Forêts aléatoires : aspects théoriques, sélection de variables et applications

Robin Genuer

N° d'ordre : 10016

THÈSE

Présentée pour obtenir

LE GRADE DE DOCTEUR EN SCIENCES
DE L'UNIVERSITÉ PARIS-SUD XI

Spécialité : Mathématiques

par

Robin GENUER

Forêts aléatoires : aspects théoriques, sélection de variables et applications

Soutenue le 24 Novembre 2010 devant la Commission d'examen :

M.	Philippe BESSE	(Rapporteur)
M.	Gérard BIAU	(Rapporteur)
M.	Gilles CELEUX	
M.	Pascal MASSART	(Président du jury)
M.	Jean-Michel POGGI	(Directeur de thèse)
M.	Bertrand THIRION	

Thèse préparée au
Département de Mathématiques d'Orsay
Laboratoire de Mathématiques (UMR 8628), Bât. 425
Université Paris-Sud 11
91 405 Orsay CEDEX

Résumé

Cette thèse s'inscrit dans le cadre de l'apprentissage statistique et est consacrée à l'étude de la méthode des forêts aléatoires, introduite par Breiman en 2001. Les forêts aléatoires sont une méthode statistique non paramétrique, qui s'avère être très performante dans de nombreuses applications, aussi bien pour des problèmes de régression que de classification supervisée. Elles présentent également un bon comportement sur des données de très grande dimension, pour lesquelles le nombre de variables dépasse largement le nombre d'observations. Dans une première partie, nous développons une procédure de sélection de variables, basée sur l'indice d'importance des variables calculée par les forêts aléatoires. Cet indice d'importance permet de distinguer les variables pertinentes des variables inutiles. La procédure consiste alors à sélectionner automatiquement un sous-ensemble de variables dans un but d'interprétation ou de prédiction. La deuxième partie illustre la capacité de cette procédure de sélection de variables à être performante pour des problèmes très différents. La première application est un problème de classification en très grande dimension sur des données de neuroimagerie, alors que la seconde traite des données génomiques qui constituent un problème de régression en plus petite dimension. Une dernière partie, théorique, établit des bornes de risque pour une version simplifiée des forêts aléatoires. Dans un contexte de régression, avec une seule variable explicative, nous montrons d'une part que les estimateurs associés à un arbre et à une forêt atteignent tous deux la vitesse minimax de convergence, et d'autre part que la forêt apporte une amélioration en réduisant la variance de l'estimateur d'un facteur de trois quarts.

Mots-clefs : apprentissage statistique, forêts aléatoires, sélection de variables, régression non paramétrique, classification supervisée, statistique en grande dimension.

Random Forests: elements of theory, variable selection and applications

Abstract

This thesis deals with statistical learning and is dedicated to the random forests method, which has been proposed by Breiman in 2001. Random forests are a non-parametric statistical method, which is very powerful in many applications, for regression problems as well as for supervised classification ones. They also succeed to handle very high dimensional data, where the number of variables largely exceeds the number of observations. In a first part, we develop a variable selection procedure, based on the variable importance index computed by random forests. This importance index allows to highlight relevant variables from useless ones. The proposed procedure consists to automatically select a variables set for interpretation or prediction purpose. The second part shows the ability of the variable selection procedure to deal with very different problems. The first application is a classification problem in very high dimension for neuroimaging data, while the second one covers genomic data which constitute a regression problem in smaller dimension. A last theoretical part, establishes some risk bounds for a simplified version of random forests. In the context of regression problems with a one-dimensional predictor space, we prove that both tree and forest estimators achieved the minimax rate of convergence. In addition we prove that forests improve accuracy by reducing the estimator variance by a factor of three fourths.

Keywords : statistical learning, random forests, variable selection, non-parametric regression, supervised classification, high dimension statistics.

Remerciements

Mes trois années de thèse à Orsay ont été très enrichissantes et très agréables. Ceci est dû à de nombreuses personnes que j'ai côtoyées une ou plusieurs années. Je voudrais donc les remercier ici.

Tout d'abord, je voudrais dire un énorme merci à Jean-Michel, dont je qualifierais de parfait l'encadrement de thèse. Bien sûr mon avis est très subjectif, et par ailleurs je ne pense pas qu'il y ait de règles strictes en matière de direction de thèse. Quoi qu'il en soit, j'ai vraiment beaucoup apprécié réaliser ma thèse avec toi. Tu m'as fait découvrir le monde de la recherche, que je ne connaissais pas du tout, ainsi que celui de l'enseignement à l'Université. Tu m'as toujours soutenu, conseillé, fait avancer dans mes travaux et tu as toujours mis en valeur mon travail. Tu m'as aussi aidé à m'insérer dans le laboratoire et plus généralement dans la communauté statistique. Finalement, merci pour tous ces moments enrichissants, scientifiquement et humainement.

Je voudrais remercier Gérard et Philippe d'avoir accepté de rapporter ma thèse. Je remercie également Gilles et Bertrand, qui me font le plaisir de faire partie du jury. Une mention spéciale pour Pascal qui, en plus d'avoir été le premier à me donner le goût des statistiques en maîtrise, et de m'avoir guidé pendant l'année de M2, m'a été d'une grande aide pour les résultats du chapitre 4 de cette thèse : merci pour tout Pascal.

En espérant n'oublier personne, je remercie chaleureusement les personnes suivantes membres (ou anciens membres) du laboratoire de mathématiques d'Orsay pour leur sympathie, leur bonne humeur voire pour leurs cours intéressants (dans le désordre et sans distinguer de statut particulier) : Pierre, Cathy, Vincent R, Nicolas V, Jean-Patrick, Benoît L, Marie-Anne, Antoine, Laurent T, Cécile, Cyril, Camille, Yves M, Jean-Christophe, Bertrand, Aurélien, Claire, Ramla, Olivier, Liliane, Maud, Jérémie, Christine K, Laure, Erwan, Thanh Mai, Sylvain, Nathalie, Sébastien, Abed, Emmanuel, Oana, Guillaume, Dominque B, Bruno, Caroline, Thi Thu, Patrick, Nadine, Jean-Michel M, Besma, Dominique H, Alain, Shweta, Marianne, Aude, Juliette, Thierry, Benoît S, Bernardo, Merlin, Yves A, Jairo, Lionel, Wilson, Laurent M, Vincent B, Vladimir, Wendelin, Agnès, Raphaël, Etienne, Elisabeth, Sorin, Charles, Frédéric.

Encore une mention spéciale, cette fois-ci pour Valérie, qui a la tâche difficile d'essayer d'apporter une certaine légèreté à la trop lourde charge administrative qui accompagne la thèse, mais qui le fait très bien : c'est-à-dire avec une gentillesse, une disponibilité et une efficacité exceptionnelles. Merci beaucoup Valérie.

Un grand merci à Christine, ma grande soeur de thèse et collaboratrice. En parlant de collaboration, je salue également Vincent M, mon compagnon de route neurospinien.

Je remercie les membres du MAP5 pour leur accueil lors de mon année d'intérim à Paris Descartes : Servane, Adeline, Antoine, Fabienne, Valentine, Sandra, Bérénice, Flora, Avner, Christophe, Nicolas M, Gaëlle, Nicolas M, Jérôme, Olivier, Mélina, Vincent.

Je salue également les gens que j'ai pu croiser au détour d'un congrès, d'un séminaire ou d'une conférence : Joseph, Guillem, Fanny, Gilles S, Arnaud, Aude, Mahendra, Stéphane, Yannick, Patricia, Xavier, Christophe, Aurélien.

Merci aux courageux pour leurs précieuses remarques sur l'exposé : Servane, Adeline, Pierre, Maud et Vincent.

Orsay signifie aussi pour moi quatre années d'études motivantes. Je remercie mes camarades étudiants de l'époque : Adeline, Alain, Othilie, Manu, Chip et Vianney.

Je remercie ma famille et mes amis pour leur soutien, et particulièrement mes parents qui m'ont toujours laissé très libre dans mes choix et m'ont toujours encouragé dans ceux-là.

Enfin, je remercie ma femme Marjolaine, pour son amour, son soutien, sa capacité à me supporter quand je lui parle de maths, et pour tous ces évènements riches en émotions que nous avons connus pendant ces trois ans : le dernier en date étant la venue au monde de notre fils. Elouen, pour finir cette page, je te remercie pour ton oeil coquin et pour tes sourires matinaux qui ont transformé nos réveils...

Table des matières

Chapitre 1

Présentation générale

Sommaire

Introduction

Cette présentation générale est composée de quatres parties. Nous introduisons tout d'abord le cadre statistique de la thèse ainsi que la problématique de sélection de variables pour des données de grande dimension. Nous présentons également une famille de méthodes statistiques à laquelle appartiennent les forêts aléatoires : les méthodes d'ensemble.

Dans une deuxième partie, nous décrivons en détail l'algorithme des forêts aléatoires de Leo Breiman, ainsi que le calcul d'importance des variables donné par la méthode.
Nous présentons ensuite les différentes variantes de forêts aléatoires que l'on rencontre dans la littérature. Nous discutons également des motivations de l'introduction de ces différentes variantes.
Dans la dernière partie, nous regroupons tous les résultats théoriques sur les forêts aléatoires dont nous avons connaissance. Pour cela, nous présentons les différentes approches qui existent pour analyser de façon théorique cette méthode.

1.1 Contexte

Les forêts aléatoires sont une méthode statistique non-paramétrique aux performances exceptionnelles. Elles ont été introduites par Breiman (2001) et sont de plus en plus utilisées pour traiter de nombreux et divers jeux de données réelles. Les domaines d'application sont nombreux, citons par exemple l'étude des biopuces (c.f. Díaz-Uriarte and Alvarez de Andrés (2006)), l'écologie (c.f. Prasad et al. (2006)) ou encore la génomique (c.f. Goldstein et al. (2010)).

Dans cette section, nous introduisons le contexte général dans lequel s'inscrivent les forêts aléatoires, les problèmes pour lesquels elles peuvent être utilisées, ainsi que la famille de méthodes dont elles font partie.

1.1.1 Apprentissage statistique

Le cadre mathématique de l'apprentissage statistique est le suivant. Soit $\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ un échantillon d'apprentissage, c'est-à-dire une suite de vecteurs aléatoires indépendants et identiquement distribués (i.i.d.), de même loi qu'un vecteur aléatoire (X, Y). Le vecteur (X, Y) est indépendant de $\mathcal{L}_n$ et sa loi est inconnue. L'entier naturel n désigne le nombre d'observations de l'échantillon d'apprentissage.

Nous recommandons vivement le livre de Hastie et al. (2009) qui est une excellente référence pour une introduction à l'apprentissage statistique.

Le but d'une méthode statistique est d'"apprendre" la loi inconnue de (X, Y), au travers de l'échantillon d'apprentissage $\mathcal{L}_n$ dont on dispose. Notons $\mathcal{X}$ et $\mathcal{Y}$ les espaces mesurables dans lesquels vivent respectivement les variables aléatoires X et Y. X est vue comme la variable d'entrée et Y comme celle de sortie. En particulier, le but de la méthode est d'apprendre le lien entrée-sortie, c'est-à-dire le lien qui existe entre X et Y. Étant donné une nouvelle entrée $x \in \mathcal{X}$ (une valeur x fixée qui n'est pas présente dans l'échantillon d'apprentissage), la méthode statistique doit être capable de "prédire" la sortie $\hat{y} \in \mathcal{Y}$ correspondante. La prédiction $\hat{y}$ doit être la plus proche possible de la vraie

sortie y, associée à x. Nous parlons alors d'un problème de prédiction.

Il existe une autre façon de voir le problème, c'est le point de vue de l'estimation. Il s'agit alors d'estimer la fonction (inconnue) qui à X associe Y. Bien entendu, ce problème est relié au problème de prédiction précédent. En effet, si nous disposons d'une "bonne" estimation du lien entre X et Y, nous pourrons a fortiori "bien" prédire une nouvelle entrée x. Néanmoins, comme nous le verrons dans le paragraphe ci-dessous concernant la classification, il est parfois possible de bien prédire alors que l'estimation de la fonction qui relie X et Y n'est pas très bonne.

Il existe deux principaux cadres en apprentissage statistique : la régression et la classification. Ces deux cadres diffèrent par la nature de la sortie Y.

Régression

Le cadre de la régression est celui où la réponse Y est continue, typiquement lorsque $\mathcal{Y} = \mathbb{R}$. Le modèle statistique s'écrit alors sous la forme suivante :

$$Y = s(X) + \varepsilon \tag{1.1}$$

La fonction $s : \mathcal{X} \to \mathbb{R}$ est la fonction inconnue que nous cherchons à estimer. Elle est appelée fonction de régression. La variable ε est une variable aléatoire réelle. Elle est appelée variable de bruit : les mesures Y_i dont nous disposons dans l'échantillon $\mathcal{L}_n$ sont des observations de $s(X_i)$ bruitées par des variables aléatoires ε_i.

Pour des raisons d'identifiabilité, nous supposons que la variable de bruit est centrée conditionnellement à X : $\mathbb{E}[\varepsilon|X] = 0$. En effet, il existe alors une unique fonction s qui satisfait $s(X) = \mathbb{E}[Y|X]$.

Ce modèle statistique est appelé modèle de régression non-paramétrique. Ce nom vient du fait, qu'a priori, nous n'imposons aucune contrainte sur la fonction de régression s, contrairement aux modèles paramétriques, comme par exemple le modèle de régression linéaire. Dans le modèle linéaire, on cherche une fonction de régression s sous la forme d'une combinaison linéaire des coordonnées de X. Ce sont alors les coefficients de cette combinaison linéaire qui sont appelés les paramètres du modèle et qu'il faut estimer.

Dans le cadre du modèle (1.1), nous introduisons deux mesures de qualité : l'une pour le problème de prédiction, l'autre pour le problème d'estimation.

– Étant donné un prédicteur $\hat{h}$, c'est-à-dire une fonction de $\mathcal{X}$ dans $\mathbb{R}$, construite sur l'échantillon d'apprentissage $\mathcal{L}_n$. Le but de $\hat{h}$ est de prédire la sortie y associée à une entrée x. Nous mesurons la qualité de $\hat{h}$ par son erreur de généralisation, définie par :

$$\mathbb{E}[(\hat{h}(X) - Y)^2] \, .$$

Il existe d'autres mesures de qualité d'un prédicteur, obtenues notamment en modifiant la fonction dans l'espérance ci-dessus. Néanmoins dans cette thèse, nous

nous limitons à cette erreur, qui est souvent appelée erreur de généralisation des moindres carrés (en référence à la fonction carrée utilisée dans l'espérance).

– Pour le problème d'estimation, nous disposons d'un estimateur $\hat{s}$ de la fonction de régression s, c'est-à-dire une fonction de $\mathcal{X}$ dans $\mathbb{R}$, construite sur l'échantillon d'apprentissage $\mathcal{L}_n$. Le but de $\hat{s}$ est d'estimer au mieux la fonction s. Nous mesurons la qualité de $\hat{s}$ par son risque, défini par :

$$\mathbb{E}[(\hat{s}(X) - s(X))^2] \, .$$

De même, il existe d'autres risques dans la littérature. Nous nous limitons au risque ci-dessus, souvent appelé risque quadratique.

Ces deux mesures de qualité dépendent donc du point de vue (prédiction ou estimation). De plus, comme nous avons supposé que $\mathbb{E}[\varepsilon|X] = 0$, ces deux mesures satisfont la relation suivante : pour un prédicteur $\hat{h}$,

$$\mathbb{E}[(\hat{h}(X) - Y)^2] = \mathbb{E}[(\hat{h}(X) - s(X))^2] + \mathbb{E}[\varepsilon^2] \, .$$

Ainsi, en régression, la différence entre prédiction et estimation est essentiellement une différence de point de vue et de vocabulaire. Comme nous allons maintenant le voir, ce n'est pas le cas en classification.

Classification

En classification (appelée plus précisément classification supervisée), la réponse Y est discrète et désigne la classe à laquelle appartient l'entrée X associée. Ici, $\mathcal{Y} = \{1, \ldots, L\}$, où L désigne le nombre de classes. Nous codons l'ensemble des classes de façon ordonnée pour faciliter les notations, mais l'ensemble des classes peut être non-ordonné (les classes {bleu,rouge,vert} sont par exemple codées {1,2,3}).

En régression, le but est d'estimer la fonction de régression, qui n'est autre que l'espérance conditionnelle de Y sachant X. En classification, nous ne pouvons pas écrire le modèle sous une forme équivalente au modèle (1.1). Mais le but est maintenant d'estimer les probabilités a posteriori définies, pour un $x \in \mathcal{X}$ fixé, par :

$$\forall c \in \{1, \ldots, L\} \quad \mathrm{P}(Y = c|X = x)$$

c'est-à-dire les probabilités pour Y d'appartenir à chacune des classes, conditionnellement à X.

Le fait que nous traitions un échantillon d'observations bruitées se traduit par le fait que pour un x fixé, il n'y a pas forcément une probabilité a posteriori égale à 1 et les autres égales à 0. Donc, pour certaines observations, la classe correspondante à x devrait être c, mais se retrouve altérée en c' dans l'échantillon. En régression, le bruit vient du fait que nous n'observons pas exactement $s(X)$, mais $s(X)+\varepsilon$. En classification, le bruit provient du fait que certaines classes sont altérées.

En classification, nous avons également deux mesures de qualité : l'une pour la prédiction, l'autre pour l'estimation.

- Nous mesurons la qualité d'un prédicteur $\hat{h}$ par son erreur de généralisation, définie par :

$$\mathrm{P}(\hat{h}(X) \neq Y) \ .$$

- Le prédicteur qui minimise l'erreur de généralisation est appelé prédicteur de Bayes. Ce prédicteur prédit pour un x fixé la quantité suivante : $\underset{c\in\{1,\dots,L\}}{\operatorname{argmax}} \mathrm{P}(Y = c|X = x)$. Bien entendu, ce prédicteur n'est calculable que si l'on connaît la loi du couple (X, Y). C'est un estimateur idéal qu'on cherche à approcher. Notons $\hat{p}(x, c)$ un estimateur de la probabilité a posteriori $\mathrm{P}(Y = c|X = x)$. Une façon d'approcher le prédicteur de Bayes est alors de proposer un prédicteur $\hat{h}$ qui prédit, pour un x donné, la quantité $\underset{c\in\{1,\dots,L\}}{\operatorname{argmax}} \hat{p}(x, c)$. Nous pouvons alors mesurer la capacité du prédicteur $\hat{h}$ à bien estimer le prédicteur de Bayes, par exemple, par la quantité suivante :

$$\mathbb{E}\left[\sum_{c=1}^{L} |\, \hat{p}(X, c) - \mathrm{P}(Y = c|X) \,|\right] .$$

Dans ce cadre, il n'est pas nécessaire de très bien estimer les probabilités a posteriori pour bien prédire. En effet, prenons un problème à deux classes, notées 1 et 2. Si $\mathrm{P}(Y = 1|X = x) = 0.99$, l'estimateur de Bayes prédit alors la classe 1 pour l'observation x. Mais alors, si $\hat{p}(x, 1) = 0.51$, le prédicteur $\hat{h}$ prédit lui aussi la classe 1 pour l'observation x, alors que l'estimation de la probabilité a posteriori est assez mauvaise.

Une des particularités des forêts aléatoires est qu'elles peuvent être utilisées dans des cadres de régression et de classification, et seules quelques légères adaptations sont nécessaires pour passer d'un cadre à l'autre. De plus, elles présentent de très bonnes performances en prédiction (c'est-à-dire en terme d'erreur de généralisation) dans les deux cas (c.f. Section 1.2).

L'espace d'entrée

Dans cette thèse, nous nous limitons à l'étude de problème d'apprentissage statistique lorsque l'espace d'entrée $\mathcal{X}$ est égal à $\mathbb{R}^p$. L'entier naturel p désigne le nombre de coordonnées de X, et nous appelons ces coordonnées les variables. Par la suite, nous noterons X^j pour désigner la j-ième variable.

Le rapport entre le nombre d'observations n et le nombre de variables p est crucial en statistiques, et peut mener à des problèmes très différents. Nous développons ce point dans la sous-section qui suit.

1.1.2 Données de grande dimension et sélection de variables

Classiquement, les problèmes statistiques comportaient beaucoup d'observations (n de l'ordre de quelques centaines ou milliers) et peu de variables (p seulement de l'ordre de la dizaine). Les progrès technologiques ont fait que l'acquisition de données est devenue de plus en plus facile techniquement et de nos jours des bases de données gigantesques sont collectées quasi-quotidiennement. Les techniques classiques de statistiques ne suffisent plus pour traiter ces nouvelles données. Le nombre de variables p, peut maintenant atteindre des dizaines voire des centaines de milliers. Dans le même temps, pour beaucoup d'applications, le nombre d'observations n, se trouve réduit à quelques dizaines. Le domaine typique de telles situations est le domaine biomédical où l'on peut maintenant faire énormément de mesures sur un individu donné (mesures d'expression de gènes par exemple), mais le nombre d'individus sur lequel on fait l'expérience est réduit (dans le cas d'étude d'une maladie, le nombre de porteurs de la maladie qui participent à une étude est souvent limité).

Dans cette thèse, nous dirons que les données considérées sont de grande dimension quand $n << p$, ce qui signifie que le nombre de variables est très grand devant le nombre d'observations. Nous avons en tête des problèmes où n est de l'ordre de 100 et p de l'ordre de plusieurs milliers.

Comme nous le verrons au Chapitre 2, un des avantages des forêts aléatoires est qu'elles sont très performantes aussi bien pour des problèmes classiques (où $n >> p$) que pour des problèmes de grande dimension (où $n << p$).

Dans de nombreux problèmes et en particulier dans le cas de données de grande dimension ($n << p$), en plus de vouloir un bon prédicteur, les praticiens souhaitent également avoir des informations supplémentaires sur les variables du problème. En effet, pour mieux comprendre le phénomène étudié, ils veulent connaître les variables effectivement utiles pour expliquer le lien entrée-sortie. Ils désirent donc que le statisticien leur propose une sélection de variables. De plus, dans de nombreuses données de grande dimension, il est naturel de penser que relativement peu de variables (disons au maximum de l'ordre de n) agissent réellement sur la sortie. En effet, si le nombre de "vraies" variables dépasse largement n, le problème devient alors très mal posé et quasiment impossible à résoudre.

Relativement peu de méthodes de sélection de variables pour des données de grande dimension existent. Nous en citons quelques unes dont nous avons connaissance.
Poggi and Tuleau (2006) ont introduit une méthode basée sur le score d'importance des variables fourni par l'algorithme CART (concernant CART, voir la Section 1.2.1), tandis que Guyon et al. (2002); Rakotomamonjy (2003); Ben Ishak and Ghattas (2008) utilisent le score calculé par les Support Vector Machines (SVM : Vapnik (2000)). Díaz-Uriarte and Alvarez de Andrés (2006) proposent une procédure de sélection de variables basée sur l'importance des variables des forêts aléatoires.

Toutes ces méthodes calculent tout d'abord un score pour chacune des variables, puis procèdent à une introduction séquentielle de variables (méthodes "forward'), ou une élimination séquentielle de variables (méthodes "backward"), voire exécutent des méthodes pas-à-pas (méthodes "stepwise") mêlant introduction et élimination de variables. Fan and Lv (2008) proposent une méthode en deux temps : une première étape d'élimination de variables pour atteindre une situation raisonnable où p est de l'ordre de n, puis une deuxième étape de type forward basée par exemple sur le Least Absolute Shrinkage and Selection Operator (Lasso : Tibshirani (1996)).
Lê Cao et al. (2007) proposent quant à eux un schéma général pour calculer un score d'importance pour les variables, puis utilisent ce schéma avec comme méthode de base CART et SVM. Leur idée est d'apprendre un vecteur de poids sur toutes les variables (leur méta-algorithme se nomme Optimal Feature Weighting (OFW)) : une variable avec un poids fort est importante, tandis qu'une variable avec un poids faible est inutile.
Enfin, très récemment, des méthodes d'amélioration du Lasso, pour la sélection de variables, ont été mises au point. Ces dernières ont des points communs avec les méthodes d'ensemble (dont nous discuterons dans la Section 1.1.3). En effet, au lieu de chercher à faire de la sélection "en un coup" avec un Lasso classique, elles cherchent à construire plusieurs sous-ensembles de variables et à les mettre ensuite en commun. Dans Bolasso (pour Bootstrap-enhanced Lasso), introduit par Bach (2008), on génère plusieurs échantillons bootstrap puis on lance sur chacun d'eux la méthode du Lasso. Bolasso est donc à mettre en parallèle avec la méthode du Bagging (c.f. Section 1.1.3) de Breiman (1996). Dans Randomized Lasso, Meinshausen and Bühlmann (2010) choisissent de générer plusieurs échantillons par sous-échantillonnage et rajoutent un aléa supplémentaire dans la construction même du Lasso. Randomized Lasso est donc lui à rapprocher des forêts aléatoires Random Forests-RI (c.f. Section 1.2.2) de Breiman (2001).

Dans le Chapitre 2, nous proposons une méthode de sélection de variables, basée sur l'importance des variables calculée par les forêts aléatoires. Cette indice d'importance nous fournit un classement des variables, de la plus importante à la moins importante. Après avoir étudié sur des données simulées le comportement de cette indice importance, nous mettons au point un procédé automatique de sélection de variables. Le terme "automatique" signifie ici qu'il n'y a aucun a priori à apporter pour faire la sélection. Par exemple, il n'est pas nécessaire de préciser le nombre de variables que l'on souhaite obtenir : la procédure s'adapte aux données pour donner le sous-ensemble de variables final. De plus, notre méthode procède en deux étapes : la première (assez grossière) consiste à seuiller sur l'importance des variables dans le but d'éliminer un grand nombre de variables inutiles, tandis que la seconde (plus fine) consiste en une introduction de variables dans des modèles de forêts aléatoires.

Nous insistons, ici, également sur deux objectifs distincts en sélection de variables. En effet, nous distinguons l'objectif d'interprétation de l'objectif de prédiction.

1. Pour un but d'interprétation, nous cherchons à sélectionner toutes les variables X^j fortement reliées à la variable réponse Y (même si les variables X^j sont corrélées

entre elles).

2. Alors que pour un but de prédiction, nous cherchons à sélectionner un petit sous-ensemble de variables suffisant pour bien prédire la variable réponse.

Typiquement, un sous-ensemble construit pour satisfaire (1) pourra contenir beaucoup de variables, qui seront potentiellement très corrélées entre elles. Au contraire, le sous-ensemble de variables satisfaisant (2) contiendra peu de variables, avec très peu de corrélations entre elles.

Notre méthode de sélection de variables tente de satisfaire les deux objectifs précédents.

Le Chapitre 3 illustre l'application de cette procédure de sélection de variables pour deux problèmes issus de données réelles. La première application (Section 3.1) consiste à analyser des données d'Imagerie à Résonance Magnétique fonctionnelle (IRMf) dans un cadre de classification en grande dimension ($n = 72$, $p = 10^5$). La seconde application (Section 3.2) traite de données génomiques dans une étude sur la transmission du paludisme, dans un cadre de régression en dimension plus raisonnable, où le nombre de variables et le nombre d'observations sont du même ordre ($n = 110$, $p = 88$). Les sous-ensembles de variables sélectionnées par notre procédure sont très satisfaisants dans les deux cas. Précisons que la méthode est exactement la même dans les deux situations. Ceci montre la capacité d'adaptation de la procédure à des problèmes très différents.

Données de neuroimagerie

Détaillons ici le cadre de l'application sur les données d'IRMf de la Section 3.1, qui est en réalité le problème de départ qui a motivé la mise au point de la méthode de sélection de variables proposée. Les données ont été obtenues de la façon suivante. Un sujet humain est placé dans un IRM, et les images de l'activité cérébrale du sujet, suite à une stimulation externe, sont enregistrées. Les stimulations sont variées : calcul mental, présentation d'images, écoute de paroles, etc. Le but de l'étude est alors de déterminer quelles sont les zones du cerveau qui s'activent le plus pour telle ou telle tâche exécutée par le sujet.

Dans notre cas, la stimulation est la présentation, au sujet dans l'IRM, de quatre formes différentes d'un même objet. La Figure 1.1 présente un schéma résumant l'acquisition des données. Les entrées sont les images d'activation cérébrale en trois dimensions (3D) et les sorties sont les formes de l'objet. Les différentes formes de l'objet représente les classes. Les variables sont ici les voxels de l'image (un voxel est comme un pixel, mais en trois dimensions), parties élémentaires de l'image qui contiennent l'information codant l'activation cérébrale. Elles sont au nombre de 10^5, alors que nous n'avons que 72 observations. Nous cherchons à construire un prédicteur, qui, à une nouvelle image d'activation cérébrale, prédit la forme de l'objet qui a induit cette image lorsque qu'elle a été présentée au sujet. Nous cherchons également à sélectionner les variables qui expliquent la sortie, c'est-à-dire nous cherchons à détecter les zones responsables de la

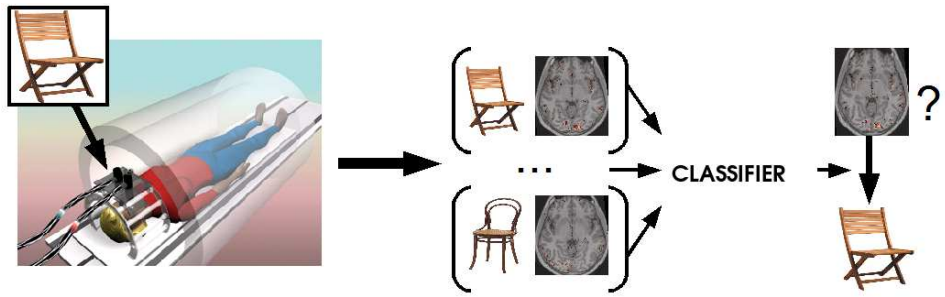

FIGURE 1.1 – Acquisition des données d'IRM lors d'une présentation de formes différentes d'un objet.

reconnaissance de formes dans le cerveau humain. Pour illustrer le type d'image d'entrée que nous devons traiter, la Figure 1.2 présente une coupe des images d'activation 3D obtenues lors de la présentation de deux formes différentes de l'objet.

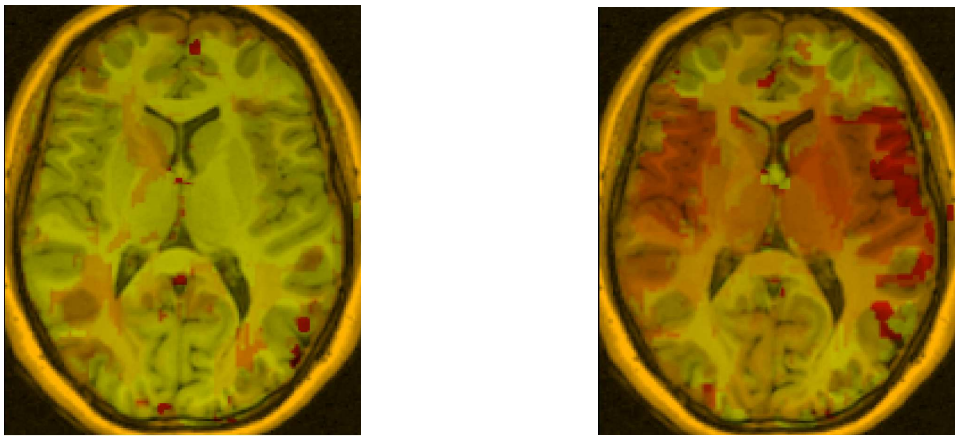

FIGURE 1.2 – Deux images d'IRM en réponses à la présentation de deux formes différentes d'un objet pour un même sujet. Les zones jaunes sont peu activées, oranges moyennement activées et rouges très activées.

Nous avons ce type d'images pour toutes les coupes du cerveau, et des images encore différentes pour la présentation des autres formes de l'objet. Ceci illustre la difficulté des données à traiter.

Comme nous le verrons dans la Section 3.1, notre procédure de sélection de variables donne des résultats très intéressants. La force de la méthode est de, sans aucun a priori, faire ressortir peu de zones où l'information se concentre. Ceci est d'une grande aide pour l'interprétation des résultats par les spécialistes. Nous déclarons que l'information se concentre dans les zones que nous avons sélectionnées, car un prédicteur construit en utilisant uniquement ces variables atteint des performances remarquables en prédiction.

Étude de la transmission du paludisme

Les données de la Section 3.2 sont issues d'une étude de la transmission du parasite responsable du paludisme. Le paludisme représente un problème de santé majeur, notamment dans les régions tropicales. Le parasite se transmet d'homme à homme via des piqûres de moustique. Cette étude tente d'expliquer le développement (ou le non-développement) du parasite à l'intérieur du moustique après que le moustique a piqué un porteur humain. Notre procédure de sélection de variables tente alors de déterminer quels sont les facteurs qui favorisent, ou au contraire inhibent, la capacité de transmission du parasite de l'homme vers le moustique.

Grossièrement, nous trouvons que les facteurs les plus influents sont la "concentration" du parasite dans le porteur humain, ainsi que la "multiplicité d'infection". Le fait de sélectionner cette multiplicité d'infection est assez nouveau, et est très intéressant pour les biologistes. En effet plus la multiplicité d'infection est faible, plus le parasite s'agrège dans le sans du porteur, et plus la probabilité que le moustique n'ingère pas le parasite au moment de la piqûre est grande (le moustique ne peut alors a fortiori pas transmettre le parasite).

Avant de décrire en détails la méthode des forêts aléatoires, nous présentons une famille plus large de méthodes statistiques à laquelle elle appartient : les méthodes d'ensemble.

1.1.3 Méthodes d'ensemble

Le principe général des méthodes d'ensemble (voir, par exemple Dietterich (2000)) est de construire une collection de prédicteurs, pour ensuite agréger l'ensemble de leurs prédictions. Dans un cadre de régression, agréger les prédictions de q prédicteurs revient par exemple à en faire la moyenne : chaque prédicteur fournit un $\hat{y}_l$, et la prédiction finale est alors $\frac{1}{q}\sum_{l=1}^{q}\hat{y}_l$. Dans un cadre de classification, l'agrégation revient par exemple à faire un vote majoritaire parmi les classes fournies par les prédicteurs. Soulignons le fait que l'étape d'agrégation de ces méthodes est toujours très simple et n'est pas optimisée, contrairement aux méthodes dites d'"agrégation de modèles", qui pour une famille de prédicteurs donnée, cherche la meilleure façon de combiner ces prédicteurs pour obtenir un bon prédicteur agrégé (voir par exemple les travaux de Lecué (2007)).

Au lieu d'essayer d'optimiser une méthode "en un coup", les méthodes d'ensemble génèrent plusieurs règles de prédiction et mettent ensuite en commun leurs différentes réponses. L'heuristique de ces méthodes est qu'en générant beaucoup de prédicteurs, on explore grandement l'espace des solutions, et qu'en agrégeant toutes les prédictions, on récupère un prédicteur qui rend compte de toute cette exploration. Du coup, on s'attend à ce que le prédicteur final soit meilleur que chacun des prédicteurs individuels : en quelque sorte, on croit à l'idée que l'union fait la force. En effet, illustrons sur un cas

simple cette heuristique. Plaçons-nous dans un cadre de classification à deux classes. Pour que le classifieur agrégé commette une erreur pour un x donné, il faut qu'au moins la moitié des classifieurs individuels se soient également trompés pour ce même x. Moralement, ceci n'arrive pas très souvent, car même si les classifieurs individuels commettent des erreurs, il est peu probable qu'ils commettent les mêmes erreurs pour les mêmes entrées. Ici, surgit l'idée que les prédicteurs individuels doivent être différents les uns des autres : la majorité ne doit pas se tromper pour un même x. Pour que cela soit possible, il faut également que les prédicteurs individuels soient relativement bons. Et là où un prédicteur se trompe, les autres doivent prendre le relais en ne se trompant pas.

L'heuristique expliquant le succès de ces méthodes d'ensemble se résume ainsi :

- Chaque prédicteur individuel doit être relativement bon.
- Les prédicteurs individuels doivent être différents les uns des autres.

Le premier point est nécessaire, car agréger des prédicteurs tous mauvais ne pourra vraisemblablement pas donner un bon prédicteur. Le deuxième point est également naturel, car agréger des prédicteurs tous quasiment pareils donnera encore un prédicteur semblable et n'améliorera pas les prédictions.

Finalement, pour qu'une méthode d'ensemble soit performante, elle doit réussir à construire une collection de prédicteurs qui vérifie les deux points ci-dessus.

Nous citons maintenant quelques exemples de méthodes d'ensemble qui sont historiquement apparues avant les forêts aléatoires (que nous détaillerons dans la Section 1.2).

Bagging

La méthode du Bagging a été introduite par Breiman (1996). Le mot Bagging est la contraction des mots **B**ootstrap et **Agg**regat**ing**. Étant donné un échantillon d'apprentissage $\mathcal{L}_n$ et une méthode de prédiction (appelée règle de base), qui construit sur $\mathcal{L}_n$ un prédicteur $\hat{h}(., \mathcal{L}_n)$. Le principe du Bagging est de tirer indépendamment plusieurs échantillons bootstrap $(\mathcal{L}_n^{\Theta_1}, \ldots, \mathcal{L}_n^{\Theta_q})$, d'appliquer la règle de base sur chacun d'eux pour obtenir une collection de prédicteurs $(\hat{h}(., \mathcal{L}_n^{\Theta_1}), \ldots, \hat{h}(., \mathcal{L}_n^{\Theta_q}))$, et enfin d'agréger ces prédicteurs de base.

L'idée du Bagging, et qu'en appliquant la règle de base sur différents échantillons bootstrap, on en modifie les prédictions, et donc on construit à terme une collection de prédicteurs variés. L'étape d'agrégation permet alors d'obtenir un prédicteur performant.

Un échantillon bootstrap $\mathcal{L}_n^{\Theta_l}$ est, par exemple, obtenu en tirant aléatoirement n observations avec remise dans l'échantillon d'apprentissage $\mathcal{L}_n$, chaque observation ayant une probabilité $1/n$ d'être tirée. La variable aléatoire Θ_l représente alors ce tirage aléatoire. Une deuxième façon classique d'obtenir un échantillon bootstrap est de tirer aléatoirement k observations sans remise dans $\mathcal{L}_n$, avec $k < n$.

Initialement, le Bagging a été introduit avec comme règle de base, un arbre de décision (nous détaillons ce cas dans la Section 1.2, car on peut alors voir le Bagging comme un cas particulier de forêts aléatoires). Cependant, le schéma est très général et peut-être appliqué à d'autres règles de bases : par exemple, la règle du plus proche voisin. Cette méthode du plus proche voisin "baggé" a été récemment étudiée, dans un cadre de régression, par Biau and Devroye (2010), puis Biau et al. (2010) (voir également les références de cet article). Le premier article établit la consistance de la méthode du plus proche voisin "baggé" (l'estimateur obtenu converge vers la vraie fonction de régression quand n tend vers $+\infty$), à condition que le nombre d'observations k dans les échantillons bootstrap (avec ou sans remise) tende vers $+\infty$, mais moins vite que n : $\frac{k}{n} \to +\infty$. Ces conditions de convergence sur k et n sont des conditions que nous retrouverons dans la Section 1.4. Le deuxième article va plus loin et montre que l'estimateur atteint la vitesse optimale de convergence sous les mêmes conditions sur k et n.

Cette étude illustre à merveille les bienfaits des méthodes d'ensemble : partant d'une règle basique assez pauvre (la règle du plus proche voisin n'est pas consistante), le Bagging la transforme en une règle aux très bonnes propriétés asymptotiques (consistance et vitesse optimale de convergence). L'idée, ici, est que la méthode du plus proche voisin, n'explore pas assez l'espace : elle assigne à un x donné, le y correspondant à l'observation de $\mathcal{L}_n$ la plus proche de x. Le fait de lancer la méthode, sur un échantillon bootstrap permet d'aller prendre en compte les sorties des observations plus éloignées de x (ce qui arrive lorsque les plus proches voisins de x ne sont pas présents dans l'échantillon bootstrap courant). Le plus proche voisin "baggé" met alors un poids sur chacune des données de $\mathcal{L}_n$ et le prédicteur agrégé est finalement une moyenne pondérée des Y_i de l'échantillon d'apprentissage. Les résultats théoriques nous assure en fait que la méthode règle automatiquement et de façon optimale ces poids.

Boosting

Introduit par Freund and Schapire (1996), le Boosting est une des méthodes d'ensemble les plus performantes à ce jour. Étant donné un échantillon d'apprentissage $\mathcal{L}_n$ et une méthode de prédiction (appelée règle de base), qui construit sur $\mathcal{L}_n$ un prédicteur $\hat{h}(., \mathcal{L}_n)$. Le principe du Boosting est de tirer un premier échantillon boostrap $\mathcal{L}_n^{\Theta_1}$, où chaque observation a une probabilité $1/n$ d'être tirée, puis d'appliquer la règle de base pour obtenir un premier prédicteur $\hat{h}(., \mathcal{L}_n^{\Theta_1})$. Ensuite, l'erreur de $\hat{h}(., \mathcal{L}_n^{\Theta_1})$ sur l'échantillon d'apprentissage $\mathcal{L}_n$ est calculée. Un deuxième échantillon boostrap $\mathcal{L}_n^{\Theta_2}$, est alors tiré mais la loi du tirage des observations n'est maintenant plus uniforme. La probabilité pour une observation d'être tirée dépend de la prédiction de $\hat{h}(., \mathcal{L}_n^{\Theta_1})$ sur cette observation. Le principe est d'augmenter la probabilité de tirer une observation mal prédite, et de diminuer celle de tirer une observation bien prédite. Une fois le nouvel échantillon $\mathcal{L}_n^{\Theta_2}$ obtenu, on applique à nouveau la règle de base $\hat{h}(., \mathcal{L}_n^{\Theta_2})$. On tire alors un troisième échantillon $\mathcal{L}_n^{\Theta_3}$, qui dépend des prédictions de $\hat{h}(., \mathcal{L}_n^{\Theta_2})$ sur $\mathcal{L}_n$ et ainsi de suite. La collection de prédicteurs obtenus est alors agrégée en faisant une moyenne pondérée.

Le Boosting est donc une méthode séquentielle, chaque échantillon étant tiré en fonction des performances de la règle de base sur l'échantillon précédent. En cela, le Boosting diffère de façon importante du Bagging, où les échantillons sont tirés indépendamment les uns des autres, et peuvent être obtenus en parallèle.

L'idée du Boosting est de se concentrer de plus en plus sur les observations mal prédites par la règle de base, pour essayer d'apprendre au mieux cette partie difficile de l'échantillon, en vue d'améliorer les performances globales.

Pour dénommer les méthodes de Boosting, Breiman (1998) parle d'algoritmes **Arc**ing, pour Adaptively Resample and Combine. L'idée est bien qu'au lieu de ré-échantillonner de façon indépendante comme dans le Bagging, on ré-échantillonne de façon adaptative dans le Boosting. Contrairement aux autres méthodes d'ensemble, le Boosting a beaucoup été étudié théoriquement : voir par exemple Bartlett and Traskin (2007) et les références de cet article.

