COMMUNAUTE FRANÇAISE DE BELGIQUE
FACULTE UNIVERSITAIRE DES SCIENCES AGRONOMIQUES DE
GEMBLOUX

Etude du classement par forêts aléatoires d'échantillons perturbés à forte structure d'interaction

YVES BROSTAUX

Dissertation originale présentée en vue de l'obtention du grade de docteur en sciences agronomiques et ingénierie biologique

Promoteur : Pr J.-J. Claustriaux

2005

BROSTAUX Yves [2005]. *Etude du classement par forêts aléatoires d'échantillons perturbés à forte structure d'interaction* (thèse de doctorat). Gembloux, Faculté Universitaire des Sciences agronomiques, 168 p., 23 tabl., 25 fig.

Résumé. Parmi les méthodes de classement, les forêts d'arbres de décision (*Random Forests*, BREIMAN, 2001) offrent une souplesse indéniable tant en ce qui concerne la nature des variables descriptives et de la cible du classement que sur la forme du concept à modéliser. Leur diffusion en agronomie se heurte à un manque de connaissance concernant leur aptitude à apprendre des modèles fortement marqués par les interactions, en utilisant des échantillons de taille modeste et caractérisés par un bruit de fond aléatoire et des attributs diversement pertinents. La présente recherche a pour but de combler ce vide au moyen d'une exploration systématique de l'effet de ces différents facteurs ainsi que des paramètres des forêts, réalisée par simulation, en prenant comme base de comparaison des arbres de décision issus de la méthode *CART* (BREIMAN et al., 1984). Les résultats montrent que les forêts aléatoires les plus efficaces sont basées sur une sélection des attributs partiellement déterministe et une taille de forêt égale à au moins 100 voire 500 arbres. Ces forêts présentent globalement un avantage significatif en terme d'erreur de prédiction et ce dès les effectifs d'apprentissage faibles (50 individus). Cet avantage se réduit avec le niveau de perturbation général de l'échantillon (bruit et variables parasites) mais augmente avec la taille de celui-ci, les forêts aléatoires n'étant pas affectées par la limitation asymptotique de l'apprentissage affichée par la méthode *CART*.

BROSTAUX Yves [2005]. *Study of random forests classification of perturbed learning sets with high interaction structure* (thèse de doctorat), Gembloux, Belgium, Faculté Universitaire des Sciences agronomiques, 168 p., 23 tabl., 25 fig.

Summary. Amongst classification methods, forests of decision trees (*Random Forests*, BREIMAN, 2001) are highly versatile concerning descriptive attributes' or target variable's nature and shape of the concept to estimate. Their diffusion in agronomical sciences is slowed by a lack of information about their ability to learn models with high interaction structures using learning samples with few examples and affected by random noise and irrelevant attributes. This research aim to fill this gap by a systematic exploration of those factors' effects and of the parameters of the Random Forests method, which is done by computer simulations, taking as a reference the classification trees generated by Breiman's *CART* method (1984). Results show that generating random forests with a partially deterministic attributes selection and a forest size of at least 100 or 500 trees give the best prediction accuracy. Those random forests show a significant increase in prediction accuracy on *CART* trees, even for low learning sample size (50 examples). This advantage reduce with the global perturbation level (noise and irrelevant attributes) but increase with the learning sample size, as random forests aren't affected by the asymptotic limitation of the learning curve showed by *CART* method.

Un ami, c'est quelqu'un qui vous connaît bien et qui vous aime quand même.

Remerciements

Au terme de ce travail, je voudrais remercier tous ceux qui m'ont aidé et soutenu tout au long de ces années :

M. le Recteur A. Théwis, pour son dévouement à la cause statistique lié à la présidence de quatre jurys de thèse dans ce domaine en quelques mois ;

Mme Charles et MM. Gouet, Jijakli et Lebailly, qui par leurs relectures attentives et leurs remarques pertinentes m'ont permis d'améliorer grandement ce manuscrit ;

M. le Professeur Claustriaux, promoteur de cette thèse, pour m'avoir lancé sur les traces de ces arbres bizarres qui poussent les racines en l'air, mais également pour m'avoir ouvert la voie dans le monde de la statistique en m'accueillant dans son service ;

M. le Professeur Palm, pour ses talents d'analyse pointus qui ont non seulement rendu cette thèse meilleure mais qui m'ont appris la rigueur dans le travail scientifique ;

Mes collègues et amis de l'Unité de Statistique et informatique appliquées, pour avoir su créer une ambiance de travail chaleureuse et agréable ; Mme Josiane Austraet dite "Tantine" qui grâce à son sens de l'organisation nous décharge de nombreux soucis au quotidien, Mme Guylaine Delaplace-Melon, qui malgré son magnétisme irréfutablement funeste aux machines modernes retrouve en un clin d'œil la moindre parcelle de documentation cachée dans sa bibliothèque, Isabelle Carletti, qui m'a définitivement convaincu que les brunes ne comptent pas pour des prunes, Catherine Rasse, qui par sa sympathie et sa joie de vivre m'a empêché de tomber dans l'excès capillichrome inverse et Valérie Clissen, qui malgré son exil volontaire dans le cyberespace pédagogique est toujours restée proche de ses ex(collègues) ;

Cécile, ma compagne de (presque) tous les jours, pour m'avoir supporté lors des phases de bouclage de ce document, en avoir relu patiemment les premières versions et avoir réussi à tempérer ma tendance à vouloir tout expliquer en une seule phrase ;

Mes parents et grands-parents, tout d'abord parce que je ne serais pas là sans eux, mais aussi pour leur suivi très attentif de l'évolution de cette thèse ("Et alors, le doctorat… ça avance ?") ;

Et enfin mon chat Vanille, qui sait d'instinct quand venir ronronner sur mes genoux pour calmer ma tension.

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE[1]

De par ses compétences, l'ingénieur agronome est souvent amené à intervenir dans des processus de décision de natures très diverses. Nombre de ces décisions, comme par exemple l'identification d'une espèce animale, végétale ou d'un pathogène, l'interprétation d'imagerie satellitaire ou l'évaluation des chances de succès d'un projet de développement, peuvent être rattachées à des problèmes de *classement*[2†]. Ces problèmes consistent en l'attribution d'une classe préexistante à un individu ou une situation donnée, qui pourraient être dans les exemples précités le nom d'espèce, l'occupation du sol correspondant à un pixel ou la prévision du succès ou de l'échec d'un projet d'irrigation, à ne pas confondre avec la *classification*[3†], dont l'objet est la création *ex nihilo* de telles classes par groupement d'individus présentant des caractéristiques semblables.

Pour mener à bien ces tâches de classement†, l'agronome peut utiliser deux voies distinctes mais non exclusives :

- un raisonnement formel, basé sur une théorie clairement établie et formulée,
- une réflexion empirique, résultat de la synthèse des expériences passées.

[1] les termes signalés dans le texte par le symbole † sont définis dans le glossaire figurant à la fin de ce document.

[2] en anglais : *classification.*

[3] en anglais : *clustering*.

Par rapport à d'autres domaines appliqués tels que la physique ou la chimie, l'agronomie, située à la croisée de la biologie, des sciences environnementales et des sciences humaines, dispose d'un arsenal beaucoup plus restreint de théories de type déterministe. En effet, la multiplicité des facteurs intervenant dans les processus naturels, la complexité de leurs interactions et la variabilité du matériel biologique brouillent leur mise en équation. Cet état de fait limite notre recours à la voie purement théorique et nous conduit à extraire principalement nos connaissances de l'examen minutieux des expériences passées, ce qui a d'ailleurs été à l'origine de l'avènement de l'expérimentation et de l'inférence statistique (FISHER, 1925).

Cependant, l'inférence statistique n'offre qu'une utilité marginale dans les procédures de synthèse conduisant aux classements†, car elle repose sur des hypothèses de travail, qui doivent elles-mêmes être au préalable extraites des expériences passées. Elle intervient donc éventuellement en phase de confirmation, à la suite du classement, pour en vérifier les hypothèses et non pour les élaborer.

