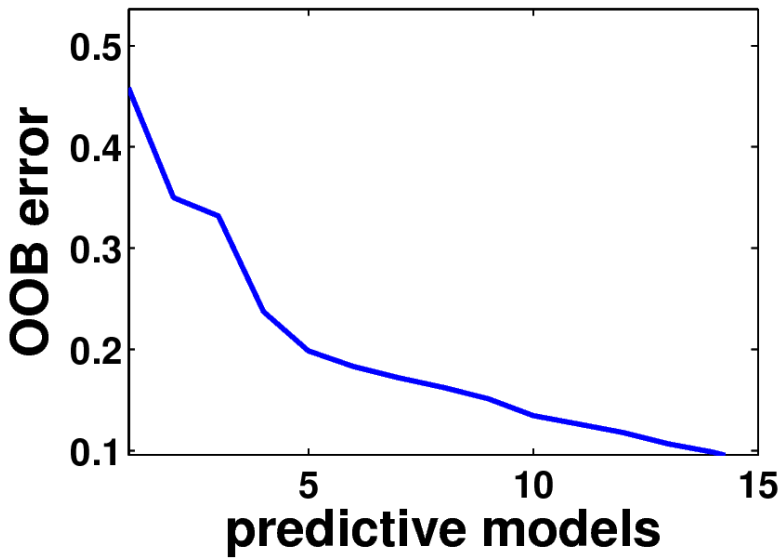
Etape d'interprétation $\rightarrow$ 50 variables