Randomizing Outputs

Le Bagging et le Boosting construisent une collection de prédicteurs en ré-échantillonnant $\mathcal{L}_n$. Breiman (2000a) introduit la méthode Randomizing Outputs, qui est une méthode d'ensemble de nature différente. Le principe est, ici, de construire des échantillons indépendants dans lesquels on altère les sorties de l'échantillon d'apprentissage. La modification que subissent les sorties est obtenue en rajoutant une variable de bruit à chaque Y_i de $\mathcal{L}_n$. On obtient alors une collection d'échantillons "à sorties randomisées", puis on applique une règle de base sur chacun et on agrège enfin l'ensemble des prédicteurs obtenus.

L'idée de Randomizing Outputs est, encore, qu'en appliquant une règle de base sur des échantillons à sorties randomisées, on obtient une collection de prédicteurs différents les uns des autres.

Random Subspace

Ho (1998) introduit un autre type de méthode d'ensemble. Il n'est ici plus question de jouer sur l'échantillon, mais plutôt sur l'ensemble des variables considérées. Le principe de la méthode Random Subspace est de tirer aléatoirement un sous-ensemble de variables et d'appliquer une règle de base sur $\mathcal{L}_n$ qui ne prend en compte que les variables sélectionnées. On génère alors une collection de prédicteurs chacun construit en utilisant des variables différentes, puis on agrège ces prédicteurs. Les sous-ensembles de variables sont tirés indépendamment pour chaque prédicteur.

L'idée de cette méthode est de construire plusieurs prédicteurs chacun étant bon dans un sous-espace de $\mathcal{X}$ particulier, pour ensuite en déduire un prédicteur sur l'espace d'entrée tout entier.

Conclusion

Les quatre méthodes d'ensemble évoqués ont toutes un principe général commun. Il s'agit de partir d'une règle de prédiction de base, puis de perturber cette règle de base. On construit alors une collection de prédicteurs issus de différentes perturbations de la règle de base. Enfin on agrège l'ensemble des prédicteurs obtenus. Les perturbations peuvent portés sur l'échantillon (ré-échantillonnage, sorties randomisées) ou le sous-espace d'entrée dans lequel on construit le prédicteur, et les différentes perturbations sont générées indépendamment les unes des autres, ou non.

Pour chacune de ces méthodes, les auteurs montrent sur des simulations que le prédicteur agrégé final fait systématiquement mieux (en terme d'erreur de généralisation) que la règle de prédiction de base. Donc, en pratique, il apparaît que "perturber puis agréger" améliore les performances d'une méthode de prédiction donnée.

La remarque précédente ne vaut que si dans la collection construite, les prédicteurs sont différents les uns des autres. C'est pourquoi ces méthodes sont appliqués sur des méthodes dites "instables". Une méthode est instable si de petites perturbations de l'échantillon d'apprentissage peuvent engendrer de grandes modifications du prédicteur obtenu. Par exemple, les arbres de décision (que nous détaillons dans la Section 1.2.1 ci-après) sont instables. Par contre, les méthodes linéaires, qui elles sont stables, ne sont jamais utilisés dans les méthodes d'ensemble.

Nous détaillons à présent la méthode des forêts aléatoires, qui est le sujet principal de cette thèse.

1.2 Les forêts aléatoires de Leo Breiman

Les forêts aléatoires ont été introduites par Breiman (2001). Cet article fondateur de la méthode a été le point de départ de cette thèse. Voici la définition générale des forêts aléatoires, donnée dans cet article :

Définition 1. *Soit* $\{\hat{h}(.,\Theta_1),\ldots,\hat{h}(.,\Theta_q)\}$ *une collection de prédicteurs par arbre, où* $(\Theta_1,\ldots,\Theta_q)$ *est une suite de variables aléatoires i.i.d., indépendante de l'échantillon d'apprentissage* $\mathcal{L}_n$*. Le prédicteur des forêts aléatoires est obtenu par agrégation de cette collection de prédicteurs.*

Le terme forêt aléatoire vient du fait que les prédicteurs individuels sont, ici, explicitement des prédicteurs par arbre, et du fait que chaque arbre dépend d'une variable aléatoire supplémentaire (c'est-à-dire en plus de $\mathcal{L}_n$). Une forêt aléatoire est l'agrégation d'une collection d'arbres aléatoires.

Nous détaillons les procédures d'arbre de décision dans le paragraphe suivant.

Les forêts aléatoires font bien partie de la famille des méthodes d'ensemble. Remarquons d'ailleurs que, parmi les méthodes d'ensemble précédemment cités (lorsque l'on

choisit comme règle de base un arbre de décision), seul le Boosting ne rentre pas dans la définition de forêts aléatoires. En effet, les arbres individuels du Boosting ne dépendent pas d'aléas indépendants les uns des autres. Le Bagging, Randomizing Outputs et Random Subspace sont alors des cas particuliers de forêts aléatoires, avec respectivement pour aléa supplémentaire le tirage de l'échantillon bootstrap, la modification aléatoire des sorties de $\mathcal{L}_n$ et le tirage des sous-ensembles de variables. En plus de ces trois méthodes, il existe de nombreux cas particuliers de forêts aléatoires dans la littérature. Nous listons tous ceux dont nous avons connaissance dans la Section 1.3.

Concernant le vocabulaire des forêts aléatoires, il existe une ambiguïté dans la littérature. En effet, Leo Breiman, dans son article de 2001, définit les forêts aléatoires comme ci-dessus. Les forêts aléatoires sont donc pour lui une famille de méthodes. Or, dans le même article, il présente un cas particulier de forêts aléatoires, appelées Random Forests-RI, qu'il a implémentées (voir Breiman and Cutler (2005)). Par suite, ce sont ces Random Forests-RI qui ont été quasi-systématiquement utilisées dans de très nombreuses applications réelles. Et pour cause, le programme est accessible à tous, est facile d'utilisation et la méthode atteint des performances exceptionnelles. Finalement, la dénomination "forêts aléatoires" désigne maintenant très souvent les Random Forests-RI. On trouve également le terme de "forêts aléatoires de Leo Breiman" pour désigner les Random Forests-RI.

Dans la suite de cette section, nous détaillons tout d'abord une méthode d'arbre de décision (les arbres CART) largement utilisée dans les forêts aléatoires et plus généralement dans les méthodes d'ensemble. Ensuite nous décrivons en détails les Random Forests-RI. Nous terminons par définir l'importance des variables, qui est une sortie très utile de l'algorithme Random Forests-RI.

1.2.1 CART

L'acronyme CART signifie Classification And Regression Trees. Il désigne une méthode statistique, introduite par Breiman et al. (1984) qui construit des prédicteurs par arbre aussi bien en régression qu'en classification. Le principe général de CART est de partitionner récursivement l'espace d'entrée $\mathcal{X}$ de façon dyadique, puis de déterminer une sous-partition optimale pour la prédiction. Nous renvoyons au Chapitre 2 de la thèse de Gey (2002), pour un exposé en français, concis et très clair de la méthode en régression. Nous n'en rappelons ici que les grandes lignes.

A chaque étape du partitionnement, on découpe une partie de l'espace en deux sous-parties. On associe alors naturellement un arbre binaire à la partition construite. Les noeuds de l'arbre sont associés aux éléments de la partition. Par exemple, la racine de l'arbre est associé à l'espace d'entrée tout entier. Ses deux noeuds fils sont associés aux deux sous-parties obtenues par la première découpe du partitionnement, et ainsi de suite. La Figure 1.3 illustre la correspondance entre une partition dyadique et un arbre binaire.

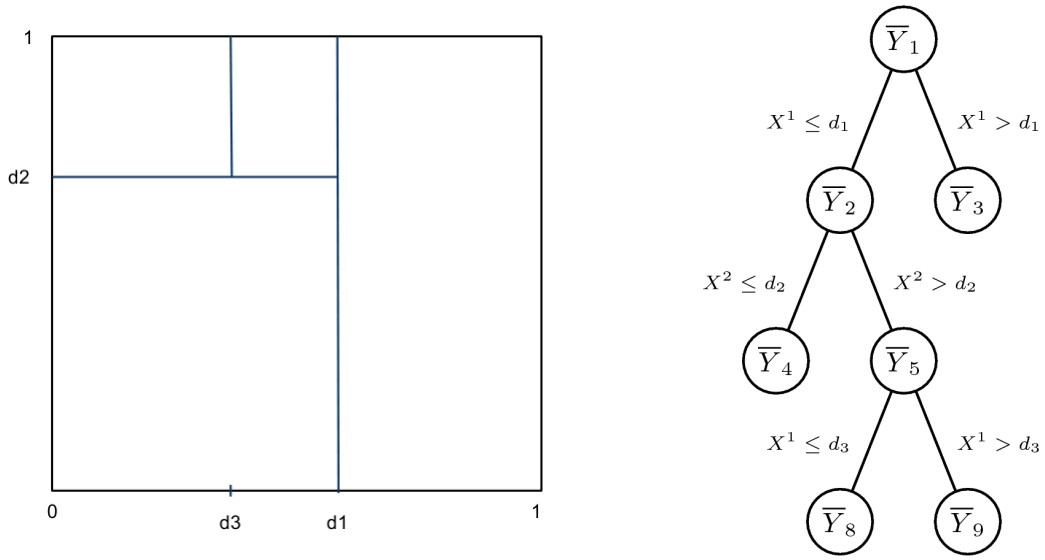

FIGURE 1.3 – Une partition dyadique du carré unité et son arbre CART associé.

Détaillons maintenant la règle de découpe. Nous nous restreignons à des variables continues, l'espace d'entrée est alors $\mathbb{R}^p$, où p est le nombre de variables. Partons de la racine de l'arbre (associée à $\mathbb{R}^p$ tout entier), qui contient toutes les observations de l'échantillon d'apprentissage $\mathcal{L}_n$. La première étape de CART consiste à découper au mieux cette racine en deux noeuds fils. Nous appelons coupure un élément de la forme

$$\{X^j \leqslant d\} \cup \{X^j > d\} \ ,$$

où $j \in \{1, \ldots, p\}$ et $d \in \mathbb{R}$. Découper suivant $\{X^j \leqslant d\} \cup \{X^j > d\}$ signifie que toutes les observations avec une valeur de la j-ième variable plus petite que d vont dans le noeud fils de gauche, et toutes celles avec une valeur plus grande que d vont dans le noeud fils de droite. La méthode sélectionne alors la meilleure découpe, c'est-à-dire le couple (j, d) qui minimise une certaine fonction de coût :

- En régression, on cherche à minimiser la variance des noeuds fils. La variance d'un noeud t est définie par $\sum_{i:X_i \in t} (Y_i - \overline{Y}_t)^2$ où $\overline{Y}_t$ est la moyenne des Y_i des observations présentes dans le noeud t.
- En classification (où l'ensemble des classes est $\{1, \ldots, L\}$), on cherche à minimiser l'indice de Gini des noeuds fils. L'indice de Gini d'un noeud t est défini par $\sum_{c=1}^{L} \hat{p}_t^c(1 - \hat{p}_t^c)$, où $\hat{p}_t^c$ est la proportion d'observations de classe c dans le noeud t.

En régression, on cherche donc des découpes qui tendent à diminuer la variance des noeuds obtenus. En classification, on cherche à diminuer l'indice de Gini, et donc à augmenter l'homogénéité des noeuds obtenus (un noeud étant parfaitement homogène s'il ne contient que des observations de la même classe).

Une fois la racine de l'arbre découpée, on se restreint à chacun des noeuds fils et on recherche alors, suivant le même procédé, la meilleure façon de les découper en deux nouveaux noeuds, et ainsi de suite. Les arbres sont ainsi développés, jusqu'à atteindre une règle d'arrêt. Une règle d'arrêt classique consiste à ne pas découper des noeuds qui contiennent moins qu'un certain nombre d'observations. Les noeuds terminaux, qui ne sont plus découpés, sont appelés les feuilles de l'arbre. A noter, que l'on ne découpe pas un noeud pur, c'est-à-dire un noeud ne contenant que des observations dont les sorties sont les mêmes (typiquement en classification). On appelle arbre maximal, l'arbre pleinement développé. Dans le même temps, on associe à chaque noeud t de l'arbre une valeur ($\overline{Y}_t$ en régression, la classe majoritaire des observations présentes dans le noeud t en classification). Donc, à un arbre est associée une partition (définie par ses feuilles) et également des valeurs attachées à chaque élément de cette partition. Le prédicteur par arbre est alors la fonction constante par morceaux, associée à l'arbre maximal.

La deuxième étape de l'algorithme CART, s'appelle l'élagage et consiste à chercher le meilleur sous-arbre élagué de l'arbre maximal (meilleur au sens de l'erreur de généralisation). L'idée est que l'arbre maximal possède une très grande variance et un biais faible. A contrario, un arbre constitué uniquement de la racine (qui engendre alors un prédicteur constant) a une très petite variance mais un biais élevé. L'élagage est une procédure de sélection de modèles (où les modèles sont les sous-arbres élagués de l'arbre maximal) qui minimise un critère pénalisé, la pénalité étant proportionnelle au nombre de feuilles de l'arbre. Pour plus de détails et des résultats théoriques sur cette étape, voir Gey and Nedelec (2005). Comme nous le verrons au paragraphe suivant, les forêts aléatoires sont la plupart du temps des forêts d'arbres non élagués. Cependant, c'est grâce à l'étape d'agrégation que l'on peut se dispenser d'élaguer les arbres individuels. Nous insistons sur le fait qu'un arbre CART, s'il est utilisé seul, doit être élagué.

Il existe plusieurs façons de construire des arbres CART, par exemple en changeant la famille de coupures autorisées, la fonction de coût ou encore la règle d'arrêt. Nous nous limitons à la façon, couramment utilisée, présentée ci-dessus dans le cadre de cette thèse et renvoyons à Breiman et al. (1984) pour des compléments. Signalons également l'existence d'autres méthodes construisant des arbres de décision, comme par exemple l'algorithme $C4.5$ introduit par Quilan (1993). De plus, il existe d'autres méthodes de partitionnement récursif qui construisent des prédicteurs plus réguliers que les prédicteurs par arbres constants par morceaux. Citons, par exemple, l'algorithme MARS introduit par Friedman (1991). Dans le cadre de cette thèse, nous nous limitons à l'utilisation des arbres CART dans les forêts aléatoires.

1.2.2 Random Forests-RI

Random Forests-RI signifie "forêts aléatoires à variables d'entrée aléatoires" (Random Forests with Random Inputs). Le principe des Random Forests-RI est tout d'abord de générer plusieurs échantillons bootstrap $\mathcal{L}_n^{\Theta_1}, \ldots, \mathcal{L}_n^{\Theta_q}$ (comme dans le Bagging). Ensuite, sur chaque échantillon $\mathcal{L}_n^{\Theta_l}$, une variante de CART est appliquée. Plus précisément,

un arbre est, ici, construit de la façon suivante. Pour découper un noeud, on tire aléatoirement un nombre m de variables, et on cherche la meilleure coupure uniquement suivant les m variables sélectionnées. De plus, l'arbre construit est complètement développé (arbre maximal) et n'est pas élagué. La collection d'arbres obtenus est enfin agrégée (moyenne en régression, vote majoritaire en classification) pour donner le prédicteur Random Forests-RI.

Ainsi, les Random Forests-RI peuvent être vues comme une variante du Bagging, où la différence intervient dans la construction des arbres individuels (les étapes de bootstrap et d'agrégation étant les mêmes).

Le tirage, à chaque noeud, des m variables se fait, sans remise, et uniformément parmi toutes les variables (chaque variable a un probabilité $1/p$ d'être choisie). Le nombre m ($m \leqslant p$) est fixé au début de la construction de la forêt et est donc identique pour tous les arbres. C'est un paramètre très important de la méthode. Une forêt construite avec $m = p$ revient à faire du Bagging d'arbres CART non élagués, alors qu'une forêt construite avec $m = 1$ est très différente du Bagging. En effet, lorsque $m = 1$, le choix de la variable, suivant laquelle est découpé un noeud, est complètement aléatoire (les coupures suivant cette variable ne sont pas mises en compétition avec des coupures utilisant d'autres variables).

Le tirage des m variables à chaque noeud représente un aléa supplémentaire, par rapport au Bagging. Pour les Random Forests-RI, il y a donc deux sources d'aléas pour générer la collection des prédicteurs individuels : l'aléa dû au bootstrap et l'aléa du choix des variables pour découper chaque noeud d'un arbre. Ainsi, on perturbe à la fois l'échantillon sur lequel on lance la règle de base, et à la fois le coeur de la construction de la règle de base. Ce tirage aléatoire de variables pour découper un noeud avait déjà été utilisé par Amit and Geman (1997) dans des problèmes de reconnaissance d'image. Leur méthode a beaucoup influencé Leo Breiman dans sa mise au point de Random Forests-RI. Pour leur problème, le nombre de coupures candidates était tellement gigantesque qu'ils étaient obligés de réduire le nombre de possibilités, par exemple en effectuant un choix aléatoire préliminaire à la découpe.

En pratique, les Random Forests-RI améliorent les performances du Bagging (voir la comparaison des méthodes sur des données de référence dans Breiman (2001)). L'explication heuristique de ces améliorations est que le fait de rajouter un aléa supplémentaire pour construire les arbres, rend ces derniers encore plus différents les uns des autres, sans pour autant dégrader de façon significative leurs performances individuelles. Le prédicteur agrégé est alors meilleur. Nous avons vu que la plupart des méthodes d'ensemble construisent une collection de prédicteurs qui sont des versions perturbées d'une règle de base. La perturbation introduite doit alors réaliser un compromis entre deux situations : une trop grande perturbation dégrade les prédicteurs individuels et le prédicteur agrégé est alors mauvais, une trop petite perturbation induit des prédicteurs individuels trop similaires entre eux et le prédicteur agrégé n'apporte alors aucune amélioration. Les excellents résultats des Random Forests-RI en pratique laissent penser qu'elles (avec le paramètre m bien choisi) réalisent un bon compromis, en injectant la "bonne dose"

d'aléa.

Le paquet R `randomForest`

L'algorithme des Random Forests-RI a été codé par Breiman and Cutler (2005). Il a ensuite été importé dans le logiciel libre R par Liaw and Wiener (2002), via le paquet `randomForest`. Ce paquet est librement utilisable et est utilisé dans le traitement de très nombreuses applications réelles. Nous l'avons exclusivement utilisé pour toutes les simulations présentes dans cette thèse. Il existe deux principaux paramètres dans ce programme.

- Le paramètre le plus important est le nombre m de variables choisies aléatoirement à chacun des noeuds des arbres. Il est nommé `mtry` dans le paquet. Il peut varier de 1 à p et possède une valeur par défaut : $\sqrt{p}$ en classification, $p/3$ en régression.
- Nous pouvons également jouer sur le nombre d'arbres q de la forêt. Ce paramètre est nommé `ntree` et sa valeur par défaut est 500.

Le programme permet également de régler d'autres aspects de la méthode : le nombre minimum d'observations (nommé `nodesize`) en dessous duquel on ne découpe pas un noeud, ou encore la façon d'obtenir les échantillons bootstrap (avec ou sans remise, ainsi que le nombre d'observations tirées). Nous laissons pour ces éléments les valeurs par défaut, i.e. un `nodesize` de 1 en classification et 5 en régression, et les échantillons bootstrap considérés sont tous obtenus en tirant n observations avec remise dans l'échantillon d'apprentissage $\mathcal{L}_n$.

Dans l'Annexe 2.A, nous étudions le comportement de l'erreur de généralisation des Random Forests-RI vis-à-vis des deux paramètres `mtry` et `ntree`. Les simulations sont effectuées sur des jeux de données de référence, réels et simulés, de petite et de grande dimension. Cette partie est un bilan des études précédentes sur ce sujet (Breiman (2001), Díaz-Uriarte and Alvarez de Andrés (2006)) avec l'ajout de graphiques qui permettent d'illustrer le comportement de l'erreur en fonction de ces paramètres. Nous donnons également des conseils sur le réglage des ces paramètres.

L'erreur Out-Of-Bag

En plus de construire un prédicteur, l'algorithme des Random Forests-RI calcule une estimation de son erreur de généralisation : l'erreur Out-Of-Bag (OOB). "Out-Of-Bag" signifie ici "en dehors du bootstrap". Cette erreur était déjà calculée par l'algorithme du Bagging, d'où la présence du mot "Bag". Le procédé de calcul de cette erreur est le suivant.

Fixons une observation (X_i, Y_i) de l'échantillon d'apprentissage $\mathcal{L}_n$. Considérons maintenant l'ensemble des arbres construits sur les échantillons bootstrap ne contenant pas cette observation, c'est-à-dire pour lesquels cette observation est "Out-Of-Bag". Nous agrégeons alors uniquement les prédictions de ces arbres pour fabriquer notre pré-

diction $\hat{Y}_i$ de Y_i. Après avoir fait cette opération pour toutes les données de $\mathcal{L}_n$, nous calculons alors l'erreur commise par nos prédictions : l'erreur quadratique moyenne en régression $\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2\right)$, et la proportion d'observations mal classées en classification $\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\hat{Y}_i \neq Y_i}\right)$. Cette quantité est appelée erreur OOB du prédicteur Random Forests-RI.

Cette estimation s'impose les mêmes contraintes que celles des estimateurs classiques de l'erreur de généralisation (échantillon test, validation croisée), au sens où les données prédites sont des données qui n'ont pas été rencontrées au préalable par le prédicteur utilisé. Un avantage de l'erreur OOB par rapport aux estimateurs classiques est qu'elle ne nécessite pas de découpage de l'échantillon d'apprentissage. Ce découpage est en quelque sorte inclus dans la génération des différents échantillons bootstrap. Cependant, il faut bien noter que pour chaque observation ce n'est pas le même ensemble d'arbres qui est agrégé. En conséquence, cette erreur estime l'erreur de généralisation d'une forêt, mais elle n'utilise jamais les prédictions de la forêt elle-même, mais plutôt celles de prédicteurs qui sont des agrégations d'arbres de cette forêt. Un inconvénient de l'erreur OOB est qu'elle est souvent considérée comme étant trop optimiste. Une erreur OOB corrigée a été introduite par Efron and Tibshirani (1997). Néanmoins, dans cette thèse, nous utilisons uniquement l'erreur OOB pour comparer des prédicteurs entre eux, et non pour obtenir une estimation précise de leurs erreurs de généralisation.

1.2.3 L'importance des variables au sens des forêts aléatoires

Comme rappelé dans la Section 1.1.2, il est très utile, en pratique, d'avoir des informations sur les variables des données que l'on étudie. Quelles sont les variables vraiment nécessaires pour expliquer la sortie ? De quelles variables peut-on se passer ? Ces informations peuvent être d'une grande aide pour l'interprétation des données. Elles peuvent également servir à construire de meilleurs prédicteurs : un prédicteur construit en utilisant uniquement les variables utiles pourra être plus performant qu'un prédicteur construit avec en plus des variables de bruit.

Un aspect qui a fait le succès des arbres CART est qu'ils sont facilement interprétables. En effet, une fois l'arbre CART construit, on se dit naturellement que les variables qui interviennent effectivement dans les découpes des noeuds de l'arbre (et particulièrement les noeuds les plus proches de la racine) sont les variables les plus utiles pour le problème considéré. C'est une façon heuristique pratique d'avoir une information sur l'importance des variables. En réalité, cette première intuition donne des résultats biaisés et un indice d'importance des variables plus élaboré est fournit par les arbres CART. Cette indice est définit dans Breiman et al. (1984), et est, par exemple, utilisé par Poggi and Tuleau (2006) dans une procédure de sélection de variables.

Cette facilité d'interprétation est perdue avec les forêts aléatoires. En effet, une forêt

est l'agrégation de toute une collection d'arbres, donc on perd l'aspect structuré du prédicteur obtenu. Pour palier à ce manque, un autre indice d'importance des variables, spécifique aux forêts, est introduit par Breiman (2001). Comme pour l'erreur OOB, le calcul de cet indice d'importance utilise pleinement les échantillons bootstrap. Le principe en est le suivant.

Fixons $j \in \{1, \ldots, p\}$ et détaillons le calcul de l'indice d'importance de la variable X^j. Considérons un échantillon bootstrap $\mathcal{L}_n^{\Theta_l}$ et l'échantillon OOB_l associé, c'est-à-dire l'ensemble des observations qui n'apparaissent pas dans $\mathcal{L}_n^{\Theta_l}$. Calculons errOOB_l, l'erreur commise sur OOB_l par l'arbre construit sur $\mathcal{L}_n^{\Theta_l}$ (erreur quadratique moyenne en régression, proportion de mal classés en classification). Permutons alors aléatoirement les valeurs de la j-ième variable dans l'échantillon OOB_l. Ceci donne un échantillon perturbé, noté $\widetilde{\mathrm{OOB}_l}^j$. Calculons enfin $\mathrm{err}\widetilde{\mathrm{OOB}_l}^j$, l'erreur sur l'échantillon $\widetilde{\mathrm{OOB}_l}^j$. Nous effectuons ces opérations pour tous les échantillons bootstrap. L'importance de la variable X^j, $\mathrm{VI}(X^j)$, est définie par la différence entre l'erreur moyenne d'un arbre sur l'échantillon OOB perturbé et celle sur l'échantillon OOB :

$$\mathrm{VI}(X^j) = \frac{1}{q} \sum_{l=1}^{q} \left(\mathrm{err}\widetilde{\mathrm{OOB}_l}^j - \mathrm{errOOB}_l \right) .$$

Ainsi, plus les permutations aléatoires de la j-ième variable engendrent une forte augmentation de l'erreur, plus la variable est importante. A l'inverse, si les permutations n'ont quasiment aucun effet sur l'erreur, la variable est considérée comme une variable très peu importante.

Précisons que ce calcul de l'importance est exactement celui qui est exécuté dans le paquet `randomForest`. La définition qui est donné de l'importance dans Breiman (2001) est légèrement différente. Les échantillons OOB perturbés sont obtenus de la même manière. Par contre, l'importance d'une variable est alors définie par la différence entre l'erreur OOB sur les échantillons OOB perturbés, et l'erreur OOB initiale. Nous pensons que la raison de ce changement est que dans le calcul de l'erreur OOB, il y a une étape d'agrégation. L'agrégation a tendance à amoindrir les erreurs des arbres individuels, donc l'utilisation de l'erreur OOB dans le calcul tend à atténuer les effets des permutations aléatoires des variables. Or, dans le calcul de l'importance, nous voulons au contraire, que les effets de ces permutations soient les plus visibles possible. C'est pourquoi l'erreur OOB est remplacée par l'erreur moyenne sur tous les arbres, dans laquelle aucune étape d'agrégation n'est effectuée.

Ishwaran (2007) présente une première approche pour tenter d'expliquer de façon théorique le comportement de l'importance des variables, en étudiant une version simplifiée cette importance. Des études, à l'aide de simulations, ont également été menées pour illustrer le comportement de l'importance des variables des forêts aléatoires. Citons, par exemple, les travaux de Strobl et al. (2007, 2008); Archer and Kimes (2008).

Grâce à cet indice d'importance, nous récupèrons des informations sur les variables, très utiles pour interpréter les données que l'on traite. Nous étudions le comportement de cet indice d'importance dans le Chapitre 2. Nous utilisons ensuite le classement

des variables obtenu par le calcul d'importance pour mettre au point une procédure automatique de sélection de variables.

1.3 Revue des versions de forêts aléatoires

La Définition 1 des forêts aléatoires est très générale et définit une grande classe de prédicteurs. On peut trouver de nombreux cas particuliers de forêts aléatoires dans la littérature. Certains ont été introduits pour faciliter l'étude théorique des forêts aléatoires, d'autres pour augmenter la vitesse de calcul ou encore améliorer les performances. Nous regroupons dans cette section tous ceux dont nous avons connaissance.

Nous distinguons deux grandes familles de forêts aléatoires : les forêts purement aléatoires et les forêts aléatoires classiques. Dans les forêts purement aléatoires, la construction des partitions de l'espace d'entrée (associées aux arbres individuels) se fait complètement aléatoirement, c'est-à-dire sans prendre en compte les données de l'échantillon d'apprentissage $\mathcal{L}_n$. Nous appelons forêts aléatoires classiques, les versions de forêts qui utilisent effectivement $\mathcal{L}_n$ pour partitionner l'espace d'entrée.

Enfin, nous insistons sur le fait que l'étape d'agrégation des arbres est la même pour toutes les versions de forêts aléatoires présentées ici, à savoir la moyenne des prédictions en régression et le vote majoritaire en classification. De plus, lorsque cela n'est pas précisé, les arbres individuels sont toujours pleinement développés.

1.3.1 Les forêts aléatoires classiques

Le représentant majeur de cette famille est bien entendu Random Forests-RI, qui est l'algorithme de référence dans ce domaine.

Les méthodes d'ensemble précédemment évoquées dans la Section 1.1.3 font également partie de la famille des forêts aléatoires classiques : Bagging, Randomizing Outputs, Random Subspace.

Citons également une méthode introduite par Dietterich (1999) et appelée Randomization. Le principe de Randomization est de calculer, à chaque noeud d'un arbre, les 20 meilleures coupures de ce noeud, puis de découper le noeud suivant une coupure choisie aléatoirement (et de façon uniforme) parmi ces 20 candidates.

Enfin, Geurts et al. (2006) introduisent la méthode Extra-Trees (pour Extremly Randomized Trees). Le principe est, ici, de tirer aléatoirement m variables à chaque noeud, puis de choisir aléatoirement un point de coupure pour chaque variable : pour une variable continue X^j, on tire le seuil d de façon uniforme dans le segment délimité par la plus petite et la plus grande valeur de X^j des observations présentent dans le noeud courant. La coupure est alors $\{X^j \leqslant d\} \cup \{X^j > d\}$. Nous récupérons alors m coupures, et choisissons la meilleure coupure parmi celles-ci, c'est-à-dire celle qui mi-

nimise la fonction de coût considérée. Le choix des m variables à chaque noeud et le choix de la meilleure coupure sont les mêmes que dans les Random Forests-RI. Cependant, ici un aléa supplémentaire est introduit au niveau du point de coupure. Là où les Random Forests-RI optimisent le point de coupure sur $\mathcal{L}_n$, les Extra-Trees tire ce point de coupure aléatoirement. Geurts et al. (2006) illustrent dans leur article les très bonnes performances des Extra-Trees, qui apportent même parfois des améliorations par rapport à la méthode de référence Random Forests-RI.

Nous présentons un résumé du type de randomisation (ou aléa supplémentaire) utilisée pour construire ces forêts aléatoires. Cette présentation est en grande partie basée sur celle de Liu et al. (2005).

1. Randomisation préalable à la construction de l'arbre :
   (a) Ré-échantillonnage : e.g. Bagging, Random Forests-RI
   (b) Tirage d'un sous-ensemble de variables (fixe pour tout l'arbre) : e.g. Random Subspace
   (c) Perturbation des sorties : e.g. Randomizing Outputs
2. Randomisation au coeur de la construction de l'arbre :
   (a) Tirage d'un sous-ensemble de variables à chaque noeud : e.g. Random Forets-RI, Extra-Trees
   (b) Tirage d'une coupure parmi de bonnes coupures candidates : e.g. Randomization
   (c) Tirage du point de coupure : e.g. Extra-Trees

Bien entendu, il existe pour chaque type d'aléa, plusieurs randomisations possibles : ré-échantillonnage avec ou sans remise sous-échantillonné ou non, tirage uniforme ou suivant une autre loi des sous-ensembles de variables, etc. Il est également possible de combiner plusieurs randomisations dans une même méthode, comme c'est déjà le cas dans Random Forets-RI ou Extra-Trees. Néanmoins, plusieurs associations de ces randomisations ont déjà été testées par les auteurs des méthodes citées ici. Finalement, seules les méthodes présentant les meilleurs résultats ont été retenus. Par exemple, Geurts et al. (2006) ont essayé de coupler le Bagging et Extra-Trees, mais ont montré qu'en pratique cela n'apporte pas d'amélioration.

1.3.2 Les forêts purement aléatoires

Les forêts purement aléatoires n'utilisent pas l'échantillon d'apprentissage pour partitionner l'espace d'entrée, ou de façon équivalente, pour développer les arbres individuels. Elles génèrent donc une partition de $\mathcal{X}$ complètement aléatoire, i.e. indépendamment de $\mathcal{L}_n$, puis elles se servent de $\mathcal{L}_n$ pour assigner une valeur à chaque élément de la partition finale, c'est-à-dire à chaque feuille de l'arbre.

La première forêt purement aléatoire a été introduite par Breiman (2000b) en vue d'une analyse théorique. Il introduit la méthode PRF (Purely Random Forests) qui construit un ensemble d'arbres de la façon suivante. On choisit tout d'abord aléatoirement une feuille de l'arbre. On tire, ensuite, aléatoirement une variable de coupure X^j. Enfin, on tire aléatoirement et de façon uniforme un point de coupure. Cette opération est répétée k fois pour obtenir un arbre. k est un paramètre important qui régit la profondeur de l'arbre. Nous renvoyons à la Section 1.4.1 pour des résultats théoriques concernant PRF.

Une autre version, proche de PRF, est apparue au même moment dans Cutler and Zhao (2001) avec l'algorithme PERT (Perfect Random Tree Ensemble). La règle pour découper un noeud est la suivante. Tout d'abord on tire aléatoirement une variable de coupure X^j. On tire ensuite aléatoirement un point de coupure (comme dans Extra-Trees) : de façon uniforme dans le segment délimité par la plus petite et la plus grande valeur de X^j des observations présentent dans le noeud. PERT est alors équivalent à une forêt Extra-Trees lorsque le nombre m de variables tirées à chaque noeud est égal à 1. Geurts et al. (2006) nomme ce cas extrême les Totally Radomized Trees Ensembles.

Remarquons que PERT utilise l'échantillon d'apprentissage $\mathcal{L}_n$ lors du tirage du point de coupure (entre la plus petite et la plus grande valeur de la variable considérée). Néanmoins, ceci est une condition pratique pour ne pas faire apparaître de feuille vide (i.e. ne contenant aucune donnée de $\mathcal{L}_n$) et la dépendance avec l'échantillon d'apprentissage n'est que très faible. Ce qui importe en réalité, c'est de ne pas utiliser les sorties $Y_1, \ldots, Y_n$ dans la construction de l'arbre. PERT a été introduit initialement dans un cadre de classification, mais il se généralise naturellement en régression. La principale différence avec PRF est que dans PERT tous les noeuds de l'arbre sont découpés jusqu'au bout, alors que PRF choisit k fois une feuille de l'arbre à découper. Les performances de PERT sont de façon surprenante assez bonnes (c.f. Cutler and Zhao (2001)). PERT n'est pas meilleur que Random Forests-RI, mais il reste comparable. Il faut par ailleurs noter que la complexité algorithmique est beaucoup moins grande pour PERT que pour Random Forests-RI. Effectivement, dans PERT rien n'est optimisé : pour construire un arbre nous tirons juste des variables aléatoires et nous ne comparons jamais deux coupures entre elles.

Citons également les travaux de Fan et al. (2003) et Fan et al. (2006) qui ont mené des études de comparaison par simulation de forêts purement aléatoires avec des forêts aléatoires classiques. Leur message est que les forêts purement aléatoires sont une bonne alternative, car elles sont très rapides à exécuter tout en présentant des performances comparables.

Dans le Chapitre 4, nous introduisons une autre version de forêts purement aléatoires. Elle se restreint au cas où $\mathcal{X} = [0,1]$ (donc $p = 1$) et nous analysons les propriétés théoriques des estimateurs obtenus dans un cadre de régression. Nous appelons cette variante PURF (Purely Uniformly Random Forests). Le principe de la construction d'un arbre est le suivant. Tout d'abord, on tire aléatoirement k variables de loi uniforme sur $[0,1]$. Ceci nous donne directement la partition de l'espace $\mathcal{X}$. On assigne ensuite

à chaque intervalle de notre partition une valeur grâce à $\mathcal{L}_n$. La structure d'arbre est alors perdue, car nous n'obtenons pas la partition de l'espace d'entrée de façon récursive. Nous gardons néanmoins le vocabulaire des arbres et forêt pour désigner les prédicteurs individuels et le prédicteur agrégé.

Dans cette famille de forêts aléatoires, nous pourrions encore penser à d'autres façons d'obtenir des partitions aléatoires, par exemple en changeant la loi des tirages de la variable de coupure, du point de coupure ou encore de la feuille à découpée. Cependant, la loi uniforme est ici naturelle, car vu que nous construisons les arbres sans prendre en compte l'échantillon d'apprentissage, nous n'avons a priori pas de raison de privilégier une variable ou un point de coupure en particulier.

Le principal intérêt de ce type de forêt est que le fait de tirer la partition de $\mathcal{X}$ indépendamment des données d'apprentissage facilite grandement l'analyse théorique des estimateurs obtenus. Comme nous le verrons dans la Section 1.4, cette indépendance entre partition de l'espace d'entrée et $\mathcal{L}_n$ paraît indispensable pour étudier théoriquement les forêts aléatoires.

1.3.3 Une forêt aléatoire intermédiaire

Nous clôturons cette section, en présentant une version de forêt aléatoire intermédiaire entre les forêts aléatoires classiques et les forêts purement aléatoires. Cette version a été introduite par Breiman (2004) et a été très récemment analysée par Biau (2010) (voir Section 1.4.3). Nous la qualifions d'intermédiaire, car elle préserve la propriété d'indépendance entre les coupures des noeuds et $\mathcal{L}_n$ des forêts purement aléatoires. Cependant, elle utilise un échantillon test pour déterminer ces coupures, ce qui la rapproche des forêts aléatoires classiques, car elle se sert bien de données pour choisir les coupures.

Notons $\mathcal{T}_n$ un échantillon test de même taille que l'échantillon d'apprentissage $\mathcal{L}_n$, i.e. une variable aléatoire indépendante et de même loi que $\mathcal{L}_n$. Le principe d'obtention d'un arbre de cette version intermédiaire est le suivant. Pour découper un noeud, on tire tout d'abord aléatoirement m variables. Pour chaque variable X^j , le point de coupure est alors fixé à la moitié du segment correspondant aux valeurs possibles de X^j (on découpe une partie de $\mathcal{X}$ au milieu de la longueur de sa j-ième composante). On choisit alors parmi les m coupures candidates, la coupure qui minimise la fonction de coût considérée, mais calculée maintenant sur l'échantillon $\mathcal{T}_n$.

1.4 Analyse théorique des forêts aléatoires

Dans cette section, nous recensons tous les résultats théoriques existants (à notre connaissance) sur les forêts aléatoires. Nous précisons, pour chaque étude, quelle variante de forêts est considérée.

Introduisons, tout d'abord, un peu de vocabulaire qui interviendra dans cette sec-

tion. Nous discuterons, en effet, beaucoup du biais et de la variance des estimateurs associés aux forêts aléatoires considérées. Plaçons-nous dans un cadre de régression et notons s la fonction de régression inconnue. Les notions de biais et variance diffèrent parmi les différentes études. Cependant, elles traduisent toujours la même idée : le biais est l'écart entre une approximation "idéale" et la fonction s ; la variance est la variation de l'estimateur autour de l'approximation idéale lorsqu'on change l'échantillon d'apprentissage. Pour le biais on parle souvent d'erreur d'approximation et pour la variance d'erreur d'estimation. L'approximation idéale ne dépend pas de l'échantillon $\mathcal{L}_n$. Elle est souvent définie comme la meilleur approximation que l'on pourrait construire suivant une certaine méthode si l'on connaissait la loi du couple (X, Y). Par exemple, un arbre idéal est un arbre pour lequel, au lieu d'assigner à un noeud la moyenne des Y_i des observations de ce noeud, nous assignons l'espérance conditionnelle de Y sachant que X appartient à la région de $\mathcal{X}$ définie par le noeud.

Précisons également ce que signifie le terme "forêt infinie" que certains auteurs utilisent. Reprenons les notations de la Définition 1 : une forêt aléatoire est l'agrégation d'une collection $\{\hat{h}(.,\Theta_1),\ldots,\hat{h}(.,\Theta_q)\}$ de q prédicteurs par arbre. En régression, par exemple, l'étape d'agrégation revient à faire la moyenne des arbres individuels. Faisons maintenant tendre q vers l'infini, nous obtenons une "forêt infinie", qui n'est autre que $\mathbb{E}[\hat{h}(.,\Theta)]$ par la loi des grands nombres (rappelons que $(\Theta_1,\ldots,\Theta_q)$ est une suite de variables i.i.d.). Cette dernière espérance porte uniquement sur la loi de Θ. Certains des résultats suivants sont établis pour une "forêt infinie" d'autre pour une forêt de q arbres. Cependant, la plupart des résultats s'avèrent vérifiés pour les deux notions de forêts. Enfin les différences qui existent en considérant une forêt finie ou infinie ne sont pas des différences cruciales dans l'étude théorique des forêts aléatoires.

1.4.1 Consistance de PRF et prolongements

Des résultats de consistance ont été établis par Biau et al. (2008). Ils concernent la méthode PRF dans un cadre de classification, la première forêt purement aléatoire évoquée en Section 1.3.2. Rappelons que k désigne le nombre de découpes effectuées (complètement aléatoirement) et n le nombre d'observations de l'échantillon d'apprentissage. Le résultat est le suivant :

Theorem 1. *Si* $k \xrightarrow[n\to+\infty]{} +\infty$ *et* $\frac{k}{n} \xrightarrow[n\to+\infty]{} 0$, *alors le classifieur donné par PRF est consistant.*

Ceci signifie que le classifieur PRF converge (lorsque $n \to +\infty$) en probabilité vers le classifieur de Bayes. Ce résultat est très général et les conditions sur k et n sont importantes. Nous les retrouverons régulièrement dans cette section. Concernant ces conditions, l'idée est qu'il faut beaucoup découper les arbres ($k \xrightarrow[n\to+\infty]{} +\infty$) pour réduire le biais, mais il faut également qu'il reste assez d'observations dans les feuilles des arbres $\left(\frac{k}{n} \xrightarrow[n\to+\infty]{} 0\right)$ pour contrôler la variance.

Pour montrer ce résultat, Biau et al. (2008) montre en réalité qu'un arbre purement aléatoire est consistant. Ils utilisent ensuite une propriété qui établit que la consistance se transmet d'un arbre aléatoire à une forêt aléatoire. Donc, du point de vue de la consistance, les deux estimateurs arbre et forêt sont performants. Mais ils sont également indiscernables : on ne voit pas d'amélioration apportée par la forêt comparé à l'arbre.