Historiquement, cette dernière tâche de synthèse et de recherche des processus qui sous-tendent l'appartenance des individus à différentes classes ou catégories a d'abord été l'apanage d'experts humains, spécialistes du domaine concerné. Mais avec l'augmentation du volume des données et de la complexité des problèmes traités, ces experts sont aujourd'hui souvent assistés voire remplacés par des algorithmes informatiques. Ces derniers sont en effet mieux adaptés aux représentations multidimensionnelles nécessaires à la résolution de ces problèmes que l'esprit humain, au détriment toutefois de la plasticité qui caractérise celui-ci.

Pour extraire des règles de classement† d'un ensemble d'exemples existants, une catégorie particulière de ces algorithmes utilise la construction d'arbres hiérarchisés. Les branches† de ces arbres sont formées par des tests logiques portant sur les caractéristiques des individus étudiés et choisis de manière à discriminer au mieux les différentes classes existantes, formant au final des classificateurs dont

l'usage rappelle celle des clés de détermination botaniques. Ces classificateurs sont appelés arbres de décision[4†].

Ces procédures offrent de nombreux avantages, notamment l'absence d'hypothèses concernant la distribution des populations cibles[†] et la possibilité de traiter conjointement des données numériques ou qualitatives, ce qui leur confère une grande faculté d'adaptation. Elles fournissent également une représentation synthétique claire du processus de classement[†], dont l'interprétation est aisée pour l'utilisateur même néophyte dans le domaine.

Cependant, dans leur forme classique matérialisée notamment par l'algorithme *CART* développé par BREIMAN, FRIEDMAN, OLSHEN et STONE, 1984, les méthodes de construction des arbres de décision présentent certaines lacunes dans la détection des structures d'interaction. Cette limitation est notamment liée au mode d'extension de l'arbre basé sur des tests univariés et donc aveugles à bon nombre de ces interactions. Or, les problèmes associés au domaine agronomique dépendent régulièrement de l'action combinée et réciproque de multiples facteurs, dessinant ainsi des structures d'interactions variées et complexes.

Parmi les solutions envisagées à ce problème, beaucoup apportent une réponse partielle, limitée à certaines catégories de données ou d'interactions. Néanmoins une méthode récente, les *forêts aléatoires*[5†] (BREIMAN, 2001), offre des perspectives intéressantes dans le traitement des interactions complexes. Celle-ci est basée sur l'agrégation des résultats de plusieurs arbres et fut développée initialement dans l'objectif distinct d'améliorer la stabilité des prédictions.

Peu d'informations sont toutefois disponibles sur l'évolution des performances en prédiction de ces méthodes sur une gamme de situations proche des conditions rencontrées dans les domaines agronomiques et biologiques, à savoir des échantillons d'effectifs faibles à modérés (quelques dizaines à quelques centaines), caractérisés par un bruit de fond non négligeable, des variables de pertinences très inégales

[4] en anglais : *decision trees*.
[5] en anglais : *random forests*.

pour le classement† final et illustrant un processus dépendant de nombreuses interactions.

L'objectif de la présente recherche est de combler cette lacune et de vérifier par une étude systématique l'influence de ces derniers paramètres sur la qualité des résultats fournis par les forêts aléatoires†, et par là leur adéquation aux tâches de classement† rencontrées par l'agronome. Dans ce but, une étude par simulation a été conduite, de manière à pouvoir contrôler chacun des paramètres relatifs aux échantillons utilisés, ce qui est impossible avec des jeux de données réels, aux caractéristiques internes généralement inconnues.

Nous présenterons tout d'abord les méthodes de classement† basées sur la génération d'arbres de décision sur un plan bibliographique, en les replaçant dans leur contexte et en décrivant leurs principes de base et leurs limitations. Nous en profiterons pour détailler plus avant le mode de fonctionnement des deux algorithmes qui serviront de base à notre étude par simulation, à savoir les algorithmes *CART* (BREIMAN *et al.*, 1984) et *Random Forest* (BREIMAN, 2001) et exposer les résultats de leur application sur un cas agronomique réel ayant trait à la phytopathologie (Chapitre I).

Ensuite, nous aborderons la description de l'étude par simulation et des paramètres retenus lors de son élaboration, qu'ils aient trait à l'échantillon (effectif, niveau du bruit de fond, taux de présence de variables non pertinentes), aux interactions caractérisant le concept† sous-jacent (complexité d'apprentissage) ou aux paramètres des algorithmes *Random Forest* (taille de la forêt et mode de présélection des variables). Les étapes de génération des données, d'exécution des procédures de classement† et de compilation des résultats, constituant de manière globale l'algorithme de simulation, seront également passées en revue au cours de ce chapitre (Chapitre II).

Le Chapitre III sera consacré à la description des procédures générales utilisées pour analyser les données générées par la simulation, et plus particulièrement les effets des facteurs de la simulation sur le taux d'erreur des classificateurs générés. Les analyses ont été divisées en deux phases, la première traitant des paramètres liés aux algorithmes de génération des forêts aléatoires† (§ III.2), tandis que la seconde aura

pour objet les paramètres caractérisant les échantillons d'apprentissage† (§ III.3).

Les résultats de ces analyses seront ensuite exposés et interprétés au cours du Chapitre IV. La première phase analytique nous permettra de sélectionner la combinaison optimale des paramètres de l'algorithme de génération des forêts aléatoires† (§ IV.2). Ces paramètres seront ensuite réutilisés pour établir le comportement de ces classificateurs face aux variations des caractéristiques des échantillons d'apprentissage† tels que leur taille, le niveau de bruit et/ou le taux de variables parasites† les affectant et la nature du concept† duquel ils sont issus (§ IV.3). Cette seconde analyse sera réalisée en utilisant l'algorithme *CART* comme base de référence.

Sur base de ces interprétations, nous conclurons enfin sur les possibilités d'utilisation des forêts aléatoires† dans le domaine agronomique et sur les éventuelles précautions ou restrictions à leur usage dans ce cadre. Nous examinerons également les prolongements envisageables de cette étude, toujours dans l'optique de sa diffusion et de sa mise en application en agronomie.

CHAPITRE I. MÉTHODES DE SEGMENTATION DE DONNÉES PAR ARBRES DE DÉCISION, APPROCHE BIBLIOGRAPHIQUE

I.1. INTRODUCTION

De tous temps, l'homme a cherché à organiser le monde qui l'entoure afin d'améliorer sa compréhension des phénomènes auxquels il assiste. On trouve dans toutes les sciences des traces de ces grands travaux de classification†, basés sur le regroupement des entités semblables en ensembles présentant des propriétés homogènes, depuis la biologie avec Aristote, Linné et Darwin, à la chimie de Mendeleïev, en passant par la psychologie avec les typologies de Myers-Briggs et autres théories jungiennes[6].

Une fois cette phase accomplie se pose alors le problème de l'attribution d'une classe existante aux nouveaux individus dans ces systèmes préétablis. Parfois la définition intrinsèque des classes suffit à résoudre cet écueil. Cependant, suite à l'indisponibilité de certaines informations à un moment précis ou à la méconnaissance du processus de classification†, on peut être amené à redéfinir une ou plusieurs règles pour ces nouvelles attributions. Sur quel(s) critère(s) puis-je affirmer

[6] JUNG (Gustav Carl, 1875-1961), psychiatre suisse, auteur de travaux sur l'inconscient et les types psychologiques humains.

que cet individu appartient à telle ou telle classe préétablie ? Cette question forme la base des problèmes de *classement*[7]†.

Dans les problèmes de classement†, la notion de *concept*† est définie comme un ensemble de règles caractérisant l'appartenance d'un individu à une classe d'objets donnée. Les méthodes de classement ont donc pour objectif la recherche de tels concepts, le plus souvent sur base d'un jeu de données composé d'exemples illustrant ceux-ci, selon un processus dit d'*apprentissage*[8].