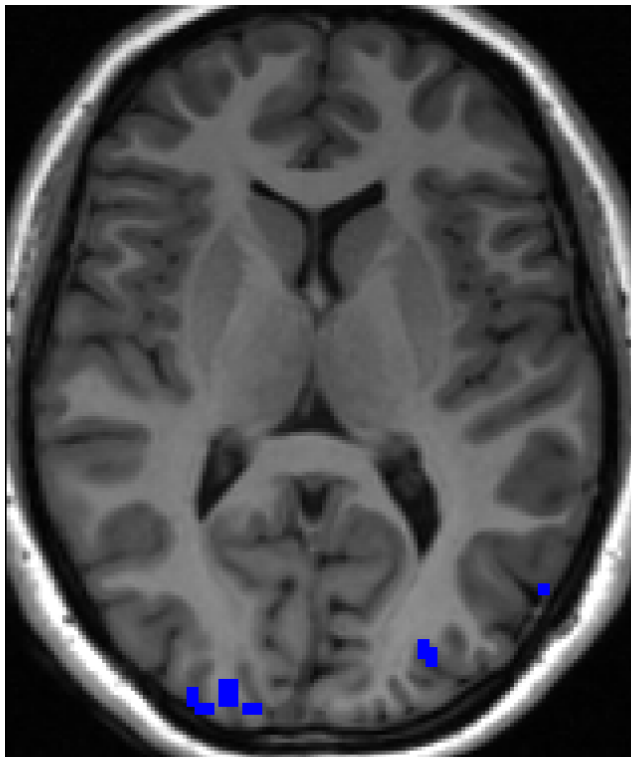






Etape de prédiction $\rightarrow$ 15 variables





Le résultat final : une vue 3D

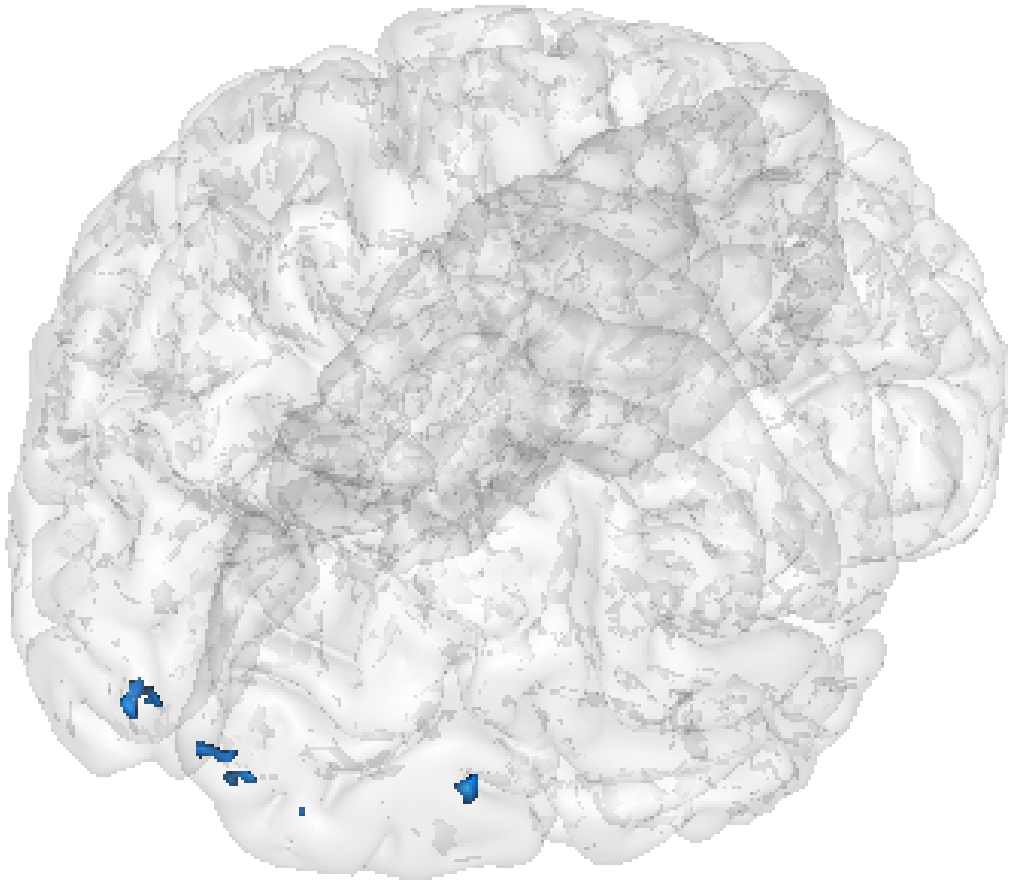


FIGURE : Regions sélectionnées pour au moins 3 sujets parmi 12 par la dernière étape de la procédure

Une comparaison

	Initiale	Elim.	Interp.	Préd.	Référence
Erreur	34 %	29 %	27 %	30 %	31 %
Nombre var.	1000	146	23	8	350

FIGURE : Résultats sur les 12 sujets de l'étude

- Méthode de référence : SVM linéaire (F-test + validation croisée)
- Taux d'erreurs comparables
- Beaucoup moins de variables

Extension : Importance de groupes de variables

Gregorutti et al. (2015)

$\mathbf{X^J}$: groupe de variables et $\bar{\mathbf{L}}_k^J$ l'échantillon OOB permuté de $\bar{\mathbf{L}}_k$ résultant de la permutation aléatoire du groupe $\mathbf{X^J}$

L'importance par permutation du groupe $\mathbf{X^J}$:

$$I(\mathbf{X^J}) = \frac{1}{ntree} \sum_{k=1}^{ntree} \left[\hat{R}(\hat{f}_k, \bar{\mathbf{L}}_k^J) - \hat{R}(\hat{f}_k, \bar{\mathbf{L}}_k) \right]$$

X_1	...	X_{j_1}	...	X_{j_2}	...	X_p	Y
$x_{1,1}$		$x_{\pi_J(1),j_1}$	...	$x_{\pi_J(1),j_{k(J)}}$	...	$x_{1,p}$	y_1
$\vdots$		$\vdots$		$\vdots$		$\vdots$	$\vdots$
$x_{i,1}$		$x_{\pi_J(i),j_1}$	...	$x_{\pi_J(i),j_{k(J)}}$	...	$x_{i,p}$	y_i
$\vdots$		$\vdots$		$\vdots$		$\vdots$	$\vdots$
$x_{n,1}$		$x_{\pi_J(n),j_1}$	...	$x_{\pi_J(n),j_{k(J)}}$	...	$x_{n,p}$	y_n

et l'extension à la sélection de variables fonctionnelles en découle

Contexte et stratégies pour analyser des Big Data

- Genuer et al. (2016) RF for BD
- Big Data (BD) :
  - Données massives, hétérogènes, changeant fréquemment et arrivant en flux
  - Un défi majeur pour les statisticiens : le passage à l'échelle des méthodes statistiques pour l'analyse des BD. Cf. Jordan (2013)
- Stratégies pour analyser des Big Data, Wang et al. (2015)
  - Sous-échantillonnage : choisir un sous-ensemble de données traitable, lui appliquer une analyse classique, et répéter ceci plusieurs fois (e.g. Bag of Little Bootstrap, Kleiner et al. 2012)
  - Diviser pour régner : découper les données en sous-ensembles de données traitables, appliquer une analyse classique à chacun d'eux et combiner les résultats obtenus (e.g. MapReduce)
  - Mise à jour séquentielle pour les flux de données : conduire les analyses en ligne, en mettant à jour des quantités ou des objets convenablement choisis au fur et à mesure de l'arrivée des données (e.g. Schifano et al. 2014)

Variante 1 : Sous-échantillonnage BLB et RF

- Tirer un échantillon aléatoire de taille m (non trivial dans le contexte Big Data) et construire une RF sur cet ensemble. Problème : le m parmi n bootstrap