Le Chapitre 4 se situe dans le prolongement naturel de ces résultats de consistance. En effet, nous étudions les vitesses de convergence des estimateurs associés à un arbre et à une forêt. Cependant cette étude n'est pas faite pour PRF, mais pour une autre variante de forêt purement aléatoire, appelée PURF (décrite en Section 1.3.2). De plus, nous étudions cette variante, dans un cadre de régression, avec une seule variable d'entrée ($\mathcal{X} = [0,1]$). Le cadre est simplifié en vue de donner une analyse fine du risque des estimateurs. Nous prouvons alors que les estimateurs arbre et forêt atteignent tous deux la vitesse minimax de convergence sur la classe des fonctions Lipschitziennes. De plus, nous montrons que la variance de la forêt est majorée par la variance de l'arbre multipliée par 3/4. Ceci illustre donc une amélioration apportée par la forêt. La forêt conserve donc la vitesse de convergence atteinte par un arbre, et de plus a un effet de réduction de variance.

Signalons enfin que dans nos majorations du risque, apparaissent un terme de variance en k/n et un terme de biais en $1/k^2$. On retrouve donc bien des résultats de consistance en supposant que $k \xrightarrow[n\to+\infty]{} +\infty$ et $\frac{k}{n} \xrightarrow[n\to+\infty]{} 0$ (en régression, la consistance signifie que le risque de l'estimateur tend vers 0).

1.4.2 Lien avec les estimateurs de plus proches voisins et à noyaux

Lien avec les estimateurs de plus proches voisins

Lin and Jeon (2006) établissent un lien entre les forêts aléatoires et les estimateurs des plus proches voisins. La notion de voisin est alors la notion de Layered Nearest Neighbors (LNN) : une observation X_i de $\mathcal{L}_n$ est un LNN d'un point x de $\mathcal{X}$ fixé, si l'hyperrectangle défini par x et X_i ne contient pas d'autre observation de $\mathcal{L}_n$. L'idée de cette correspondance est que si on considère un arbre aléatoire pleinement développé (c'est-à-dire qu'il ne reste qu'une observation dans chaque feuille de l'arbre), alors le X_i associé à la feuille dans laquelle tombe un point x lorsqu'on le fait descendre dans l'arbre, est un LNN de x. En effet, étant donné que nous nous limitons à des coupures parallèles aux axes, les feuilles de nos arbres définissent des hyperrectangles. Et le fait qu'on considère un arbre pleinement développé entraîne qu'il n'y a pas d'observation autre que X_i dans la feuille associée à X_i. Lors d'une deuxième réalisation de l'arbre aléatoire, il peut arriver que x tombe alors dans la feuille associée à une autre observation X_j dans le nouvel arbre. L'estimateur des forêts aléatoires est alors vu comme une estimateur LNN à poids.

Biau and Devroye (2010), dans un article mathématiquement plus abouti, montre la consistance de estimateurs LNN à poids. De plus, ils prouvent que pour une forêt d'arbres purement aléatoires, complètement développés et obtenus sans ré-échantillonnage préalable, le risque quadratique est minoré par $\frac{1}{(\log n)^{p-1}}$. Cette faible vitesse de convergence peut être améliorée en arrêtant de découper un arbre, par exemple, quand un noeud contient moins qu'un certain nombre d'observations. Elle peut également être améliorée en rajoutant une étape de ré-échantillonnage, comme dans le Bagging. Cependant ce résultat remet en question le fait d'utiliser toujours des arbres pleinement développés dans les méthodes de forêts aléatoires classiques. Des améliorations sont, en théorie, possibles en construisant des arbres moins profonds via une certaine règle d'arrêt de découpage ou une étape d'élagage.

Lien avec les estimateurs à noyaux

Initialement introduit par Breiman (2000b) pour PRF, le lien entre forêts aléatoires et estimateurs à noyaux est explicité plus généralement dans Geurts et al. (2006). Nous exposons ici ce lien dans un cadre de régression.

Soit un arbre $\hat{s}_l$, nous introduisons les fonctions caractéristiques associées aux feuilles de $\hat{s}_l$. Pour A_r la r-ième feuille de $\hat{s}_l$, la fonction $x \mapsto \mathbb{1}_{l,r}(x)$ vaut 1 si $x \in A_r$ et 0 sinon. Notons alors $n_{l,r}$ le nombre d'observations de $\mathcal{L}_n$ inclues dans la feuille A_r : $n_{l,r} = \sum_{i=1}^{n} \mathbb{1}_{l,r}(X_i)$. Définissons enfin, le vecteur des fonctions caractéristiques normalisées de $\hat{s}_l$:

$$\mathbb{1}_l^{R_l}(x) = \left(\frac{\mathbb{1}_{l,1}(x)}{\sqrt{n_{l,r}}}, \ldots, \frac{\mathbb{1}_{l,R_l}(x)}{\sqrt{n_{l,R_l}}} \right)^T$$

R_l désignant le nombre de feuille de l'arbre $\hat{s}_l$. La notation v^T désigne le vecteur v transposé.

On peut alors écrire $\hat{s}_l$ comme un estimateur à noyaux :

$$\hat{s}_l(x) = \sum_{i=1}^{n} K_l(X_i, x) Y_i$$

où

$$K_l(x', x)(\mathbb{1}_l^{R_l}(x))^T \mathbb{1}_l^{R_l}(x)$$

désigne le noyau associé à l'arbre $\hat{s}_l$.

Le noyau associé à une forêt de q arbres est donc défini par :

$$K(x', x) = \frac{1}{q} \sum_{l=1}^{q} K_l(x', x) \ .$$

On a finalement,

$$\hat{s}(x) = \frac{1}{q}\sum_{l=1}^{q}\hat{s}_l(x) = \sum_{i=1}^{n} K(X_i, x)Y_i \ .$$

Geurts et al. (2006) montrent que lorsque q tend vers $+\infty$, le noyau $K(x', x)$ d'une forêt infine d'Extra-Trees est continue et bilinéaire.

Elias (2009) adopte également ce point de vue d'estimateurs à noyaux dans son étude des forêts aléatoires. Il montre que l'application $d(x', x) = 1 - K(x', x)$ définit une pseudo-distance de l'espace $\mathcal{X}$.

1.4.3 Analyse d'une forêt aléatoire intermédiaire

Biau (2010) analyse le modèle de forêt aléatoire intermédiaire (entre forêts aléatoires classiques et forêts purement aléatoires) introduit en Section 1.3.3. Il introduit tout d'abord une variante théorique des forêts aléatoires où, à chaque noeud, chaque variable X_j a une probabilité p_{nj} d'être choisie. Précisons que les probabilités p_{nj} sont indépendantes de l'échantillon d'apprentissage $\mathcal{L}_n$. Cette méthode diffère des méthodes introduites précédemment pour lesquelles le choix des variables à chaque noeud était systématiquement fait de façon uniforme sur l'ensemble des p variables. La coupure est ensuite réalisé au milieu du noeud considéré, le long de la variable choisie. A chaque étape, toutes les feuilles de l'arbre sont découpées, et on réalise $\log_2 k$ étapes.

Le premier résultat est la consistance de ces forêts :

Theorem 2. *Si pour tout $j \in \{1, \ldots, p\}$, $p_{nj}\log k \xrightarrow[n\to+\infty]{} +\infty$ et si $\dfrac{k}{n} \xrightarrow[n\to+\infty]{} 0$, alors l'estimateur associé à ces forêts aléatoires est consistant.*

Le deuxième résultat établit que l'estimateur s'adapte à la parcimonie, au sens où sa vitesse de convergence dépend seulement du nombre de "vraies" variables et non du nombre de variables de bruit. L'hypothèse est que la fonction de régression est en réalité une fonction d'uniquement S vraies variables, les $p - S$ variables restantes étant des variables de bruit. Plus précisément, le résultat obtenu est le suivant :

Theorem 3. *Si le poids de chaque vraie variable tend (à une certaine vitesse) vers $1/S$ quand n tend vers $+\infty$, et le poids de chaque variable de bruit vers 0, alors la vitesse de convergence de l'estimateur est de l'ordre de*

$$n^{\frac{-0.75}{S\log 2 + 0.75}} \ .$$

Ainsi, si une méthode forêt aléatoire parvient à régler convenablement les poids pour qu'ils se concentrent assez rapidement sur les vraies variables, alors l'estimateur a une vitesse de convergence qui ne dépend que du nombre de vraies variables S. Ceci aide à comprendre les performances exceptionnelles, par exemple, de l'algorithme Random

Forests-RI sur des données de grande dimension (où $n << p$). En effet, on peut conjecturer que la règle de découpe des Random Forests-RI (i.e. choix d'un nombre m de variables, puis sélection de la coupure qui minimise une certaine fonction de coût sur $\mathcal{L}_n$) parvient à se focaliser rapidement sur les vraies variables (pour une valeur de m bien choisie).

Cependant, les Random Forests-RI ne rentrent pas dans le cadre de la variante théorique de forêt étudiée ici, car elles utilisent $\mathcal{L}_n$ pour déterminer la variable de coupure à chaque noeud. Biau (2010), dans la dernière section de l'article, donne, par contre, des éléments pour justifier que la forêt aléatoire intermédiaire décrite en Section 1.3.3 évalue correctement les poids et donc satisfait les conditions du Théorème 3. Dans cette forêt intermédiaire, la variable de coupure d'un noeud est sélectionnée en utilisant un échantillon test indépendant de $\mathcal{L}_n$. Elle rentre bien dans le cadre de la variante théorique étudiée plus haut. De plus, le fait de mettre en compétition m variables à chaque noeud pousse la forêt à se focaliser uniquement sur les vraies variables (pour une bonne valeur de m).

1.4.4 Analyse du Bagging

Nous évoquons maintenant un article sur l'analyse du Bagging. Bühlmann and Yu (2002) détaillent, en dimension 1 ($\mathcal{X} = \mathbb{R}$), la convergence des trois paramètres qui définissent un arbre à deux feuilles (le point de coupure et les deux valeurs assignées à chacune des feuilles). Cet arbre à deux feuilles, appelé stump, est obtenu en suivant exactement la première étape de l'algorithme CART. En régression, on cherche alors le point de coupure qui minimise la variance des deux feuilles obtenues. En dimension 1, un stump est donc représenté par une fonction constante par morceaux, à deux morceaux. Les valeurs assignées à chaque feuille, sont alors appelées les hauteurs de palier.

Ils montrent alors que le point de coupure converge à vitesse $n^{1/3}$ vers une variable aléatoire W. Cependant, pour cela, ils supposent que les deux hauteurs de palier convergent à vitesse $n^{1/2}$, plus rapide. Cette hypothèse, qui est vérifiée lorsque le point de coupure est déterministe, est erronée dans notre cas où le point de coupure est également une variable aléatoire et où les trois paramètres sont reliés. En effet, Banerjee and McKeague (2007) prouvent que les hauteurs de palier convergent également à la vitesse $n^{1/3}$. Néanmoins, l'analyse du Bagging de Bühlmann and Yu (2002) reste valable, et la variable aléatoire limite W est correcte. W est définie comme le lieu du maximum d'un mouvement brownien à dérive parabolique. Cette variable limite, dont nous n'avons pas d'écriture explicite est alors difficile à manier, et les auteurs continuent leur analyse grâce à des simulations.

Bühlmann and Yu (2002) tentent ensuite d'étudier la version bootstrap d'un stump : on tire un échantillon bootstrap de $\mathcal{L}_n$, puis on construit un stump sur l'échantillon obtenu. Ils montrent que si le "bootstrap marche", le Bagging a un effet de réduction de variance. On dit que le bootstrap marche lorsque la version bootstrap converge à la même vitesse et vers la même limite que l'estimateur initial (celui construit sur $\mathcal{L}_n$).

Après avoir laisser la question du bootstrap classique des stumps comme une question ouverte, Bühlmann and Yu (2002) étudient une variante où les échantillons bootstrap sont tirées sans remise. Ils illustrent alors l'effet de réduction de variance de la variante du Bagging considérée.

Avec Cécile Durot, nous avons exploré la question du bootstrap classique pour les stumps. Grâce à sa connaissance des processus convergeant à vitesse $n^{1/3}$ vers des variables aléatoires du type de W (voir Durot (2008)), Cécile Durot conjecture que le bootstrap classique ne marche effectivement pas dans ce cas. Les techniques de Bühlmann and Yu (2002) ne peuvent alors plus être mises en oeuvre. Cette approche sur l'étude de la convergence des paramètres définissant les arbres aléatoires demeure intéressante, mais paraît très difficile à poursuivre.

1.4.5 Remarques générales

Nous terminons cette section par évoquer quelques remarques générales que l'on trouve dans la littérature.

Un fait général est que les forêts aléatoires ont un effet de réduction de variance, quand on les compare à des arbres seuls (aléatoires ou non). Cette remarque est très importante et est quasiment présente dans tous les articles qui s'intéressent aux méthodes d'ensemble de type forêts aléatoires. Une étude par simulation menée par Geurts et al. (2006) illustre très bien ce phénomène de réduction de variance. Cette réduction apparaît systématique et est très souvent spectaculaire.

La compréhension de l'effet qu'ont les forêts aléatoires sur le biais est moins claire. Pour certains les forêts réduisent également le biais : Dietterich (2000) justifie ceci par le fait qu'en construisant une forêt on augmente la famille de prédicteurs explorés et qu'on a donc une meilleure capacité d'approximation, Biau (2010) montre que la vitesse de convergence du biais d'une forêt est supérieure à la vitesse du biais d'estimateurs obtenus par partitionnement ordinaires. Pour d'autres, le fait de randomiser un arbre, augmente son biais, et l'étape d'agrégation laisse le biais inchangé, donc au final le biais augmente lorsque l'on passe d'un arbre non randomisé à un forêt aléatoire. C'est pourquoi, ils conseillent de développer les arbres randomisés individuels jusqu'au bout, pour tenter de construire des prédicteurs individuels avec le biais le plus petit possible, avant de les agréger (c.f. Hastie et al. (2009)). Dans leur étude par simulation, Geurts et al. (2006) illustre le fait que le biais augmente lors de l'étape de randomisation et reste ensuite inchangé après agrégation.

De façon générale, les études par simulation montrent que comparées à des arbres CART seuls, les forêts aléatoires apportent une amélioration plus forte en classification qu'en régression (voir par exemple Breiman (2001)). Geurts et al. (2006) justifient cette observation par le fait qu'en classification on peut bien prédire sans forcément bien estimer (voir le paragraphe concernant la classification de la Section 1.1.1). Un arbre randomisé augmente un peu le biais. Mais si cette augmentation du biais est effecti-

vement faible, elle n'altère quasiment pas l'erreur de généralisation. Par suite, comme l'agrégation n'augmente pas le biais mais diminue fortement la variance, l'erreur de la forêt aléatoire est très inférieure à celle d'un arbre non-randomisé. Et l'amélioration apportée par les forêts en régression est moindre, car, dans ce cadre, l'augmentation du biais influe forcément sur l'erreur de généralisation d'un arbre randomisé, et donc sur celle de la forêt aléatoire.

Une autre remarque générale dont nous avons déjà parlé est que pour être performante une forêt doit être composée d'arbres très différents les uns des autres. Breiman (2001) introduit la corrélation entre deux arbres et montre que réduire cette corrélation induit une réduction de l'erreur de prédiction. Il introduit également la force d'un arbre (une sorte de mesure de la qualité d'un arbre individuel) et montre que l'erreur de prédiction diminue lorsque cette force augmente. La majoration de Breiman (2001) de l'erreur de prédiction semble assez grossière, mais les idées qu'elle traduit sont très importantes dans la compréhension des forêts aléatoires : une forêts d'arbres individuellement performants et différents les uns des autres est une forêt performante.

Après cette présentation, nous donnons une courte description de la suite de la thèse.

Dans le Chapitre 2, nous présentons une étude méthodologique de la méthode des forêts aléatoires. Nous étudions le comportement de l'indice d'importance des variables, et présentons une procédure de sélection de variables basée sur les forêts aléatoires.
Le travail de ce chapitre a été réalisé en collaboration avec Jean-Michel Poggi et Christine Tuleau. Il a fait l'objet d'un article à paraître dans le journal *Pattern Recognition Letters*. La référence Genuer et al. (2010a) désigne cet article tout au long de la thèse. Le contenu de l'annexe ne figure pas dans l'article, mais dans le rapport de recherche Genuer et al. (2008).

Le Chapitre 3 est dédié à l'application sur des données réelles de la procédure de sélection de variables introduites au Chapitre 2. La première application concerne des données de neuroimagerie, alors que la deuxième traite des données de génétique.
La première section de ce chapitre est le fruit d'une collaboration avec Vincent Michel, Evelyn Eger et Bertrand Thirion du CEA Neurospin. Ce travail (Genuer et al., 2010b) est publié dans les actes de la conférence avec comité de lecture *COMPSTAT'2010*.
La suite du chapitre concerne une collaboration avec Wilson Toussile et Isabelle Morlais, qui a donné lieu à un article actuellement soumis.

Dans le Chapitre 4, nous présentons des résultats théoriques obtenus sur une variante simple de forêts aléatoires. Nous démontrons que les arbres et les forêts aléatoires atteignent la vitesse minimax de convergence, et prouvons une réduction de variance apportée par les forêts.
Le contenu de ce chapitre a donné lieu à un article (Genuer, 2010c) qui est actuellement soumis.

Chapitre 2

Forêts aléatoires : aspects méthodologiques

Sommaire

Résumé

Ce chapitre étudie de façon méthodologique la méthode des forêts aléatoires introduite par Breiman (2001). Tout d'abord, nous étudions le comportement de l'indice d'importance des variables des forêts aléatoires sur des simulations. Nous nous intéressons au comportement de

cet indice vis-à-vis du nombre de variables de bruit des données, vis-à-vis des paramètres de la méthode, ainsi qu'à son comportement en présence de groupes variables très corrélées entre elles.

Ensuite, nous mettons au point une procédure automatique de sélection de variables entièrement basée sur les forêts aléatoires. Une première étape consiste à classer les variables par ordre décroissant d'importance, puis à éliminer les variables de faible importance. Une deuxième étape compare alors les erreurs de généralisation de modèles de forêts aléatoires emboîtés (à chaque pas nous rajoutons une variable dans le modèle) et sélectionne le modèle réalisant l'erreur la plus faible. Le sous-ensemble de variables obtenu à cette étape est appelée sous-ensemble d'interprétation, car il tente de fournir toutes les variables reliés à la variable réponse (même si celles-ci sont corrélées entre elles). Une dernière étape a pour but de trouver un petit sous-ensemble de variables, suffisant pour bien prédire la variable réponse, que nous appelons sous-ensemble de prédiction. Cette étape n'ajoute (à chaque pas) une variable dans le modèle que si elle fait suffisamment diminuer l'erreur de généralisation. Nous appliquons cette procédure sur des données simulées ainsi que sur des données réelles de biopuces et comparons les résultats avec d'autres méthodes existantes.

L'annexe du chapitre contient une étude par simulation du comportement de l'erreur de généralisation des forêts aléatoires en fonction des principaux paramètres de la méthode. Nous faisons une étude systématique illustrée par des graphiques sur de nombreuses données de références, de petite et de grande dimension, dans des cadres de classification et de régression. Nous donnons enfin quelques conseils sur le réglage des paramètres des forêts aléatoires, en fonction du type de données traitées.

Le travail de ce chapitre a été réalisé en collaboration avec Jean-Michel Poggi et Christine Tuleau. Les sections 2.1 *à* 2.5 *font l'objet d'un article (Genuer et al., 2010a) à paraître dans le journal* Pattern Recognition Letters. *Le contenu de l'annexe* 2.A *ne figure pas dans l'article, mais dans le rapport de recherche Genuer et al. (2008).*

2.1 Introduction

This paper is primarily interested in random forests for variable selection. Mainly methodological the main contribution is twofold : to provide some experimental insights about the behavior of the variable importance index based on random forests and to use it to propose a two-steps algorithm for two classical problems of variable selection starting from variable importance ranking. The first problem is to find important variables for interpretation and the second one is more restrictive and try to design a good

parsimonious prediction model. The general strategy involves a ranking of explanatory variables using the random forests score of importance and a stepwise ascending variable introduction strategy. Let us mention that we propose an heuristic strategy which does not depend on specific model hypotheses but based on data-driven thresholds to take decisions.

Before entering into details, we introduce the three main topics of this paper (random forests, variable importance, variable selection) and we sketch a typical high dimensional classification problem motivating this work.

Random forests

Random forests (RF henceforth) is a popular and very efficient algorithm, based on model aggregation ideas, for both classification and regression problems, introduced by Breiman (2001). It belongs to the family of ensemble methods, appearing in machine learning at the end of nineties (see for example Dietterich (1999) and Dietterich (2000)). Let us briefly recall the statistical framework by considering a learning set $L = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ made of n i.i.d. observations of a random vector (X, Y). Vector $X = (X^1, ..., X^p)$ contains predictors or explanatory variables, say $X \in \mathbb{R}^p$, and $Y \in \mathcal{Y}$ where $\mathcal{Y}$ is either a class label or a numerical response. For classification problems, a classifier t is a mapping $t : \mathbb{R}^p \to \mathcal{Y}$ while for regression problems, we suppose that $Y = s(X) + \varepsilon$ with $E[\varepsilon|X] = 0$ and s the so-called regression function (for more background on statistical learning, see e.g. Hastie et al. (2009)). Random forests is a model building strategy providing estimators of either the Bayes classifier, which is the mapping minimizing the classification error $P(Y \neq t(X))$, or the regression function.

The principle of random forests is to combine many binary decision trees built using several bootstrap samples coming from the learning sample L and choosing randomly at each node a subset of explanatory variables X. More precisely, with respect to the well-known CART [1] model building strategy (see Breiman et al. (1984)) performing a growing step followed by a pruning one, two differences can be noted. First, at each node, a given number (denoted by $mtry$) of input variables are randomly chosen and the best split is calculated only within this subset. Second, no pruning step is performed so all the trees of the forest are maximal trees.

In addition to CART, bagging, another well-known related tree-based method, is to be mentioned (see Breiman (1996)). Indeed random forests with $mtry = p$ reduce simply to unpruned bagging. The associated R packages [2] are respectively `randomForest` (intensively used in the sequel of the paper), `rpart` and `ipred` for CART and bagging respectively (cited here for the sake of completeness).

RF algorithm becomes more and more popular and appears to be very powerful in a lot of different applications (see for example Díaz-Uriarte and Alvarez de Andrés (2006) for gene expression data analysis) even if it is not clearly elucidated from a mathematical

1. Classification And Regression Trees
2. see http ://www.r-project.org/

point of view (see the recent paper by Biau et al. (2008) about purely random forests and Bühlmann and Yu (2002) about bagging). Nevertheless, Breiman (2001) sketches an explanation of the good performance of random forests related to the good quality of each tree (at least from the bias point of view) together with the small correlation among the trees of the forest, where the correlation between trees is defined as the ordinary correlation of predictions on so-called out-of-bag (OOB henceforth) samples. The OOB sample is the set of observations which are not used for building the current tree. It is used to estimate the prediction error and then to evaluate variable importance.

The R package about random forests is based on the seminal contribution of Breiman and Cutler (2005) and is described in Liaw and Wiener (2002). In this paper, we focus on the `randomForest` procedure. The two main parameters are *mtry*, the number of input variables randomly chosen at each split and *ntree*, the number of trees in the forest. Some details about numerical and sensitivity experiments can be found in Appendix 2.A.

In addition, we will concentrate on the prediction performance of RF focusing on out-of-bag (OOB) error (see Breiman (2001)). We use this kind of prediction error estimate for three reasons : the main is that we are mainly interested in comparing models instead of assessing models, the second is that it gives fair estimation compared to the usual alternative test set error even if it is considered as a little bit optimistic and the last one, is that it is a default output of the `randomForest` procedure, so it is used by almost all users.

Variable importance

The quantification of the variable importance (VI henceforth) is an important issue in many applied problems complementing variable selection by interpretation issues.

In the random forests framework, the most widely used score of importance of a given variable is the increasing in mean of the error of a tree (mean square error (MSE) for regression and misclassification rate for classification) in the forest when the observed values of this variable are randomly permuted in the OOB samples (it could be slightly negative). Often, such random forests VI is called permutation importance indices in opposition to total decrease of node impurity measures already introduced in the seminal book about CART by Breiman et al. (1984).

For regression problems, two other measures of VI are to be mentioned. In the linear regression framework it is examined for example by Grömping (2007), making a distinction between various variance decomposition based indicators : "dispersion importance", "level importance" or "theoretical importance" quantifying explained variance or changes in the response for a given change of each regressor. An extension to nonlinear regression models focusing on the input-output analysis (avoiding model estimation) is provided by Sobol sensitivity indices (Sobol' (1993)).

A comparison between RF variable importance and linear regression based importance indices has been carried out by Grömping (2009) and the conclusion is that the results are in agreement. A preliminary comparison with sensitivity indices (not repor-

ted here) lead to the same conclusion. A more intensive comparison could be of interest but it is out of the scope of this paper which focuses on RF. We emphasize that, due to the versatility of the RF framework, RF variable importance can be computed for standard ($n >> p$) or high dimensional ($n << p$) problems, as well as for classification or regression problems. In addition, the computational burden is acceptable.

Even if only little investigation is available about RF variable importance, some interesting facts are collected for classification problems when this index is based on the average loss of heterogeneity criterion, derived for example from the Gini impurity function used for growing classification trees. First, the RF Gini importance is not fair in favor of predictor variables with many categories (see Strobl et al. (2007)) while the RF permutation importance is a more reliable indicator. So we restrict our attention to this last one. Then, it seems that permutation importance overestimates the variable importance of highly correlated variables and a conditional variant is proposed by Strobl et al. (2008). In this paper, we do not diagnose such a critical phenomenon for variable selection. Finally, the recent paper by Archer and Kimes (2008), focusing more specifically on the VI topic is also of interest. We give some experimental insights about variable importance behavior in presence of groups of highly correlated variables. This is the first goal of this paper.

Variable selection

Many variable selection procedures are based on the cooperation of variable importance for ranking and model estimation to generate, evaluate and compare a family of models. Following Kohavi and John (1997) and Guyon et al. (2003), it is usual to distinguish three types of variable selection methods : "filter" for which the score of variable importance does not depend on a given model design method ; "wrapper" which include the prediction performance in the score calculation ; and finally "embedded" which combine more closely variable selection and model estimation.

Let us briefly mention some of them, in the classification case, which are potentially competing tools : of course the wrapper methods based on VI coming from CART, and from random forests. Then some examples of embedded methods : Poggi and Tuleau (2006) propose a method based on CART scores and using stepwise ascending procedure with elimination step ; Guyon et al. (2002) and Rakotomamonjy (2003), propose methods based on Support Vector Machines (SVM) scores and using descending elimination. More recently, Ben Ishak and Ghattas (2008) propose a stepwise variant while Park and Hastie (2007) propose a LARS [3] type strategy (see Efron et al. (2004)) for classification problems. Finally we mention a mixed approach, see Fan and Lv (2008) in regression, ascending in order to avoid to select redundant variables or, for the case $n << p$, descending first using a screening procedure to reach a classical situation $n \sim p$, and then ascending using LASSO [4] or SCAD [5], see Fan and Li (2001). We propose in this

3. Least Angle Regression
4. Least Absolute Shrinkage and Selection Operator
5. Smoothly Clipped Absolute Deviation

paper, a two-steps procedure, the second one depends on the objective (interpretation or prediction) while the first one is common. The key point is that it is entirely based on random forests, so fully non parametric and then free from the usual linear framework.

A typical situation

Let us close this section by introducing a typical situation which can be useful to capture the main ideas of this paper. We consider a high dimensional ($n << p$) classification problem for which the predictor variables are associated to a pixel in an image or a 3D location in the brain like in fMRI brain activity classification problems. In such situations, of course it is clear that there is a lot of useless variables and that there exists unknown groups of highly correlated predictors corresponding to brain regions. We emphasize that two distinct objectives about variable selection can be identified : (1) to find important variables highly related to the response variable for interpretation purpose ; (2) to find a small number of variables sufficient for a good prediction of the response variable. Key tools combine variable importance thresholding, variable ranking and stepwise introduction of variables. Turning back to our typical situation, an example of the first kind of problem is the determination of entire regions in the brain or a full parcel in an image while an instance of the second one is to exhibit a parsimonious subset of the most discriminant variables within the previously highlighted groups.

Outline

The paper is organized as follows. After this introduction, Section 2.2 illustrates RF variable importance behavior, especially in presence of groups of highly correlated explanatory variables. Section 2.3 proposes an ascending procedure for two classical variable selection problems starting from an initial ranking based on the random forests score of importance. Section 2.4 examines some experimental results, by focusing mainly on high dimensional classification datasets and, in order to illustrate the general value of the strategy, it is applied to a standard ($n >> p$) regression dataset. Finally Section 2.5 opens discussion about future work.

2.2 Variable importance

The quantification of the variable importance is a crucial issue not only for ranking the variables before a stepwise estimation model but also to interpret data and understand underlying phenomenons in many applied problems.

RF variable importance of X^j is defined as follows. For each tree t of the forest, consider the associated OOB_t sample (data not included in the boostrap sample used to construct t). Denote by $errOOB_t$ the error (MSE for regression and misclassification rate for classification) of a single tree t on this OOB_t sample. Now, randomly permute the values of X^j in OOB_t to get a perturbed sample denoted by $\widetilde{OOB_t}^j$ and compute

$\widetilde{errOOB_t}^j$, the error of predictor t on the perturbed sample. Variable importance of X^j is then equal to :

$$VI(X^j) = \frac{1}{ntree} \sum_t (\widetilde{errOOB_t}^j - errOOB_t)$$

where the sum is over all trees t of the RF and *ntree* denotes the number of trees of the RF.

In this section, we examine the RF variable importance behavior according to three different issues. The first one deals with the sensitivity to the sample size n and the number of variables p. The second examines the sensitivity to method parameters *mtry* and *ntree*. This is of interest since a good choice of parameters of RF can help to better discriminate between important and useless variables. In addition, it can increase the stability of VI scores. The third one deals with the variable importance in presence of groups of highly correlated variables.

To illustrate this discussion, we examine a simulated dataset for the case $n << p$, introduced by Weston et al. (2003) and called "toys data" in the sequel. It is an equiprobable two-class problem, $Y \in \{-1, 1\}$, with 6 true variables, the others being some noise. This example is interesting since it constructs two independent groups of 3 significant variables (highly, moderately and weakly correlated with response Y) and an additional group of noise variables, uncorrelated with Y. A forward reference to the plots on the left side of Figure 2.1 allows to see the variable importance picture : importances of variables 1 to 3 are higher than the ones of variables 4 to 6, i.e. $VI(X^j) > VI(X^{j+3})$ for $j = 1, 2, 3$. The simulation model is defined through the conditional distribution of the X^i for $Y = y$:

- For the six first variables : with probability 0.7, $X^i \sim \mathcal{N}(yi, 1)$ for $i = 1, 2, 3$ and $X^i \sim \mathcal{N}(0, 1)$ for $i = 4, 5, 6$; with probability 0.3, $X^i \sim \mathcal{N}(0, 1)$ for $i = 1, 2, 3$ and $X^i \sim \mathcal{N}(y(i-3), 1)$ for $i = 4, 5, 6$.
- Remaining variables are noise : $X^i \sim \mathcal{N}(0, 1)$ for $i = 7, \ldots, p$.

After simulation, the obtained variables are finally standardized.

Remark 1. *Variable importance is computed conditionally to a given realization even for simulated datasets. This choice which is criticizable if the objective is to reach a good estimation of an underlying constant, is consistent with the idea of staying as close as possible to the experimental situation dealing with a given dataset.*

2.2.1 Sensitivity to n and p

Figure 2.1 illustrates the behavior of variable importance for several values of n and p. Parameters *ntree* and *mtry* are set to their default values ($ntree = 500$ and $mtry = \sqrt{p}$ for the classification case). Boxplots are based on 50 runs of the RF algorithm and for visibility, we plot the variable importance only for a few variables.

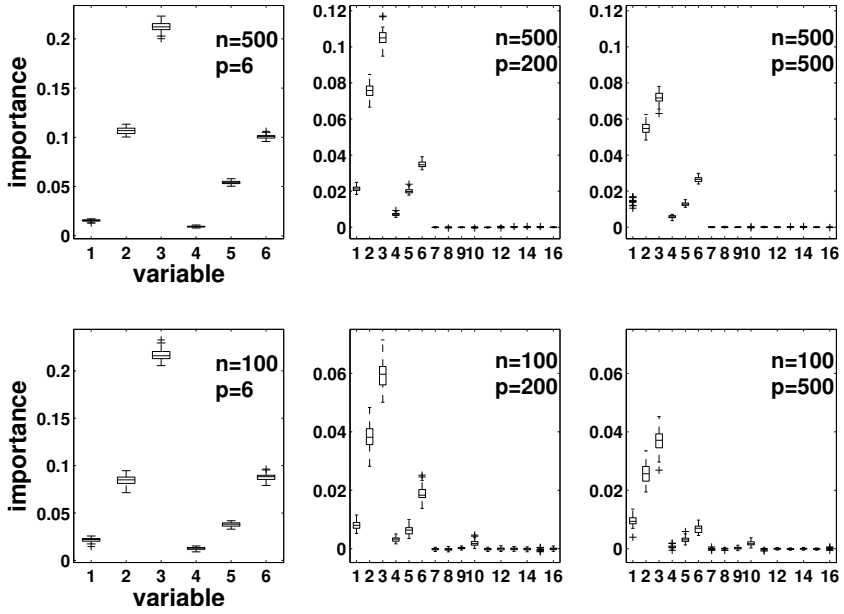

FIGURE 2.1 – Variable importance sensitivity to n and p (toys data)

On each row, the first plot is the reference one for which we observe a convenient picture of the relative importance of the initial variables. Then, when p increases tremendously, we try to check if : (1) the situation between the two groups remains readable ; (2) the situation within each group is stable ; (3) the importance of the additional noise variables is close to 0.

The situation $n = 500$ (graphs at the top of the figure) corresponds to an "easy" case, where a lot of data are available and $n = 100$ (graphs at the bottom) to a harder one. For each value of n, three values of p are considered : $6, 200$ and 500. When $p = 6$ only the 6 true variables are present. Then two very difficult situations are considered : $p = 200$ with a lot of noisy variables and $p = 500$ is even harder. We consider such very high values for p to mimic the situation of very high dimensional real data considered in the sequel. Graphs are truncated after the 16th variable for readability (importance of noisy variables left are of the same order of magnitude as the last plotted).

Let us comment on graphs on the first row ($n = 500$). When $p = 6$ we obtain concentrated boxplots and the order is clear, variables 2 and 6 having nearly the same importance. When p increases, the order of magnitude of importance decreases (note that the y-axis scale is different for $p = 6$ and for $p \neq 6$). The order within the two groups of variables (1-3 and 4-6) remains the same, while the overall order is modified (variable 6 is now less important than variable 2). In addition, variable importance is more unstable for huge values of p. But what is remarkable is that all noisy variables have a zero VI. So one can easily recover variables of interest.

In the second row ($n = 100$), we note a greater instability since the number of observations is only moderate, but the variable ranking remains quite the same. What

differs is that in the difficult situations ($p = 200, 500$) importance of some noisy variables increases, and for example variable 4 cannot be distinguished from noise. The same holds even for variable 5 for $p = 500$. This is due to the decreasing behavior of VI with p growing, coming from the fact that when $p = 500$ the algorithm randomly choose only 22 variables at each split (with the *mtry* default value). The probability of choosing one of the 6 true variables is really small and the less a variable is chosen, the less it can be considered as important. We will see the benefits of increasing *mtry* in the next paragraph.

In addition, it should be noted that the variability of VI is large for true variables with respect to useless ones. This remark can be used to build some kind of test for VI (see Strobl et al. (2007)) but of course ranking is better suited for variable selection.

We now study how this VI index behaves when changing values of the main method parameters.

2.2.2 Sensitivity to *mtry* and *ntree*

The choice of *mtry* and *ntree* can be important for the VI computation. We fix $n = 100$ and $p = 200$ and, in Figure 2.2, we plot variable importance obtained using three values of *mtry* (14 the default, 100 and 200) and two values of *ntree* (500 the default, and 2000).

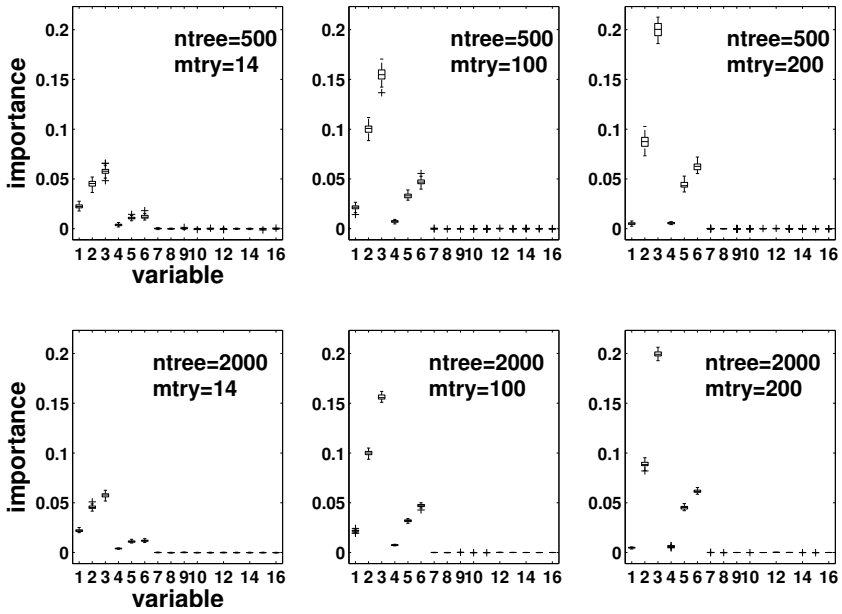

FIGURE 2.2 – Variable importance sensitivity to *mtry* and *ntree* (toys data)

The effect of taking a larger value for *mtry* is obvious. Indeed the magnitude of VI is more than doubled starting from $mtry = 14$ to $mtry = 100$, and it again increases

whith $mtry = 200$. The effect of $ntree$ is less visible, but taking $ntree = 2000$ leads to better stability. What is difficult to see but interesting in the bottom right graph is that we get the same order for all true variables in every run of the procedure.

2.2.3 Sensitivity to highly correlated predictors

We now illustrate an important issue : how does variable importance behave in presence of several highly correlated variables ? We take as basic framework the previous context with $n = 100$, $p = 200$, $ntree = 2000$ and $mtry = 100$. Then we add to the dataset highly correlated replications of some of the 6 true variables.

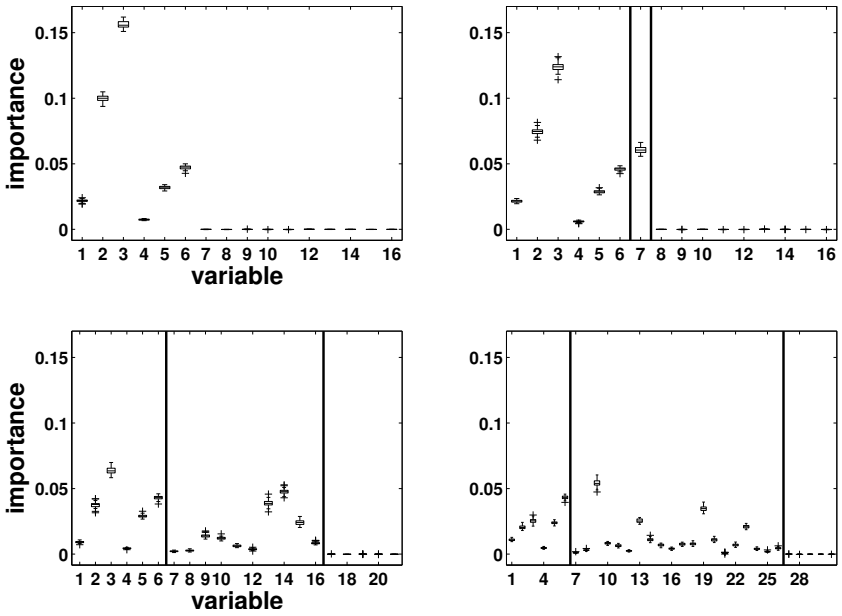

FIGURE 2.3 – Variable importance in presence of a group of correlated variables (augmented toys data)

The first graph of Figure 2.3 is the reference one : the situation is the same as previously. Then for the three other cases, we simulate 1, 10 and 20 variables with a correlation of 0.9 with variable 3 (the most important one). These replications are plotted between the two vertical lines.

VIs in the group $1, 2, 3$ are steadily decreasing when adding more replications of variable 3. On the other hand, VIs in the group $4, 5, 6$ are unchanged. Notice that the importance is not divided by the number of replications. Indeed in our example, even with 20 replications the maximum of VIs in the group containing variable 3 (that is variables 1 to 3 and all replications of variable 3) is only three times lower than the initial VI of variable 3. Finally, note that even if some variables in this group have low importance, they cannot be confused with noise.

Let us briefly comment on similar experiments (see Figure 2.4) obtained by perturbing the basic situation not only by introducing highly correlated versions of the third variable but also of the sixth, leading to replicate the most important of each group.

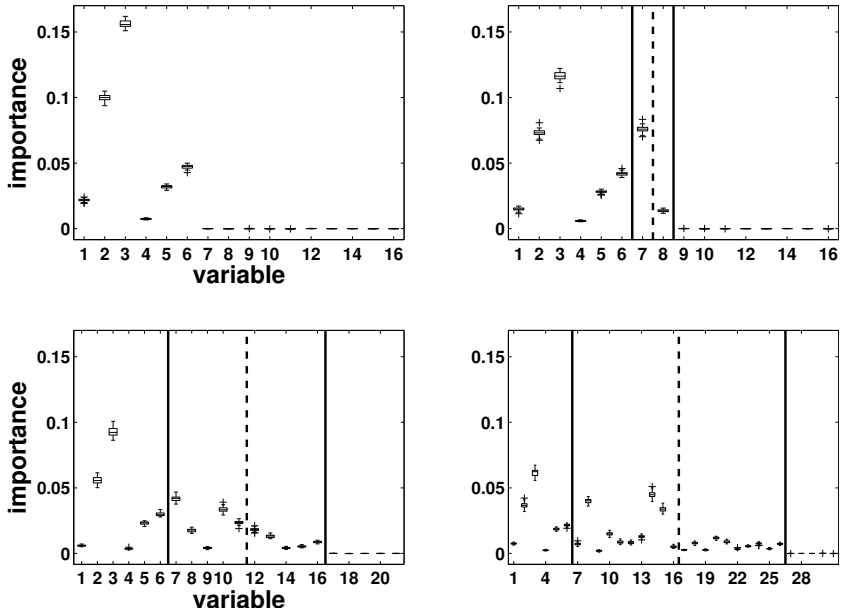

FIGURE 2.4 – Variable importance in presence of two groups of correlated variables (augmented toys data)

Again, the first graph is the reference one. Then for the three other cases, we simulate, in each group ($1, 2, 3$ and $4, 5, 6$ respectively) 1, 5 and 10 variables of correlation about 0.9 with variable 3 and variable 6 respectively. Replications of variable 3 are plotted between the first vertical line and the dashed line, and replications of variable 6 between the dashed line and the second vertical line.

VIs within each group are steadily decreasing when adding more replications. Nevertheless, the relative position between the two groups is preserved.