La *segmentation de données par arbres de décision* regroupe une série de méthodes par lesquelles l'apprentissage est réalisé au travers de la construction d'une structure hiérarchique arborescente. Celle-ci est établie par partition récursive de l'espace défini par l'ensemble des caractères décrivant les exemples étudiés, de manière à délimiter des zones d'homogénéité croissante sur le plan de la variable cible. La nature de cette variable peut être qualitative (arbre de classement[9]) ou quantitative (arbre de régression[10]).

Ces méthodes présentent l'avantage de fournir une représentation graphique claire et intelligible des concepts† découverts sous forme d'un a*rbre de décision*[11]†. Elles sont utilisées dans de nombreux domaines dans lesquels le classement† des individus doit être accompagné d'une compréhension synthétique des facteurs qui sous-tendent ces décisions, tels que le diagnostic médical, la segmentation de la clientèle dans les secteurs bancaires et du marketing, l'interprétation d'imageries satellitaires, etc.

Après un tour d'horizon des principales méthodes de classement† existantes (paragraphe I.1), nous décrirons plus précisément la famille à laquelle appartiennent les algorithmes qui fondent la présente recherche, à savoir la segmentation de données par arbres de décision.

---

[7] en anglais : *classification*, par opposition aux travaux de classification (en anglais, *clustering*), qui construisent de nouvelles classes sur base des caractéristiques communes des individus.

[8] en anglais : *concept learning*.

[9] en anglais : *classification tree*.

[10] en anglais : *regression tree*.

[11] en anglais : *decision tree*.

Malgré une certaine diversité technique parmi les algorithmes mettant en oeuvre cette dernière méthode, la plupart se fondent sur des principes théoriques communs. Après avoir présenté ces principes (paragraphe I.2), nous passerons en revue divers problèmes courants dans les travaux de classement par arbres de décision et les solutions apportées par quelques méthodes spécifiques (paragraphe I.4). Nous terminerons par un bref historique des algorithmes de génération d'arbres de décision (paragraphe I.5), en se concentrant principalement sur deux d'entre eux, les méthodes *CART* (BREIMAN *et al.*, 1984) et *Random Forests* (BREIMAN, 2001), qui sont au centre des travaux de recherche qui prolongent ce chapitre. Un exemple d'application agronomique basé sur un problème de diagnostic phytosanitaire illustrera l'usage et les résultats de ces deux algorithmes.

I.2. PRINCIPALES MÉTHODES DE CLASSEMENT

Les méthodes de classement† ont pour objectif commun la découverte d'un estimateur assurant l'affectation d'une classe parmi c classes disponibles à un individu inconnu sur base de la connaissance d'un ensemble de m caractères le décrivant (appelés *attributs*[12] descripteurs†).

Cet estimateur est généralement construit au départ d'un jeu de n individus représentatifs de la population globale, décrits par ces mêmes caractères mais dont l'appartenance de classe est connue, et qui va servir d'exemple lors de la construction des règles d'attribution.

Les méthodes de classement† se distinguent essentiellement par la forme conceptuelle de ces règles et par les procédures de construction qui en découlent. Parmi les principales méthodes existantes, nous rappellerons brièvement les principes de l'analyse factorielle discriminante (§ I.2.1), des méthodes du noyau et des plus proches voisins (§ I.2.2) et des réseaux neuronaux (§ I.2.3) afin de comparer leurs approches respectives à celle de la segmentation par arbres, qui sera l'objet du paragraphe I.3.

[12] en anglais : *attributes.*

I.2.1. Analyse factorielle discriminante

L'analyse factorielle discriminante[13] fait partie des méthodes dites paramétriques. Elle s'appuie en effet sur une série d'hypothèses concernant la distribution des attributs descripteurs† au sein des différentes classes à attribuer.

La forme générale de la règle d'affectation se base sur la connaissance hypothétique des fonctions de densité de probabilité des populations à discriminer. Un individu est ainsi affecté à la classe dont la densité de population, compte tenu des caractéristiques de cet individu (probabilité *a posteriori*), est la plus élevée.

Cette méthodologie revient à diviser l'espace des m attributs† au moyen de $c(c - 1)/2$ hypercourbes, issues d'autant de fonctions discriminantes[14], isolant chacune des c classes dans une région distincte de cet espace. Ces hypercourbes sont les lieux des points d'égale densité de probabilité des différentes classes considérées deux par deux. Leur forme dépend des hypothèses retenues concernant la distribution des attributs au sein des classes. Lorsque les probabilités de classes sont *a priori* égales, elles constituent des hyperplans dans le cas de populations multinormales à m dimensions de même matrice de variances et covariances (analyse discriminante linéaire), ou des hyperparaboles lorsque la condition d'égalité des variances et covariances n'est pas retenue (analyse discriminante quadratique). Il existe également des extensions de cette méthode assouplissant la condition de multinormalité des distributions (analyse discriminante logistique) (PALM, 1994).

L'analyse factorielle discriminante est une méthode de classement† globale, en ce sens que chaque règle d'affectation qu'elle génère s'applique à l'ensemble des individus des différentes populations, sur base de l'ensemble de leurs m caractéristiques respectives. Etant donné la nature des hypothèses concernant la distribution des attributs, cette méthode est principalement adaptée aux attributs numériques. Néanmoins, le traitement de données binaires ou qualitatives est rendu possible par certains prétraitements visant à leur conversion en scores

[13] en anglais : *discriminant analysis*.
[14] en anglais : *discriminant function*.

numériques continus (par une analyse factorielle des correspondances, après éventuelle binarisation des attributs qualitatifs), plus connus sous le nom de procédure DISQUAL (SAPORTA, 1990).

I.2.2. Méthodes du noyau et plus proches voisins

Les méthodes du noyau[15], si elles se basent également sur des estimations des fonctions de densité, ne font aucune hypothèse concernant la forme de ces fonctions et appartiennent donc au domaine non paramétrique.

La règle d'affectation est identique à celle de l'analyse discriminante, un individu étant affecté à la classe dont la fonction de densité estimée au point correspondant dans l'espace des attributs† est la plus élevée. La distinction provient de la procédure d'estimation de cette densité de probabilité.

La densité de probabilité de la classe *i* est estimée en deux phases. Chaque individu de cette classe est d'abord entouré d'un certain volume fixe (ROSENBLATT, 1956), ou plus efficacement d'un halo de densité qui décroît au fur et à mesure que l'on s'éloigne de cet individu (PARZEN, 1962), entraînant un effet de lissage multidimensionnel de cette estimation. La densité de la classe est ensuite calculée en prenant la valeur moyenne de ces *n* densités locales en un point donné.

Parmi les fonctions de densité les plus couramment utilisées, appelées fonctions du noyau[16], on retrouve la loi normale multidimensionnelle, estimée cette fois au niveau local à la différence de l'analyse discriminante.

Plutôt qu'utiliser une zone d'influence de taille fixe autour de chaque point, une autre approche consiste à étendre le voisinage du point représentant l'individu à classer jusqu'à ce qu'il contienne *k* points de l'échantillon d'apprentissage†. Le point est alors affecté à la classe la plus représentée parmi ces *k* points. Cette règle, conceptuellement très simple puisque aucune fonction complexe de

[15] en anglais : *kernel methods*.
[16] en anglais : *kernel functions*.

densité ne doit être estimée, porte le nom de méthode des *k* plus proches voisins[17] (FIX et HODGES, 1989).

Dans ces deux méthodes, l'attribution d'une classe est donc réalisée au travers d'une règle d'affectation basée sur l'ensemble des attributs descripteurs†, mais estimée localement sur un sous-ensemble de l'échantillon de base. La règle d'affectation doit d'ailleurs être générée pour chaque nouvel individu à étiqueter, ce qui peut entraîner des coûts importants en temps de calcul lorsque le nombre d'individus et d'attributs descripteurs augmente.

A nouveau, ces méthodes sont destinées avant tout à être utilisées sur des données quantitatives, les distances entre les modalités de variables qualitatives non ordinales ou mêlant différents types d'attributs étant plus délicates (mais pas impossibles) à définir et à interpréter.