- Bag of Little Bootstrap (BLB) : échantillons bootstrap de taille n, chacun contenant seulement $m \ll n$ données différentes
- La taille de l'échantillon reste classique (n), évitant ainsi le problème du biais impliqué par le bootstrap m parmi n, conséquence directe de la stratégie de MR (chaque fragment contient une partie de l'ensemble de données et entraîne ainsi un échantillon bootstrap de taille m)
- Le traitement de cet échantillon est simplifié par une pondération intelligente et est donc gérable même pour n très grand puisqu'il ne contient qu'une petite fraction (m/n) d'observations différentes de l'ensemble de données d'origine

Variante 2 : MapReduce RF

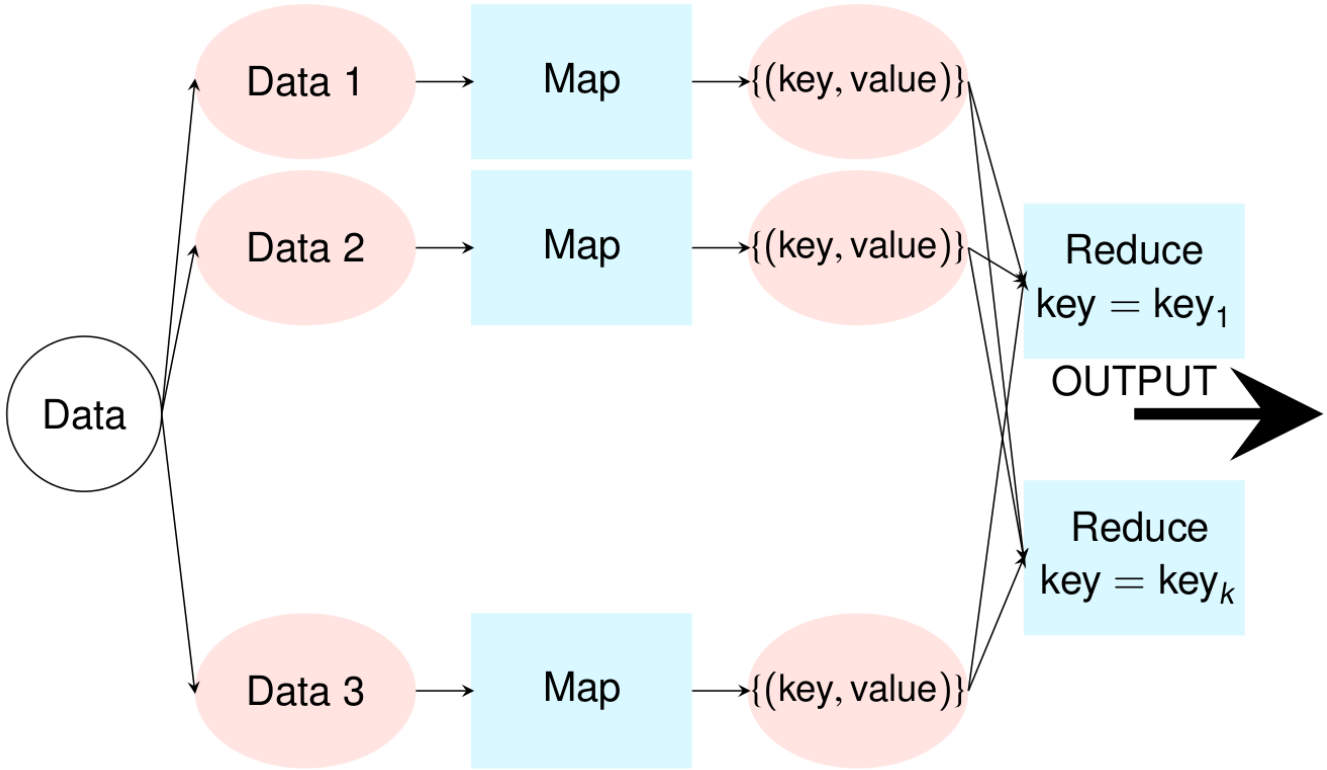


FIGURE : Construction parallèle de sous-forêts d'arbres obtenues sur des sous-échantillons bootstrap des sous-ensembles de données propres à chaque tâche Map. Chaque bloc est envoyé à un travail Map dans lequel une RF (qui peut avoir un nombre modéré d'arbres) est construite
Les sous-forêts obtenues, fusionnées, donnent la forêt finale

Variante 3 : RF en ligne (en classification)

- Conjugue le Bagging en ligne Oza (2005) et les ERT de Geurts et al. (2006)
- Traite les flux de données (données arrivant séquentiellement) en ligne (c-à-d. sans mémoire) : Saffari et al. (2009)

- Ceci permet aussi de traiter des données massives (répondant simultanément aux caractéristiques Volume et Vélocité), mais aussi aux données massives (statiques), en les considérant séquentiellement
- Adaptation en profondeur des RF de Breiman y compris dans la construction de chacun des arbres de la RF.
  Idée principale : penser l'algorithme uniquement en termes des proportions des classes de Y, plutôt qu'en termes des observations

- Un résultat de consistance dans Denil et al. (2013)