2.2.4 Prostate data variable importance

To end this section, we illustrate the behavior of variable importance on a high dimensional real dataset : the microarray data called Prostate, for which $n = 102$ and $p = 6033$ (see Singh et al. (2002) for a detailed presentation). The global picture is the following : two hugely important variables, about twenty moderately important variables and the others of small importance. So, more precisely, Figure 2.5 compares VI obtained for parameters set to their default values (graphs of the left column) and those obtained for $ntree = 2000$ and $mtry = p/3$ (graphs of the right column). Graphs are truncated after the 250th variable for readability (importance of noisy variables left are of the same order of magnitude as the last plotted).

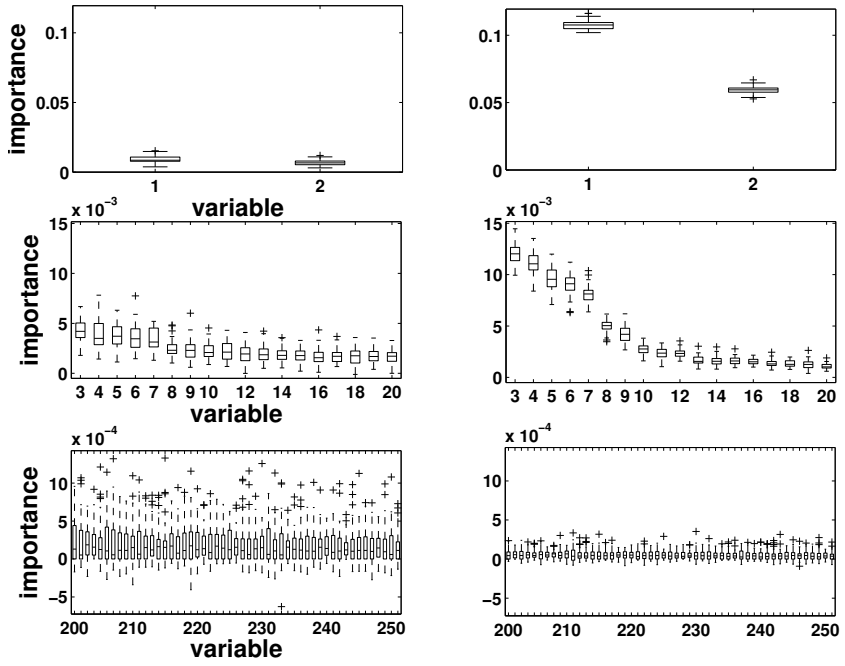

FIGURE 2.5 – Variable importance for Prostate data (using $ntree = 2000$ and $mtry = p/3$, on the right and using default values on the left)

For the two most important variables (first row), the magnitude of importance obtained with $ntree = 2000$ and $mtry = p/3$ is much larger than to the one obtained with default values. In the second row, the increase of magnitude is still noticeable from the third to the 9th most important variables and from the 10th to the 20th most important variables, VI is quite the same for the two parameter choices. In the third row, we get VI closer to zero for the variables with $ntree = 2000$ and $mtry = p/3$ than with default values. In addition, note that for the less important variables, boxplots are larger for default values, especially for unimportant variables (from the 200th to the 250th).

2.3 Variable selection

We distinguish two variable selection objectives :

1. to find important variables highly related to the response variable for interpretation purpose ;
2. to find a small number of variables sufficient to a good parsimonious prediction of the response variable.

The first one is to magnify all the important variables, even with high redundancy, for interpretation purpose and the second one is to find a sufficient parsimonious set of important variables for prediction.

As mentioned at the end of the introduction, we are guided in this paper by a typical situation matching two characteristics. The first one is high dimensionality, or at least when the number of true variables is much less than p, and the second one is the presence of groups of highly correlated predictors. They are also specifically addressed in two earlier works by Díaz-Uriarte and Alvarez de Andrés (2006) and Ben Ishak and Ghattas (2008). We briefly recall these contributions.

Díaz-Uriarte, Alvarez de Andrés propose a strategy based on recursive elimination of variables. More precisely, they first compute RF variable importance. Then, at each step, they eliminate the 20% of the variables having the smallest importance and build a new forest with the remaining variables. They finally select the set of variables leading to the smallest OOB error rate of a forest, defined by

$$errOOB = \frac{1}{n} Card\{i \in \{1, \ldots, n\} \mid y_i \neq \hat{y}_i\}$$

where $\hat{y}_i$ is the most frequent label predicted by trees t for which (x_i, y_i) is in the OOB_t sample. The proportion of variables to eliminate is an arbitrary parameter of their method and does not depend on the data.

Ben Ishak, Ghattas choose an ascendant strategy based on a sequential introduction of variables. First, they compute some SVM-based variable importance. Then, they build a sequence of SVM models invoking at the beginning the k most important variables, by step of 1. When k becomes too large, the additional variables are invoked by blocks. They finally select the set of variables leading to the model of smallest error rate. The way to introduce variables is not data-driven since it is fixed before running the procedure. They also compare their procedure with a similar one using RF instead of SVM.

2.3.1 Procedure

We propose the following two-steps procedure, the first one is common while the second one depends on the objective :

1. Preliminary elimination and ranking :
   - Sort the variables in decreasing order of RF scores of importance.
   - Cancel the variables of small importance. Denote by m the number of remaining variables.
2. Variable selection :
   - For *interpretation* : construct the nested collection of RF models involving the k first variables, for $k = 1$ to m, and select the variables involved in the model leading to the smallest OOB error ;
   - For *prediction* : starting from the ordered variables retained for interpretation, construct an ascending sequence of RF models, by invoking and testing the variables stepwise. The variables of the last model are selected.

Of course, this is a sketch of procedure and more details are needed to be effective. The next paragraph answer this point but we emphasize that we propose an heuristic

strategy which does not depend on specific model hypotheses but based on data-driven thresholds to take decisions.

Remark 2. *Since we want to treat in an unified way all the situations, we will use for finding prediction variables the somewhat crude strategy previously defined. Nevertheless, starting from the set of variables selected for interpretation (say of size K), a better strategy could be to examine all, or at least a large part, of the 2^K possible models and to select the variables of the model minimizing the OOB error. But this strategy becomes quickly unrealistic for high dimensional problems so we prefer to experiment a strategy designed for small n and large K which is not conservative and even possibly leads to select fewer variables.*

2.3.2 Starting example

To both illustrate and give more details about this procedure, we apply it on a simulated learning set of size $n = 100$ from the classification toys data model with $p = 200$. The results are summarized in Figure 2.6. The true variables (1 to 6) are respectively represented by $(\triangleright, \triangle, \circ, \star, \triangleleft, \square)$. We compute, thanks to the learning set, 50 forests with $ntree = 2000$ and $mtry = 100$, which are values of the main parameters previously considered as well adapted for VI calculations (see Section 2.2.2).

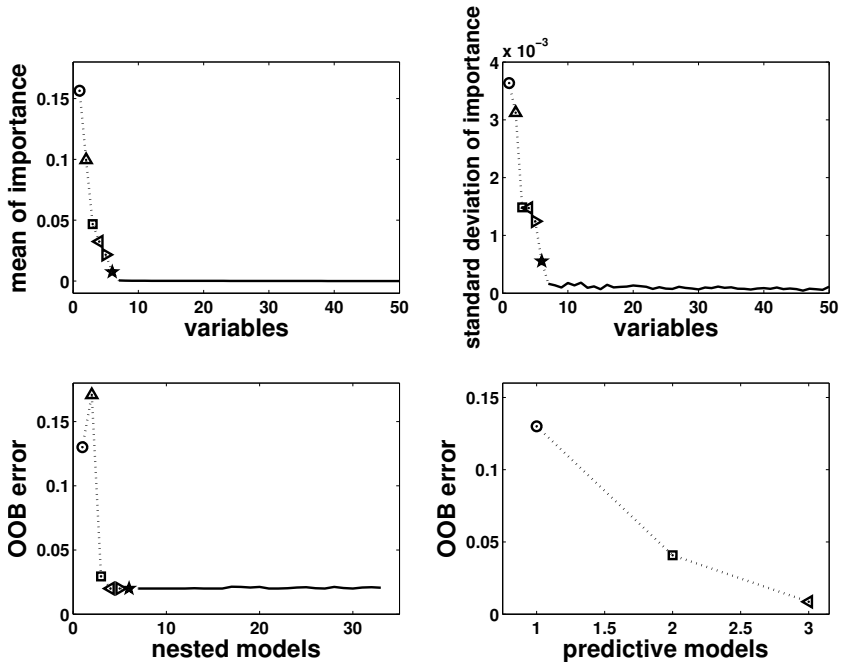

FIGURE 2.6 – Variable selection procedures for interpretation and prediction for toys data

Let us detail the main stages of the procedure together with, in italics, the results obtained on toys data :

– Variable ranking.
First we rank the variables by sorting the VI (averaged from the 50 runs) in descending order.
The result is drawn on the top left graph for the 50 *most important variables (the other noisy variables having an importance very close to zero too). Note that true variables are significantly more important than the noisy ones.*
– Variable elimination.
We keep this order in mind and plot the corresponding standard deviations of VI. We use this graph to estimate some threshold for importance. More precisely, we set the threshold as the minimum prediction value given by a CART model fitting this curve (see Figure 2.7). Then we keep only the variables with an averaged VI exceeding this level. This rule is, in general, conservative and leads to retain more variables than necessary, in order to make a careful choice later.

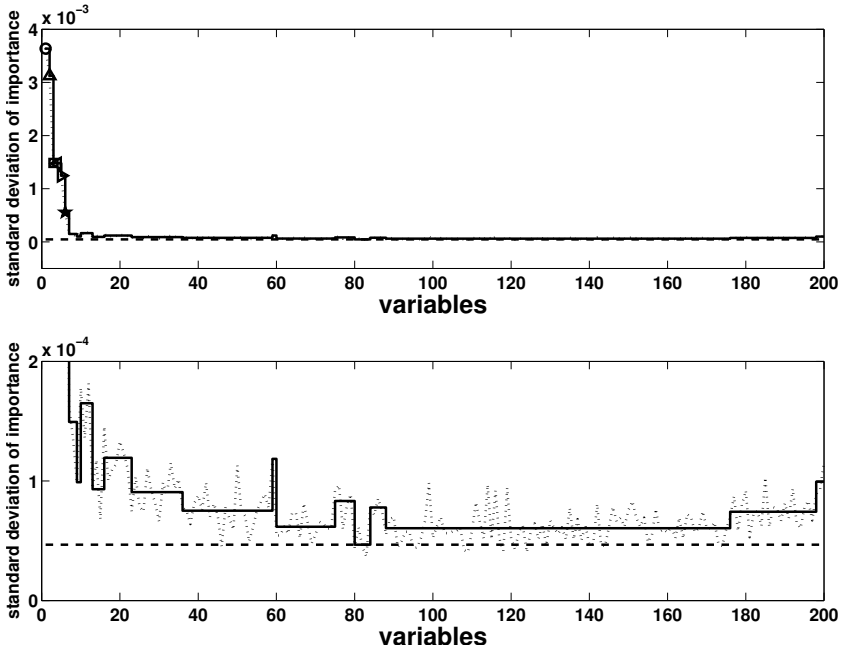

FIGURE 2.7 – Selecting the threshold for variable elimination using CART. Bold line refers to the CART estimation of the dotted line and the horizontal dashed line indicates the threshold (the bottom graph being a zoom of the top one)

The standard deviations of VI can be found in the top graph of Figure 2.7. We can see that true variables standard deviation is large compared to the noisy variables one, which is close to zero. The threshold leads to retain 33 *variables. Note that the threshold value is based on VI standard deviations while the effective thresholding is performed on VI mean (top left graph).*
Of course, this strategy is sensible when there exist irrelevant variables. Otherwise, a classical alternative is to select the threshold according to some elbow finding strategy on the VI mean curve. Some ideas can be found among those used for selecting the number of principal components in PCA (see Jolliffe (2002)).

- Variable selection procedure for interpretation.
  We compute OOB error rates of random forests (averaged on 50 runs and using default parameters) of the nested models starting from the one with only the most important variable, and ending with the one involving all important variables kept previously. Ideally, the variables of the model leading to the smallest OOB error are selected. In fact, in order to deal with instability, we use a classical trick : we select the smallest model with an OOB error less than the minimal OOB error augmented by its empirical standard deviation (based on 50 runs).
  Note that in the bottom left graph the error decreases quickly and reaches its minimum when the first 4 *true variables are included in the model. Then it remains nearly constant. We select the model containing* 4 *of the* 6 *true variables, while the actual minimum is reached with 24 variables.*
- Variable selection procedure for prediction.
  We perform a sequential variable introduction with testing : a variable is added only if the error gain exceeds a threshold. The idea is that the error decrease must be significantly greater than the average variation obtained by adding noisy variables.
  The bottom right graph shows the result of this step, the final model for prediction purpose involves only variables 3, 6 *and* 5. *The threshold is set to the mean of the absolute values of the first order differentiated OOB errors between the model with* $p_{interp} = 4$ *variables (the model we selected for interpretation, see the bottom left graph) and the one with all the* $p_{elim} = 33$ *variables :*

$$\frac{1}{p_{elim} - p_{interp}} \sum_{j=p_{interp}}^{p_{elim}-1} |\, errOOB(j+1) - errOOB(j) \,| \; .$$

  where $errOOB(j)$ *is the OOB error of the RF built using the* j *most important variables.*

It should be noted that if one wants to estimate the prediction error, since ranking and selection are made on the same set of observations, of course an error evaluation on a test set or using a cross validation scheme should be preferred. It is taken into account in the next section when our results are compared to others.

To evaluate fairly the different prediction errors, we prefer here to simulate a test set of the same size than the learning set. The test error rate with all (200) variables is about 6% while the one with the 4 variables selected for interpretation is about 4.5%, a little bit smaller. The model with prediction variables 3, 6 and 5 reaches an error of 1%. Repeating the global procedure 10 times on the same data always gave the same interpretation set of variables and the same prediction set, in the same order.

2.3.3 Highly correlated variables

We now apply the procedure on toys data with replicated variables : a first group of variables highly correlated with variable 3 and a second one replicated from variable 6

(the most important variable of each group). The situations of interest are the same as those considered to produce Figure 2.4.

number of replications	interpretation set	prediction set
1	3 7^3 2 6 5	3 6 5
5	3 2 7^3 10^3 6 11^3 5 12^6	3 6 5
10	3 14^3 8^3 2 15^3 6 5 10^3 13^3 20^6	3 6 5 10^3

TABLE 2.1 – Variable selection procedures in presence of highly correlated variables (augmented toys data) where the expression i^j means that variable i is a replication of variable j

Let us comment on Table 2.1, where the expression i^j means that variable i is a replication of variable j.

Interpretation sets do not contain all variables of interest. Particularly we hardly keep replications of variable 6. The reason is that even before adding noisy variables to the model the error rate of nested models do increase (or remain constant) : when several highly correlated variables are added, the bias remains the same while the variance increases. However the prediction sets are satisfactory : we always highlight variables 3 and 6 and at most one correlated variable with each of them.

Even if all the variables of interest do not appear in the interpretation set, they always appear in the first positions of our ranking according to importance. More precisely the 16 most important variables in the case of 5 replications are : (3 2 7^3 10^3 6 11^3 5 12^6 8^3 13^6 16^6 1 15^6 14^6 9^3 4), and the 26 most important variables in the case of 10 replications are : (3 14^3 8^3 2 15^3 6 5 10^3 13^3 20^6 21^6 11^3 12^3 18^6 1 24^6 7^3 26^6 23^6 16^3 25^6 22^6 17^6 19^6 4 9^3). Note that the order of the true variables (3 2 6 5 1 4) is always the same.

2.4 Experimental results

In this section we experiment the proposed procedure on four high dimensional classification datasets and then finally we examine the results on a standard regression problem to illustrate the versatility of the procedure.

2.4.1 Prostate data

We apply the variable selection procedure on Prostate data (for which $n = 102$ and $p = 6033$, see Singh et al. (2002)). The graphs of Figure 2.8 are obtained as those of Figure 2.6, except that for the RF procedure, we use $ntree = 2000$, $mtry = p/3$ and for the bottom left graph, we only plot the 100 most important variables for visibility. The procedure leads to the same picture as previously, except for the OOB error rate

along the nested models which is less regular. The first point is to notice that the elimination step leads to keep only 270 variables. The key point is that the procedure selects 9 variables for interpretation, and 6 variables for prediction. The number of selected variables is then very much smaller than $p = 6033$.

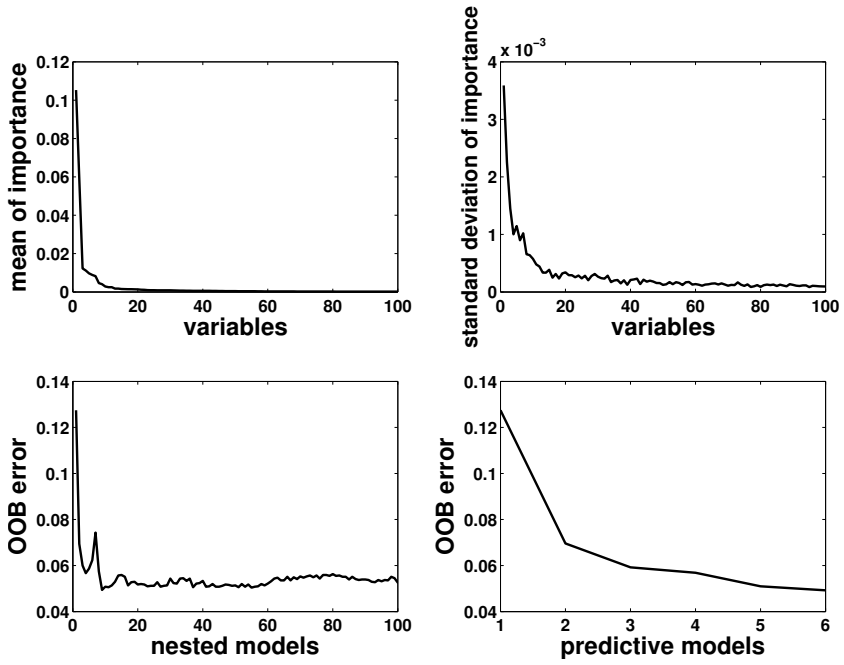

FIGURE 2.8 – Variable selection procedures for interpretation and prediction for Prostate data

In addition, to examine the variability of the interpretation and prediction sets the global procedure is repeated five times on the entire Prostate dataset. The five prediction sets are very close to each other. The number of prediction variables fluctuates between 6 and 10, and 5 variables appear in all sets. Among the five interpretation sets, 2 are identical and made of 9 variables and the 3 other are made of 25 variables. The 9 variables of the smallest sets are present in all sets and the biggest sets (of size 25) have 23 variables in common.

So, although the sets of variables are not identical for each run of the procedure, the most important variables are included in all of the sets.

2.4.2 Four high dimensional classification datasets

Let us consider the four well known high dimensional real datasets called Colon ($n = 62, p = 2000$), see Alon et al. (1999), Leukemia ($n = 38, p = 3051$), see Golub et al. (1999), Lymphoma ($n = 62, p = 4026$), see Alizadeh (2000) and Prostate ($n = 102, p = 6033$), see Singh et al. (2002). We apply the global variable selection procedure on these four benchmark high dimensional real datasets, and we want to get an estimation of

prediction error rates. Since these datasets are of small size and in order to be comparable with the results obtained by other authors, we use a 5-fold cross-validation to estimate the error rate. So we split the sample in 5 stratified parts, each part is successively used as a test set, and the remaining of the data is used as a learning set. Note that the set of variables selected vary from one fold to another. So, we give in Table 2.2 the misclassification error rate, given by the 5-fold cross-validation, for interpretation and prediction sets of variables respectively. The number into brackets is the average number of selected variables. In addition, one can find the original error which stands for the misclassification rate given by the 5-fold cross-validation achieved with random forests using all variables. This error is calculated using the same partition in 5 parts and again we use $ntree = 2000$ and $mtry = p/3$ for all datasets.

Dataset	interpretation	prediction	original
Colon	0.16 (35)	0.20 (8)	0.14
Leukemia	0 (1)	0 (1)	0.02
Lymphoma	0.08 (77)	0.09 (12)	0.10
Prostate	0.085 (33)	0.075 (8)	0.07

TABLE 2.2 – Variable selection procedure for four high dimensional real datasets. CV-error rate and into brackets the average number of selected variables

The number of interpretation variables is hugely smaller than p, at most tens to be compared to thousands. The number of prediction variables is very small (always smaller than 12) and the reduction can be very important with respect to the interpretation set size. The errors for the two variable selection procedures are of the same order of magnitude as the original error (but a little bit larger).

We compare these results with the results obtained by Ben Ishak, Ghattas (see tables 9 and 11 in Ben Ishak and Ghattas (2008)) which have compared their method with 5 competitors (mentioned in the introduction) for classification problems on these four datasets. Error rates are comparable. With the prediction procedure we always select fewer variables than their procedures (except for their method GLMpath which select less than 3 variables for all datasets).

One can notice that the results for the dataset Prostate differ from Section 2.4.1 to Section 2.4.2. This difference can mainly be explained by the use of 5-fold cross-validation in Section 2.4.2. Indeed the fact that n is very small ($n = 62$) makes the method quite unstable with respect to resampling.

2.4.3 Ozone data

Before ending the paper, we consider a standard regression dataset. Since it is far from matching the two main characteristics which have guided the algorithm principle, it allows us to check that it still work well. We apply the entire procedure to the easy to

interpret ozone dataset (it can be retrieved from the R package `mlbench` and detailed information can be found in the corresponding description file). It consists of $n = 366$ observations of the daily maximum one-hour-average ozone together with $p = 12$ meteorologic explanatory variables. Let us first examine, in Figure 2.9 the VI obtained with RF procedure using $mtry = p/3 = 4$ and $ntree = 2000$.

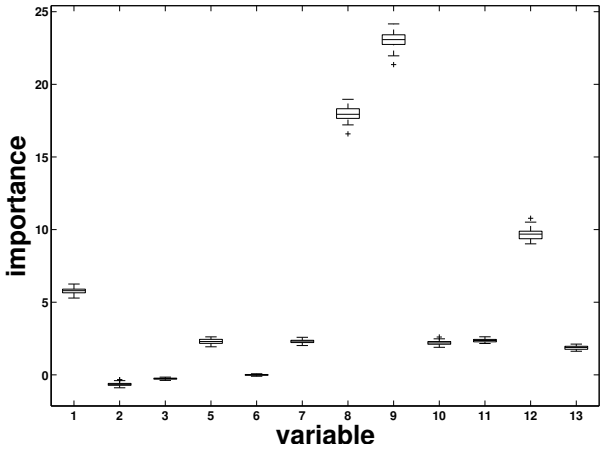

FIGURE 2.9 – Variable importance for ozone data

From the left to the right, the 12 explanatory variables are 1-Month, 2-Day of month, 3-Day of week, 5-Pressure height, 6-Wind speed, 7-Humidity, 8-Temperature (Sandburg), 9-Temperature (El Monte), 10-Inversion base height, 11-Pressure gradient, 12-Inversion base temperature, 13-Visibility. Variables are numbered exactly as in `mlbench`, so the 4th variable is the response one.

Three very sensible groups of variables appear from the most to the least important. First, the two temperatures (8 and 9), the inversion base temperature (12) known to be the best ozone predictors, and the month (1), which is an important predictor since ozone concentration exhibits an heavy seasonal component. A second group of clearly less important meteorological variables : pressure height (5), humidity (7), inversion base height (10), pressure gradient (11) and visibility (13). Finally three unimportant variables : day of month (2), day of week (3) of course and more surprisingly wind speed (6). This last fact is classical : wind enter in the model only when ozone pollution arises, otherwise wind and pollution are weakly correlated (see for example Cheze et al. (2003) highlighting this phenomenon using partial estimators).

Let us now examine the results of the selection procedures.

After the first elimination step, the 2 variables of negative importance are canceled, as expected.

Then the interpretation procedure leads to select the model with 7 variables, which contains all the most important variables : (9 8 12 1 11 7 5).

Finally, only one more variable is eliminated (humidity (7)) by the prediction pro-

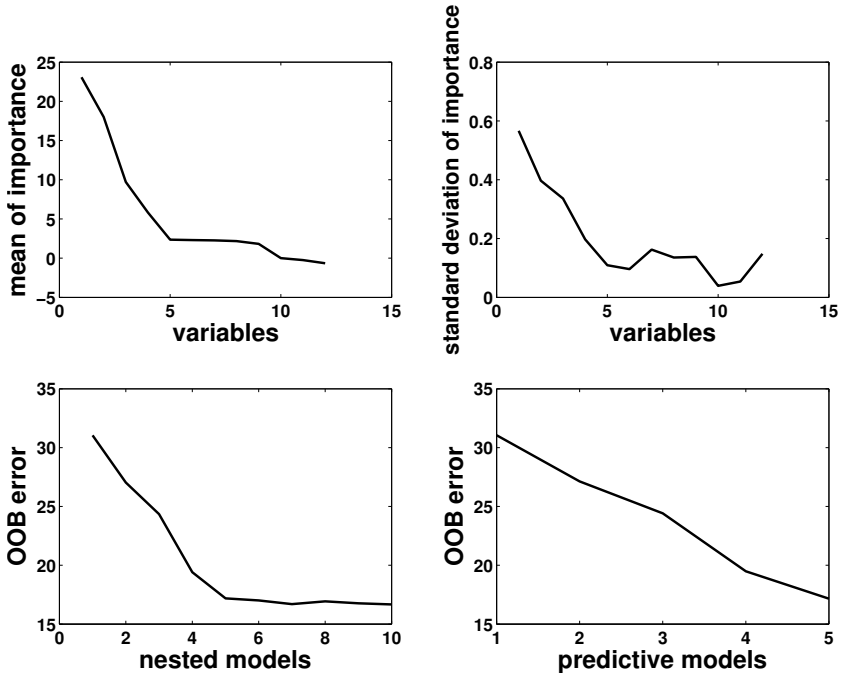

FIGURE 2.10 – Variable selection procedures for interpretation and prediction for ozone data

cedure.

2.5 Discussion

Of course, one of the main open issues about random forests is to elucidate from a mathematical point of view its exceptionally attractive performance. In fact, only a small number of references deal with this very difficult challenge and, in addition to bagging theoretical examination by Bühlmann and Yu (2002), only purely random forests, a simple version of random forests, is considered. Purely random forests have been introduced by Cutler and Zhao (2001) for classification problems and then studied by Breiman (2004), but the results are somewhat preliminary. More recently Biau et al. (2008) obtained the first well stated consistency type results.

From a practical perspective, surprisingly, this simplified and essentially not data-driven strategy seems to perform well, at least for prediction purpose (see Cutler and Zhao (2001)) and, of course, can be handled theoretically in a easier way. Nevertheless, it should be interesting to check that the same conclusions hold for variable importance and variable selection tasks.

In addition, it could be interesting to examine some variants of random forests which, at the contrary, try to take into account more information. We propose two ideas. The

first one is about pruning : why pruning is not used for individual trees ? Of course, from the computational point of view the answer is obvious and for prediction performance, averaging eliminate the negative effects of individual overfitting. But from the two other previously mentioned statistical problems, prediction and variable selection, it remains unclear. The second remark is about the random feature selection step. The most widely used version of RF selects randomly $mtry$ input variables according to the discrete uniform distribution. Two variants can be suggested : the first is to select random inputs according to a distribution coming from a preliminary ranking given by a pilot estimator ; the second one is to adaptively update this distribution taking profit of the ranking based on the current forest which is then more and more accurate.

Finally, let us mention an application for fMRI brain activity classification (see Genuer et al. (2010b)). This is a typical situation where $n << p$, with a lot of highly correlated variables and where the two objectives have to be addressed : find the most activated (whole) regions of the brain, and build a predictive model involving only a few voxels of the brain. An interesting aspect for us will be the feedback given by specialists, needed to interpret the set of variables found by our algorithm. In addition a lot of well known methods have already been used for these data, so fair comparisons will be easy and fruitful.

2.A Appendix : Selecting method parameters

2.A.1 Experimental framework

RF procedure

The R package about random forests is based on the the seminal contribution of Breiman and Cutler (2005) and is described in Liaw and Wiener (2002). In this section, we focus on the `randomForest` procedure. The two main parameters are *mtry*, the number of input variables randomly chosen at each split and *ntree*, the number of trees in the forest [2].

A third parameter, denoted by *nodesize*, allows to specify the minimum number of observations in a node. We retain the default value (1 for classification and 5 for regression) of this parameter for all of our experimentations, since it is close to the maximal tree choice.

OOB error

In this section, we concentrate on the prediction performance of RF focusing on out-of-bag (OOB) error (see Breiman (2001)). We use this kind of prediction error estimate for three reasons : the main is that we are mainly interested in comparing results instead of assessing models, the second is that it gives fair estimation compared to the usual alternative test set error even if it is considered as a little bit optimistic and the last one, but not the least, is that it is a default output of the procedure. To avoid unsignificant sampling effects, each OOB errors is actually the mean of OOB error over 10 runs.

Datasets

We have collected information about the data sets considered in this section : the name, the name of the corresponding data structure (when different), n, p, the number of classes c in the multiclass case, a reference, a website or a package. The two next tables contain synthetic information while details are postponed in the Appendix. We distinguish standard and high dimensional situations and, in addition, the three problems : regression, 2-class classification and multiclass classification.

Table 2.3 displays some information about standard problems datasets : for classification at the top and for regression at the bottom.

Table 2.4 displays high dimensional problems datasets : for classification at the top and for regression at the bottom.

2. In all the section, $mtry = m$ with $m \in \mathbb{R}$ stands for $mtry = \lfloor m \rfloor$

Name	Observations	Variables	Classes
Ionosphere	351	34	2
Diabetes	768	8	2
Sonar	208	60	2
Votes	435	16	2
Ringnorm	200	20	2
Threenorm	200	20	2
Twonorm	200	20	2
Glass	214	9	6
Letters	20000	16	26
Sat-images	6435	36	6
Vehicle	846	18	4
Vowel	990	10	11
Waveform	200	21	3
BostonHousing	506	13	
Ozone	366	12	
Servo	167	4	
Friedman1	300	10	
Friedman2	300	4	
Friedman3	300	4	

TABLE 2.3 – Standard problems : data sets for classification at the top, and for regression at the bottom

Name	Observations	Variables	Classes
Adenocarcinoma	76	9868	2
Colon	62	2000	2
Leukemia	38	3051	2
Prostate	102	6033	2
Brain	42	5597	5
Breast	96	4869	3
Lymphoma	62	4026	3
Nci	61	6033	8
Srbct	63	2308	4
toys data	100	100 to 1000	2
PAC	209	467	
Friedman1	100	100 to 1000	
Friedman2	100	100 to 1000	
Friedman3	100	100 to 1000	

TABLE 2.4 – High dimensional problems : data sets for classification at the top, and for regression at the bottom

2.A.2 Regression

About regression problems, even if it seems at first inspection that the seminal paper by Breiman (2001) closes the debate about good advice, it remains that the experimental results are about a variant which is not implemented in the universally used R package. Moreover, except this reference, at our knowledge, no such a general paper is available, so we develop again the Breiman's study both for real and simulated data corresponding to the case $n >> p$ and we provide some additional study on data corresponding to the case $n << p$ (such examples typically come from chemometrics).

We observe that the default value of *mtry* proposed by the R package is not optimal, and that there is no improvement by using random forests with respect to unpruned bagging (obtained for $mtry = p$).

Standard problems

Let us briefly examine standard ($n >> p$) regression datasets. In Figure 2.11 for real ones and for simulated ones in Figure 2.12. Each plot gives for $mtry = 1$ to p the OOB error for three different values of $ntree = 100, 500$ and 1000. The vertical solid line indicates the value $mtry = p/3$, the default value proposed by the R package for regression problems, the vertical dashed line being the value $mtry = \sqrt{p}$.

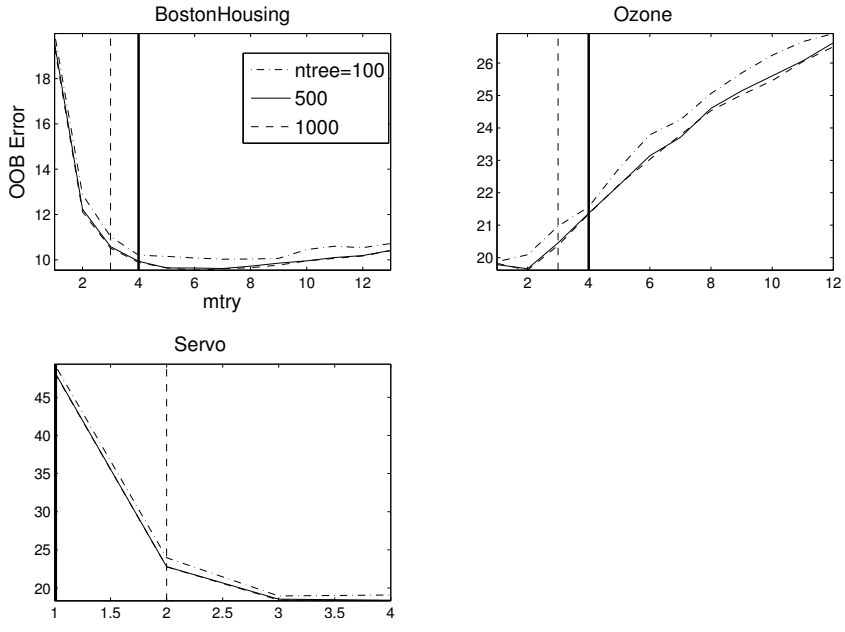

FIGURE 2.11 – Standard regression : 3 real data sets

Three remarks can be formulated. First, the OOB error is maximal for $mtry = 1$ and then decreases quickly (except for the ozone dataset, for reasons not clearly elucidated), then as soon as $mtry > \sqrt{p}$, the error remains the same. Second, the choice $mtry = \sqrt{p}$

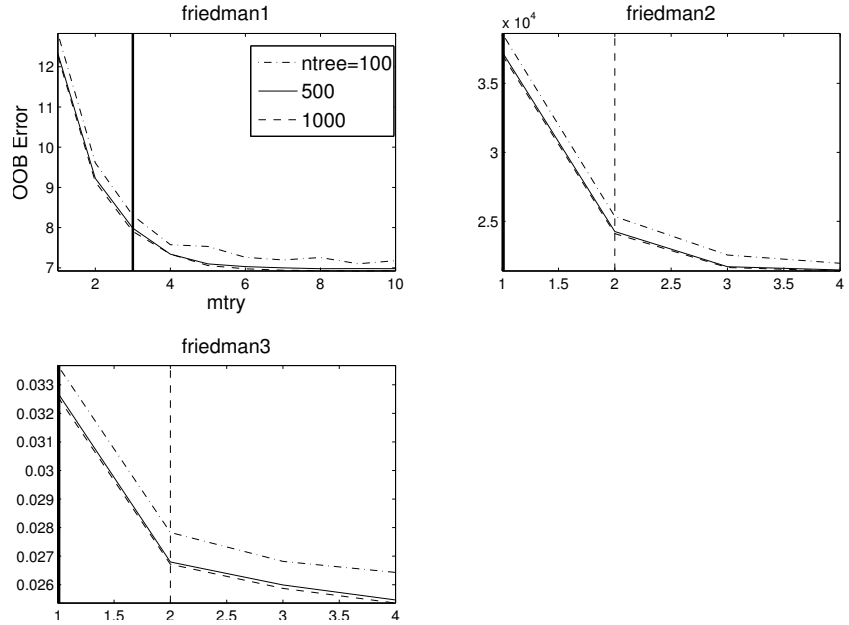

FIGURE 2.12 – Standard regression : 3 simulated data sets

gives always lower OOB error than $mtry = p/3$, and the gain can be important. So the default value proposed by the R package seems to be often not optimal, especially when $\lfloor p/3 \rfloor = 1$. Lastly, the default value $ntree = 500$ is convenient, but a much smaller one $ntree = 100$ leads to comparable results.

So, for standard ($n >> p$) regression problems, it seems that there is no improvement by using random forests with respect to unpruned bagging (obtained for $mtry = p$).

High dimensional problems

Let us start with a simulated data set for the high dimensional case $n << p$. This example is built by adding extra noisy variables (independent and uniformly distributed on $[0, 1]$) to the Friedman1 model defined by :

$$Y = 10 \sin(\pi X^1 X^2) + 20(X^3 - 0.5)^2 + 10X^4 + 5X^5 + \epsilon$$

where $X^1, \ldots, X^5$ are independent and uniformly distributed on $[0, 1]$ and $\epsilon \sim \mathcal{N}(0, 1)$. So we have 5 variables related to the response Y, the others being noise. We set $n = 100$ and let p vary.

Figure 2.13 contains four plots corresponding to 4 values of p (100, 200, 500 and 1000) increasing the nuisance space dimension. Each plot gives for ten values of $mtry$ (1, $\sqrt{p}/2$, $\sqrt{p}$, $2\sqrt{p}$, $4\sqrt{p}$, $p/4$, $p/3$, $p/2$, $3p/4$, p) the OOB error for three different values of $ntree = 100, 500$ and 1000. The x-axis is in log scale and the vertical solid line indicates $mtry = p/3$ the default value proposed by the R package for regression, the vertical dashed line being the value $mtry = \sqrt{p}$.

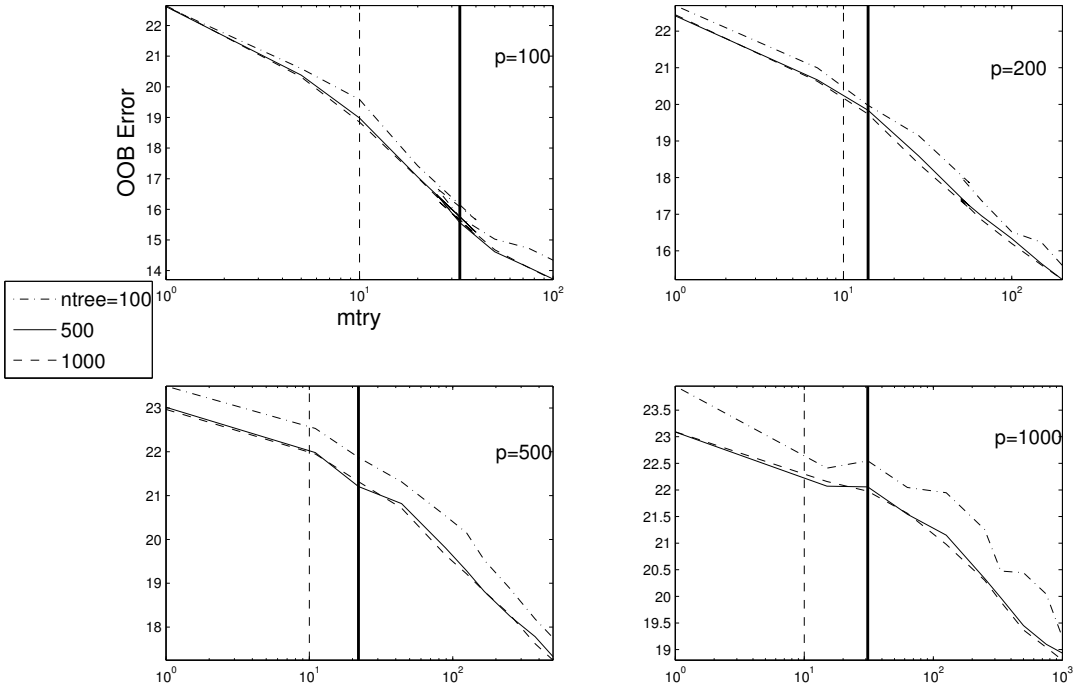

FIGURE 2.13 – High dimensional regression simulated data set : Friedman1. The x-axis is in log scale

Let us give four comments. All curves have the same shape : the OOB error decreases while $mtry$ increases. While p increases, both OOB errors of unpruned bagging (obtained with $mtry = p$) and random forests with default value of $mtry$ increase, but unpruned bagging performs better than RF (about 25% of improvement). The choice $mtry = \sqrt{p}$ gives always worse results than those obtained for $mtry = p/3$. Finally, the default choice $ntree = 500$ is convenient, but a much smaller one $ntree = 100$ leads to comparable results.

Figure 2.14 and 2.15 show the results of the same study for the Friedman2 and Friedman3 models. The previous comments remain valid. Let us just note that the difference between unpruned bagging and random forests with $mtry$ default value is even more pronounced for these two problems.

To end, let us now examine the high dimensional real data set PAC. Figure 2.16 gives for same ten values of $mtry$ the OOB error for four different values of $ntree = 100, 500, 1000$ and 5000 (x-axis is in log scale). The general behavior is similar except for the shape : as soon as $mtry > \sqrt{p}$, the error remains the same instead of still decreasing. The difference of the shape of the curves between simulated and real datasets can be explained by the fact that, in simulated datasets we considered, the number of true variables is very small compared to the total number of variables. One may expect that in real datasets, the proportion of true variables is larger.

So, for high dimensional ($n << p$) regression problems, unpruned bagging seems to

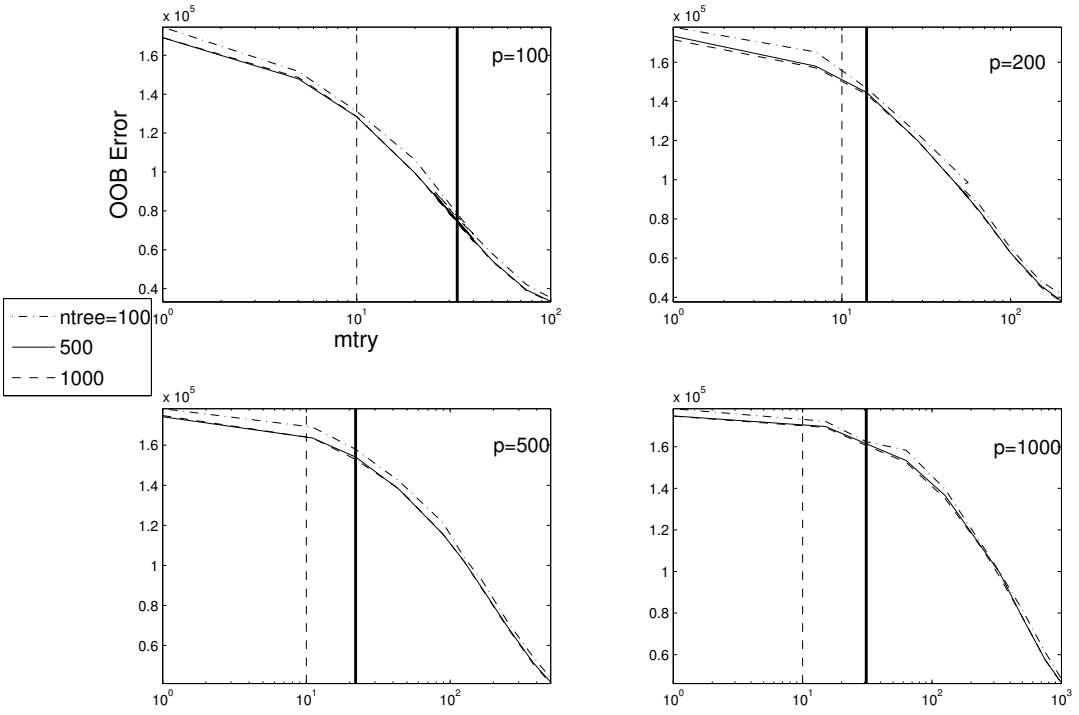

FIGURE 2.14 – High dimensional regression simulated data set : Friedman2. The x-axis is in log scale

perform better than random forests and the difference can be large.