I.2.3. Réseaux neuronaux

Lorsque l'on compare les performances du cerveau humain et des processeurs informatiques, force est de constater que si ces derniers présentent un avantage écrasant et en constante augmentation en terme de vitesse de calcul, le cerveau humain demeure plus efficace dans la réalisation de tâches complexes telles que la reconnaissance de structures et le classement†. De cette constatation découle l'idée de concevoir un algorithme mimant le fonctionnement de base de notre système nerveux.

L'avantage décisif de l'homme sur la machine peut être relié à la structure de son système de traitement de l'information, formé de milliards d'unités élémentaires fortement interconnectées. Ces neurones pris isolément ont une capacité de traitement réduite, mais leur fonctionnement coordonné permet la réalisation de tâches extraordinairement complexes. L'unité de base de cette nouvelle méthode d'estimation, utilisable notamment dans les problèmes de classement†, est donc le neurone, défini pour la première fois par MCCULLOCH et PITTS, 1943.

---

[17] en anglais : *k-nearest neighbours method*.

Chaque neurone informatique forme une entité recevant un ou plusieurs signaux d'entrée, accompagnés de leur pondération respective, et délivrant à son tour un ou plusieurs signaux de sortie issus de l'exécution d'une fonction d'activation sur les premiers (Figure 1). Cette fonction peut prendre des formes diverses, seuil, combinaison linéaire, sigmoïde, …, cette dernière étant la plus usitée car elle autorise une grande souplesse dans les concepts† qui peuvent être modélisés.

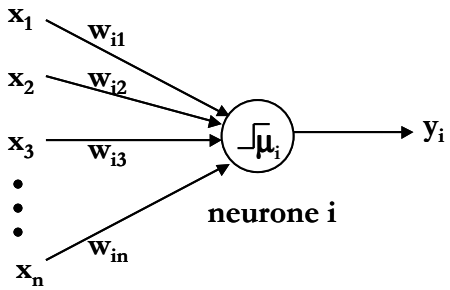


Figure 1. Schéma général d'un neurone informatique (McCulloch et Pitts, 1943).

Un réseau neuronal[18] est un algorithme d'apprentissage formé de telles unités de traitement interconnectées. Les réseaux neuronaux peuvent être organisés selon de nombreuses topologies différentes, dont une des plus courantes est le perceptron multicouches[19]. Les neurones y sont regroupés en trois classes, une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie, chaque entité d'une couche étant reliée exclusivement à toutes les entités de la couche suivante (Figure 2).

Une fois les fonctions d'activation et la structure du réseau choisies, l'apprentissage consiste en une modification dynamique des pondérations appliquées aux différentes liaisons, selon un algorithme conçu pour minimiser l'erreur finale de prédiction, au fur et à mesure de la présentation séquentielle des individus de l'échantillon, qui peut être répétée plusieurs fois (Prévot, 2004).

[18] en anglais : *neural network*.
[19] en anglais : *multilayer feedforward neural network*.

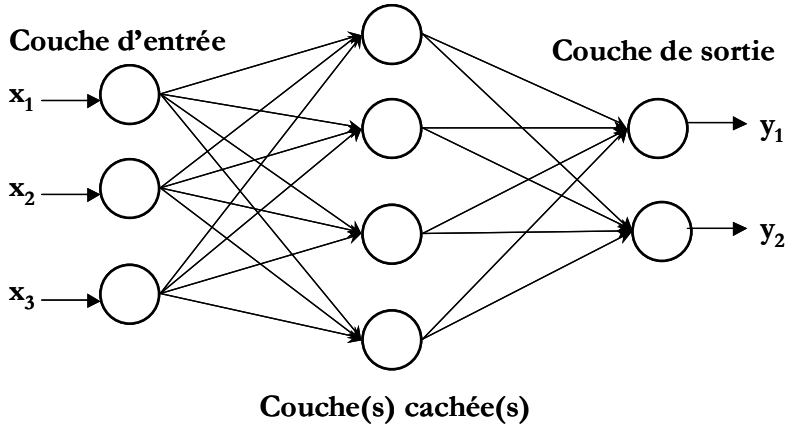


Figure 2. Structure générale d'un perceptron multicouche.

Ces méthodes extrêmement souples présentent toutefois certains inconvénients majeurs par rapport aux méthodes « traditionnelles ». Tout d'abord, la phase d'apprentissage consomme une puissance de calcul non négligeable. De plus, elles présentent certains risques de surapprentissage[20], modélisant non seulement le concept† mais également le bruit de fond qui l'accompagne. Enfin, l'interprétation synthétique des estimateurs qu'elles délivrent est rendue malaisée par leur structure interne complexe, phénomène connu sous le nom de syndrome de la boîte noire[21].

I.3. SEGMENTATION PAR ARBRES : PRINCIPES THÉORIQUES GÉNÉRAUX

I.3.1. Introduction

Les méthodes de segmentation par arbres utilisent l'information disponible concernant les individus non plus globalement, mais de manière hiérarchisée. Leurs procédures de classement† sont basées sur la partition récursive de l'espace des attributs† de manière à constituer des entités d'homogénéité croissante en regard de la variable cible. Cette homogénéisation est atteinte par un cheminement au travers d'une structure de décision hiérarchique. Chaque étape de ce

[20] en anglais : *overfitting*.
[21] en anglais : *black box syndrom*.

cheminement prend la forme d'un nœud de décision, matérialisé par un test dont le résultat permet de diviser le groupe d'individus I_0 ayant atteint ce nœud en une série de sous-groupes distincts I_1, I_2, …, I_n, eux-mêmes sujets à une éventuelle partition, et ainsi de suite (Figure 3).

Ces estimateurs issus des méthodes de segmentation récursive sont donc naturellement représentés sous forme d'arbres, définis par la théorie mathématique comme des "graphes orientés acycliques possédant une racine† unique" (SAFAVIAN et LANDGREBE, 1991).

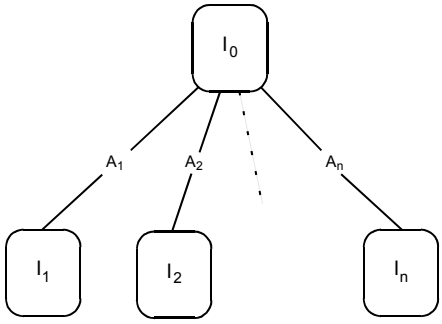


Figure 3. Partition d'un ensemble d'individus I_0.

On retrouve en association avec cette terminologie un vocabulaire descriptif en analogie avec son homologue végétal. Le point d'entrée de l'estimateur, constitué par le premier test, forme ainsi la *racine*[22]† de l'arbre, lequel test dirige les individus vers différentes *branches*[23]† selon son résultat, branches qui se subdivisent à leur tour grâce à d'autres tests, chaque point de connexion entre plusieurs branches portant le nom de *nœud*[24], pour aboutir enfin aux nœuds terminaux appelés *feuilles*[25]†, qui dans le cas des arbres de décision portent la prédiction finale.

Par rapport aux méthodes évoquées au paragraphe I.2, ce type de structure a le double avantage d'être aisément interprétable et d'être très flexible quant à la forme du concept† sous jacent.

---

[22] en anglais : *root node*.
[23] en anglais : *branches*, *edges*.
[24] en anglais : *node, vertex*.
[25] en anglais : *leaves*, *end-nodes*.

La première raison est qu'elle est formée d'une succession d'étapes décisionnelles élémentaires, donc simples à maîtriser et à transmettre aux praticiens.

Ces étapes sont en outre indépendantes et estimées localement sur base d'un sous-ensemble du jeu d'apprentissage et uniquement au travers des attributs jugés les plus pertinents. Ceci lui permet de s'adapter aux singularités locales des concepts de manière plus efficace que des méthodes globales de classement† à une seule phase, et donc de pouvoir aisément prendre en charge les interactions qui sont monnaies courantes dans les problèmes agronomiques.

Enfin, aucune hypothèse préalable ne doit être formulée concernant la structure du concept à modéliser (méthode non paramétrique), ni concernant la nature des attributs descripteurs† qu'ils soient quantitatifs, qualitatifs, binaires ou un mélange de ces différents types, comme cela est souvent le cas au cours des phases exploratoires des recherches, où un maximum d'informations de tous bords sont collectées ensemble.