2.A.3 Classification

About standard classification problems, we check that Breiman's conclusions remain valid for the considered variant and that the *mtry* default value proposed in the R package is good. However for high dimensional classification problems, we observe that larger values of *mtry* give sometimes much better results.

Standard problems

For classification problems for which $n >> p$, again the paper by Breiman is interesting and we just quickly check the conclusions.

Let us first examine in Figure 2.17 standard ($n >> p$) classification real data sets. Each plot gives for $mtry = 1$ to p the OOB error for three different values of $ntree = 100, 500$ and 1000. The vertical solid line indicates the value $mtry = \sqrt{p}$, the default value proposed by the R package for classification.

Three remarks can be formulated. The default value $mtry = \sqrt{p}$ is convenient for

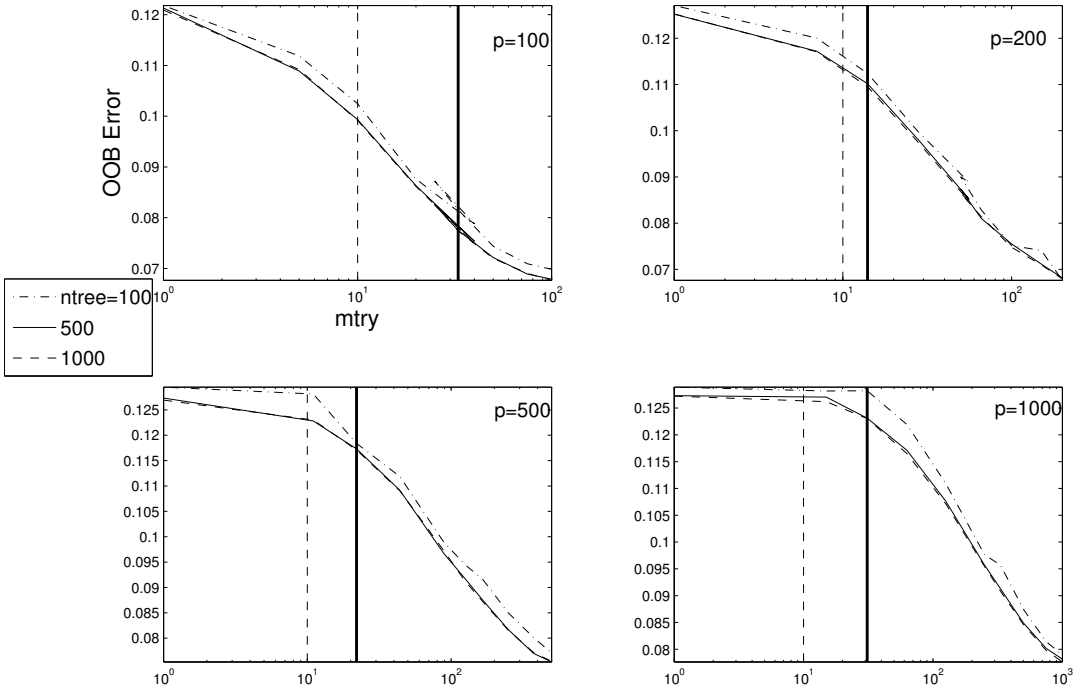

FIGURE 2.15 – High dimensional regression simulated data set : Friedman3. The x-axis is in log scale

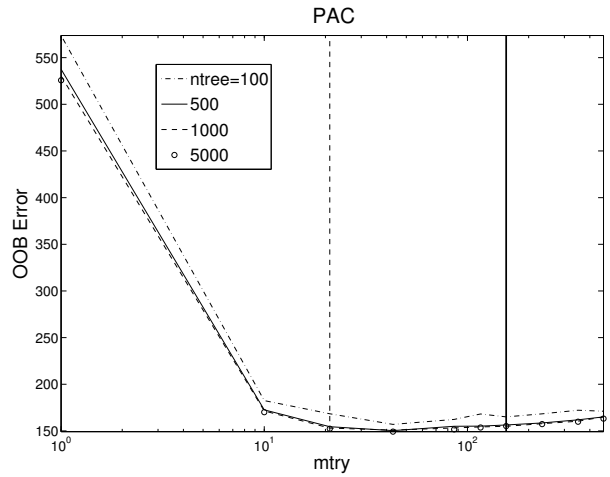

FIGURE 2.16 – High dimensional regression : PAC data. The x-axis is in log scale

all the examples. The default value $ntree = 500$ is sufficient and a much smaller one $ntree = 100$ is not convenient and can leads to significantly larger errors. The general shape is the following : the errors for $mtry = 1$ and for $mtry = p$ (corresponding to the unpruned bagging) are of the same "large" order of magnitude and the minimum is reached for the value $\sqrt{p}$. The gain can be about 30 or 50%.

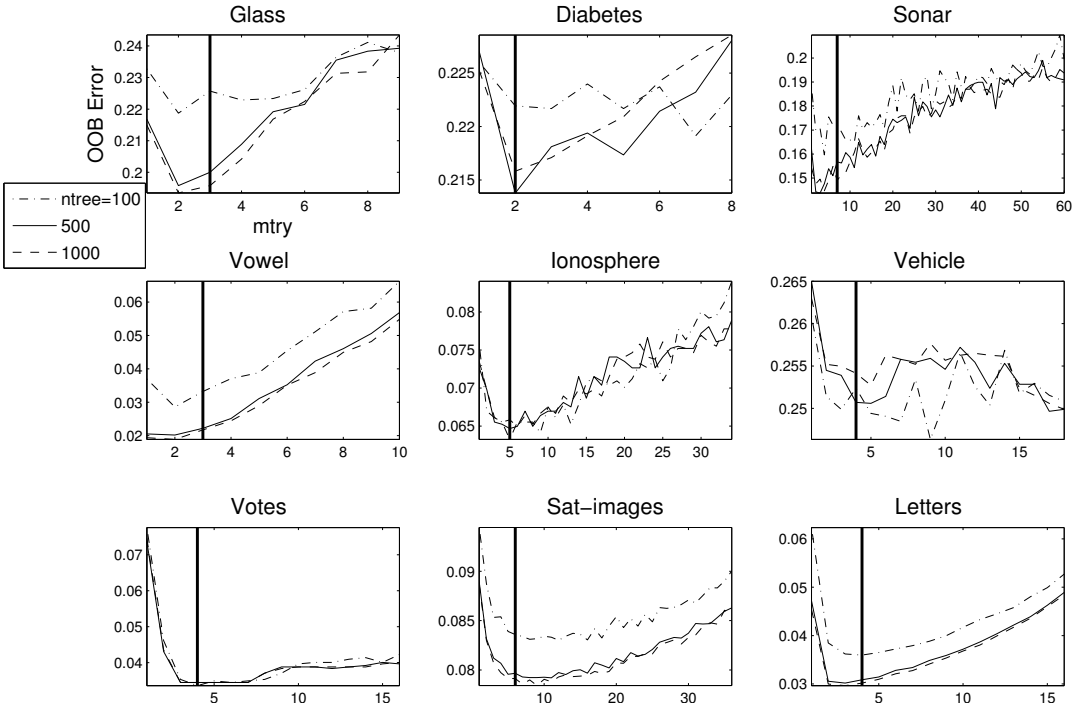

FIGURE 2.17 – Standard classification : 9 real data sets

So, for these 9 examples, the default value proposed by the R package is quite optimal.

Let us now examine in Figure 2.18 standard ($n >> p$) classification simulated datasets. As it can be seen, $ntree = 500$ is sufficient and, except for the ringnorm already pointed out as a somewhat special dataset (see Cutler, Zhao (2001) Cutler and Zhao (2001)) the value $mtry = \sqrt{p}$ is good. Here, the general shape of the error curve is quite different compared to real datasets : the error increases with $mtry$. So for these four examples, the smaller $mtry$, the better.

High dimensional problems

Let us now consider the case $n << p$ for which Díaz-Uriarte and Alvarez de Andrés (2006) give numerous advice. We complete the study by trying larger values of $mtry$, which give interesting results. One can found in Figure 2.19 the OOB errors for nine high dimensional real datasets. Each plot gives for nine values of $mtry$ (1, $\sqrt{p}/2$, $\sqrt{p}$, $2\sqrt{p}$, $4\sqrt{p}$, $p/4$, $p/2$, $3p/4$, p) the OOB error for four different values of $ntree = 100, 500, 1000$ and 5000. The x-axis is in log scale. The vertical solid line indicates the default value proposed by the R package $mtry = \sqrt{p}$.

Again the default value $ntree = 500$ is sufficient, and at the contrary the value $ntree = 100$ can leads to significantly larger errors. The general shape is the following : it decreases in general and the minimum value is obtained or is close to the one reached using $mtry = p$ (corresponding to the unpruned bagging). The difference with standard

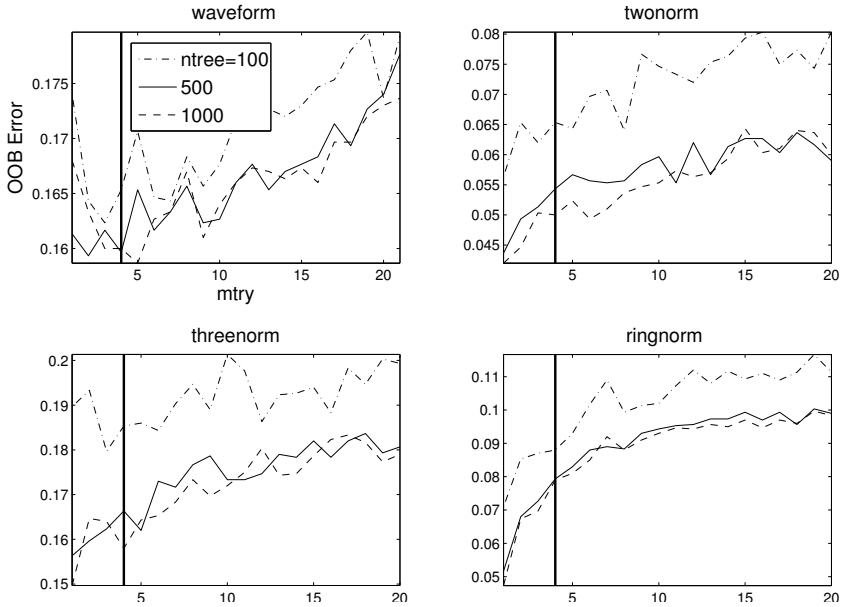

FIGURE 2.18 – Standard classification : 4 simulated data sets

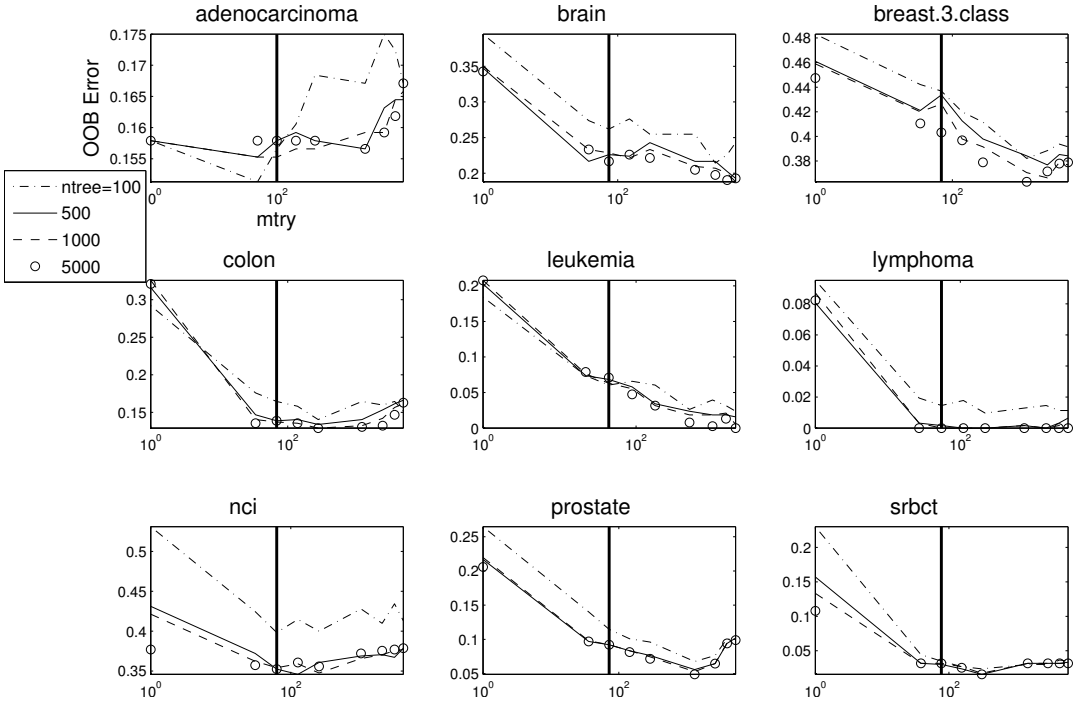

FIGURE 2.19 – High dimensional classification : 9 real data sets. The x-axis is in log scale

problems is notable, the reason is that when p is large, *mtry* must be sufficiently large in order to have a high probability to capture important variables (that is variables highly

related to the response) for defining the splits of the RF. In addition, let us mention that the default value $mtry = \sqrt{p}$ is still reasonable from the OOB error viewpoint but of course, since $\sqrt{p}$ is small with respect to p, it is a very attractive value from a computational perspective (notice that the trees are not too deep since n is not too large).

Let us examine a simulated dataset for the case $n << p$, introduced by Weston et al. (2003), called "toys data" in the sequel, and described in Section 2.2. It is an equiprobable two-class problem, $Y \in \{-1, 1\}$, with 6 true variables, the others being some noise. Let us fix $n = 100$.

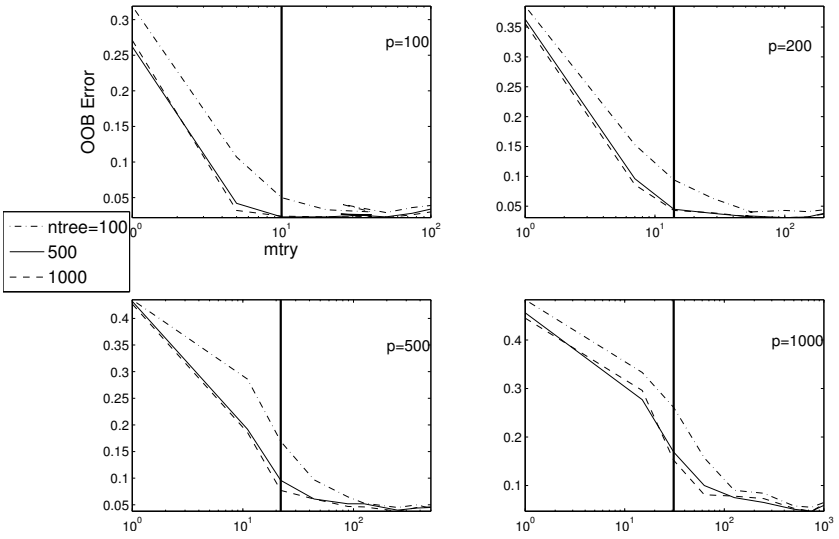

FIGURE 2.20 – High dimensional classification simulated data set : toys data for 4 values of p. The x-axis is in log scale

The plots of Figure 2.20 are organized as previously, four values of p are considered : 100, 200, 500 and 1000 corresponding to increasing nuisance space dimension. For $p = 100$ and $p = 200$, the error decreases hugely until $mtry$ reaches $\sqrt{p}$ and then remains constant, so the default values work well and perform as well as unpruned bagging, even if the true dimension $\tilde{p} = 6 << p$. For larger values of p ($p \geq 500$), the shape of the curve is close to the one for high dimensional real data sets (the error decreases and the minimum is reached when $mtry = p$). Whence, the error reached by using random forests with default $mtry$ is about 70% to 150% larger than the error reached by unpruned bagging which is close to 3% for all the considered values of p.

Finally, for high dimensional classification problems, our conclusion is that it may be worthwhile to choose $mtry$ larger than the default value $\sqrt{p}$.

Chapitre 3

Forêts aléatoires : sélection de variables et applications

Sommaire

Résumé

Ce chapitre présente deux travaux de nature appliquée. Dans la première section, nous traitons des données d'IRM fonctionnelle. On mesure, pour plusieurs sujets humains, l'activation cérébrale induite par la présentation de différentes formes d'un même objet. Les données obtenues sont des données de classification de très grande dimension (le nombre de variables est de 100 000 et le nombre d'observations de 72). Le but est de construire un prédicteur qui réussit, au vue d'une carte d'activation, à prédire la forme de l'objet qui a été présenté au sujet. De plus, un deuxième objectif est de déterminer quelles sont les zones du cerveau humain les plus sollicitées pour la reconnaissance de

formes. Après une première étape de réduction de dimension (le nombre de variables est alors de 1000), nous appliquons la procédure de sélection de variables présentée au chapitre précédent. Nous interprétons les résultats obtenus et les comparons avec une méthode de référence pour traiter ce type de données. Notre méthode présente l'avantage de fournir des zones actives très localisées, tout en donnant des erreurs de généralisation légèrement meilleures que la méthode de référence. Ceci facilite grandement l'interprétation des résultats.

Les données traitées dans la deuxième section de ce chapitre sont de nature très différente. Ce sont des données de régression en dimension plus raisonnable (le nombre de variables est ici de 88 et le nombre d'observations de 110). Elles proviennent d'une étude sur la transmission du paludisme d'homme à homme, par l'intermédiaire de moustiques. L'enjeu est ici de comprendre quelles sont les variables qui favorisent ou inhibent la capacité de transmission du parasite responsable de la maladie. Notre méthode de sélection de variables confirme l'importance de la variable décrivant la concentration initiale du parasite dans le porteur de la maladie. De plus, elle met en évidence une seconde variable qui décrit la multiplicité d'infection du porteur. Ce dernier résultat intéresse beaucoup les biologistes spécialistes de cette étude, car contrôler cette multiplicité d'infection pourrait permettre de contrôler la capacité de transmission du parasite.

Le contenu de la section 3.1 *est le fruit d'une collaboration avec Vincent Michel, Evelyn Eger et Bertrand Thirion du CEA Neurospin. Ce travail (Genuer et al., 2010b) est publié dans les actes de la conférence avec comité de lecture* COMPSTAT'2010.

La suite du chapitre (sections 3.2 *et annexes* 3.A *et* 3.B*) concerne une collaboration avec Wilson Toussile et Isabelle Morlais, qui a donné lieu à un article actuellement soumis.*

3.1 Random forests based feature selection for decoding fMRI data

3.1.1 Introduction

A new way of analyzing neuroimaging data consists in assessing how well behavioral information or cognitive states can be predicted from brain activation images such as those obtained with functional magnetic resonance imaging (fMRI) (Cox and Savoy (2003)). This approach opens the way to understanding the mental representation of various perceptual and cognitive parameters. Indeed, certain neuronal populations are thought to activate specifically when a certain perceptual or cognitive parameter reaches

a given value. The accuracy of the prediction of the target behavioral variable, as well as the spatial layout of predictive regions can provide valuable information about functional brain organization ; in short, it helps to *decode* the brain system (Dayan and Abbott (2001)).
The main difficulty in this procedure is the huge dimensionality of the data, with far more features than samples. In this article, the samples will refer to the activation parameter maps resulting from a General Linear Model (GLM), the features being the voxel-based activation values. The large number of features leads to overfitting and thus to a dramatic decrease in prediction accuracy. Feature selection is thus mandatory, and is often performed by a mass-univariate selection based on F-test statistics. However, this classical approach is not well suited for neuroimaging as it does not cope with the multivariate structure of the data.
In order to improve the predictive framework, we introduce a new multivariate method of feature selection based on Random Forests (RF henceforth). RF is an increasingly used statistical method introduced in Breiman (2001). It gives outstanding results in prediction for lots of diverse applications. In addition, it computes a variable importance that can be used to select variables. Our RF-based algorithm uses the variable importance index in a feature selection framework. This variable selection procedure comes from Genuer et al. (2010a), where one can find more information about RF variable importance.
After introducing the Random Forests and the RF-based algorithm, we show that our self-calibrated method performs an accurate feature selection, yielding a little bit better classification score than the reference technique, while keeping much less jointly informative variables. And this very sparse aspect of our variable selection method can help understanding functional brain organization. Let us finally emphasize that all along this paper, we distinguish two objectives : interpretation, which aims at selecting all the variables the most related to the response variable (even if they are correlated to each other) ; and prediction, which focuses on building a model involving the smallest subset of variables sufficient to make accurate predictions.

3.1.2 Methods

Let $(Y_1, \ldots, Y_n)$ represent the behavioural data to be fitted ($\forall i$, $Y_i \in \{1, \ldots, c\}$, where c is the number of classes) related to a set of n parameter maps obtained with a GLM, where each image corresponds to one stimulus presentation ; $(X_1, \ldots, X_n)$ are the p-dimensional activation maps ($X \in \mathbb{R}^p$) and p is the number of features (voxels or parcels). In fMRI data, we have $n \ll p$, so that feature selection is mandatory.

Random Forests

The principle of random forests is to aggregate many binary decision trees built on several bootstrap samples drawn from the learning set. The bootstrap samples are obtained by uniformly drawing n samples among the learning set with repetition. The decision trees are fully developed binary trees and the split rule is the following :

First, the whole dataset (also called the root of the tree) is split into two subsets of data (called two children nodes). To do that, one randomly chooses a given number m_{try} of variables, and computes all the splits only for the previously selected variables. A split is of the form $\{X^i \leqslant s\} \cup \{X^i > s\}$, which means that data with the i-th variable value less than the threshold s go to the left child node and the others to the right one. Finally the selected split is the one leading to the most homogeneous children nodes (i.e. subsets associated to the same class).
Then, one restraints to one child node, randomly chooses another set of m_{try} variables and calculates the best split. And so on, until each node is a terminal node, i.e. it comprises observations associated with the same class.
A new data item X, starting in the root of the tree, goes down the tree following the splits and falls in a terminal node. Then the tree predicts for X, the common class $\hat{Y}$ of the data in this terminal node. To finally get the RF classifier, one aggregates all the tree classifiers through a majority vote heuristic : for a new observation, each tree predicts a class and RF finally returns the most popular class.
Inside the variable selection procedure, we use an estimation of the prediction error directly computed by the RF algorithm. This is the Out Of Bag (OOB) error and is calculated as follows. Fix one data in the learning sample, and consider all the bootstrap samples which do not contain this data (i.e. for which the data is "out of bag"). Now perform a majority vote only among trees built on these bootstrap samples. After doing this for all data, compare to the true classes and get an estimation of the prediction error (which is a cross-validated error estimate).
Let us now detail the computation of the RF variable importance for the first variable X^1. For each tree, one has a bootstrap sample associated with an OOB sample. Predict the OOB data with the tree classifier. Now, randomly permute the values of the first variable of the OOB observations, predict these modified OOB data with the tree classifier. The variable importance (VI henceforth) of X^1 is defined as the mean increase of prediction errors after permutation. The more the error increases, the more important the variable is (note that it can be slightly negative, typically for irrelevant variables).

Variable selection procedure

Let us give (following Genuer et al. (2010a)) some details about the variable selection procedure that we use here. We apply it on a simulated learning set of size $n = 100$ from the classification toys data model, introduced in Weston et al. (2003) and described in Section 2.2, with $p = 200$. It is an equiprobable two-class problem, $Y \in \{-1, 1\}$, with 6 true variables, the others being some noise.
The results are summarized in Figure 3.1. The true variables (1 to 6) are respectively represented by $(\triangleright, \triangle, \circ, \star, \triangleleft, \square)$. Based on to the learning set, we compute 50 forests with $n_{tree} = 2000$ and $m_{try} = 100$, which are values of the main parameters considered as well adapted for VI estimation (for more details, see Genuer et al. (2010a)).

Let us detail the four main steps of the procedure :

Variable ranking : First the variables are sorted according to the VI (averaged

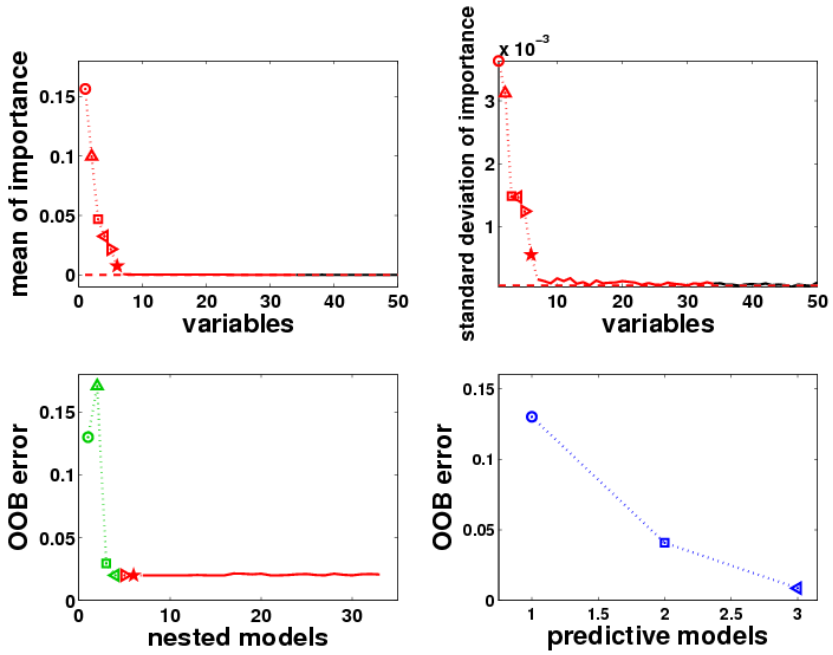

FIGURE 3.1 – Variable selection procedure for a toy dataset. The top left graph shows the variable ranking. The curve of the top right graph is used to determine the threshold (represented by the horizontal dashed line) needed in Elimination step. OOB errors of the nested models are plotted in the bottom left graph to illustrate the Interpretation step. The bottom right graph stands for Prediction step.

from the 50 runs) in descending order. Note that true variables are significantly more important than the noisy ones.

Elimination step : Keeping this order in mind, the corresponding standard deviations of VI are plotted. A threshold for importance is computed using this graph, and only the variables with an importance exceeding this level are kept. More precisely, the threshold is set as the minimum prediction value given by a CART model fitting this curve (for details, see Breiman et al. (1984)).

Interpretation step : Then, OOB error rates of the nested random forests models are computed ; starting from the one with only the most important variables, and ending with the one involving all important variables kept previously. The set of variables leading to the smallest OOB error is selected.

Prediction step : Finally a sequential variable introduction with testing is performed : a variable is added only if the error gain exceeds a data-driven threshold (see Genuer et al. (2010a)). The rationale is that the error decrease must be significantly greater than the average variation obtained by adding noisy variables.

3.1.3 Experiments and Results

Real Data

We used a real dataset related to an experiment on the representation of objects Eger et al. (2008). During the experiment, twelve healthy volunteers viewed objects of three different sizes and four different shapes, with 6 repetitions of each stimulus (refering to 6 sessions), resulting in a total of $n = 72$ images by subject. Functional images were acquired on a 3-T MR system with eight-channel head coil (Siemens Trio, Erlangen, Germany) as T2*-weighted echo-planar image (EPI) volumes. Twenty transverse slices were obtained with a repetition time of 2 s (echo time, 30 ms ; flip angle, 70° ; $2 \times 2 \times 2$-mm voxels ; 0.5-mm gap). Realignment, normalization to MNI space and GLM fit were performed with the SPM5 software. For our analysis we used the resulting session-wise parameter estimate images. The four different shapes of objects are pooled across the three sizes, and we are interested in discrimination between shapes. We used parcellation as a preprocessing, which allows important unsupervised reduction of dimensions. Parcellation uses Ward's algorithm (hierarchical agglomerative clustering) to create groups of voxels which have similar activity across trials. Thus, the signal is averaged in each parcel. The number of parcels created is fixed to 1000 for the whole brain.

Feature selection results for one subject

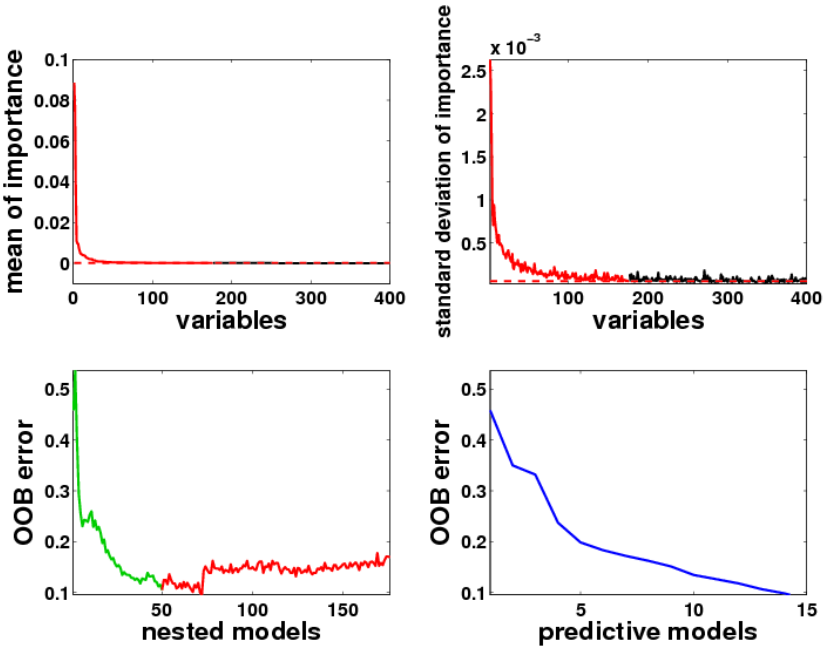

FIGURE 3.2 – Variable selection procedure for one subject. The graphs follow the exact same description as in Figure 3.1.

We apply the procedure described in Section 3.1.2 for the subject 2 of the study. The results are plotted in Figure 3.2. The horizontal dotted line of the top graphs indicates

the threshold, computed using standard deviations of VI (see the top right graph) and used in the top left graph to eliminate variables of small importance. Starting with all the 1000 variables, this elimination step retains 176 variables. The minimum OOB error rate in the bottom left graph is obtained by the RF model involving 50 variables, which constitute the interpretation set. Finally, the prediction procedure, illustrated by the bottom right graph, selects 15 variables.

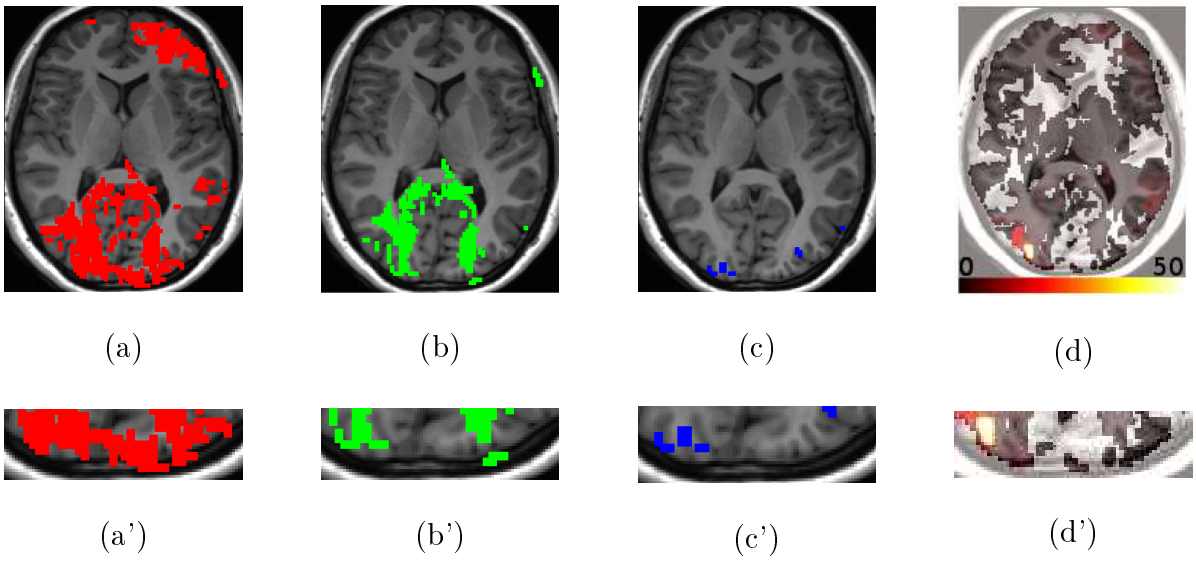

FIGURE 3.3 – Example of the different steps of the framework for one subject (slice z=6 mm). (a) Selected parcels after Elimination Step. (b) Selected parcels after Interpretation Step. (c) Selected parcels after Prediction Step. (d) Shows the parcels selected by the reference method, and their F-test values. (a'), (b'), (c') and (d') are magnifications of the occipital part.

Figure 3.3 shows the selected parcels for the different steps of the algorithm in one axial slice for subject 2 : sub-figures (a), (b) and (c) represent the variables selected in the Elimination step, Interpretation step and Prediction step, and (d) represent the variables selected by the reference method. Sub-figures (a'), (b'), (c') and (d') are magnifications of the occipital part. During the interpretation step, our algorithm keeps only three regions of the occipital cortex, reducing the features to a much smaller sets while keeping an accurate prediction (see Figure 3.4). In addition, the prediction step (c) allows to avoid redundancy in the features. The selected regions are different between the two hemispheres, while the interpretation step retained more symmetric regions. Finally, comparison whith sub-figure (d) highlights the most beneficial aspect of our method : we select very localised informative regions, while the reference method keeps lots of regions distributed in all brain.

Prediction results for the whole data

We perform a leave-one-session-out cross-validation : we successively train the classifier with all the sessions except one, and report the performance of the trained classifier

on the left out session. Importance-based feature selection was applied independently on the twelve datasets. The results are shown in Figure 3.4. The first row represents the classification score of RF for each subject (from left to right : all parcels, after Elimination step, after Interpretation step and after Prediction step). The average number of selected parcels across subjects is noted above each histogram, with the average classification score across all subjects.
The first graph of the second row shows the results of a cross-validated linear SVM : the optimal number of parcels to be kept (from 50 to 1000 parcels with a step of 50) for the linear SVM is selected using the F-statistic, by leave-one-out validation on the training set. The average number of selected parcels across subjects is equal to 350. The three last histograms of the second row show the results of a linear SVM : the parcels are selected by using a F-statistics, and the number of features used is equal to the number of parcels found by the three different steps of the RF-based algorithm. We can see that our algorithm gives better accuracy for the three steps of selection than the reference method (cross-validated linear SVM). And the three last histograms of the second row illustrate the fact that a linear SVM (coupled with F-test) do not manage to keep good accuracy with as few features as selected by our method.

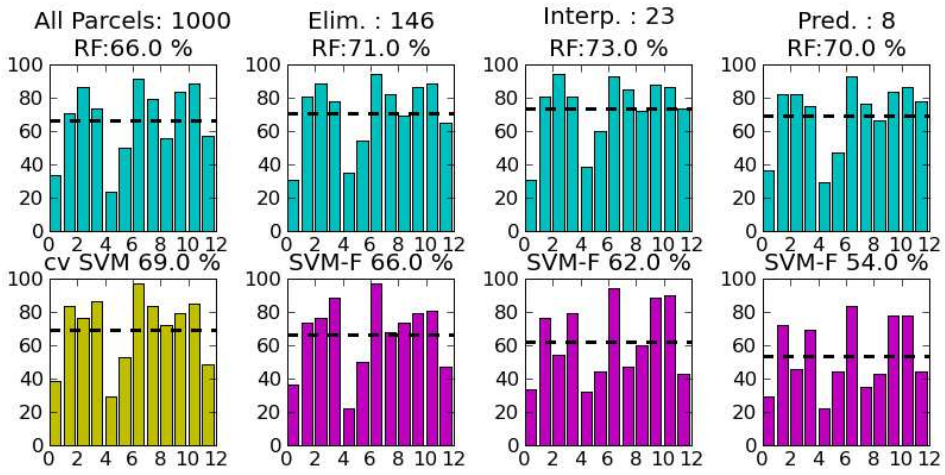

FIGURE 3.4 – Results on real data : rate of correct object identification, using the mean signal of 1000 parcels of the brain volume (chance level=25 %). The first row shows the prediction accuracy in each individual dataset, the mean classification score and the number of selected parcels for (from left to right) the whole brain, the Elimination step, the Interpretation step and the Prediction step. The first graph of the second row represents the results for the reference method. The three last histograms of the second row show the prediction accuracy of a linear SVM trained with the same number of parcels as above, but selected by F-statistics.

3.1.4 Discussion

This work presents the first application of a RF-based feature selection technique to brain state decoding. We show that it is competitive with state of the art method (univariate selection followed by linear SVM classifier). More importantly, the insensitivity of the correct classification rates along the different steps of feature reduction that is observed in Figure 3.4 for the RF model shows that our strategy manages to extract the statistical information of the data : it keeps much of the information while significantly reducing the dimension. This suggests that the multivariate RF variable importance index performs better than the classical univariate F-test score to detect the most predictive variables. Another noticeable aspect of the proposed procedure is that it is entirely data-driven : at each step of the procedure, thresholds are computed using only the data. So this procedure can adapt to lots of different applications, without the need of adding prior information (like e.g. a number of variables to be selected).

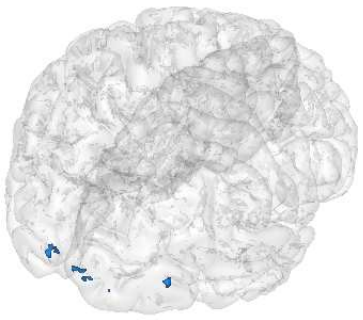

FIGURE 3.5 – Regions selected in at least 3 subjects among 12 by the last step of the RF-based selection. The MNI coordinates are :
[18, -102, 6]mm
[10, -100, 4]mm
[-12, -96, 0]mm
[50, -78, 6]mm.

From a neuroscientific point of view, we can notice that the spatial distribution of the selected parcels is quite informative : first, by avoiding redundancy, the algorithm is able to focus on few extremely precise regions of the brain without loss of accuracy. Moreover, starting from whole brain, the algorithm selects very few parcels in the occipital cortex, corresponding to visual areas. If we look at the regions selected for 3 subjects or more among the 12 subjects by the last step of the RF-based selection (see Figure 3.5), there are only few regions in the early visual cortex, and a slightly more anterior parcel. This is consistent with the fact that early visual cortex contains highly reliable signals discriminative of feature/shape differences between object exemplars, as long as no generalization across image changes is required (Cox and Savoy (2003), Eger et al. (2008)).

Conclusion

In this article, a multivariate and threshold-free feature selection algorithm based on Random Forests, yields an accurate selection for fMRI data analysis, and creates a highly informative set of very few features. Results on real data show the benefits of our approach for both interpretation and prediction, with higher accuracy and higher sparsity than the reference method.

3.2 Gametocytes infectiousness to mosquitoes : variable selection using random forests, and zero inflated models

3.2.1 Introduction

Malaria still represents a major health problem in more than one hundred tropical countries. The disease is caused by the parasite *Plasmodium* and its transmission occurs through the bite of an infective *Anopheles* female mosquito. In the last decades, insecticide and drug resistance has seriously hampered its control and alternative measures are urgently needed. Because *Plasmodium* transmission relies on the success of its development within the mosquito vector, called the sporogonic development, new strategies to fight malaria aim at controlling *Plasmodium* during the mosquito life cycle. Within the mosquito vector, malaria parasites undergo several life-stages and their successful development from one transition stage to an other will determine the outcome of infection. When ingested with the blood meal, male and female gametocytes fuse to form a zygote that differentiates into a mobile ookinete. The ookinete then traverses the midgut epithelium and encysts as an oocyst along the basal lamina. The oocyst, after several days of maturation, will release large number of sporozoites into the hemocoel. Sporozoites that will reach salivary glands will then be transmitted to a new host at a subsequent blood meal. *Plasmodium* parasites encounter severe losses during these successive phases and factors controlling parasite densities are not yet completely understood. Blood digestion processes and mosquito immune responses account for parasite decrease, but also the complex interplay between vector and parasite genotypes (Vaughan, 2007; Jaramillo-Gutierrez et al., 2009).

Transmission of *Plasmodium falciparum* sexual stages, the gametocytes, to the mosquito mainly depends on their maturity and density in the human host at the time of the mosquito bite. Even if it has been demonstrated that high gametocyte densities do not guarantee high mosquito infection, a greater infection of mosquitoes is generally observed with higher gametocyte densities (Hogh et al., 1998; Drakeley et al., 1999; Targett et al., 2001; Boudin et al., 2004; Paul et al., 2007; Nwakanma et al., 2008). Gametocyte densities vary greatly between human hosts, due to host acquired immunity, genetic factors of the parasite strain and other environmental parameters (blood quality, fever, anemia, anti-malarial drug uptake). In malaria endemic areas, human hosts are typically infected with multiple genotypes of parasites (Day et al., 1992; Babiker et al., 1999; Anderson et al., 2000; Nwakanma et al., 2008) and within-host competition of parasite genotypes is likely to drive transmission success. Indeed, from experiments using *Plasmodium* animal models, it has been shown that different genotypes of parasites in mixed infections have distinct ability to transmit, the more virulent strain having a competitive advantage (de Roode et al., 2005; Bell et al., 2006; Wargo et al., 2007). If different models have been proposed to correlate the gametocyte density to the transmission success of wild isolates of *Plasmodium falciparum* (Pichon et al., 2000;

Boudin et al., 2005; Paul et al., 2007), to date no study related the outcome of infection to parasite complexity within the gametocyte population. Understanding relationships between co-infecting genotypes and how they influence the disease transmission is however of great importance as these might help to predict the spread of resistant strains of parasites and guide strategies for malaria control.

In this paper, we investigate how density and genetic diversity of gametocytes impact on infectiousness to mosquitoes. We analyze mosquito infection data consisted of oocyst counts with corresponding gametocyte data : densities and genotypes at 7 microsatellite loci. Data were obtained from experiments of membrane feeding of a local colony of *Anopheles gambiae* mosquitoes on blood from volunteers naturally infected by *Plasmodium falciparum* isolates from Cameroon. Gametocyte genotypes are occurrences of several unordered categorical variables, each having numerous levels. Therefore the number of variables plus attendant interactions is at least of order of the sample size. We considered as response variables : the intensity of infection as measured by the mean of oocyst counts in infected mosquitoes, and the infection prevalence defined by the proportion of mosquitoes that became infected. The high number of variables in our data set will obviously lead to over-fitting of many familiar regression techniques such as general linear model (GLM). In addition, we deal with unordered categorical variables with several levels and potentially accompanying interactions. Therefore, following Segal et al. (2001), we use regression trees techniques.

We address the problem of selecting the most influent variables related to the response variable by applying a variable selection procedure, which comes from Genuer et al. (2010a), and is based on variable importance from random forests (Breiman, 2001). The resulting method is completely non-parametric and thus can be used on data with a large number of variables of various types. Moreover, it solves the two following constraints about variable selection : 1) to find all variables highly related to the response variable ; and 2) to find a small number of variables sufficient for a good prediction of the response variable. The selected variables are then assessed in a modeling for oocyst count which takes into account the complexity of the experiment we deal with. The key point of our modeling is the introduction of a new unobserved variable that enables to distinguish two possible sources of non infected mosquitoes. Indeed, the heterogeneity in the quantity and quality of gametocytes in blood-meal (Vaughan, 2007), and natural variation in mosquito susceptibility (Riehle et al., 2006) are well known phenomena. We then suggest here that mosquitoes with no oocyst can be non infected either because they did not ingest enough gametocytes with the blood-meal, or because they were refractory to the ingested parasites. We fitted a model, called Zero-Inflated (ZI) model, which is a two components mixture model combining a point mass at zero with a proper count model. Since we deal with count data, the typical candidate models were Zero-Inflated Poisson (ZIP) and Zero-Inflated Negative Binomial (ZINB) ; ZINB having a slight advantage because it captures over-dispersion which is likely to appear in such data.

The rest of the paper is organized as follows. Section 3.2.2 presents the data to

be analyzed in Subsection 3.2.2, the principle of variable selection based on variable importance from random forests in Subsection 3.2.2, and the modeling of oocyst count in Subsection 3.2.2. Section 3.2.3 is devoted to the application of these methods on our data. Finally a discussion is given in Section 3.2.4.

3.2.2 Material and methods

Data collection and description

The data we considered consist of parasite densities and genotypes at 7 microsatellite loci for gametocyte isolates of *Plasmodium falciparum* on one hand, and oocyst counts 7 days post feeding for each engorged females on the other hand. *Plasmodium falciparum* gametocyte carriers were identified among asymptomatic children aged from 5 to 11 in primary schools of the locality of Mfou, a small town located 30 km apart from Yaounde, the Cameroon capital city. Volunteers were enrolled upon signature of an informed consent form by their parents or legal guardian. The protocol was approved by the National Ethics Committee of Cameroon. Gametocyte densities were expressed as the number of parasites seen against 1 000 leukocytes in a fresh thick blood smear, assuming a standard concentration of 8 000 leukocytes per μl (see Table 3.1 for summary of log-transformed gametocyte densities). Venous blood (2 to 3 mL) was taken from consenting gametocyte carriers, centrifuged and the serum replaced by a non-immune AB serum. This procedure avoids the introduction of human transmission blocking factors in the experiment. 3 to 5 old females of a laboratory strain of *Anopheles gambiae* mosquito were used for the membrane feeding assays placed in cups of approximately 60-80 mosquitoes. Females were allowed to feed for 20 minutes through a Parafilm membrane on glass feeders maintained at $37°C$ and fully engorged females were kept in insectar until dissections 7 days post-infection. Midguts were removed, stained in a 0.4% Mercurochrome solution and the number of developed oocysts counted by light microscopy ($X20$ *lens*). A total of 7 364 mosquitoes (see Table 3.1) were dissected, giving a mean of 39 females per experiment.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
N	11.000	29.000	38.000	39.380	47.000	79.000
log_gameto	1.816	3.156	3.832	3.973	4.612	7.742

TABLE 3.1 – Summary of the numbers of mosquitoes per isolate (N) and log-transformed of gametocyte densities (**log_gameto**).

Gametocytes were separated from 1 mL of serum free blood using MACS® columns as previously described (Ribaut et al., 2008). DNA extractions from purified gametocytes were performed with DNAzol® and 20 ng of gametocyte DNA were subjected to whole-genome amplification (WGA) using the GenomiPhi V2 DNA Amplification Kit to generate sufficient amounts of DNA for microsatellite genotyping. Genetic polymor-

phism was assessed at 7 microsatellite loci as previously described (Annan et al., 2007). Their chromosome location and GenBank accession number are as follows : POLYa (chr. 4, G37809), TA60 (chr. 13, G38876), ARA2 (chr. 11, G37848), Pfg377 (chr. 12, G37851), PfPK2 (chr. 12, G37852), TA87 (chr. 6, G38838), and TA109 (chr.6, G38842). Alleles were analyzed using `GeneMapper`® software. Multiple alleles were scored when minor peaks were at least 20% of the height of the predominant allele. The number of observed alleles per locus is 21, 9, 10, 5, 15, 10 and 17 respectively (see Figure 3.6).

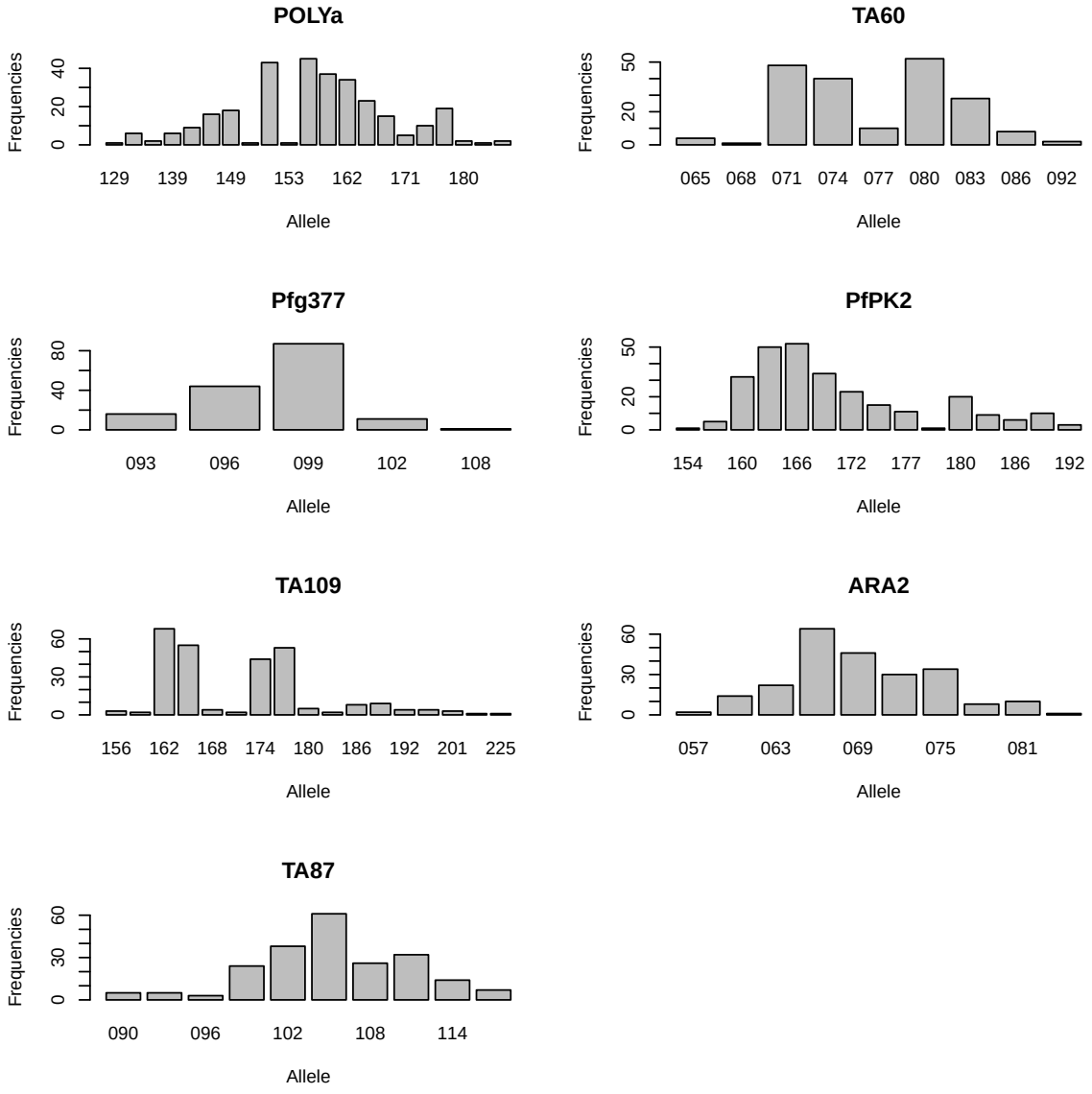

FIGURE 3.6 – Alleles detected for the 7 microsatellite loci and their frequencies in *Plasmodium falciparum* gametocyte carriers.

Feedings for which the number of dissected mosquitoes was below 20 were not consi-

dered. Then 110 experiments were included in the analysis.

Variable selection procedure

The selection procedure we considered is based on variable importances (VI) from random forests (RF). The principle of RF is to aggregate regression or classification trees built on several bootstrap samples drawn from the learning set (more details are given in Appendix 3.A). It is shown to exhibit very good performance for lots of diverse applied situations (Breiman, 2001). Moreover, it computes a variable importance index, defined in Appendix 3.A. Roughly, this index is a measure of the degradation of forest predictions when values of a variable are permuted.

RF variable importance is the key point of the selection procedure (see Genuer et al. (2010a) for more backgrounds on RF variable importance). This procedure presents two main benefits. First the method is completely non-parametric and can be applied on data with lots of variables of various types. Second, it achieves two main variable selection objectives : (1) to magnify all the variables related to the response variable, even with high redundancy, for interpretation purpose ; (2) to find a parsimonious set of variables sufficient for prediction of the outcome variable.

Let us now describe the procedure, which comes from Genuer et al. (2010a), with the following algorithm. The R package `randomForest` (Liaw and Wiener, 2002; R Core Team Development, 2009) was used in all computations.

To both illustrate and give details about this procedure, we apply it on a simulated dataset with $n = 200$ observations described by 25 continuous variables and 25 binary variables. We assume standard normal distribution $\mathcal{N}(0,\ 1)$ for all continuous variables and binomial distribution $\mathcal{B}(0.5)$ for all binary variables. We consider the following linear model

$$Y = \sum_{j=1}^{25} \beta_{c_j} X_{c_j} + \sum_{j=1}^{25} \beta_{b_j} X_{b_j}$$

in which only 8 over a total of $p = 50$ variables are related to the outcome, the others being just noise. The set of significant variables is composed by the first 4 continuous variables $\left(X_{c_j}\right)_{1 \leqslant j \leqslant 4}$ and the first 4 binary ones $\left(X_{b_j}\right)_{1 \leqslant j \leqslant 4}$. Their associated coefficients are given by

$$(\beta_{c_j})_{1 \leqslant j \leqslant 25} = (\beta_{b_j})_{1 \leqslant j \leqslant 25} = (4, 4, 2, 2, 0, \ldots, 0).$$

We also assume a 0.9 correlation between X_{c_1} and X_{c_2}, X_{c_3} and X_{c_4}, X_{b_1} and X_{b_2}, and X_{b_3} and X_{b_4}.

The selection process uses a certain number $nfor$ of random forests. In addition of this number, the user has also to provide the number $ntree$ of trees in each random forest, and the number $mtry$ of variables among which to select the best split at each node. The default parameters in the **R** package `randomForest` we used are $mtry = p/3$, $ntree = 500$. In our example, we choose the following parameters : $mtry = p/3$, and

we choose $nfor = 50$ and $ntree = 1000$ to increase the VI stability. The results are summarized in Figure 3.7.

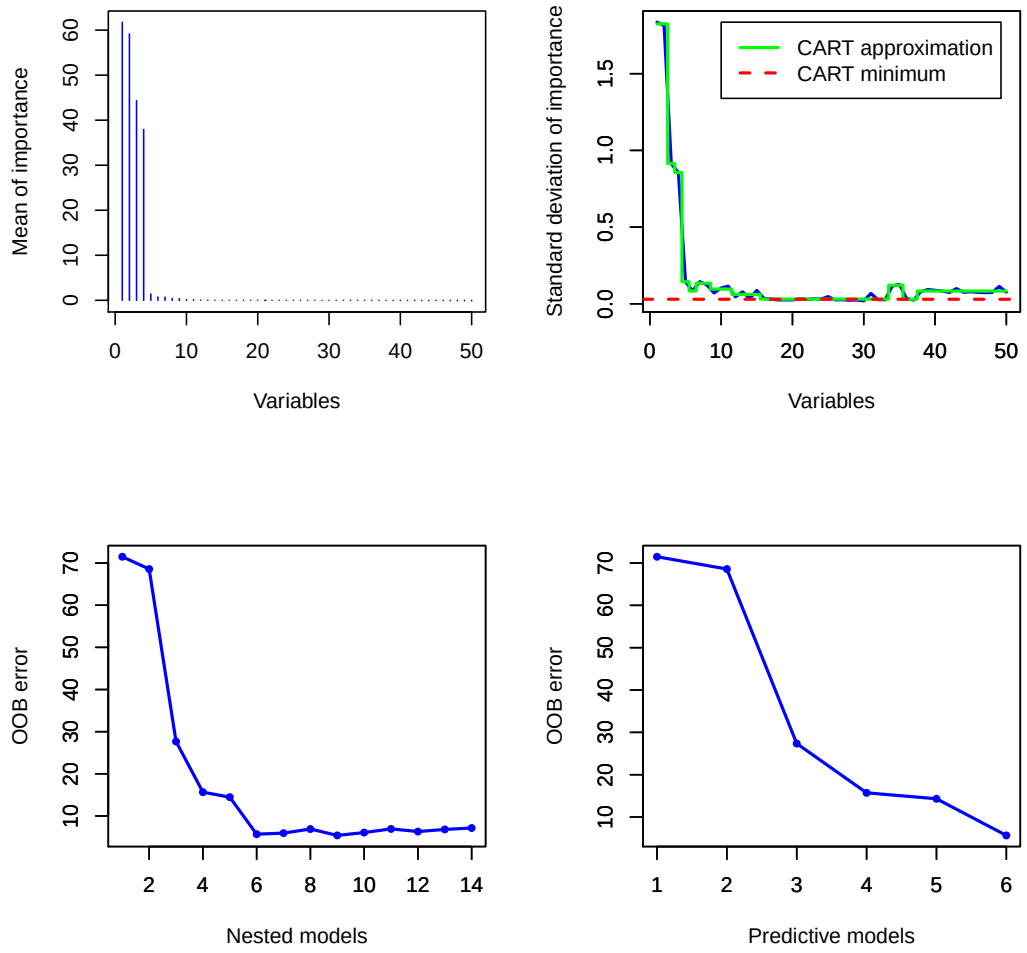

FIGURE 3.7 – Variable selection procedures for interpretation and prediction for simulated data

Let us detail the main stages of the procedure together with, in italics, the results obtained on simulated data. In the following, out of bag (OOB) error refers to an estimation of the prediction error (which is defined in Appendix 3.A and is close to a cross-validation estimate).

– **Elimination step**
First the variables are sorted in descending order according to VI (averaged from the $nfor$ runs).
The result is drawn on the top left graph. The 8 *variables of interest arrive in the first* 8 *positions of the ranking.*

Keeping this order in mind, the corresponding standard deviations of VI are plotted. A threshold for importance is computed using this graph. More precisely, the threshold is set as the minimum prediction value given by a Classification And Regression Tree (CART) model fitting this curve (for details about CART, see Breiman et al. (1984)). Then only variables with an averaged VI exceeding this level are kept. This rule is, in general, conservative and leads to retain more variables than necessary, in order to make a careful choice later.
The standard deviations of VI can be found in the top right graph. We can see that true variables standard deviation is large compared to the noisy variables one, which is very close to zero. The threshold leads to retain $p_{elim} = 14$ *variables. Note that the threshold value is based on VI standard deviations (top right panel of Figure 3.7) while the effective thresholding is performed on VI mean (top left panel of Figure 3.7).*

– **Interpretation step**
Then, OOB error rates (averaged on $nfor$ runs and using default parameters) of the nested random forests models are computed ; starting from the one with only the most important variable, and ending with the one involving all important variables kept previously. The set of variables leading to the smallest OOB error is selected.
Note that in the bottom left graph the error decreases and reaches its minimum when the first $p_{interp} = 9$ *variables are included in the model. This set of selected variables for interpretation contains the* 8 *true variables plus one noisy one. Note that the associated error is closed to the one of the model with the* 6 *first variables (see bottom left panel of Figure 3.7) suggesting that a smaller model should be preferred for prediction purposes.*

– **Prediction step**
Finally a sequential variable introduction with testing is performed : a variable is added only if the error gain exceeds a data-driven threshold. The rationale is that the error decrease must be significantly greater than the average variation obtained by adding noisy variables.
The bottom right graph shows the result of this step, the final model for prediction purpose involves 6 *out of the* 8 *true variables. It is of interest that each of the two true variables non-selected is correlated to one selected variable. The threshold is set to twice the mean of the absolute values of the first order differentiated OOB errors between the model with* $p_{interp} = 9$ *variables (the model we selected for interpretation, see the bottom left graph) and the one with all the* $p_{elim} = 14$ *variables :*

$$ave_{jump} = \frac{1}{p_{elim} - p_{interp}} \sum_{j=p_{interp}}^{p_{elim}-1} | \, errOOB(j+1) - errOOB(j) \, |$$

where $errOOB(j)$ *is the OOB error of the RF built using the* j *most important variables.*

Since the number of variables after the variable elimination step is small (14), we

tried some variants more computationally expensive, in order to validate the two last steps of the algorithm. Instead of the interpretation step, we launch a forward procedure. The principle is, at each time, to seek the best variable (in terms of OOB error rate, averaged on $nfor$ runs and using default parameters) to add in the current variable set. The set of variables leading to the smallest OOB error is then selected.

For our example, it leads, as the interpretation step, to retain the 8 *true variables plus one noisy variable (this last noisy variable being different from the one selected by interpretation step). We remark however that the initial ranking according to VI is quite changed with this procedure.*

To validate the prediction step, we tried an exhaustive procedure, i.e. we compute the OOB error rate (averaged on $nfor$ runs and using default parameters) for all models formed with the variables selected by the forward procedure. The set of variables leading to the smallest OOB error is then selected.

This procedure selects all 9 *variables selected previously.*

This validates the interpretation and the prediction step of our algorithm, since the variables sets in these variants are close to ours. In addition the errors reached by the two procedures are comparable. However this comparison was done on the easy simulated dataset we considered in this section.

Modeling oocyst count with Zero-Inflated models

The key point of our modeling is to consider that there are two possible sources of non-infected mosquitoes. First, some mosquitoes may not ingest enough parasites with sufficient sex-ratio to ensure fertilization. The reason is seemingly the high heterogeneity in the number of gametocytes in blood-meals (Pichon et al., 2000). Second, some other mosquitoes may not be genetically susceptible to the parasites ingested (Riehle et al., 2006). We introduce a new variable U materializing this situation of non-infected mosquitoes : for mosquito j fed with blood coming from gametocytes carrier i,

$$U_{i,j} = \begin{cases} 1 & \text{if enough parasites are present in its blood-meal} \\ 0 & \text{otherwise.} \end{cases}$$

$U_{i,j}$ is an unobserved variable in our experiment. We assume that for a given i, $U_{i,1}, \ldots, U_{i,n_i}$ are independent and identically distributed. Here n_i is the number of mosquitoes associated to gametocytes carrier i. For any gametocytes carrier i, denote by

$$\pi_i := P\left(U_{i,j} = 0\right)$$

the probability that mosquito j does not ingest enough gametocytes in its blood-meal. Let $Y_{i,j}$ be the number of oocysts developed in mosquito j associated to gametocytes carrier i. The probability distribution of $Y_{i,j}$ is given by

$$P\left(Y_{i,j} = y_{i,j}\right) = \pi_i \mathbb{1}_{(y_{i,j}=0)} + \left(1 - \pi_i\right) P\left(Y_{i,j} = y_{i,j} | \; U_{i,j} = 1\right), \tag{3.1}$$

where $P(Y_{i,j} = y_{i,j} | U_{i,j} = 1)$ is a suitable count probability distribution.

Consequently, for any gametocytes carrier i, the zero class is a mixture of two components with π_i and $1 - \pi_i$ as the mixture proportions. The resulting model of probability distribution is known as a zero-inflated count model. Such a model is a two components mixture model combining a point mass at zero with a count distribution such as Poisson, geometric or negative binomial (see Zeileis and Jackman (2008) and references therein). Thus there are two sources of zeros : zeros may come from point mass or from count component. In our framework, the zeros coming from the point mass are assumed to represent mosquitoes which did not ingest enough gametocytes to produce an infection.

Let $\lambda_i := \mathbb{E}(Y_{i,j} | U_{i,j} = 1)$ be the conditional mean of the count component. In the regression setting, both the mean λ_i and the excess zero proportion π_i are related to covariates vectors $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,p})$ and $\mathbf{z}_i = (z_{i,1}, \ldots, z_{i,q})$, respectively. The components of these covariates are typically the observations of the previously selected variables. They contain gametocyte density and / or their genetic profile. We consider canonical link functions **log** and **logit** for the mean of count component and the point mass component respectively. The corresponding regression equations are

$$\begin{cases} \lambda_i &= \exp(\beta_0 + \beta_1 x_{i,1} + \ldots + \beta_p x_{i,p}) \\ \pi_i &= \dfrac{\exp(\gamma_0 + \gamma_1 z_{i,1} + \ldots + \gamma_p z_{i,q})}{1 + \exp(\gamma_0 + \gamma_1 z_{i,1} + \ldots + \gamma_p z_{i,q})}, \end{cases}$$

where $\beta := (\beta_0, \ldots, \beta_p)$ and $\gamma := (\gamma_0, \ldots, \gamma_q)$ are the parameters to be estimated. Note that different sets of regressors can be specified for the zero inflated component and count component. In the simplest case, only an intercept is used for modeling the unobserved state (zero vs. count).

Typical candidate of zero-inflated models for count data are zero inflated Poisson (ZIP) and zero-inflated negative binomial (ZINB) (see Xiang et al. (2007) and references therein). ZINB and ZIP specifications are given in Appendix 3.B. For the estimation of the parameters of these models, we used the package named `pscl` (Zeileis and Jackman, 2008) in $\mathbb{R}$ statistical software (R Core Team Development, 2009).

3.2.3 Application on the real data

Variable selection

Here, the results are given following the main stages of the selection procedure given in Subsection 3.2.2. The details are given once, in the case where the response variable is the infection prevalence of mosquitoes measured by proportion of infected mosquitoes. We will just give the selected variables at each stage in the other case where the response variable is the mean number of oocysts per infected mosquitoes. In these results, the binary variables associated to the observed alleles are coded as `locus_allele`. For

example, $Pfg377\_093$ is allele 093 at locus $Pfg377$. In addition to the log-transformed of gametocytes density ($log_g ameto$), we also consider the multiplicity of infection (MOI) defined as the maximum number of observed alleles across the considered microsatellite loci.

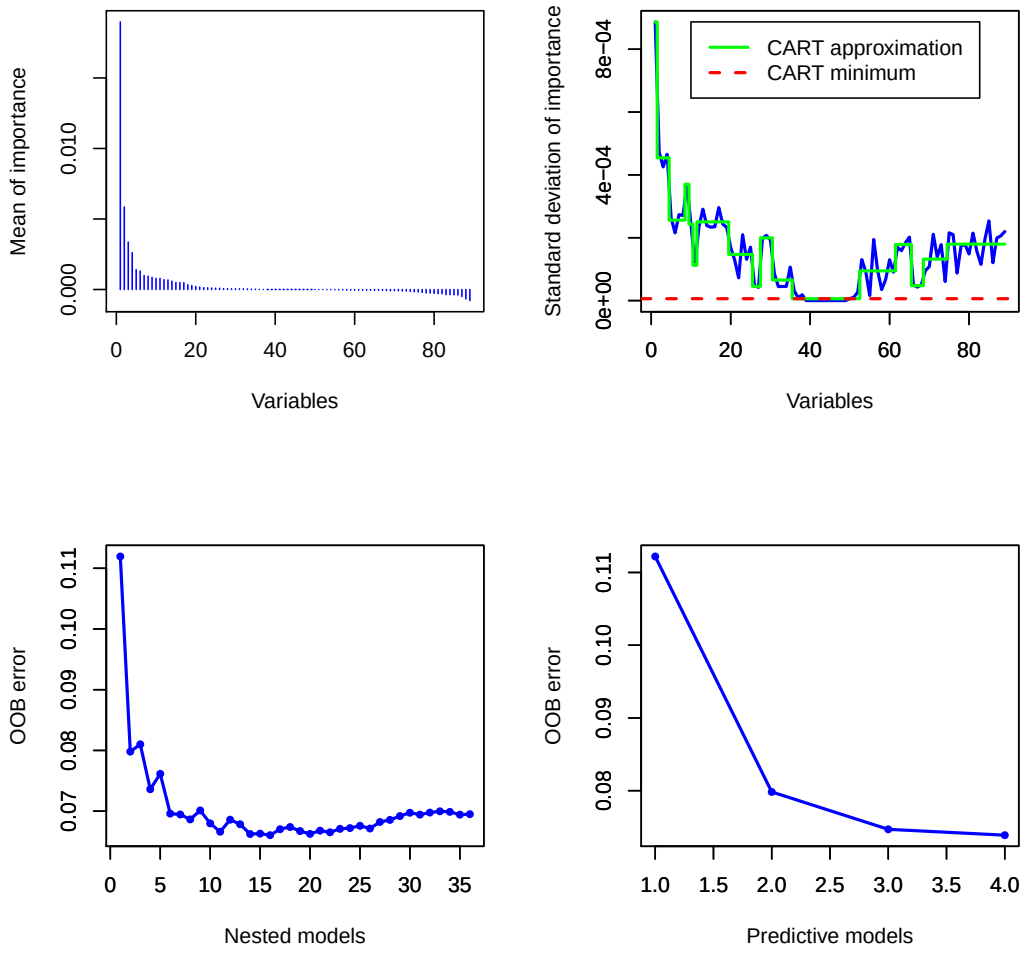

FIGURE 3.8 – Variable selection for interpretation and prediction. The response variable is the infection prevalence measured by the proportion of infected mosquitoes.

Here are the main stages of the procedure.

- **Elimination Step**
  - The top left panel in Figure 3.8 gives the VI mean of all the 88 variables sorted in decreasing order.
  - The top right panel of Figure 3.8 plots the standard deviations of VI and the fitted CART model. The threshold $\min_{CART}$ represented by the horizontal dashed line leads to retain $p_{elim} = 36$ variables over 88.

– **Interpretation Step**
This step is illustrated in the bottom left panel of Figure 3.8 in which the minimum OOB error rate is reached with $p_{interp} = 11$ variables for interpretation :

$$S_{interp} = \{log\_gameto,\ Pfg377\_093,\ PfPK2\_180,\ MOI,\ Pfg377\_102,\ PfPK2\_183,\ Pfg377\_099,\ TA60\_071,\ PfPK2\_169,\ PfPK2\_166,\ POLYa\_135\}.$$

– **Prediction Step**
The bottom right panel in Figure 3.8 shows the behavior of the OBB error of the nested models corresponding to the selected variables for prediction :

$$S_{pred} = \{log\_gameto,\ Pfg377\_093,\ MOI,\ PfPK2\_183\}.$$

The 4 selected variables in S_{pred} lead to the OOB error of 0.074. We also launch the variant based on forward and exhaustive search of the selection procedure. Finally it retains a set of 9 variables containing S_{pred}. The associated OOB error is 0.062 which is not far from 0.074. So we prefer a model with variables in S_{pred} which is more parsimonious.

The same procedure was applied when the outcome variable is the infection intensity as measured by the mean number of oocysts in infected mosquitoes. Figure 3.9 gives the behavior of VI and the OOB error at each stage of the selection procedure. 25 variables were selected by thresholding the VI in the first stage, the 2 most important being *log_gameto* and *MOI*. Even if only *log_gameto* is selected in the interpretation and prediction stages, we also keep *MOI*. Indeed, as can be seen in the bottom left graph of Figure 3.9, the model with these two variables is still competitive compared with the model built with *log_gameto* only.

Zero-Inflated models fitting oocyst count

Zero-Inflated negative binomial (ZINB) and Poisson (ZIP) were fitted to the data in two situations : (i) using only log-transformed of the gametocyte density as covariate, (ii) using the set of variables selected for prediction for both infection prevalence and infection intensity (see Subsection 3.2.3). The estimates of the parameters of ZINB and ZIP models are given in Table 3.2 and 3.3.

In situation (i), it is of interest how the zero counts are captured by the two models : they perfectly predict the observed number of non infected mosquitoes (see the left panel of Figure 3.10). Also, the mean oocyst estimates from both two models are similar (see the right panel of Figure 3.10). But according to the $\mathcal{X}^2$ goodness-of-fit test ($\mathcal{X}^2 = 48.162$, $df = 45$, $p.value \geq 0.3461$ for ZINB against $\mathcal{X}^2 = 2964.606$, $df = 46$, $p.value = 0$ for ZIP model), ZINB model is more adapted to our data. Over-dispersion is probably the main reason : there are more mosquitoes with no or few oocysts than the ones with high oocyst loads. ZIP model underestimates the number of mosquitoes with lower oocyst loads (see the left panel of Figure 3.10). We then consider the ZINB model in the rest of the analysis.

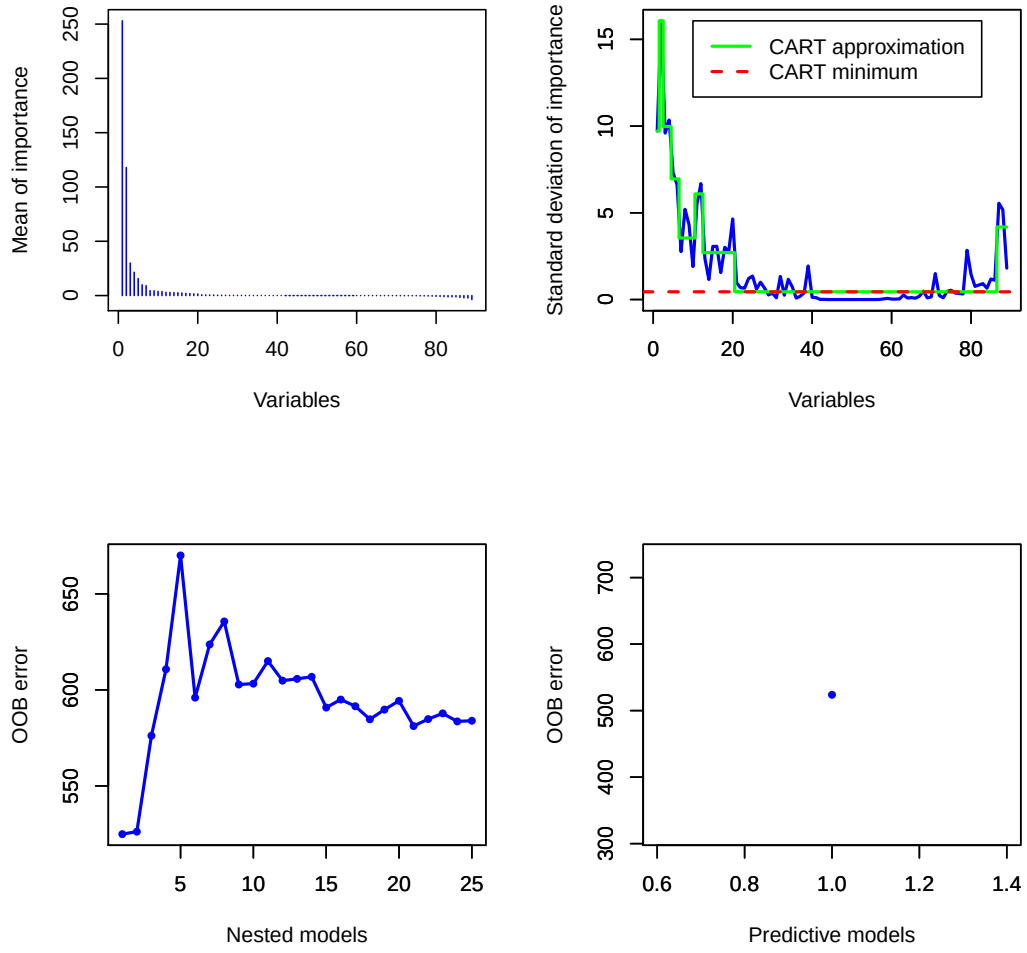

FIGURE 3.9 – Variable selection for interpretation and prediction. The response variable is the mean number in infected mosquitoes.

In situation (ii), since the data are over-dispersed, only ZINB is considered. The selected variables in the prediction step of our variable selection process using the infection prevalence as response variable are used in point mass component, and the ones using the infection intensity as response variable are used in the count component. Recall that the infection prevalence is measured by the proportion of mosquitoes that became infected, and the infection intensity by the mean number of oocysts in infected mosquitoes. We found that allele PfPK2_183 is the only variable not significant ($Z = -0.8329$, $p.value \geq 0.40$). In contrary, gametocyte density log_gameto, gametocyte genetic complexity MOI and allele 093 of locus $Pfg377$ significantly influence the oocyst load in mosquitoes. The significance of the gametocyte density confirms the result obtained by ZINB model in situation (i). The significance of the effect of MOI in both zero and

		Estimate	Std. Error	z value	Pr(>|z|)	
ZINB						
Count	(Intercept)	-1.3021	0.1163	-11.1985	4.1E-29	***
	log_gameto	0.8402	0.0257	32.6835	2.7E-234	***
	Log(theta)	-0.5693	0.0557	-10.2235	1.6E-24	***
Zero	(Intercept)	0.0029	0.2405	0.0119	9.9E-01	
	log_gameto	-0.2618	0.0531	-4.9294	8.2E-07	***
ZIP						
Count	(Intercept)	-0.7941	0.0199	-40.0016	0.0E+00	***
	log_gameto	0.7717	0.0036	213.9455	0.0E+00	***
zero	(Intercept)	1.4508	0.1284	11.2996	1.3E-29	***
	log_gameto	-0.4383	0.0316	-13.8930	7.0E-44	***

TABLE 3.2 – Maximum likelihood estimates of the parameters of ZINB and ZIP models with data from 110 gametocyte carriers using only log_gameto as the variable. Significant codes : 0 '***'; 0.001 '**'; 0.01 '*'; 0.05 '.'; 0.1 ' '. $\mathcal{X}^2$ Goodness-of-fit test : $\mathcal{X}^2 = 47.0992$, $df = 45$, $p.value \geq 0.3866$ for ZINB against $\mathcal{X}^2 = 2834.848$, $df = 46$, $p.value = 0$ for ZIP model

		Estimate	Std. Error	z value	Pr(>|z|)	
ZINB						
Count	(Intercept)	-0.9985	0.1436	-6.9539	3.6E-12	***
	log_gameto	0.8009	0.0261	30.6432	3.3E-206	***
	MOI	-0.0333	0.0158	-2.1058	3.5E-02	*
	Log(theta)	-0.5210	0.0500	-10.4296	1.8E-25	***
Zero	(Intercept)	0.9651	0.2679	3.6030	3.1E-04	***
	log_gameto	-0.3769	0.0534	-7.0615	1.6E-12	***
	Pfg377_093	1.2242	0.1204	10.1717	2.7E-24	***
	MOI	-0.1499	0.0328	-4.5711	4.9E-06	***
	PfPK2_183	-4.5225	5.4301	-0.8329	4.0E-01	

TABLE 3.3 – Maximum likelihood estimates of the parameters of ZINB and ZIP models with the data from 110 gametocytes carriers, using $S_{pred} = \{log\_gameto, Pfg377\_093, MOI, PfPK2\_183\}$ as variables. Significant codes : 0 '***'; 0.001 '**'; 0.01 '*'; 0.05 '.'; 0.1 ' '.

count components is very interesting. The significance is more important in the zero component (t-test $Z = -4.5711$, $p.value < 4.9e-06$) than in the count one (t-test $Z = -2.1058$, $p.value < 3.5e-02$); also note that the correlation is negative in both two components ($\widehat{\beta}_{MOI} = -0.0333$ and $\widehat{\gamma}_{MOI} = -0.1499$ in count and zero components respectively). So mono infected gametocyte isolates increase the probability that a mosquito do not ingest enough parasites to ensure the transmission success of *Plasmodium*

through its vector mosquito. Hence, low values of MOI tend to decrease the infection prevalence. In contrary, a lower genetic diversity of gametocytes in an isolate increases the mean number of oocysts in infected mosquitoes. Also note that the presence of allele 093 of the genetic marker Pfg377 increases the proportion of non-infected mosquitoes ($\widehat{\gamma}_{Pfg377\_093} = 1.2242$, $SE = 0.1204$; t-test $Z = 10.177$, $p - value < 2.7e - 24$).

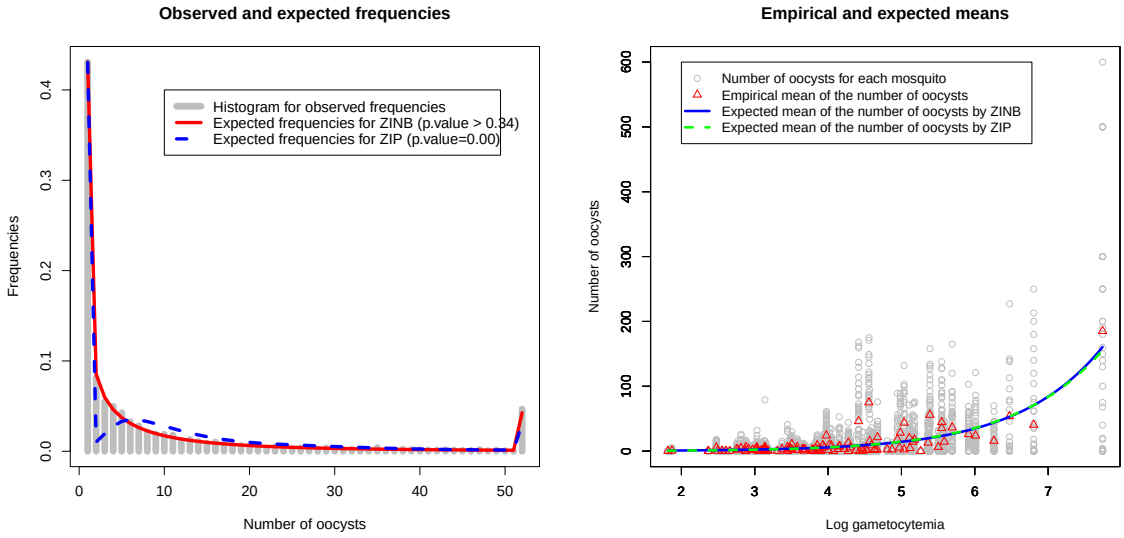

FIGURE 3.10 – The right panel gives observed and predicted frequencies from ZINB and ZIP models, and the right one the empirical and predicted mean number of oocysts versus log-gametocytemia.

3.2.4 Discussion

Plasmodium development within its vector mosquito follows complex biological processes and factors controlling parasite dynamics are not well understood. In the rodent malaria parasite *Plasmodium berghei*, it has been previously shown that the efficiency of parasite transmission from one developmental stage to another followed density-dependent models and the best fitted mathematical model differed from one developmental transition to the other one (Sinden et al., 2007). For natural populations of *Plasmodium falciparum*, the human malaria parasite, modeling becomes more challenging because of unknown genetic factors and uncontrolled environmental parameters. Nonetheless, Paul et al. (2007) found a sigmoid relationship between *Plasmodium falciparum* gametocyte density and mosquito transmission and the authors argued that parasite aggregation within mosquitoes represents an adaptive mechanism for transmission efficiency. The great variability in *Plasmodium falciparum* oocyst numbers observed in natural *Anopheles gambiae* populations suggests that parasite transmission is the result of complex interactions between vectors and parasites, which rely on both genetic and environmental factors. Understanding factors that determine transmission intensity

and then parasite population structure is of crucial importance in predicting the impact of current malaria control strategies.

In this study, we analyzed patterns of mosquito infection from experiments performed with field isolates of *Plasmodium falciparum* from Cameroon, an area of high malaria endemicity. The infection prevalence for each parasite isolate was scored using two variables, the mean number of oocysts developed within the mosquito midgut 7 days post-infection and the number of mosquitoes carrying at least one oocyst. Gametocyte isolates were genetically characterized at seven microsatellite loci, allowing estimation of the number of co-infecting parasite clones, the MOI, and of the genetic polymorphism, given by the number of alleles at each locus. In such a situation with potentially a high number of unordered categorical variables with numerous levels and accompanying interactions, many familiar statistical techniques such as GLM over-fit the data. Then we had to face the problem of selecting the most important variables related to the outcome variables. We have addressed this issue with a selection procedure based on variable importance from random forests. The procedure has two main benefits. First, it is completely non-parametric and thus can be used on data with lots of variables of various types. Second, it answers the two distinct objectives about variable selection : (1) to find all variables related to the outcome variable and (2) to find a small number of variables sufficient for a good prediction of the outcome variable.

Recall that we are in a critical situation with the number of variables of the order of the sample size ($n = 110$). The application of the variable selection procedure on our data revealed that only 4 among the 88 variables we considered suffice to predict the infectiousness of *Plasmodium falciparum* to *Anopheles gambiae* in our experimental settings. The procedure indicates that the log-transformed of gametocyte density is the most influent variable for both infection prevalence and infection intensity. But whereas gametocyte density was positively correlated with mean oocyst burden, it was negatively correlated with mosquito infection prevalence. This contrasting feature of transmission parameters probably reflects that *Plasmodium* parasites have developed complex and diverse strategies to ensure their transmission through the mosquito vector. The fact that higher oocyst counts are found for higher gametocyte densities conforms to previous observations showing that infectiousness generally increases with gametocytemia. Interestingly, Paul et al. (2007) described upper gametocyte densities at which mosquito infection rates level off, which is consistent with our results. In their models, mosquitoes with no oocyst were treated as non infected without further consideration about the putative factors responsible of the non infected status. However, a mosquito population fed on the same gametocyte carrier results in individuals carrying high number of parasites while others do not have any. Failure to infection of a mosquito can result from various factors such the heterogeneity of gametocyte environment (Vaughan, 2007) and natural variation in mosquito susceptibility in the other hand (Riehle et al., 2006). We have described in this article an approach based on that the non-infected mosquitoes represent two distinct populations : one genetically refractory vector population and another population for which the no-oocyst status results from other biological or interacting factors. Further study to quantify the gametocyte uptake in mosquitoes fed on

a single carrier would help to determine the individual variation of gametocyte density between blood-meals, and thus the real part of mosquitoes that are refractory and those that did not develop any oocyst because of other environmental factors. Nonetheless, our model perfectly predicts the number of non infected mosquitoes. Our fitting models revealed that over-dispersion of oocysts affects mosquito infection intensity. In addition, a higher over-dispersion of oocysts is observed for mosquitoes fed on blood with high gametocyte density (over 90 gametocytes/μl). The over-dispersed distribution of oocysts has often been explained as the result of the aggregation of gametocytes in the capillary blood at the time of the mosquito bite (Pichon et al., 2000). In this study, mosquitoes were membrane fed and membrane feeding is thought to suppress gametocyte over dispersion (Vaughan, 2007). Nonetheless, the fact that the maximum aggregation is found for high gametocyte densities is indicative of aggregation of sexual stages ; aggregation may occur within the mosquito midgut after parasite intake and genetic factors from the parasites may play a role in parasite recognition. This speculation is consistent with the hypothesis of adaptive aggregation, where gamete aggregation would favor fertilization and then increase infection intensity (Paul et al., 2007; Pichon et al., 2000).However, this increased oocyst burden coincided with a lower infection prevalence, possibly indicating that other factors operate in limiting mating (see below).