Toutes ces propriétés autorisent une grande latitude dans les problèmes qui peuvent être traités par ces méthodes, qui posent moins de conditions d'applications que les méthodes paramétriques du type analyse factorielle discriminante, tout en restant plus faciles à interpréter et à transmettre que les méthodes neuronales ou des plus proches voisins.

Le recours aux méthodes de segmentation par arbres de décision implique le passage par différentes phases de travail. Un classificateur hiérarchique est d'abord construit par une série de partitions récursives de l'espace des attributs† (§ I.3.2), dont le choix est basé sur un critère de pertinence évaluant l'homogénéisation des espaces résultants (§ I.3.3). L'arbre de classification ainsi généré est ensuite éventuellement simplifié pour éliminer les risques de sur-apprentissage (§ I.3.4), avant de pouvoir fournir une prédiction quant à l'appartenance de classe de nouveaux individus (§ I.3.5).

I.3.2. Partitions de l'espace

I.3.2.1. Principes

Afin d'illustrer la méthodologie qui sous-tend la classification par arbre, nous reprenons ici un exemple tiré de QUINLAN, 1986. Nous disposons ainsi d'un échantillon d'apprentissage† de quatorze individus, repris dans le tableau 1, décrivant une série de situations météorologiques au travers de quatre attributs†, ainsi qu'une variable binaire spécifiant si les conditions mentionnées sont compatibles avec une activité non précisée par l'auteur.

Tableau 1. Exemple d'échantillon d'apprentissage (QUINLAN, 1986).

	Attributs				Classe
Situation	Prévision	Température	Humidité	Vent	Activité
1	ensoleillé	chaud	élevée	non	non
2	ensoleillé	chaud	élevée	oui	non
3	couvert	chaud	élevée	non	oui
4	pluie	doux	élevée	non	oui
5	pluie	froid	normale	non	oui
6	pluie	froid	normale	oui	non
7	couvert	froid	normale	oui	oui
8	ensoleillé	doux	élevée	non	non
9	ensoleillé	froid	normale	non	oui
10	pluie	doux	normale	non	oui
11	ensoleillé	doux	normale	oui	oui
12	couvert	doux	élevée	oui	oui
13	couvert	chaud	normale	non	oui
14	pluie	doux	élevée	oui	non

L'objectif de la segmentation par arbre est donc de construire une structure arborescente, utilisant une série de tests basés sur les attributs descripteurs†, qui permettrait de reclasser chaque individu représentant une situation météorologique donnée en apte ou non à l'activité, et ce de manière la plus exacte possible.

Un tel arbre construit par la méthode ID3, un algorithme de segmentation par arbre développé par QUINLAN, 1986 qui constitue avec la méthode CART de BREIMAN *et al.*, 1984 une des fondations principales des recherches ultérieures dans ce domaine, est représenté par la figure 4.

La procédure de classement† d'un exemple donné suit alors un cheminement simple décrit par l'arbre de décision†. La racine† de cet arbre est formée par un test concernant les prévisions. L'individu à reclasser est dirigé selon la valeur de cet attribut† vers la branche† adéquate, pour y être soumis le cas échéant à un nouveau test, et ce jusqu'à atteindre une feuille† terminale. L'attribution de l'étiquette de classe est effectuée via un vote à majorité simple basé sur le sous-ensemble de l'échantillon d'apprentissage† appartenant à cette feuille. Les feuilles de l'arbre représenté en figure 4 étant toutes pures, l'échantillon d'apprentissage est reclassé sans équivoque et sans erreur.

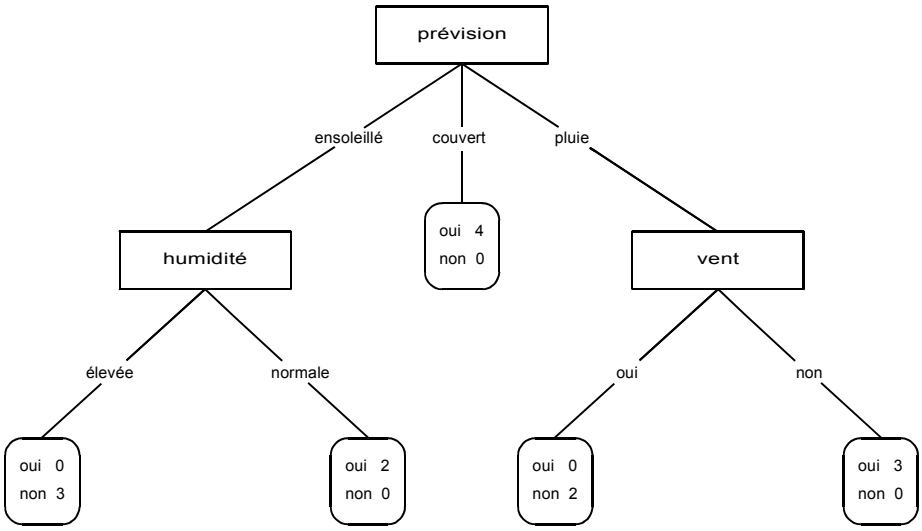


Figure 4. Arbre de décision† ID3 basé sur l'exemple de QUINLAN, 1986.

Remarquons l'absence de l'attribut température dans cet estimateur, non nécessaire à la détermination de l'activité dans l'exemple présenté, ce qui illustre bien la sélection réalisée par les estimateurs par arbres de décision au niveau des attributs† sur base de leur pertinence.

La partition proposée ici est toutefois loin d'être unique. Un grand nombre d'alternatives parviennent au même résultat de classement†.

Pour effectuer un choix parmi cet ensemble de solutions apparentes il convient de définir une mesure de qualité des partitions, première étape de la démarche conduisant à leur optimisation.

I.3.2.2. Qualité d'une partition

a) Erreur en généralisation

Comme tout estimateur, les méthodes de construction d'arbres de décision visent la prédiction d'une variable **Y** sur base d'un ensemble d'attributs† prédicteurs **X**, et ce avec une erreur de prédiction globale minimale sur la population dont est issue l'échantillon d'apprentissage†. L'arbre de décision† optimal doit donc conduire à une partition présentant un taux d'erreur en généralisation[26]† (TEG) minimal sur le domaine considéré.

Cette erreur engendrée par l'approximation de **Y** par $\hat{\mathbf{Y}}$ peut être traduite par une fonction mathématique, appelée fonction de coût[27] $l(y, \hat{y})$ dont la forme est liée à la nature de la variable dépendante et de son approximation (CHOU, 1991). A titre d'exemple, on peut citer parmi les fonctions de coût les plus simples :

- l'erreur de classification ($\hat{Y}$ est une variable de classe) ;

$$l(y,\hat{y}) = \begin{cases} 0 & si\ y = \hat{y} \\ 1 & si\ y \neq \hat{y} \end{cases}$$

- la somme des carrés des écarts résiduelle ($\hat{\mathbf{Y}}$ est une variable numérique) ;

$$l(\mathbf{y},\hat{\mathbf{y}}) = \left\|\mathbf{y} - \hat{\mathbf{y}}\right\|^2$$

- le logarithme de la vraisemblance ($\hat{\mathbf{Y}}$ est une matrice de probabilités conditionnelles**).**

$$l(y,\hat{\mathbf{y}}) = -\log \hat{p}(y)$$

Cette erreur, pour être correctement estimée, doit évidemment être calculée sur base d'un échantillon indépendant de celui ayant servi à la

[26] en anglais : *true misclassification rate*.
[27] en anglais : *loss function*.

construction de l'estimateur[28]. Le plus souvent, l'échantillon global est alors scindé aléatoirement en deux sous-ensembles, le premier étant utilisé pour la construction de l'arbre (échantillon d'apprentissage[29†]) et le second pour l'estimation de l'erreur (échantillon test[30†]).