In malaria endemic areas, intensive use of treatments for malaria has led to the emergence of drug-resistant parasites. Despite their low efficacy, malaria therapies such as chloroquine (CQ) and sulphadoxine-pyrimethamine (SP) are still widely used in sub Saharan Africa. It has been shown that, upon treatment, drug-resistant parasites have a selective advantage, leading to higher transmission by the vector (Hallett et al., 2004, 2006). Our samples originated from an area with high drug pressure and volunteers carrying single parasite genotype may have received an early anti malarial treatment that cured them from drug-sensitive genotypes, thus allowing an optimal growth and transmission of a resistant genotype. However, children who received a malaria treatment in the one month period preceding the gametocyte carriage detection were not included in the study and genotyping of pfcrt-K76T mutation in a subset of our gametocyte samples identified single infections both as CQ resistant or sensitive parasite strains. This result indicates that other factors contribute to the better transmission capacity of the mono-infected *Plasmodium falciparum* isolates.

We found that the Multiplicity Of Infection is negatively correlated to the response variable in both zero and count components. This indicates that the genetic complexity of gametocyte populations modulates the mosquito infection outcomes in an opposite manner : while gametocyte isolates containing a single clone of *Plasmodium falciparum* resulted in a higher mean number of oocysts in infected mosquitoes, gametocyte isolates with multiple genotypes gave rise to a higher infection prevalence. These results may suggest that malaria parasites use kin discrimination to adapt strategies allowing optimal parasite transmission.

Our results showed that the genetic complexity of gametocyte isolates affects the mosquito infection intensity. Mosquito infections with isolates of lower complexity resul-

ted in higher oocyst counts. This may reflects a higher virulence of genotypes in these infections, where the gametocyte genotypes in the mono-infected isolates could have suppressed their competitors in a prior step of the infection, within the human host. Nonetheless, the lower infection prevalence in mono clonal infections indicates that the higher number of oocysts arises at the cost of a reduced ability to infect the mosquito vector population. This could result from blood quality/quantity such as agglutinating antibodies or anaemia. It was shown that mixed infections resulted in increased anaemia, a possible adaptive response for sex ratio adjustment (Taylor and Read, 1998; Paul et al., 2004). Sex allocation theory predicts that sex ratio becomes less female-biased as clone number increases (Read et al., 1992; Paul et al., 2002; Reece et al., 2008; Schall, 2009). Then, if parasite aggregation is an adaptive trait to promote gamete fertilization, by contrast the highly female biased sex ratio in mono infected isolates will affect infection prevalence because male availability will constitute a limiting factor for mating.

Our results may have important implications for the genetic structuring of *Plasmodium falciparum* populations. For *Plasmodium falciparum*, fertilization of gametes can occur between genetically-identical gametes (inbreeding) or between different gametes (outbreeding). Levels of inbreeding differ from one malaria area to another but they roughly correlate with the disease endemicity (Anderson et al., 2000). In areas of high malaria endemicity, inbreeding levels are generally more reduced, mostly because parasite genetic diversity is high and multiple infections predominant. However, population genetics studies, after genotyping of oocysts from wild mosquitoes collected in intense malaria transmission areas, gave rise to conflicting results and the extent of inbreeding in natural settings remains controversial (Razakandrainibe et al., 2005; Annan et al., 2007; Mzilahowa et al., 2007). The higher fitness of inbred parasites, as suggested in this study and others (Hastings and Wedgwood-Oppenheim, 1997; Razakandrainibe et al., 2005), could explain the departs from panmixia frequently found in areas of high malaria transmission.

Finally, our results comfort the idea that malaria parasites are able to discriminate the genetic complexity of their infections and to adjust accordingly adaptive traits implicated in transmission (aggregation, sex ratio). Deciphering specific processes involved in parasite recognition and competition within the mosquito vector would help for our understanding of within host behaviour of malaria parasites. This may have important implications for future malaria interventions strategies.

3.A Random Forests

RF estimator

The principle of random forests is to aggregate a given number *ntree* of binary decision trees built on several bootstrap samples drawn from the learning set. The bootstrap samples are obtained by uniformly drawing n samples among the learning set with repetition. The decision trees are fully developed binary trees and the split rule is the following.

First, the whole dataset (also called the root of the tree) is split into two subsets of data (called two children nodes). To do that, one randomly chooses a given number *mtry* of variables, and computes all the splits only for the previously selected variables. A split is of the form $\{X^i \leqslant s\} \cup \{X^i > s\}$, which means that data with the i-th variable value less than the threshold s go to the left child node and the others to the right one. Finally the selected split is the one minimizing the variance children nodes.

Then, one restraints to one child node, randomly chooses another set of *mtry* variables and calculates the best split. And so on, until each node is a terminal node, i.e. it comprises less than 5 observations.

A new data item X, starting in the root of the tree, goes down the tree following the splits and falls in a terminal node. Then the tree predicts for X, $\bar{Y}$ the mean of response of data in this terminal node. To finally get the RF predictor, one aggregates all the tree predictors by averaging their predictions.

RF error estimate : the OOB error

Inside the variable selection procedure, we use an estimation of the prediction error directly computed by the RF algorithm. This is the Out Of Bag (OOB) error and is calculated as follows. Fix one data in the learning sample, and consider all the bootstrap samples which do not contain this data (i.e. for which the data is "out of bag"). Now perform an aggregation only among trees built on these bootstrap samples. After doing this for all data, compare to the true response and get an estimation of the prediction error (which is a kind of cross-validated error estimate).

RF variable importance

Let us now detail the computation of the RF variable importance for the first variable X^1. For each tree, one has a bootstrap sample associated with an OOB sample. Predict the OOB data with the tree predictor. Now, randomly permute the values of the first variable of the OOB observations, predict these modified OOB data with the tree predictor. The variable importance of X^1 is defined as the mean increase of prediction errors after permutation. The more the error increases, the more important the variable is (note that it can be slightly negative, typically for irrelevant variables).

3.B $\mathcal{ZIP}$ and $\mathcal{ZINB}$ specifications

These two models are defined by equation (3.1) with the count model given by :

– $\mathcal{ZIP}$:

$$\begin{cases} P\left(Y_{i,j}=y_{i,j}|\ U_{i,j}=1\right) &= \exp\left(-\lambda_i\right)\dfrac{\lambda_i^{y_{i,j}}}{y_{i,j}!} \\ \lambda_i &:= \mathbb{E}\left(Y_{i,j}|\ U_{i,j}=1\right) \\ &= \mathrm{Var}(Y_{i,j}|\ U_{i,j}=1) \end{cases}$$

– $\mathcal{ZINB}$:

$$\begin{cases} P\left(Y_{i,j}=y_{i,j}|\ U_{i,j}=1\right) &= \dfrac{\Gamma\left(y_{i,j}+\theta\right)}{\Gamma\left(\theta\right)\cdot y_{i,j}!}\dfrac{\lambda_i^{y_{i,j}}\cdot\theta^{\theta}}{\left(\lambda_i+\theta\right)^{y_{i,j}+\theta}} \\ \lambda_i &:= \mathbb{E}\left(Y_{i,j}|\ U_{i,j}=1\right) \\ \mathrm{Var}(Y_{i,j}|\ U_{i,j}=1) &= \lambda_i+\dfrac{1}{\theta}\lambda_i^2 \end{cases}$$

where $\Gamma\left(t\right)=\displaystyle\int_0^{\infty}x^{t-1}\mathbf{e}^{-x}\mathbf{d}x$, and θ is the over-dispersion parameter. The expectation and the variance of $Y_{i,j}$ are given by :

$$\begin{aligned} \mu\left(x\right) &:= \mathbb{E}\left(Y_{i,j}|\right)=\left(1-\pi_i\right)\lambda_i \\ \mathrm{Var}(Y_{i,j}) &= \begin{cases} \left(1-\pi_i\right)\left(\lambda_i+\pi_i\lambda_i^2\right) & ZIP \\ \left(1-\pi_i\right)\left(\lambda_i+\left(\dfrac{1}{\theta}+\pi_i\right)\lambda_i^2\right) & ZINB. \end{cases} \end{aligned}$$

Chapitre 4

Bornes de risque pour une variante des forêts aléatoires

Sommaire

Résumé

Ce chapitre présente des résultats théoriques obtenus sur une version simple de forêts aléatoires. Cette version fait partie de la famille des forêts purement aléatoires (voir la présentation générale). Le cadre de cette étude est celui de la régression avec une seule variable explicative (l'espace d'entrée est de dimension 1). Nous montrons alors que les estimateurs par arbres aléatoires et forêts aléatoires atteignent tous

deux la vitesse de convergence minimax pour la classe des fonctions Lipschitziennes. Nous établissons également un deuxième résultat qui illustre une amélioration apportée par une forêt comparée à un arbre. En effet, nous montrons qu'une forêt réduit la variance d'estimation, d'un facteur de trois quarts. Ce résultat est intéressant car il traduit théoriquement l'observation que l'on fait en pratique : une forêt aléatoire améliore systématiquement les performances d'un arbre aléatoire. Nous insistons enfin sur le fait que le résultat précédent est obtenu en analysant de façon précise un terme de covariance entre deux arbres. Grossièrement, nous montrons que la covariance entre deux arbres est majorée par trois quarts fois la variance d'un arbre, et ceci nous permet de conclure. Le contrôle de ce terme de covariance illustre également théoriquement l'heuristique expliquant les performances des forêts aléatoires : pour qu'une forêt soit performante il faut que les arbres individuels soient différents les uns des autres.

Le contenu de ce chapitre constitue l'article Genuer (2010c) qui est actuellement soumis.

4.1 Introduction

Random forests (RF), introduced by Breiman (2001), are a very effective statistical method. They give outstanding performances in a lot of situations for both regression and classification problems. Mathematical understanding of these good performances remains quite unknown. As defined by Leo Breiman, a random forest is a collection of tree-predictors $\{h(x, \Theta_l), 1 \leqslant l \leqslant q\}$, where $(\Theta_l)_{1 \leqslant l \leqslant q}$ are i.i.d. random vectors, and a random forest predictor is obtained by aggregating this collection of trees. In addition to consistency results, one of the main theoretical challenges is to explain why a random forest improves so much the performance of a single tree.

Breiman (2001) introduced a specific instance of random forest, called random forests-RI, which has been adopted in many fields as a reference method. Indeed, random forests-RI are simple to use, and are efficiently coded in the popular R-package `randomForest` Liaw and Wiener (2002). They are effective for a predictive goal and they can also be used for variable selection (see e.g. Díaz-Uriarte and Alvarez de Andrés (2006), Genuer et al. (2010a)).

However, forests-RI are very difficult to handle theoretically. This is why people are interested in simplified versions, called purely random forests (PRF). The main difference is that in PRF, the splits of tree nodes are randomly drawn *independently* of the learning sample ; while in random forests-RI, the splits are optimized using the learning sample. This independence between splits and learning sample makes mathematical analysis easier. Cutler and Zhao (2001) introduced PERT (Perfect Random Tree Ensemble), an algorithm which builds some purely random forests, and illustrated its good perfor-

mance on benchmark datasets. More recently Biau et al. (2008) showed that both purely random trees and purely random forests are universally consistent.

Our paper offers to examine another simple variant of random forests, which can be put in the so-called purely random forests family. We call it *purely uniformly random forests* and we analyze its risk, only in a regression framework with a one-dimensional predictor space. The main goal is to emphasize the gain of using a forest instead of a tree. The results of this paper are twofold : first we show that both purely uniformly random trees and forests risks reach minimax rate of convergence on the Lipschitz functions class ; second we show that forests improve the variance term by a factor of three fourths while not increasing the bias.

The paper is organized as follows. Section 4.2 presents the model. Section 4.3 and Section 4.4 give some risk bounds for purely uniformly random trees and purely uniformly random forests respectively. Section 4.5 concludes the paper, while proofs are collected in Section 4.6.

4.2 Framework

The framework we consider all along the paper is the classical random design regression framework.

More precisely, consider a learning set $\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ made of n i.i.d. observations of a vector (X, Y) from an unknown distribution. Y is real-valued since we are in a regression framework. We consider the following statistical model :

$$Y_i = s(X_i) + \varepsilon_i \quad \text{for } i = 1, \ldots, n \ . \tag{4.1}$$

s is the unknown regression function and the goal is to estimate s. We make the following assumptions on model (4.1) :

- $X \in [0, 1]$ *with continuous density function* μ *;*
- $(\varepsilon_1, \ldots, \varepsilon_n)$ *are i.i.d. observations of* ε*, independent of* $\mathcal{L}_n$*, with* $\mathbb{E}[\varepsilon] = 0$ *and where* $\mathrm{Var}(\varepsilon) = \sigma^2$ *is assumed to be known.*

Note that we deal only with a one-dimensional predictor space.

This paper aims at comparing performances in estimating s using a single random tree and a random forest of a special kind, described in the next section.

4.3 Risk bounds for Purely Uniformly Random Trees

4.3.1 Tree definition

The principle of Purely Uniformly Random Trees (PURT) is that we draw k uniform random variables, which form the partition of the input space $[0, 1]$. Then we build a

regressogram on this partition, that we call a tree.

Note that, unlike purely random forests or random forests-RI, the tree structure of individual predictors is not obvious. This comes from the fact that in PURT the partition is not obtained in a recursive manner. Nevertheless we keep the vocabulary of trees and forests to distinguish individual predictors from aggregated ones.

Let us mention that, all along the paper, we make a slight language abuse. Indeed, we refer to random tree, the tree himself (as a graph), the corresponding partition of $[0,1]$, as well as the corresponding estimator.

More precisely, let $\mathbb{U} = (U_1, \ldots, U_k)$ be k i.i.d. random variables of uniform distribution on $[0,1]$, where k is a natural integer which will depend on the number of observations n.

A Purely Uniformly Random Tree (PURT), associated with $\mathbb{U}$, is defined for $x \in [0,1]$ as :

$$\hat{s}_{\mathbb{U}}(x) = \sum_{j=0}^{k} \hat{\beta}_j \mathbb{1}_{U_{(j)} < x \leqslant U_{(j+1)}}$$

where

$$\hat{\beta}_j = \frac{1}{\sharp\{i :\ U_{(j)} < X_i \leqslant U_{(j+1)}\}} \sum_{i:\, U_{(j)} < X_i \leqslant U_{(j+1)}} Y_i$$

and $(U_{(1)}, \ldots, U_{(k)})$ is the ordered statistics of $(U_1, \ldots, U_k)$ and $U_{(0)} = 0$, $U_{(k+1)} = 1$. $\sharp\,\mathcal{E}$ denotes the cardinality of the set $\mathcal{E}$.

Remark 3. *Let us mention that if* $\sharp\{i :\ U_{(j)} < X_i \leqslant U_{(j+1)}\} = 0$, *we set* $\hat{\beta}_j = 0$. *However as we will see in Section 4.3.2, our assumptions on* k *and* n *will make the probability of observing such an event tend to* 0.

In addition, let us define, for $x \in [0,1]$:

$$\tilde{s}_{\mathbb{U}}(x) = \sum_{j=0}^{k} \beta_j \mathbb{1}_{U_{(j)} < x \leqslant U_{(j+1)}}$$

where

$$\beta_j = \mathbb{E}[Y \,|\, U_{(j)} < X \leqslant U_{(j+1)}] \,.$$

Conditionally on $\mathbb{U}$, $\tilde{s}_{\mathbb{U}}$ is the best approximation of s among all the regressograms based on $\mathbb{U}$, but of course it depends on the unknown distribution of (X, Y).

With these notations, we can write a bias-variance decomposition of the quadratic risk of $\hat{s}_{\mathbb{U}}$ as follows :

$$\begin{aligned} \mathbb{E}[(\hat{s}_{\mathbb{U}}(X) - s(X))^2] &= \mathbb{E}[(\hat{s}_{\mathbb{U}}(X) - \tilde{s}_{\mathbb{U}}(X))^2] + \mathbb{E}[(\tilde{s}_{\mathbb{U}}(X) - s(X))^2] \\ &= \text{variance term} + \text{bias term} \end{aligned} \tag{4.2}$$

To clarify these variance and bias terms, we emphasize that for a given partition u and a given x, we have

$$\mathbb{E}[\hat{s}_u(x)] = \tilde{s}_u(x)$$

so $\mathbb{E}[(\hat{s}_u(x) - \tilde{s}_u(x))^2]$ is the variance of the estimator $\hat{s}_u(x)$ and $\mathbb{E}[(\tilde{s}_u(x) - s(x))^2]$ is its bias. We then integrate with respect to (w.r.t) X and $\mathbb{U}$ to get decomposition (4.2).

4.3.2 Variance of a tree

We start to deal with the variance term of decomposition (4.2). First, we work conditionally on $\mathbb{U}$, then the problem reduces to the case of a regressogram on a deterministic partition, and we can apply the following proposition which comes from Arlot (2008).

Proposition 1. *Conditionally on $\mathbb{U}$, the variance term of decomposition (4.2) satisfies :*

$$\mathbb{E}[(\hat{s}_{\mathbb{U}}(X) - \tilde{s}_{\mathbb{U}}(X))^2 \,|\, \mathbb{U}] = \frac{1}{n}\sum_{j=0}^{k}(1 + \delta_{n,p_j})(\sigma^2 + (\sigma_j^d)^2) \tag{4.3}$$

where

- $p_j = \mathrm{P}(U_{(j)} < X \leqslant U_{(j+1)})$,
- $(\sigma_j^d)^2 = \mathbb{E}[(s(X) - \tilde{s}_{\mathbb{U}}(X))^2 \,|\, U_{(j)} < X \leqslant U_{(j+1)}]$,
- $\delta_{n,p} \xrightarrow[np\to+\infty]{} 0$.

■

We now integrate equation (4.3) w.r.t. $\mathbb{U}$, and we get the following equality :

$$\mathbb{E}[(\hat{s}_{\mathbb{U}}(X) - \tilde{s}_{\mathbb{U}}(X))^2] = \frac{1}{n}\sum_{j=0}^{k}\left(\sigma^2 + \sigma^2\mathbb{E}[\delta_{n,p_j}] + \mathbb{E}[(\sigma_j^d)^2] + \mathbb{E}[(\sigma_j^d)^2\delta_{n,p_j}]\right) \tag{4.4}$$

Let us stress that equation (4.4) is general, since it does not depend on the distribution of $\mathbb{U}$. Hence, it can be used for any random partition distributions.

Finally, using the fact that, in our case, $\mathbb{U}$ is made of k i.i.d. random variables of uniform distribution on $[0, 1]$, we deduce from equation (4.4) the following proposition :

Proposition 2. *If $k \xrightarrow[n\to+\infty]{} +\infty$, $\frac{k}{n} \xrightarrow[n\to+\infty]{} 0$, $\mu > 0$ and s is C-Lipschitz, the variance of a PUR Tree satisfies :*

$$\mathbb{E}[(\hat{s}_{\mathbb{U}}(X) - \tilde{s}_{\mathbb{U}}(X))^2] = \frac{\sigma^2(k+1)}{n} + \underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right) \tag{4.5}$$

where the notation $\underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right)$ denotes a function $f(n)$ such as $\frac{f(n)}{k/n} \xrightarrow[n\to+\infty]{} 0$.

■

Details of the proof of Proposition 2 can be found in Section 4.6.1.

The first two hypotheses of Proposition 2 ($k \xrightarrow[n\to+\infty]{} +\infty$, $\frac{k}{n} \xrightarrow[n\to+\infty]{} 0$) are the same natural conditions found by Biau et al. (2008) for consistency of PRF. They guarantee that the number of splits of the tree must grow to infinity but slower than the number of samples.

4.3.3 Bias of a tree

We now turn to the bias term of decomposition (4.2). Direct calculations (see Section 4.6.2 for details) lead to the following upper bound for the bias term of a PURT :

Proposition 3. *If μ is bounded by $M > 0$ and s is C-Lipschitz, the bias of a PURT is upper bounded by :*

$$\mathbb{E}[(\tilde{s}_{\mathbb{U}}(X) - s(X))^2] \leqslant \frac{6MC^2}{(k+1)^2} \tag{4.6}$$

■

4.3.4 Risk bounds for a tree

Putting together (4.5) and (4.6) leads to the following risk bound for a PURT.

Theorem 4. *If $k \xrightarrow[n\to+\infty]{} +\infty$, $\frac{k}{n} \xrightarrow[n\to+\infty]{} 0$, $0 < \mu \leqslant M$ and s is C-Lipschitz, the risk of a PURT satisfies :*

$$\mathbb{E}[(\hat{s}_{\mathbb{U}}(X) - s(X))^2] \leqslant \frac{\sigma^2(k+1)}{n} + \frac{6MC^2}{(k+1)^2} + \underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right) \tag{4.7}$$

■

The balance between the two first terms of the right hand side (r.h.s.) of (4.7) leads to take $(k+1) = n^{1/3}$, and gives the following upper bound for the risk of a PURT.

Corollary 1. *Under the assumptions of Theorem 4,*

$$\mathbb{E}[(\hat{s}_{\mathbb{U}}(X) - s(X))^2] \leqslant Kn^{-2/3} + \underset{n\to+\infty}{\mathrm{o}}(n^{-2/3})$$

where K is a positive constant.

■

Therefore, a PURT reaches the minimax rate of convergence associated with the class of Lipschitz functions (see e.g. Ibragimov and Khasminskii (1981)).

Let us now analyze purely uniformly random forests. As a result, we emphasize an improvement given by a forest compared to a single tree.

4.4 Risk bounds for Purely Uniformly Random Forests

4.4.1 Forest definition

A random forest is the aggregation of a collection of random trees. So, in the context of Purely Uniformly Random Forests (PURF), the principle is to generate several PUR Trees by drawing several random partitions given by uniform random variables, and to aggregate them.

Let $\mathbb{V} = (\mathbb{U}^1, \ldots, \mathbb{U}^q)$ be q i.i.d. random vectors of the same distribution as $\mathbb{U}$ (defined in Section 4.3.1). That is for $l = 1, \ldots, q, \quad \mathbb{U}^l = (U_1^l, \ldots, U_k^l)$ where the $(U_j^l)_{1 \leqslant j \leqslant k}$ are i.i.d. random variables of uniform distribution on $[0, 1]$.

A PURF, associated with $\mathbb{V}$, is defined for $x \in [0, 1]$ as follows :

$$\hat{s}(x) = \frac{1}{q} \sum_{l=1}^{q} \hat{s}_{\mathbb{U}^l}(x) \ .$$

Let us define, for $x \in [0, 1]$:

$$\tilde{s}(x) = \frac{1}{q} \sum_{l=1}^{q} \tilde{s}_{\mathbb{U}^l}(x) \ .$$

Again, we have a bias-variance decomposition of the quadratic risk of $\hat{s}$, given by :

$$\begin{aligned} \mathbb{E}[(\hat{s}(X) - s(X))^2] &= \mathbb{E}[(\hat{s}(X) - \tilde{s}(X))^2] + \mathbb{E}[(\tilde{s}(X) - s(X))^2] \\ &= \text{variance term} + \text{bias term} \end{aligned} \tag{4.8}$$

4.4.2 Variance of a forest

We first deal with the variance term of decomposition (4.8). We begin to show that when letting the number of trees q grow to infinity, the variance of a PURF is close to the covariance between two PURT.

Indeed, since $\hat{s}(x) = \frac{1}{q} \sum_{l=1}^{q} \hat{s}_{\mathbb{U}^l}(x)$, the variance term satisfies :

$$\mathbb{E}[(\hat{s}(X) - \tilde{s}(X))^2] = \frac{1}{q^2}\sum_{l=1}^{q}\mathbb{E}[(\hat{s}_{\mathbb{U}^l}(X) - \tilde{s}_{\mathbb{U}^l}(X))^2] + \frac{1}{q^2}\sum_{l\neq q}\mathbb{E}[(\hat{s}_{\mathbb{U}^l}(X) - \tilde{s}_{\mathbb{U}^l}(X))(\hat{s}_{\mathbb{U}^m}(X) - \tilde{s}_{\mathbb{U}^m}(X))]$$

$$= \frac{1}{q}\mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))^2] + \frac{q(q-1)}{q^2}\mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))]$$

where the last equality comes from the fact that the $((\hat{s}_{\mathbb{U}^l}(X) - \tilde{s}_{\mathbb{U}^l}(X))_{1\leqslant l\leqslant q}$ are of the same distribution.

Now, if we let q grow to infinity, we get :

$$\mathbb{E}[(\hat{s}(X) - \tilde{s}(X))^2] = \mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))] + \underset{q\to+\infty}{\mathrm{o}}(1)$$

The next step is to upper bound the covariance between two PURT

$$\mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))]$$

(it is detailed in Section 4.6.3) and it leads to the following theorem, which gives the behavior of the variance of a PURF :

Theorem 5. *If* $k \xrightarrow[n\to+\infty]{} +\infty$, $\frac{k}{n} \xrightarrow[n\to+\infty]{} 0$, $\mu > 0$, *s is C-Lipschitz and* $q \xrightarrow[n\to+\infty]{} +\infty$, *the variance of a PURF satisfies the following upper bound :*

$$\mathbb{E}[(\hat{s}(X) - \tilde{s}(X))^2] \leqslant \frac{3}{4}\frac{\sigma^2(k+1)}{n} + \underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right) \tag{4.9}$$

■

Theorem 5 is to be compared with Proposition 2 and tells us that the variance of a PUR Forest is upper bounded by three fourths times the variance of a PUR Tree. So, the rate of decay (in terms of power of n) of the PUR Forest variance is the same as the PUR Tree variance, and the actual gain appears in the multiplicative constant.

We mention that, as in the analysis of the variance of a tree (see equation (4.4)), we derive, in the proof of Theorem 5, a general statement (see equation (4.13) in Section 4.6.3), which does not depend on the distribution of the partition defining the random trees.

Let us, finally, comment the hypotheses of Theorem 5. First, note that the hypotheses on k and n are the same as in Proposition 2, which allows a fair comparison between

the two results. Second, the hypothesis on q allows to ensure that the upper bound on the covariance (given by Corollary 3 in Section 4.6.3) leads to the same upper bound for the variance of the forest. Finally, the other hypotheses ($\mu > 0$, s is C-Lipschitz) are the same as in Proposition 2 and help to control negligible terms.

4.4.3 Bias of a forest

We now deal with the bias term of decomposition (4.8). A convex inequality gives that the bias of a forest is not larger than the bias of a single tree :

$$\begin{aligned}\mathbb{E}[(\tilde{s}(X) - s(X))^2] &\leqslant \frac{1}{q}\sum_{l=1}^{q}\mathbb{E}[(\tilde{s}_{\mathbb{U}^l}(X) - s(X))^2] \\ &= \mathbb{E}[(\tilde{s}_{\mathbb{U}^1}(X) - s(X))^2] \, .\end{aligned}$$

So from Proposition 3, we deduce that :

Proposition 4. *If μ is bounded by $M > 0$ and s is C-Lipschitz, the bias of a PURF satisfies the same inequality as (4.6), that is :*

$$\mathbb{E}[(\tilde{s}(X) - s(X))^2] \leqslant \frac{6MC^2}{(k+1)^2} \tag{4.10}$$

■

4.4.4 Risk bounds for a forest

Putting together (4.9) and (4.10) leads to the following risk bound for a PURF.

Theorem 6. *If $k \xrightarrow[n\to+\infty]{} +\infty$, $\dfrac{k}{n} \xrightarrow[n\to+\infty]{} 0$, $0 < \mu \leqslant M$, s is C-Lipschitz and $q \xrightarrow[n\to+\infty]{} +\infty$, the risk of a PURF satisfies :*

$$\mathbb{E}[(\hat{s}(X) - s(X))^2] \leqslant \frac{3}{4}\frac{\sigma^2(k+1)}{n} + \frac{6MC^2}{(k+1)^2} + \underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right)$$

■

Again, taking $(k+1) = n^{1/3}$ gives the upper bound for the risk :

Corollary 2. *Under the assumptions of Theorem 6,*

$$\mathbb{E}[(\hat{s}(X) - s(X))^2] \leqslant Kn^{-2/3} + \underset{n\to+\infty}{\mathrm{o}}(n^{-2/3})$$

where K is a positive constant.

■

So, a PURF reaches the minimax rate of convergence for C-Lipschitz functions.

Secondly, as the variance of a PUR Forest is systematically reduced compared to a PUR Tree and the bias of a PUR Forest is not larger than the one of a PUR Tree, the risk of a PUR Forest is actually lower.

4.5 Conclusion

We emphasize, for a very simple version of random forests, the actual gain of using a random forest instead of using a single random tree. First, we showed that both trees and forests reach the minimax rate of convergence. Then, we manage to highlight a reduction of the variance of a forest, compared to the variance of a tree. This is, in this specific context, a proof of the well-known conjecture for random forests : "a random forest, by aggregating several random trees, reduces variance and leaves the bias unchanged" which can be found for example in Hastie et al. (2009).

An interesting open problem would be to generalize this result, which could handle more complex versions of random forests and relax the hypotheses we made here. Obviously, a more ambitious goal would be to give some precise insights explaining the outstanding performances of random forests-RI.

4.6 Proofs

4.6.1 Proof of Proposition 2

We must show that the three last terms in the sum of equation (4.4) are negligible compared to the constant term σ^2.

Let us fix $0 \leqslant j \leqslant k$. As it can be found e.g. in Chapter 6 of David and Nagaraja (2003), the probability density function of $U_{(j+1)} - U_{(j)}$ is the function $t \in [0,1] \longmapsto k(1-t)^{k-1}$.

– For the second term $\mathbb{E}[\delta_{n,p_j}]$:
from Arlot (2008) we have $\delta_{n,p_j} \leqslant \kappa_3 (np_j)^{-1/4}$, where κ_3 is a positive constant. So,

$$\begin{aligned}
\mathbb{E}[\delta_{n,p_j}] &\leqslant \kappa_3 \mathbb{E}[(np_j)^{-1/4}] \\
&= \frac{\kappa_3}{n^{-1/4}} \mathbb{E}[p_j^{-1/4}] \\
&\leqslant \frac{\kappa_3}{(mn)^{-1/4}} \mathbb{E}[(U_{(j+1)} - U_{(j)})^{-1/4}] \\
&\leqslant \frac{\kappa_4}{m^{-1/4}} \left(\frac{k}{n}\right)^{1/4}
\end{aligned}$$

where $m = \min_{[0,1]} \mu$ and κ_4 is another positive constant.

Since $\frac{k}{n} \xrightarrow[n\to+\infty]{} 0$ the last upper bound tends to 0 as n tends to infinity.

– For the third term $\mathbb{E}[(\sigma_j^d)^2]$:

$$\begin{aligned}(\sigma_j^d)^2 &= \mathbb{E}[(s(X) - \tilde{s}_{\mathbb{U}}(X))^2 \,|\, U_{(j)} < X \leqslant U_{(j+1)}] \\ &\leqslant C^2(U_{(j+1)} - U_{(j)})^2 \quad \text{because } s \text{ is } C-\text{Lipschitz}\end{aligned}.$$

So, $\mathbb{E}[\sigma_j^d)^2] \leqslant C^2\mathbb{E}[(U_{(j+1)} - U_{(j)})^2] = C^2\frac{2}{(k+1)(k+2)}$ which tends to 0 as k tends to infinity.

– For the last term, the following inequality is sufficient to conclude :
$\mathbb{E}[(\sigma_j^d)^2\delta_{n,p_j}] \leqslant C^2\mathbb{E}[\delta_{n,p_j}]$, because $U_{(j+1)} - U_{(j)} \leqslant 1$.

4.6.2 Proof of Proposition 3

Function s is supposed to be C-Lipschitz, so

$$\begin{aligned}\mathbb{E}[(\tilde{s}_{\mathbb{U}}(X) - s(X))^2] &= \mathbb{E}[(\sum_{j=0}^{k}(s(X) - \beta_j)\mathbb{1}_{U_{(j)}<X\leqslant U_{(j+1)}})^2] \\ &= \mathbb{E}[\sum_{j=0}^{k}(s(X) - \beta_j)^2\ \mathbb{1}_{U_{(j)}<X\leqslant U_{(j+1)}}] \\ &\leqslant \mathbb{E}[\sum_{j=0}^{k} C^2(U_{(j+1)} - U_{(j)})^2\ \mathbb{1}_{U_{(j)}<X\leqslant U_{(j+1)}}] \\ &= C^2\mathbb{E}[\sum_{j=0}^{k}(U_{(j+1)} - U_{(j)})^2\ \mathrm{P}(U_{(j)} < X \leqslant U_{(j+1)})] \\ &\leqslant C^2\mathbb{E}[\sum_{j=0}^{k} M(U_{(j+1)} - U_{(j)})^3] \\ &\qquad \text{because } \mu \text{ is bounded by } M \\ &= MC^2\sum_{j=0}^{k}\mathbb{E}[(U_{(j+1)} - U_{(j)})^3] \\ &= MC^2\frac{6}{(k+2)(k+3)} \\ &\leqslant \frac{6MC^2}{(k+1)^2}\ .\end{aligned}$$

4.6.3 Proof of Theorem 5

Before entering into details of the proof of Theorem 5, we recall that in the proof of Proposition 1 (which can be found in Arlot (2008)), calculations lead to the following equality :

$$\mathbb{E}[(\hat{s}_{\mathbb{U}}(X) - \tilde{s}_{\mathbb{U}}(X))^2 \,|\, \mathbb{U}] = \sum_{j=0}^{k} p_j \mathbb{E}\Big[\frac{1}{n\hat{p}_j}\Big](\sigma^2 + (\sigma_j^d)^2) \tag{4.11}$$

where $\hat{p}_j = \dfrac{\sharp\{i :\ U_{(j)} < X_i \leqslant U_{(j+1)}\}}{n}$.

Then, an estimation of $p_j \mathbb{E}\Big[\dfrac{1}{n\hat{p}_j}\Big]$ gives the expression $\dfrac{1}{n}(1 + \delta_{n,p_j})$ in Proposition 1.

We note

$$Var_j = p_j \mathbb{E}\Big[\frac{1}{n\hat{p}_j}\Big](\sigma^2 + (\sigma_j^d)^2) \tag{4.12}$$

a generic term of the sum in the r.h.s. of (4.11).

We now address the proof of Theorem 5. We begin by introducing some notations and establish an intermediate result. The following proposition is not only useful to prove Theorem 5, but has its own interest. Indeed, it gives a general upper bound (to be compared to equation (4.3)) which does not depend on the distribution of random partitions defining the trees.

In the sequel we denote the covariance between two PURT by :

$$\mathbb{C}(\hat{s}_{\mathbb{U}^1}, \hat{s}_{\mathbb{U}^2}) = \mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))]$$

Let us consider $\mathbb{U}^1 = (U_1^1, \ldots, U_k^1)$ and $\mathbb{U}^2 = (U_1^2, \ldots, U_k^2)$ two sequences of i.i.d. uniform random variables, with respective ordered statistics $(U_{(1)}^1, \ldots, U_{(k)}^1)$ and $(U_{(1)}^2, \ldots, U_{(k)}^2)$.

Then we denote by $(V_{(1)}, \ldots, V_{(2k)})$ the ordered statistics of the complete vector $(U_1^1, \ldots, U_k^1, U_1^2, \ldots, U_k^2)$, $V_{(0)} = 0$ and $V_{(2k+1)} = 1$.

$(\Sigma_t^{d,1,2})^2$ denotes a sum of terms $\mathbb{E}[(\tilde{s}_{\mathbb{U}^1}(X) - s(X))(\tilde{s}_{\mathbb{U}^2}(X) - s(X)) \,|\, V_{(t')} < X \leqslant V_{(t'+1)}]$ for several consecutive values of t'.

Finally $\tilde{p}_t$ denotes for some $j \in \{0, \ldots, k\}$ either p_j^1 or p_j^2 depending on the relative positions between the $(U_1^1, \ldots, U_k^1)$ and the $(U_1^2, \ldots, U_k^2)$ in $(V_{(1)}, \ldots, V_{(2k)})$ (see details below).

Proposition 5. *The covariance between two PURT satisfies the following upper bound :*

$$\mathbb{C}(\hat{s}_{\mathbb{U}^1}, \hat{s}_{\mathbb{U}^2}) \leqslant \frac{1}{n}\mathbb{E}\left[\sum_{t=0}^{N_{1,2}}(1 + \delta_{n,\tilde{p}_t})(\sigma^2 + (\Sigma_t^{d,1,2})^2)\right] \tag{4.13}$$

where $N_{1,2} = k+1-\sum_{r=1}^{k-2}\sum_{s=1}^{k-1} \mathbb{1}_{U^2_{(s)}<U^1_{(r)}<U^1_{(r+1)}<U^1_{(r+2)}<U^2_{(s+1)}}$.

■

Remark 4. *The gain in variance for a PURF comes from the fact that the number of terms in the sum of equation (4.13) is smaller than* $k+1$*. Indeed, it is* $k+1-M_{1,2}$ *where* $M_{1,2}$ *is the number of times that* 3 *consecutive ordered statistics of* $\mathbb{U}^1$ *are included in* 2 *consecutive ordered statistics of* $\mathbb{U}^2$*.*

We now prove inequality (4.13) of Proposition 5. The term $(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))$ equals, by definition, to :

$$\begin{aligned}&\left(\sum_{r=0}^{k}(\hat{\beta}^1_r-\beta^1_r)\mathbb{1}_{U^1_{(r)}<X\leqslant U^1_{(r+1)}}\right)\left(\sum_{s=0}^{k}(\hat{\beta}^2_s-\beta^2_s)\mathbb{1}_{U^2_{(s)}<x\leqslant U^2_{(s+1)}}\right)\\&=\sum_{t=0}^{2k}(\hat{\beta}^1_{t,r}-\beta^1_{t,r})(\hat{\beta}^2_{t,s}-\beta^2_{t,s})\mathbb{1}_{V_{(t)}<X\leqslant V_{(t+1)}}\end{aligned}\tag{4.14}$$

where $(V_{(1)},\dots,V_{(2k)})$ is the ordered statistics of the vector $(U^1_1,\dots,U^1_k,U^2_1,\dots,U^2_k)$, $V_{(0)}=0$, $V_{(2k+1)}=1$, and

$$\begin{cases}\hat{\beta}^1_{t,r}=\hat{\beta}^1_r \text{ and } \beta^1_{t,r}=\beta^1_r, & \text{if }]V_{(t)},V_{(t+1)}]\subset]U^1_{(r)},U^1_{(r+1)}]\\ \hat{\beta}^2_{t,s}=\hat{\beta}^2_s \text{ and } \beta^2_{t,s}=\beta^2_s, & \text{if }]V_{(t)},V_{(t+1)}]\subset]U^2_{(s)},U^2_{(s+1)}]\end{cases}$$

For $l=1,2$ and $j=0,\dots,k$, we define $\hat{p}^l_j=\dfrac{\sharp\{i:\ U^l_{(j)}<X_i\leqslant U^l_{(j+1)}\}}{n}$.

Now, let us give some details for the first term of (4.14), denoted by $S_1(X)$. Without loss of generality, we suppose that $V_{(1)}=U^1_{(1)}$ (i.e. $U^1_{(1)}<U^2_{(1)}$). So,

$$\begin{aligned}S_1(X)&=(\hat{s}_{\mathbb{U}^1}(X)-\tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X)-\tilde{s}_{\mathbb{U}^2}(X))\mathbb{1}_{0<X\leqslant U^1_{(1)}}\\&=(\hat{\beta}^1_1-\beta^1_1)(\hat{\beta}^2_1-\beta^2_1)\mathbb{1}_{0<X\leqslant U^1_{(1)}}\\&=\Big(\frac{1}{n\hat{p}^1_1}\sum_{i:\ 0<X_i\leqslant U^1_{(1)}}(Y_i-\beta^1_1)\Big)\Big(\frac{1}{n\hat{p}^2_1}\sum_{i:\ 0<X_i\leqslant U^2_{(1)}}(Y_i-\beta^2_1)\Big)\mathbb{1}_{0<X\leqslant U^1_{(1)}}\\&=\frac{1}{n\hat{p}^1_1 n\hat{p}^2_1}\sum_{i^1:\ 0<X_{i1}\leqslant U^1_{(1)}\, i^2:\ 0<X_{i2}\leqslant U^2_{(1)}}(Y_{i^1}-\beta^1_1)(Y_{i^2}-\beta^2_1)\mathbb{1}_{0<X\leqslant U^1_{(1)}}\end{aligned}$$

If we denote by $\mathbb{E}^{\Lambda_{1,2}}[.]$ the conditional expectation $\mathbb{E}[.\,|\,(\mathbb{1}_{0<X_{i^1}\leqslant U^1_{(1)}})_{1\leqslant i^1\leqslant n},(\mathbb{1}_{0<X_{i^2}\leqslant U^2_{(1)}})_{1\leqslant i^2\leqslant n}]$, we have :

$$\mathbb{E}[S_1(X) \,|\, \mathbb{U}^1, \mathbb{U}^2]$$
$$= \mathbb{E}\Big[p_1^1 \mathbb{E}\Big[\frac{1}{n\hat{p}_1^1 n\hat{p}_1^2} \sum_{i^1:\, 0<X_{i^1}\leqslant U^1_{(1)}} \sum_{i^2:\, 0<X_{i^2}\leqslant U^2_{(1)}} \mathbb{E}^{\Lambda_{1,2}}[(Y_{i^1}-\beta_1^1)(Y_{i^2}-\beta_1^2)]\Big] \,\Big|\, \mathbb{U}^1, \mathbb{U}^2\Big]$$

but

$$i^1 \neq i^2 \quad \Longrightarrow \quad \mathbb{E}^{\Lambda_{1,2}}[(Y_{i^1}-\beta_1^1)(Y_{i^2}-\beta_1^2)] = 0$$

because Y_{i^1} and Y_{i^2} are independent. Hence :

$$\begin{aligned}
&\mathbb{E}[S_1(X) \,|\, \mathbb{U}^1, \mathbb{U}^2] \\
&= \mathbb{E}\Big[p_1^1 \mathbb{E}\Big[\frac{1}{n\hat{p}_1^1 n\hat{p}_1^2} \sum_{i:\, 0<X_i\leqslant U^1_{(1)}} \mathbb{E}^{\Lambda_1}[(Y_i-\beta_1^1)(Y_i-\beta_1^2)]\Big] \,\Big|\, \mathbb{U}^1, \mathbb{U}^2\Big] \\
&= \mathbb{E}\Big[p_1^1 \mathbb{E}\Big[\frac{1}{n\hat{p}_1^1 n\hat{p}_1^2} \sum_{i:\, 0<X_i\leqslant U^1_{(1)}} \mathbb{E}[(Y_i-\beta_1^1)(Y_i-\beta_1^2) \,|\, 0<X_i\leqslant U^1_{(1)}]\Big] \,\Big|\, \mathbb{U}^1, \mathbb{U}^2\Big]
\end{aligned}$$

where $\mathbb{E}^{\Lambda_1}[.]$ denotes the conditional expectation $\mathbb{E}[. \,|\, (\mathbb{1}_{0<X_i\leqslant U^1_{(1)}})_{1\leqslant i\leqslant n}]$.