Néanmoins il arrive que le taux d'erreur soit calculé directement sur base de l'échantillon d'apprentissage†, en raison notamment d'un effectif total déjà faible. Cette valeur prend alors le nom de taux d'erreur en resubstitution[31†] ou taux d'erreur apparent[32], constituant alors une estimation biaisée et largement optimiste du taux d'erreur en généralisation†.

Afin de pallier cet inconvénient tout en évitant le recours à un échantillon indépendant, l'estimation par validation croisée[33] forme une alternative couramment utilisée (BREIMAN *et al.*, 1984). Elle consiste à diviser l'échantillon total E en k (souvent $k = 10$) sous-ensembles d'effectifs approximativement égaux $E_1, E_2, \ldots, E_k$. La procédure de construction de l'arbre et d'estimation de l'erreur est alors répétée k fois en utilisant le sous échantillon $E - E_i$ comme échantillon d'apprentissage† et E_i comme échantillon test†. L'estimation finale est obtenue en calculant la moyenne des k erreurs obtenues sur les différents échantillons tests. Cette estimation est alors non biaisée mais présente une variabilité non négligeable, dépendante de la stabilité des résultats de la méthode de prédiction (§ I.4.4) (WEHBERG et SCHUMACHER, 2004). Une variante[34] consiste à considérer autant de sous-ensembles qu'il y a d'individus dans l'échantillon ($k = n$).

b) Complexité

L'erreur minimale n'est pas la seule qualité désirable pour un estimateur. Si on reprend l'exemple de QUINLAN, 1986 traité au paragraphe I.3.2.1, l'arbre ci-dessous permet tout comme celui de la

[28] en anglais : *test sample estimate*.
[29] en anglais : *training set, learning set*.
[30] en anglais : *test set*.
[31] en anglais : *resubstitution error rate*.
[32] en anglais : *apparent error rate*.
[33] en anglais : *cross validation error rate*.
[34] en anglais : *leave-one-out error rate*.

figure 4 de prédire l'activité sans erreur. Toutefois il est clair que ce dernier modèle est plus complexe à interpréter que le précédent.

Cette complexité peut être exprimée par diverses mesures, depuis les plus simples telles que le comptage du nombre de feuilles† de l'arbre de décision† final ou de ses nœuds (égal au nombre de feuilles moins un dans le cas d'un arbre dichotomique), jusqu'à des estimations complexes basées sur la quantité minimale d'information nécessaire pour reconstruire l'arbre (principe du *Minimum Description Length*, MDL) (QUINLAN et RIVEST, 1989; WALLACE et PATRICK, 1993; LEE, 2001).

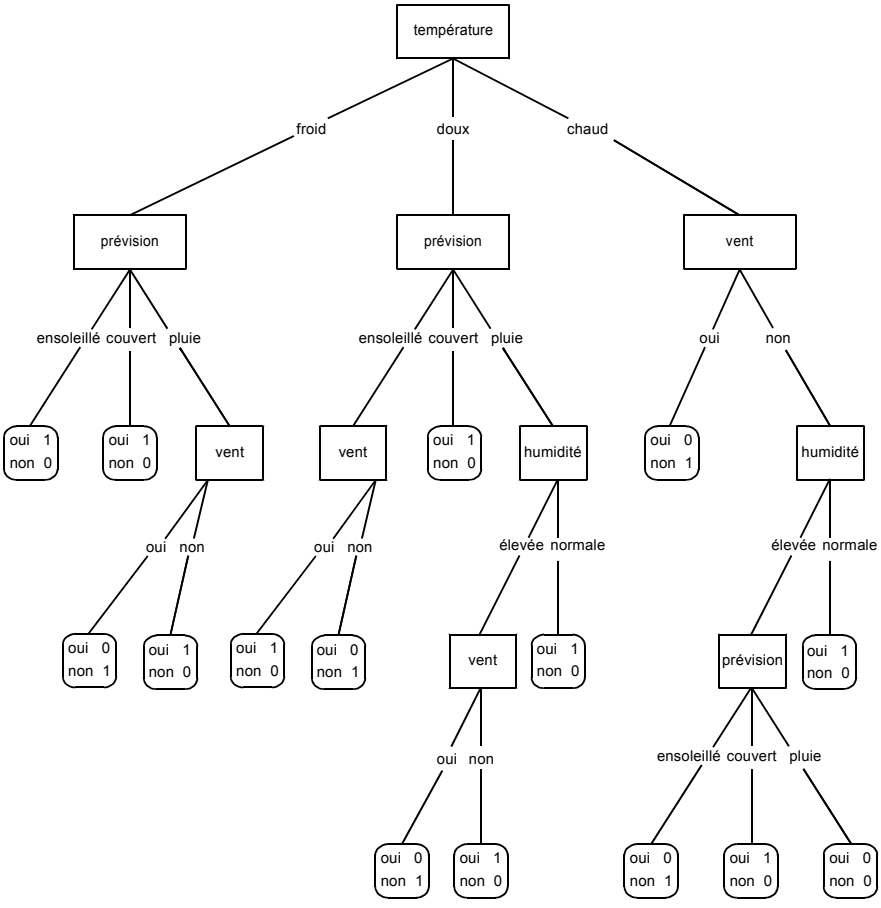


Figure 5. Arbre de décision† alternatif basé sur l'exemple de QUINLAN, 1986.

Il est bien sûr préférable d'obtenir les estimateurs les plus simples possibles pour un concept† donné. La raison la plus évidente est leur plus grande facilité d'utilisation, d'interprétation et de compréhension

(QUINLAN, 1987; BRESLOW et AHA, 1997). Il est de plus couramment établi que cette simplicité contribue à une amélioration des performances en généralisation du classificateur, et ce d'autant plus que l'on travaille avec des données réelles bruitées (QUINLAN, 1986; BLUMER, EHRENFEUCHT, HAUSSLER et WARMUTH, 1987; KOTHARI et DONG, 2001).

I.3.2.3. Exploration de l'espace des partitions

Une fois établis les instruments permettant de juger la qualité d'une partition, on peut définir le meilleur estimateur comme étant le plus petit arbre présentant une erreur globale en généralisation minimale (PAYNE et MEISEL, 1977).

L'obtention de cet optimum absolu n'est rendue certaine qu'au travers d'une recherche exhaustive parmi l'ensemble des partitions disponibles sur base des attributs† observés et de leurs combinaisons. Cependant, la taille de cet ensemble croît de manière exponentielle avec le nombre d'attributs, de leurs modalités et des opérateurs disponibles, ce qui rend cette option rapidement irréalisable pour des problèmes non triviaux (HYAFIL et RIVEST, 1976; CHOU, 1991).

Diverses restrictions doivent alors être envisagées concernant la nature des partitions admissibles afin de réduire de manière drastique l'espace de recherche. La plus courante consiste à n'autoriser que des partitions se basant sur un attribut† unique, c'est-à-dire perpendiculaires aux axes des attributs (FRIEDMAN, 1977; PAYNE et MEISEL, 1977; QUINLAN, 1986; GUEGUEN et NAKACHE, 1988). Ce nouvel espace est alors parcouru grâce à un algorithme glouton[35†] recherchant de manière récursive une succession de partitions optimales unidimensionnelles. Celles-ci forment alors une chaîne d'optima locaux dans chacun des sous domaines rectangulaires successifs créés par les partitions sélectionnées, d'où la dénomination de *partition récursive*[36] également donnée à ces méthodes. Un exemple de ce type de partition utilisant les données des iris de Fisher est donné à la figure 6.

---

[35] en anglais : *greedy algorithm*.

[36] en anglais : *recursive partition*.

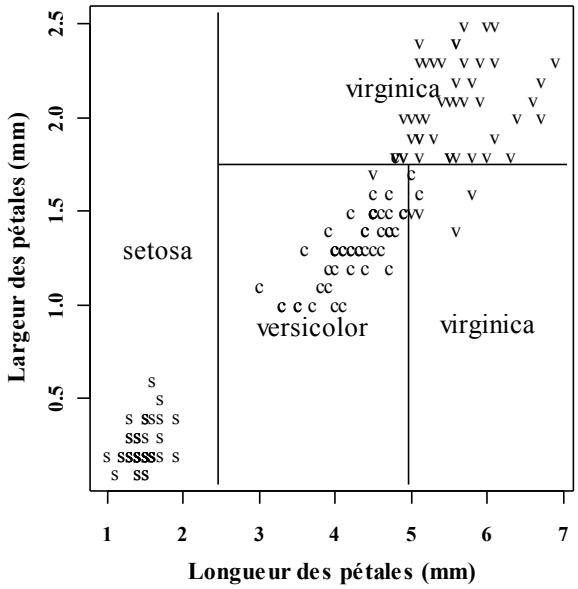


Figure 6. Partition du jeu de données des iris de Fisher par segmentation récursive unidimensionnelle (d'après VENABLES et RIPLEY, 1997).

Certaines méthodes introduisent une restriction supplémentaire en imposant que chaque partition soit de nature dichotomique, chaque nœud engendrant deux branches† filles, créées par regroupement des items de l'attribut† sélectionné (méthode CART, BREIMAN *et al.*, 1984). D'autres au contraire, comme la méthode ID3 (QUINLAN, 1986), autorisent des divisions multivaluées (une branche par item de l'attribut sélectionné). Les premières ont l'avantage de la simplicité d'interprétation et de représentation et sont moins sensibles au problème de fragmentation des données, c'est pourquoi elles tendent à s'imposer en pratique.

A chaque étape du processus de construction de l'arbre, l'algorithme analyse les données de l'échantillon et génère un jeu de partitions candidates sur base de l'ensemble des attributs†. Chaque partition est ensuite évaluée sur base de sa capacité à homogénéiser la composition des sous-ensembles ainsi créés. Les critères utilisés pour diriger ce choix seront abordés au paragraphe I.3.3. Cette procédure se répète pour chaque partition jusqu'à ce que toutes les partitions soient pures ou d'effectif faible.

Chaque nouvel ensemble issu d'une partition est traité de manière indépendante, autorisant ainsi les adaptations locales du modèle qui lui confèrent sa grande souplesse, tout en garantissant une amélioration générale de la qualité de l'estimateur. En effet, sous la condition de convexité† que respectent les fonctions de coût présentées au paragraphe I.3.2.2, on peut démontrer que chaque partition locale d'une branche† de l'arbre en cours de construction contribue à l'optimisation de l'estimateur au niveau global (CHOU, 1991), sans toutefois garantir l'obtention finale d'un optimum absolu. Cette procédure correspond à une approche inductive descendante[37] des problèmes de prédiction, partant du cas général vers le particulier.

Cette limitation drastique de l'espace de recherche aux cas unidimensionnels peut poser certains problèmes, notamment en présence d'interactions complexes entre attributs† (voir § I.4.3). Profitant de l'augmentation constante de la puissance de calcul, certaines méthodes plus récentes assouplissent ces restrictions en intégrant dans l'espace de recherche des groupes de partitions basées sur des combinaisons d'attributs par diverses catégories d'opérateurs (combinaisons linéaires, logiques, projections, etc.), mais l'immense majorité conservent le principe de partition récursive de l'espace des attributs.

I.3.3. Critères de partition

Le principe fondamental des méthodes de partition récursive nécessite le choix de tests appropriés, basés sur les attributs† du problème, permettant de faire progresser la connaissance acquise sur le concept†.

Cette progression peut être mesurée par la diminution de l'erreur associée à l'estimateur, mesurée par une fonction de coût (voir § I.3.2.2), qui devient alors un critère de partition[38]† servant d'étalon lors de la comparaison de la contribution des différents attributs† à la qualité de partition globale. Cette procédure garantit effectivement l'obtention d'une partition optimale de l'échantillon d'apprentissage†.

---

[37] en anglais : *top-down induction*.

[38] en anglais : *splitting criterion*.

Mais cette propriété intéressante n'est malheureusement pas garantie lorsque l'on passe à la généralisation de l'estimateur ainsi produit (BREIMAN *et al.*, 1984; PAZZANI, MERZ, MURPHY, ALI, HUME et BRUNK, 1994). Ce problème provient notamment de l'utilisation d'une erreur estimée sur l'échantillon d'apprentissage† et non plus sur un jeu indépendant, qui tend donc à surestimer largement les performances de l'arbre de décision† final. De plus, ces mesures ne prennent souvent en compte qu'une partie de l'information disponible dans le jeu d'apprentissage, se concentrant sur les erreurs mais négligeant leur distribution.

Ces mesures de qualité globale ne sont donc pas directement adaptées à l'élaboration d'un critère de partition†. Diverses alternatives ont alors été développées pour pallier ce défaut.

La plupart des critères couramment utilisés peuvent être associés à deux grandes familles : les dérivés de la théorie de l'information (basée sur l'entropie de Shannon) et les mesures de distance entre distributions de probabilités, auxquels s'ajoutent quelques critères particuliers (MURTHY, 1998).

I.3.3.1. Théorie de l'information

On retrouve dans cette catégorie deux critères développés pour un des premiers algorithmes de construction d'arbres de décision largement diffusés, ID3 (QUINLAN, 1986), dont nous avons déjà parlé au paragraphe I.3.2.1.

Ces deux critères sont basés sur la notion *d'impureté*[39]† d'un nœud. L'impureté I est définie comme toute fonction convexe† des probabilités de classe de ce nœud, qui est maximale lorsque toutes les classes sont équiprobables et minimale lorsqu'une classe unique a une probabilité égale à un.

Le premier critère représente le gain d'information[40] réalisé dans la description des classes lorsqu'on prend en compte un attribut† A à m modalités, l'impureté† étant mesurée par la quantité initiale

[39] en anglais : *impurity*.
[40] en anglais : *information gain*.

d'information d'un nœud estimée par la formule de l'entropie de Shannon (Équation 1), où les probabilités de classe sont estimées par les fréquences observées correspondantes.

$$I(E) = \sum_{j=1}^{c} -\frac{n_{j\cdot}}{n_{\cdot\cdot}} \log_2 \frac{n_{j\cdot}}{n_{\cdot\cdot}}$$

Équation 1. Quantité d'information du nœud *E* (*c* classes, $n_{j\cdot}$ étant l'effectif de la classe j).

Suivant la notation utilisée tout au long de ce paragraphe, n_{ji} représente ici l'effectif de la classe *j* du nœud fils *i*, le point (·) symbolisant une sommation sur l'indice associé. L'effectif total du nœud père est ainsi traduit par $n_{\cdot\cdot}$ tandis $n_{j\cdot}$ et $n_{\cdot i}$ correspondent respectivement à l'effectif de la classe *j* du nœud père et à l'effectif total du nœud fils *i*.

Le gain d'information est alors mesuré par la différence entre l'impureté† initiale du nœud parent et la somme pondérée des impuretés des p nœuds fils obtenus grâce à l'attribut† A ($p \leq m$), la pondération étant effectuée par les effectifs respectifs de ces derniers (Équation 2).

$$gain(E, A) = \Delta I = I(E) - \sum_{i=1}^{p} \frac{n_{\cdot i}}{n_{\cdot\cdot}} I(E_i)$$

Équation 2. Gain d'information lié à la partition du nœud *E* par l'attribut *A*.

Cette notion de gain informatif avait déjà été utilisée auparavant en tant que critère de partition†, sous le terme d'information mutuelle[41], dans d'autres algorithmes de classement† non paramétriques (GLESER et COLLEN, 1972; TALMON, 1986).

Ce critère ayant tendance à favoriser les attributs† présentant de nombreuses catégories, une correction a été proposée sous la forme du gain d'information relatif[42], qui rapporte le gain d'information à l'information intrinsèque de l'attribut† lui-même (QUINLAN, 1986).

[41] en anglais : *mutual information*.
[42] en anglais : *information ratio*.

$$IV(A) = -\sum_{i=1}^{p} \frac{n_{.i}}{n_{..}} \log_2 \frac{n_{.i}}{n_{..}}$$

Équation 3. Information intrinsèque de l'attribut A.

$$gain\ ratio(E, A) = \frac{gain(E, A)}{IV(A)}$$

Équation 4. Gain d'information relatif lié à un attribut A.