Now, as

$$\mathbb{E}[(Y_i-\beta_1^1)(Y_i-\beta_1^2) \,|\, 0<X_i\leqslant U^1_{(1)}] = \mathbb{E}[(Y-\beta_1^1)(Y-\beta_1^2) \,|\, 0<X\leqslant U^1_{(1)}]$$

for all i, and

$$\mathbb{E}[(Y-\beta_1^1)(Y-\beta_1^2) \,|\, 0<X\leqslant U^1_{(1)}] = \sigma^2 + (\sigma_0^{d,1,2})^2$$

where

$$(\sigma_0^{d,1,2})^2 = \mathbb{E}[(s(X)-\tilde{s}_{\mathbb{U}^1}(X))(s(X)-\tilde{s}_{\mathbb{U}^2}(X)) \,|\, 0<X\leqslant V_{(1)}]$$

we get

$$\mathbb{E}[S_1(X) \,|\, \mathbb{U}^1, \mathbb{U}^2] = p_1^1 \mathbb{E}\Big[\frac{1}{n\hat{p}_1^2}\Big](\sigma^2 + (\sigma_0^{d,1,2})^2) \,.$$

If we suppose in addition that $V_{(2)} = U^2_{(1)}$, we similarly get for the second term of (4.14) :

$$\begin{aligned}
&\mathbb{E}[S_2(X) \,|\, \mathbb{U}^1, \mathbb{U}^2] \\
&= \mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X)-\tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X)-\tilde{s}_{\mathbb{U}^2}(X))\mathbb{1}_{U^1_{(1)}<X\leqslant U^2_{(1)}} \,|\, \mathbb{U}^1, \mathbb{U}^2] \\
&= q_2 \mathbb{E}\left[\frac{n\hat{q}_2}{n\hat{p}_2^1 n\hat{p}_1^2}(\sigma^2 + (\sigma_1^{d,1,2})^2)\right]
\end{aligned}$$

where

$$q_2 = \mathrm{P}(V_{(1)} < X \leqslant V_{(2)}) = \mathrm{P}(U^1_{(1)} < X \leqslant U^2_{(1)})$$
$$n\hat{q}_2 = \sharp\{i : V_{(1)} < X_i \leqslant V_{(2)}\}$$

and

$$(\sigma_1^{d,1,2})^2 = \mathbb{E}[(s(X) - \tilde{s}_{\mathbb{U}^1}(X))(s(X) - \tilde{s}_{\mathbb{U}^2}(X)) \,|\, V_{(1)} < X \leqslant V_{(2)}] \,.$$

Since $]V_{(1)}, V_{(2)}]$ is included in $]U^1_{(1)}, U^1_{(2)}]$, we have $\hat{q}_2 \leqslant \hat{p}^1_2$, so :

$$\mathbb{E}[S_2(X) \,|\, \mathbb{U}^1, \mathbb{U}^2] \leqslant q_2 \mathbb{E}\Big[\frac{1}{n\hat{p}^2_1}\Big](\sigma^2 + (\sigma_1^{d,1,2})^2) \,.$$

Finally, by summing the two terms $S_1(X)$ and $S_2(X)$, we deduce that

$$\mathbb{E}[S_1(X) + S_2(X) \,|\, \mathbb{U}^1, \mathbb{U}^2] \leqslant p_1^2 \mathbb{E}\Big[\frac{1}{n\hat{p}^2_1}\Big](\sigma^2 + (\sigma_0^{d,1,2})^2 + (\sigma_1^{d,1,2})^2)$$

In conclusion, we succeeded to bound the sum of the first two terms of (4.14) by an expression very close to Var_j (defined in (4.12)). The only difference comes from the fact that instead of $(\sigma_j^d)^2$ we have $(\sigma_0^{d,1,2})^2 + (\sigma_1^{d,1,2})^2$. But as we saw in proof of Proposition 2, these terms are negligible, so $p_1^2 \mathbb{E}\Big[\frac{1}{n\hat{p}^2_1}\Big](\sigma^2 + (\sigma_0^{d,1,2})^2 + (\sigma_1^{d,1,2})^2)$ is of the same order than Var_j.

We can easily generalize this fact by proving the following lemma.

We denote by $S_j(X)$ the j-th term of (4.14), i.e. $S_j(X) = (\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))\mathbb{1}_{V_{(j)} < X \leqslant V_{(j+1)}}$.

Lemma 1. *Let r be in $\{0, \ldots, k\}$ and denote by t, t' the integers such that*

$$U^1_{(r)} = V_{(t)} < V_{(t'+1)} = U^1_{(r+1)} \tag{4.15}$$

then

$$\mathbb{E}\Big[\sum_{j=t}^{t'} S_j(X) \,|\, \mathbb{U}^1, \mathbb{U}^2\Big] \leqslant p_r^1 \mathbb{E}\Big[\frac{1}{n\hat{p}^1_r}\Big](\sigma^2 + (\Sigma_r^{d,1,2})^2)$$

where $(\Sigma_r^{d,1,2})^2 = \sum_{j=t}^{t'} (\sigma_j^{d,1,2})^2$.

■

Indeed for all $j \in \{t, t+1, \ldots, t'\}$,

$$\mathbb{E}[S_j(X) \,|\, \mathbb{U}^1, \mathbb{U}^2] \leqslant q_j \mathbb{E}\Big[\frac{1}{n\hat{p}^1_r}\Big](\sigma^2 + (\sigma_j^{d,1,2})^2)$$

where

$$q_j = \mathrm{P}(V_{(j)} < X \leqslant V_{(j+1)})$$

and

$$(\sigma_j^{d,1,2})^2 = \mathbb{E}[(s(X) - \tilde{s}_{\mathbb{U}^1}(X))(s(X) - \tilde{s}_{\mathbb{U}^2}(X)) \,|\, V_{(j)} < X \leqslant V_{(j+1)}] \; .$$

Thus,

$$\mathbb{E}\Big[\sum_{j=t}^{t'} S_j(X) \,|\, \mathbb{U}^1, \mathbb{U}^2\Big] \leqslant \mathrm{P}(V_{(t)} < X \leqslant V_{(t'+1)}) \; \mathbb{E}\Big[\frac{1}{n\hat{p}_r^1}\Big](\sigma^2 + (\Sigma_r^{d,1,2})^2) \; .$$

From relation (4.15) we have $\mathrm{P}(V_{(t)} < X \leqslant V_{(t'+1)}) = p_r^1$, which concludes the proof of Lemma 1.

Therefore, we can upper bound the initial sum (4.14) of $2k+1$ terms by a sum of $k+1$ terms of the same order as Var_j only involving intervals of the partition $\mathbb{U}^1$. At this stage, we get an upper bound for the variance of a forest which is of the same order as the variance of a tree. But we can do better. With similar arguments, we can prove the following lemma :

Lemma 2. *If there exist r and s such as*

$$U_{(s)}^2 < U_{(r)}^1 < U_{(r+1)}^1 < U_{(r+2)}^1 < U_{(s+1)}^2$$

the expression

$$\mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))\mathbb{1}_{U_{(r)}^1 < X \leqslant U_{(r+2)}^1} \,|\, \mathbb{U}^1, \mathbb{U}^2]$$

is upper bounded by

$$p_s^2 \mathbb{E}\Big[\frac{1}{n\hat{p}_s^2}\Big](\sigma^2 + (\Sigma_s^{d,1,2})^2) \; .$$

where $(\Sigma_s^{d,1,2})^2 = (\sigma_{r+s}^{d,1,2})^2 + (\sigma_{r+s+1}^{d,1,2})^2$.

■

Indeed,

$$\begin{aligned}
&\mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))\mathbb{1}_{U_{(r)}^1 < X \leqslant U_{(r+1)}^1} \,|\, \mathbb{U}^1, \mathbb{U}^2] \\
&\leqslant p_r^1 \mathbb{E}\Big[\frac{1}{n\hat{p}_s^2}\Big](\sigma^2 + (\sigma_{r+s}^{d,1,2})^2)
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X))\mathbb{1}_{U_{(r+1)}^1 < X \leqslant U_{(r+2)}^1} \,|\, \mathbb{U}^1, \mathbb{U}^2] \\
&\leqslant p_{r+1}^1 \mathbb{E}\Big[\frac{1}{n\hat{p}_s^2}\Big](\sigma^2 + (\sigma_{r+s+1}^{d,1,2})^2) \; .
\end{aligned}$$

Finally, since $p_r^1 + p_{r+1}^1 \leqslant p_s^2$, $(\sigma_{r+s}^{d,1,2})^2 \leqslant (\sigma_{r+s}^{d,1,2})^2 + (\sigma_{r+s+1}^{d,1,2})^2$ and $(\sigma_{r+s+1}^{d,1,2})^2 \leqslant (\sigma_{r+s}^{d,1,2})^2 + (\sigma_{r+s+1}^{d,1,2})^2$, the result is obtained by summing the two terms.

As in Proposition 1, we replace all $p_j^l \mathbb{E}\Big[\frac{1}{n\hat{p}_j^l}\Big]$ by their estimates $(1+\delta_{np_j^l})$.

By repeatedly applying this lemma for all intervals, we can upper bound

$$\mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X) - \tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X) - \tilde{s}_{\mathbb{U}^2}(X)) \,|\, \mathbb{U}^1, \mathbb{U}^2]$$

by a sum of $N_{1,2}$ terms of the form $(1+\delta_{n,\tilde{p}_t})(\sigma^2 + (\Sigma_t^{d,1,2})^2)$, where $\tilde{p}_t$ denotes for some $j \in \{0,\dots,k\}$ either p_j^1 or p_j^2 depending on the fact that we are in the situation of Lemma 1 or Lemma 2, $N_{1,2} = k+1-M_{1,2}$ and

$$M_{1,2} = \sum_{r=1}^{k-2}\sum_{s=1}^{k-1} \mathbb{1}_{U_{(s)}^2 < U_{(r)}^1 < U_{(r+1)}^1 < U_{(r+2)}^1 < U_{(s+1)}^2} \ .$$

This concludes the proof of Proposition 5. Now, using the fact that we deal with uniform partitions, we manage to prove the following corollary.

Corollary 3. *If* $k \xrightarrow[n\to+\infty]{} +\infty$, $\dfrac{k}{n} \xrightarrow[n\to+\infty]{} 0$, $\mu > 0$ *and* s *is* C*-Lipschitz, we have,*

$$\begin{aligned}\mathbb{C}(\hat{s}_{\mathbb{U}^1}, \hat{s}_{\mathbb{U}^2}) &\leqslant \frac{\sigma^2\mathbb{E}[N_{1,2}]}{n} + \underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right) \\ &\leqslant \frac{3}{4}\frac{\sigma^2(k+1)}{n} + \underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right) \ .\end{aligned}$$

■

Because of the simple draws of random partitions, the number $M_{1,2}$ is explicitly computable (we know the distribution of the two ordered statistics) and it is shown to be equivalent to $\dfrac{1}{4}(k+1)$ as k tends to $+\infty$ (see Lemma 3 below).

As in Proposition 2, we have to prove that all terms of the sum are negligible compared to the constant one σ^2. To deal with the fact that the number of terms in the sum is now random, we use the following simple inequality :

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=0}^{N_{1,2}} (\sigma^2\delta_{n,p_t} + (\Sigma_t^{d,1,2})^2 + (\Sigma_t^{d,1,2})^2\delta_{n,p_t})\right] \\ &\leqslant \sum_{t=0}^{k}\Big(\mathbb{E}[\sigma^2\delta_{n,p_t}] + \mathbb{E}[(\Sigma_t^{d,1,2})^2] + \mathbb{E}[(\Sigma_t^{d,1,2})^2\delta_{n,p_t}]\Big) \ .\end{aligned}$$

These quantities are of the same kind as the three last terms in the sum of equation 4.4. So with the same techniques we get that

$$\frac{1}{n}\mathbb{E}\left[\sum_{t=0}^{N_{1,2}}(\sigma^2\delta_{n,p_t}+(\Sigma_t^{d,1,2})^2+(\Sigma_t^{d,1,2})^2\delta_{n,p_t})\right]=\underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right)\ .$$

So, we have

$$\mathbb{E}[(\hat{s}_{\mathbb{U}^1}(X)-\tilde{s}_{\mathbb{U}^1}(X))(\hat{s}_{\mathbb{U}^2}(X)-\tilde{s}_{\mathbb{U}^2}(X))]\leqslant\frac{\sigma^2\mathbb{E}[N_{1,2}]}{n}+\underset{n\to+\infty}{\mathrm{o}}\left(\frac{k}{n}\right)\ .$$

Finally, the following technical result allows to conclude the proof of Corollary 3, and thus the proof of Theorem5.

Lemma 3.

$$\mathbb{E}[M_{1,2}]=\frac{(k-2)(k-3)}{2(2k-1)}\left(1+\frac{4}{(k+1)(k-3)}\right)\ .$$

Hence,

$$\mathbb{E}[M_{1,2}]=\frac{k+1}{4}+\underset{k\to+\infty}{\mathrm{o}}(k)\ .$$

■

We then obtain that

$$\mathbb{E}[N_{1,2}]=\frac{3}{4}(k+1)+\underset{k\to+\infty}{\mathrm{o}}(k)\ .$$

Let us demonstrate lemma 3.

$$\mathbb{E}[M_{1,2}]=\sum_{r=1}^{k-2}\sum_{s=1}^{k-1}\mathrm{P}(U^2_{(s)}<U^1_{(r)}<U^1_{(r+1)}<U^1_{(r+2)}<U^2_{(s+1)})$$

As we know the distribution of ordered statistics (see e.g. Section 2.2 of David and Nagaraja (2003)), we can compute the following probability :

$$\begin{aligned}
&\mathrm{P}(U^2_{(s)}<U^1_{(r)}<U^1_{(r+1)}<U^1_{(r+2)}<U^2_{(s+1)})\\
&=\mathrm{P}(U^2_{(s)}<U^1_{(r)}\text{ and }U^1_{(r+2)}<U^2_{(s+1)})\\
&=\sum_{j=r+2}^{k}\sum_{i=0}^{r-1}\frac{k!}{i!(j-i)!(k-j)!}\mathbb{E}[(U^2_{(s)})^i(U^2_{(s+1)}-U^2_{(s)})^{j-i}(1-U^2_{(s+1)})^{k-j}]\\
&=\sum_{j=r+2}^{k}\sum_{i=0}^{r-1}\frac{k!}{i!(k-j)!}\frac{k!}{(s-1)!(k-(s+1))!}\frac{(i+s-1)!(2k-(j+s)-1)!}{(2k)!}
\end{aligned}$$

So,

$$\begin{aligned}
&\mathbb{E}[M_{1,2}] \\
&= \frac{(k!)^2}{(2k)!} \sum_{r=1}^{k-2} \sum_{s=1}^{k-1} \left(\sum_{i=0}^{r-1} \begin{pmatrix} i+(s-1) \\ i \end{pmatrix} \right) \left(\sum_{j=r+2}^{k} \begin{pmatrix} k-j+k-(s+1) \\ k-j \end{pmatrix} \right) \\
&= \frac{(k!)^2}{(2k)!} \sum_{r=1}^{k-2} \sum_{s=1}^{k-1} \begin{pmatrix} r-1+s \\ r-1 \end{pmatrix} \begin{pmatrix} 2k-r-2-s \\ k-r-2 \end{pmatrix}
\end{aligned}$$

(by elementary properties of binomial coefficients (see e.g. Graham et al. (1989) p.160))

$$= \frac{k-2}{4(2k-1)} \sum_{t=0}^{2k-5} \sum_{r=t-k+2}^{t} \frac{\begin{pmatrix} t+1 \\ r \end{pmatrix} \begin{pmatrix} 2k-3-(t+1) \\ k-3-r \end{pmatrix}}{\begin{pmatrix} 2k-3 \\ k-3 \end{pmatrix}}$$

(by defining $t = r + s$)

$$= \frac{k-2}{4(2k-1)} \sum_{t=0}^{2k-5} [\mathrm{F}_{\mathcal{H}(2k-3,t+1,k-3)}(t) - \mathrm{F}_{\mathcal{H}(2k-3,t+1,k-3)}(t-k+1)]$$

(where $\mathrm{F}_{\mathcal{H}(N,m,n)}$ denotes the cumulative distribution function of the hyper-geometric distribution)

$$\begin{aligned}
&= \frac{k-2}{4(2k-1)} 2 \sum_{t=0}^{k-3} \mathrm{F}_{\mathcal{H}(2k-3,t+1,k-3)}(t) \\
&= \frac{k-2}{2(2k-1)} \left[\sum_{t=0}^{k-4} \left(1 - \frac{\begin{pmatrix} t+1 \\ t+1 \end{pmatrix} \begin{pmatrix} 2k-3-(t+1) \\ k-3-(t+1) \end{pmatrix}}{\begin{pmatrix} 2k-3 \\ k-3 \end{pmatrix}} \right) + 1 \right] \\
&= \frac{k-2}{2(2k-1)} \Big(k-3+\frac{4}{k+1} \Big).
\end{aligned}$$

Chapitre 5

Conclusion

Sommaire

5.1 Bilan

1. Dans cette thèse, nous avons confirmé que les forêts aléatoires étaient un outil statistique très puissant. En effet, cette méthode d'estimation non-paramétrique ne demande essentiellement le réglage que d'un paramètre pour être très performante. De plus, elle donne de très bons résultats en prédiction aussi bien en classification qu'en régression pour des données de petite ou grande dimension. Enfin, elle est rapide à exécuter et peut-être facilement mise en oeuvre grâce aux algorithmes librement disponibles.

2. Nous avons également vu, que les forêts aléatoires pouvaient s'avérer très utiles pour faire de la sélection de variables. L'indice d'importance distingue bien les bonnes variables des variables de bruit, même dans des situations extrêmes où le niveau de bruit est gigantesque. Notre procédure automatique permet alors de proposer des sous-ensembles de variables très pertinents, de façon encore relativement rapide. Nous l'avons illustré sur de jeux de données réels de natures volontairement très différentes.

3. Au niveau théorique, nous avons montré, pour un cas simple de forêts aléatoires, les améliorations apportées par une forêt comparée à un arbre individuel. En effet, nous avons exhibé que la forêt avait un effet de réduction de variance. Nous insistons, ici, que ce résultat est obtenu en étudiant un terme de covariance entre deux arbres.

5.2 Perspectives

Nous terminons ce manuscrit par donner quelques perspectives et prolongements naturels de ce travail de thèse. La numérotation correspond à celle de la section précédente.

1. La méthode des forêts aléatoires de Leo Breiman (nommée Random Forests-RI) est très largement utilisée en pratique. Elle présente des résultats en prédiction très remarquables. Cependant, Geurts et al. (2006) ont introduit une nouvelle variante de forêts aléatoires qui semble apporter des améliorations en prédiction, tout en étant plus rapide à exécuter. Une piste de recherche serait alors d'essayer encore d'autres variantes de forêts aléatoires capables d'améliorer les performances. Ceci pourrait se faire en trouvant d'autres types d'aléa à rajouter avant ou pendant la construction des arbres, ou encore d'essayer de coupler plusieurs aléas déjà introduits dans les différentes variantes de forêts aléatoires connues à ce jour.

   Un point mérite une attention particulière : c'est la question de l'élagage. En effet, dans la littérature, les auteurs qui conseillent, en pratique, de ne pas élaguer les arbres, le justifient souvent par le fait que l'élagage n'apporte pas d'amélioration en prédiction et augmente significativement le temps de calcul. Cependant les résultats de Biau and Devroye (2010) ou encore les études par simulations de Fan et al. (2003) suggèrent qu'élaguer les arbres peut s'avérer intéressant. Une étude systématique comparant toutes les variantes de forêts aléatoires avec et sans élagage serait très utile et permettrait peut-être de comprendre dans quels cas il est préférable d'élaguer les arbres ou non.

2. Concernant la procédure de sélection de variables, il serait intéressant de la tester sur d'autres jeux de données difficiles. Nous pensons notamment à des données de type génomique (Goldstein et al., 2010) qui sont de ultra grande dimension : le nombre de variables étant de l'ordre de 300 000. Il faudra alors sûrement penser à des simplifications de la méthode ou à des variantes, car le temps de calcul dans ces situations extrêmes deviendra très problématique.

   Une direction possible serait également d'essayer le même type de procédure, mais avec d'autres méthodes de prédiction performantes. En effet, les schémas des forêts aléatoires et de notre procédure de sélection de variable sont très généraux. Nous pourrions utiliser ces schémas avec des prédicteurs de type SVM ou encore Boosting, par exemple. Ceci est à mettre en parallèle avec les travaux récents de Meinshausen and Bühlmann (2010) qui transposent les idées du Bagging à des procédures de sélection de variables à l'aide du Lasso.

3. Le but ultime de l'étude théorique des forêts aléatoires serait certainement d'obtenir des résultats précis expliquant les remarquables performances des forêts aléatoires, random forests-RI, de Leo Breiman. Pour cela il faudrait réussir à gérer le fait que dans ces forêts aléatoires, la construction de la partition de l'espace d'entrée se fait à l'aide de l'échantillon d'apprentissage (et en plus de l'aléa supplémentaire). Ceci semble cependant difficile à réaliser. Cette difficulté s'illustre, par exemple, en considérant la complexité de l'étude théorique de Bühlmann and Yu (2002) rien que sur les arbres à deux feuilles en dimension 1.

Un autre axe de recherche théorique serait d'essayer d'obtenir des résultats sur l'importance des variables. En effet l'indice d'importance des variables des forêts aléatoires réussit à identifier les variables d'intérêt et les variables de bruit. Cependant, à ce jour, aucun résultat théorique n'explique ce bon comportement.

Enfin, concernant l'étude des forêts purement aléatoires (pour lesquelles la partition de l'espace d'entrée est obtenue indépendamment de l'échantillon d'apprentissage), nous pensons que la quantité d'intérêt à étudier est la covariance entre deux arbres de la forêt. En effet, dans le Chapitre 4, nous établissons que la variance d'une forêt est majorée (à des termes négligeables près) par la covariance entre deux arbres. Ceci traduit bien l'heuristique initiale de Leo Breiman : il faut que les arbres soient très peu corrélés entre eux pour donner une forêt performante. C'est également un terme de covariance entre deux arbres qu'étudie Biau (2010) dans son analyse des forêts aléatoires intermédiaires (voir Section 1.4.3). De plus, au sujet du biais des forêts aléatoires, nous pensons qu'il est possible d'obtenir des résultats traduisant le fait que la forêt a un biais strictement inférieur à celui d'un arbre. L'étude du terme de biais mériterait en tout cas une attention particulière dans l'analyse des forêts aléatoires.

Bibliographie

Alizadeh, A.A.. Distinct types of diffues large b-cell lymphoma identified by gene expression profiling. *Nature*, 403 :503-511, 2000.

Alon, U., Barkai N., Notterman, D.A., Gish, K., Ybarra, S., Mack D., Levine, A.J.. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. U.S.A.*, 96(12) :6745-6750, 1999.

Amit, Y., Geman, D.. Shape quantization and recognition with randomized trees. *Neural computation*, 9(7) :1545-1588, 1997.

Anderson, T. J., Haubold, B., Williams, J. T., Estrada-Franco, J. G., Richardson, L., Mollinedo, R., Bockarie, M., Mokili, J., Mharakurwa, S., French, N., Whitworth, J., Velez, I. D., Brockman, A. H., Nosten, F., Ferreira, M. U., and Day, K. P.. Microsatellite markers reveal a spectrum of population structures in the malaria parasite plasmodium falciparum. *Mol Biol Evol*, 17(10) :1467–82, 2000.

Annan, Z., Durand, P., Ayala, F. J., Arnathau, C., Awono-Ambene, P., Simard, F., Razakandrainibe, F. G., Koella, J. C., Fontenille, D., and Renaud, F.. Population genetic structure of Plasmodium falciparum in the two main african vectors, anopheles gambiae and anopheles funestus. *Proc Natl Acad Sci U S A*, 104(19) :7987–92, 2007.

Archer, K.J., Kimes, R.V.. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52 :2249-2260, 2008.

Arlot, S.. V-fold cross-validation improved : V-fold penalization. Preprint. arXiv :0802.0566v2, 2008.

Babiker, H. A., Ranford-Cartwright, L. C., and Walliker, D.. Genetic structure and dynamics of plasmodium falciparum infections in the kilombero region of tanzania. *Trans R Soc Trop Med Hyg*, 93 Suppl 1 :11–4, 1999.

Bach, F.R.. Bolasso : model consistent Lasso estimation through the bootstrap. In the *Proceedings of the 25th International Conference on Machine learning*, p.33-40, 2008.

Banerjee, M. and McKeague, I.W.. Confidence sets for split points in decision trees. *Annals of Statistics*, 35(2) :543-574, 2007.

Bartlett, P.L., Traskin, M.. AdaBoost is Consistent. *Journal of Machine Learning Research*, 8 :2347-2368, 2007.

Bell, A. S., de Roode, J. C., Sim, D., and Read, A. F.. Within-host competition in genetically diverse malaria infections : parasite virulence and competitive success. *Evolution*, 60 :1358–71, 2006.

Ben Ishak, A., Ghattas, B.. Sélection de variables pour la classification binaire en grande dimension : comparaisons et application aux données de biopuces. *Journal de la SFdS*, 149(3) :43-66, 2008.

Biau, G., Devroye, L., Lugosi, G.. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9 :2039-2057, 2008.

Biau, G., Cérou, F., Guyader, A.. On the rate of convergence of the Bagged Nearest Neighbor Estimate. *Journal of Machine Learning Research*, 11 :687-712, 2010.

Biau, G., Devroye, L.. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101 :2499-2518, 2010.

Biau, G.. Analysis of a Random Forests Model. Rapport de recherche, Université Paris VI, 2010.

Boudin, C., Diop, A., Gaye, A., Gadiaga, L., Gouagna, C., Safeukui, I., and Bonnet, S.. Plasmodium falciparum transmission blocking immunity in three areas with perennial or seasonal endemicity and different levels of transmission. *Am J Trop Med Hyg*, 73(6) :1090–5, 2005.

Boudin, C., Van Der Kolk, M., Tchuinkam, T., Gouagna, C., Bonnet, S., Safeukui, I., Mulder, B., Meunier, J. Y., and Verhave, J. P.. Plasmodium falciparum transmission blocking immunity under conditions of low and high endemicity in cameroon. *Parasite Immunol*, 26 :105–10, 2004.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.. *Classification And Regression Trees*. Chapman & Hall, New York, 1984.

Breiman, L.. Bagging predictors. *Machine Learning*, 26(2) :123-140, 1996.

Breiman, L.. Arcing classifiers. *Annals of statistics*, 26(3) :801-824, 1998.

Breiman, L.. Randomizing Outputs to Increase Prediction Accuracy. *Machine Learning*, 40 :229–242, 2000a.

Breiman, L.. Some infinity theory for predictor ensembles. Technical Report 577, Berkeley, 2000b.

Breiman, L.. Random Forests. *Machine Learning*, 45 :5-32, 2001.

Breiman, L.. Consistency for a simple model of Random Forests. Technical Report 670, Berkeley, 2004.

Breiman, L., Cutler, A.. Random Forests, Berkeley. Available from http ://www.stat.berkeley.edu/users/breiman/RandomForests/, 2005.

Bühlmann, P., Yu, B.. Analyzing Bagging. *The Annals of Statistics*, 30(4), 927-961, 2002.

Cheze, N., Poggi, J.M., Portier, B.. Partial and Recombined Estimators for Nonlinear Additive Models. *Statistical Inference for Stochastic Processes*, 6(2) :155-197, 2003.

Cox, D.D. and Savoy, R.L.. Functional magnetic resonance imaging (fMRI) "brain reading" : detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, 19(2) :261–270, 2003.

Cutler, A., Zhao, G.. Pert - Perfect random tree ensembles. *Computing Science and Statistics*, 33 :490-497, 2001.

David H. A., Nagaraja H. N.. *Order Statistics*. Wiley Series in Probability and Statistics, 2003.

Day, K. P., Koella, J. C., Nee, S., Gupta, S., and Read, A. F.. Population genetics and dynamics of plasmodium falciparum : an ecological view. *Parasitology*, 104 Suppl :S35–52, 1992.

Dayan, P. and Abbott, L.F.. *Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2001.

de Roode, J. C., Helinski, M. E., Anwar, M. A., and Read, A. F.. Dynamics of multiple infection and within-host competition in genetically diverse malaria infections. *Am Nat*, 166 :531–42, 2005.

Díaz-Uriarte, R., Alvarez de Andrés, S.. Gene Selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7 :3, 2006.

Dietterich, T.. An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, Boosting and randomization. *Machine Learning*, 1-22, 1999.

Dietterich, T.. Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science*, 1857 :1-15, 2000.

Drakeley, C. J., Secka, I., Correa, S., Greenwood, B. M., and Targett, G. A.. Host haematological factors influencing the transmission of plasmodium falciparum gametocytes to anopheles gambiae s.s. mosquitoes. *Trop Med Int Health*, 4 :131–8, 1999.

Durot, C.. *Inférence non-paramétrique sous contraintes de forme et sélection de modèle*. Mémoire d'habilitation à diriger des recherches, Université Paris-Sud, Orsay, 2008.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.. Least angle regression. *Annals of Statistics*, 32(2) :407-499, 2004.

Efron, B. and Tibshirani, R.. Improvements on cross-validation : The. 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438) :548-560, 1997.

Eger, E., Kell, C. and Kleinschmidt, A.. Graded size sensitivity of object exemplar evoked activity patterns in human LOC subregions. *Journal of Neurophysiology*, 100(4) :2038-47, 2008.

Elias, J.. *Randomness in tree ensemble methods*. PhD thesis, Université of Montana, 2009.

Fan, J., Li, R.. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96 :1348-1359, 2001.

Fan, J., Lv, J.. Sure independence screening for ultra-high dimensional feature space. *J. Roy. Statist. Soc. Ser. B.*, 70 :849-911, 2008.

Fan, W. and Wang, H. and Yu, P.S. and Ma, S.. Is random model better ? On its accuracy and efficiency. In *Proceedings of the Third IEEE International Conference on Data Mining*, p.51-58, 2003.

Fan, W., McCloskey, J., Yu, P.S.. A general framework for accurate and fast regression by data summarization in random decision trees. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, p.146-157, 2006.

Freund, Y., Schapire, R.E.. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, p.148-156, 1996.

Friedman, J.H.. Multivariate adaptive regression splines. *The Annals of statistics*, 19(1) :1-67, 1991.

Genuer, R., Poggi, J.-M., Tuleau, C.. Random Forests : some methodological insights. Rapport de recherche 6729, Inria, 2008.

Genuer, R., Poggi, J.-M., Tuleau, C.. Variable selection using Random Forests. *Pattern Recognition Letters*, 31(14) :2225-2236, 2010a.

Genuer, R., Michel, V., Eger, E., Thirion, B.. Random Forests based feature selection for decoding fMRI data. In *Proceedings of the 19th COMPSTAT*, Paris august 22-27, p.1079-1087, 2010b.

Genuer, R.. Risk bounds for purely uniformly random forests. Rapport de recherche 7318, Inria, 2010c.

Geurts, P., Ernst, D., Wehenkel, L.. Extremely randomized trees. *Machine Learning*, 63(1) :3-42, 2006.

Gey, S. *Bornes de risque, détection de ruptures, boosting : trois thèmes statistiques autour de CART en régression.* Thèse, Université Paris-Sud, Orsay, 2002.

Gey, S., Nedelec, E.. Model selection for CART regression trees. *IEEE Transactions on Information Theory*, 51(2) :658-670, 2005.

Goldstein, B.A. and Hubbard, A.E. and Cutler, A. and Barcellos, L.F.. An application of Random Forests to a genome-wide association dataset : Methodological considerations & new findings. *BMC Genetics*, 11(1) :49-62, 2010.

Golub, T.R., Slonim, D.K, Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.. Molecular classification of cancer : Class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-537, 1999.

Graham R.L., Knuth D.E., Patashnik O.. *Concrete mathematics.* Addison-Wesley, 1989.

Grömping, U.. Estimators of Relative Importance in Linear Regression Based on Variance Decomposition. *The American Statistician*, 61 :139-147, 2007.

Grömping, U.. Variable importance assessment in regression : linear regression versus random forest. *The American Statistician*, 63(4) :308-319, 2009.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V.N.. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3) :389-422, 2002.

Guyon, I., Elisseff, A.. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157-1182, 2003.

Hallett, R. L., Dunyo, S., Ord, R., Jawara, M., Pinder, M., Randall, A., Alloueche, A., Walraven, G., Targett, G. A., Alexander, N., and Sutherland, C. J.. Chloroquine/sulphadoxine-pyrimethamine for gambian children with malaria : transmission to mosquitoes of multidrug-resistant plasmodium falciparum. *PLoS Clin Trials*, 1 :e15, 2006.

Hallett, R. L., Sutherland, C. J., Alexander, N., Ord, R., Jawara, M., Drakeley, C. J., Pinder, M., Walraven, G., Targett, G. A., and Alloueche, A.. Combination therapy counteracts the enhanced transmission of drug-resistant malaria parasites to mosquitoes. *Antimicrob Agents Chemother*, 48 :3940–3, 2004.

Hastie, T., Tibshirani, R. and Friedman, J.. *The Elements of Statistical Learning*. Second edition. Springer, 2009.

Hastings, I. M. and Wedgwood-Oppenheim, B.. Sex, strains and virulence. *Parasitol Today*, 13 :375–83, 1997.

Ho, T.K.. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8) :832-844, 1998.

Hogh, B., Gamage-Mendis, A., Butcher, G. A., Thompson, R., Begtrup, K., Mendis, C., Enosse, S. M., Dgedge, M., Barreto, J., Eling, W., and Sinden, R. E.. The differing impact of chloroquine and pyrimethamine/sulfadoxine upon the infectivity of malaria species to the mosquito vector. *Am J Trop Med Hyg*, 58 :176–82, 1998.

Ibragimov, I.A. and Khasminskii, R.Z.. *Statistical Estimation : Asymptotic Theory*. Springer-Verlag, New York, 1981.

Ishwaran, H.. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1 :519-537, 2007.

Jaramillo-Gutierrez, G., Rodrigues, J., Ndikuyeze, G., Povelones, M., Molina-Cruz, A., and Barillas-Mury, C.. Mosquito immune responses and compatibility between plasmodium parasites and anopheline mosquitoes. *BMC Microbiol*, 9 :154, 2009.

Jolliffe, I.T.. *Principal component analysis*, Springer series in statistics, 2002.

Kohavi, R., John, G.H.. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1-2) :273-324, 1997.

Lê Cao, K.-A., Gonçalves, O., Besse, P., Gadat, S.. Selection of Biologically Relevant Genes with a Wrapper Stochastic Algorithm. *Statistical Applications in Genetics and Molecular Biology*, 6(1) :article 29, (2007).

Lecué, G.. *Méthodes d'agrégation : optimalité et vitesses rapides*. Thèse, Université Paris VI, 2007.

Liaw, A., Wiener, M.. Classification and Regression by randomForest. *R News*, 2(3) :18-22, 2002.

Lin, Y. and Jeon, Y.. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474) :578-590, 2006.

Liu, F.T., Ting, K.M., Fan, W.. Maximizing tree diversity by building complete-random decision trees. In *Proceedings of the 9th PAKDD*, Hanoi may 18-20, p.605-610, 2005.

Meinshausen, N. and Bühlmann, P.. Stability selection. *Journal of the Royal Statistical Society, Series B*, Published Online, doi :10.1111/j.1467-9868.2010.00740.x, 2010.

Mzilahowa, T., McCall, P. J., and Hastings, I. M.. population structure and genetics of the malaria agent p. falciparum. *PLoS One*, 2 :e613, 2007.

Nwakanma, D., Kheir, A., Sowa, M., Dunyo, S., Jawara, M., Pinder, M., Milligan, P., Walliker, D., and Babiker, H. A.. High gametocyte complexity and mosquito infectivity of plasmodium falciparum in the gambia. *Int J Parasitol*, 38(2) :219–27, 2008.

Park, M.Y., Hastie, T.. An L1 regularization-path algorithm for generalized linear models. *J. Roy. Statist. Soc. Ser. B.*, 69 :659-677, 2007.

Paul, R. E., Lafond, T., Muller-Graf, C. D., Nithiuthai, S., Brey, P. T., and Koella, J. C.. Experimental evaluation of the relationship between lethal or non-lethal virulence and transmission success in malaria parasite infections. *BMC Evol Biol*, 4 :30, 2004.

Paul, R. E. L., Bonnet, S., Boudin, C., Tchuinkam, T., and Robert, V.. Aggregation in malaria parasites places limits on mosquito infection rates. *Infect Genet Evol*, 7(5) :577–86, 2007.

Paul, R. E. L., Brey, P. T., and Robert, V.. Plasmodium sex determination and transmission to mosquitoes. *Trends in Parasitology*, 18 :32–38, 2002.

Pichon, G., Awono-Ambene, H. P., and Robert, V.. High heterogeneity in the number of plasmodium falciparum gametocytes in the bloodmeal of mosquitoes fed on the same host. *Parasitology*, 121 (Pt 2) :115–20, 2000.

Poggi, J.M., Tuleau, C.. Classification supervisée en grande dimension. Application à l'agrément de conduite automobile. *Revue de Statistique Appliquée*, LIV(4), 39-58, 2006.

Prasad, A.M. and Iverson, L.R. and Liaw, A.. Newer classification and regression tree techniques : bagging and random forests for ecological prediction. *Ecosystems*, 9(2) :181-199, 2006.

Quinlan, J.R.. *C4. 5 : programs for machine learning* Morgan Kaufmann, 1993.

R Core Team Development.. *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, 2009.

Rakotomamonjy, A.. Variable selection using SVM-based criteria. *Journal of Machine Learning Research*, 3, 1357-1370, 2003.

Razakandrainibe, F. G., Durand, P., Koella, J. C., Meeüs, T. D., Rousset, F., Ayala, F. J., and Renaud, F.. "clonal" population structure of the malaria agent Plasmodium falciparum in high-infection regions. *Proc Natl Acad Sci U S A*, 102(48) :17388–93, 2005.

Read, A. F., Narara, A., Nee, S., Keymer, A. E., and Day, K. P.. Gametocyte sex ratios as indirect measures of outcrossing rates in malaria. *Parasitology*, 104 (Pt 3) :387–95, 1992.

Reece, S. E., Drew, D. R., and Gardner, A.. Sex ratio adjustment and kin discrimination in malaria parasites. *Nature*, 453 :609–14, 2008.

Ribaut, C., Berry, A., Chevalley, S., Reybier, K., Morlais, I., Parzy, D., Nepveu, F., Benoit-Vical, F., and Valentin, A.. Concentration and purification by magnetic separation of the erythrocytic stages of all human plasmodium species. *Malar J*, 7 :45, 2008.

Riehle, M. M., Markianos, K., Niaré, O., Xu, J., Li, J., Touré, A. M., Podiougou, B., Oduol, F., Diawara, S., Diallo, M., Coulibaly, B., Ouatara, A., Kruglyak, L., Traoré, S. F., and Vernick, K. D.. Natural malaria infection in anopheles gambiae is regulated by a single genomic control region. *Science*, 312(5773) :577–9, 2006.

Schall, J. J.. Do malaria parasites follow the algebra of sex ratio theory ? *Trends Parasitol*, 25 :120–3, 2009.

Segal, M. R., Cummings, M. P., and Hubbard, A. E.. Relating amino acid sequence to phenotype : analysis of peptide-binding data. *Biometrics*, 57(2) :632–42, 2001.

Sinden, R. E., Dawes, E. J., Alavi, Y., Waldock, J., Finney, O., Mendoza, J., Butcher, G. A., Andrews, L., Hill, A. V., Gilbert, S. C., and Basanez, M. G.. Progression of plasmodium berghei through anopheles stephensi is density-dependent. *PLoS Pathog*, 3 :e195, 2007.

Singh, D., Febbo, P.G., Ross, K., Jackson, D.G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A.A., D'Amico, A.V., Richie, J.P., Lander, E.S., Loda, M., Kantoff, P.W., Golub, T.R., Sellers, W.R.. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1 :203-209, 2002.

Sobol', I.M.. Sensitivity analysis for nonlinear mathematical models. *Mathematical Modeling & Computational Experiment*, 1(4) :407-414, 1993.

Strobl, C., Boulesteix, A.-L., Zeileis, A., Hothorn, T.. Bias in random forest variable importance measures : illustrations, sources and a solution. *BMC Bioinformatics*, 8 :25, 2007.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., Zeileis, A.. Conditional variable importance for Random Forests. *BMC Bioinformatics*, 9 :307, 2008.

Targett, G., Drakeley, C., Jawara, M., von Seidlein, L., Coleman, R., Deen, J., Pinder, M., Doherty, T., Sutherland, C., Walraven, G., and Milligan, P.. Artesunate reduces but does not prevent posttreatment transmission of plasmodium falciparum to anopheles gambiae. *J Infect Dis*, 183 :1254–9, 2001.

Taylor, L. H. and Read, A. F.. Determinants of transmission success of individual clones from mixed-clone infections of the rodent malaria parasite, plasmodium chabaudi. *Int J Parasitol*, 28 :719–25, 1998.

Tibshirani, R.. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1) :267-288, 1996.

Vapnik, V.N.. *The nature of statistical learning theory*. Springer Verlag, 2000.

Vaughan, J. A.. Population dynamics of Plasmodium sporogony. *Trends Parasitol*, 23(2) :63–70, 2007.

Wargo, A. R., de Roode, J. C., Huijben, S., Drew, D. R., and Read, A. F.. Transmission stage investment of malaria parasites in response to in-host competition. *Proc Biol Sci*, 274 :2629–38, 2007.

Weston, J., Elisseff, A., Schoelkopf, B., Tipping, M.. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3 :1439-1461, 2003.

Xiang, L., Lee, A., Yau, K., and McLachlan, G.. A score test for overdispersion in zero-inflated poisson mixed regression model. *Statistics in medicine*, 26(7) :1608–1622, 2007.

Zeileis, A. and Jackman, C. K. S.. Regression Models for Count Data in **R**. *Journal of Statistical Software*, 27, 2008.