D'autres critères moins couramment utilisés ont également été définis dans le but de réduire ce biais en faveur des attributs† multivalués, comme la statistique *G* (MINGERS, 1987; 1989b) et la distance d_N de LÓPEZ DE MÄNTARAS, 1991, avec des performances comparables au précédent.

I.3.3.2. Distances entre distributions de probabilités

Cette famille reprend une série de critères issus d'horizons divers, dont la théorie statistique et l'informatique, qui ont en commun l'estimation d'une mesure de l'écart entre deux distributions de probabilités, estimées au départ des fréquences observées dans l'échantillon d'apprentissage†.

Parmi ces mesures, la distance de Kolmogorov-Smirnoff est certainement une des plus connues dans le domaine statistique. Elle varie en fonction de la distance maximale entre deux distributions de probabilités cumulées. Utilisée en tant que critère de partition† par FRIEDMAN, 1977 et UTGOFF et CLOUSE, 1996, ses performances sont identiques à celles obtenues par le gain d'information relatif.

Issue comme la précédente du domaine statistique, la variable χ^2 du test d'indépendance a également été utilisée dans ce domaine, tantôt directement (BELSON, 1959; KASS, 1980; LOH et SHIH, 1997), tantôt comme test de la signification d'une nouvelle partition construite sur base d'un autre critère (QUINLAN, 1986), notamment pour contrebalancer la tendance des critères basés sur l'entropie à favoriser les attributs† multivalués (WHITE et LIU, 1994).

Les critères les plus couramment employés dans cette catégorie ont été introduits par BREIMAN *et al.*, 1984 : l'indice de Gini et le critère

twoing, ce dernier étant plus particulièrement destiné aux problèmes à classes cibles multiples. Comme le gain d'information (QUINLAN, 1986), ces deux critères s'appuient sur la réduction de l'impureté† grâce à la partition du jeu d'apprentissage, mais ils diffèrent par la forme mathématique de la fonction d'impureté (Équation 5 et Équation 6), qui peut être mise en relation avec l'indice τ_b défini par GOODMAN et KRUSKAL, 1954.

$$I(E) = \sum_{j=1}^{c} \frac{n_{j.}}{n_{..}} \left(1 - \frac{n_{j.}}{n_{..}}\right)$$

Équation 5. Indice de Gini du nœud E (c classes, $n_{j.}$ étant l'effectif de la classe j).

$$\Delta I = \frac{\frac{n_{.L}}{n_{..}} \frac{n_{.R}}{n_{..}}}{4} \left(\sum_{j=1}^{c} \left| \frac{n_{jL}}{n_L} - \frac{n_{jR}}{n_R} \right| \right)^2$$

Équation 6. Critère twoing au nœud E (L et R représentent respectivement les indices des branches gauche et droite du nœud).

Parmi les autres critères développés par la suite, on retrouve le *Mean Posterior Improvement* de TAYLOR et SILVERMAN, 1993 et l'indice de séparation de classe de FAYYAD et IRANI, 1992, qui n'ont pas connu la diffusion des précédents.

I.3.3.3. Autres critères

Certains critères proposés pour la sélection des partitions n'entrent toutefois dans aucune de ces deux catégories. La plupart de ces mesures ont été construites en vue d'améliorer des aspects particuliers de la partition.

Ainsi LI et DUBES, 1986 et FRANK et WITTEN, 1998 proposent d'utiliser la valeur du test de permutation[43] en lieu et place du test χ^2 d'indépendance, en raison de son insensibilité à l'effectif du nœud. La qualité de l'approximation χ^2 diminue en effet avec la réduction de l'effectif des nœuds au fur et à mesure de la construction de l'arbre. Cette modification n'apporte toutefois pas d'amélioration du taux

[43] en anglais : *permutation statistic*.

d'erreur par rapport au gain d'information relatif. Pour la même raison, WHITE et LIU, 1994 et MARTIN, 1997 recommandent l'utilisation de la p-valeur associée au test exact de Fisher ou d'une de ses extensions comme critère de partition†.

Englobant à la fois la notion d'information et de complexité de l'estimateur, la longueur de description minimale[44], qui traduit la quantité d'information nécessaire pour coder la structure de l'arbre et son contenu (RISSANEN, 1983; LEE, 2001), a été intégrée plusieurs fois avec succès à une procédure de partition récursive (QUINLAN et RIVEST, 1989; OLIVER, 1993; WALLACE et PATRICK, 1993). Le principe général qui sous-tend la théorie du MDL veut que la meilleure théorie à induire d'un jeu de données soit celle qui minimise la somme de deux composantes :

1. la longueur de la théorie (la structure de l'arbre dans notre cas),

2. la longueur des données non couvertes par la théorie (les individus mal reclassés par l'arbre généré),

le tout exprimé en bits. La minimisation de la première composante équivaut à une simplification de l'arbre, tandis que la seconde correspond à une amélioration de l'erreur de prédiction.

Cette méthode livre des arbres de tailles plus modestes, présentant une erreur souvent réduite en comparaison avec ceux générés par un algorithme standard (BRESLOW et AHA, 1997) et moins sensible au problème des attributs† multivalués (MURTHY, 1998).

D'autres critères d'utilisation plus marginale existent également, telles les mesures de sensibilité et spécificité de KORS et HOFFMANN, 1997 ou l'algorithme de sélection d'attributs† *Strategist* de MCSHERRY, 1999, spécifiquement adaptés au domaine du diagnostic médical.

I.3.3.4. Synthèse

Outre cette classification qualitative, SHIH, 1999 propose une synthèse des principaux critères de partition† cités ci-dessus en une forme mathématique commune permettant de définir deux familles

[44] en anglais : *Minimum Description Length* (MDL).

principales, la première centrée autour de l'entropie (QUINLAN, 1986) et du χ^2 (KASS, 1980) et la seconde dérivant d'une généralisation du *Mean Posterior Improvement* (TAYLOR et SILVERMAN, 1993). Cette formalisation permet de préciser certains comportements de partition de ces critères, mais surtout met en évidence une sub-optimalité théorique de l'indice de Gini, qui ne partage pas la propriété de préférence exclusive définie par TAYLOR et SILVERMAN, 1993, propriété qui assure une disjonction optimale des classes lors de la partition.

Sur le plan empirique, MINGERS, 1989b compare les performances des critères G et χ^2, du gain d'information relatif et de l'indice de Gini à une procédure de sélection totalement aléatoire de la partition, en utilisant comme base l'algorithme ID3 (QUINLAN, 1986) sur un set de quatre jeux de données, trois réels et un artificiel. Il conclut de façon surprenante à une absence de différence significative sur le plan de l'erreur finale de prédiction de l'arbre entre les différents critères testés, en ce compris la sélection aléatoire des attributs†, la distinction se marquant par contre sur la taille de l'arbre généré.

BUNTINE et NIBLETT, 1992 reprennent une version modifiée de l'expérience de MINGERS, 1989b, limitée cette fois au gain d'information, à l'indice de Gini et à la sélection aléatoire, mais portant sur douze jeux de données et corrigeant une série de biais expérimentaux. Ils confirment ainsi l'absence de différence entre les deux critères informatifs, mais affirment leur supériorité sur la sélection aléatoire quant à la qualité des prédictions fournies.

BREIMAN, 1996c compare quant à lui l'indice de Gini, le gain d'information et le critère *twoing* sur le plan de leur comportement de partition optimal. Il apparaît que l'indice de Gini tend à produire des classes pures en isolant la classe majoritaire dans une des branches† de la partition. Ceci peut conduire à des partitions présentant un fort déséquilibre lorsque le nombre de classes est élevé et dès lors causer des problèmes de fragmentation des données. A l'opposé, le gain d'information et le critère *twoing* favorisent les partitions équilibrées en effectif, cette fois avec un risque de multiplicité des optima, et donc un problème de sélection et de stabilité, qui augmente également avec le nombre de classes.

En définitive, les diverses études sur le sujet n'ont pas pu mettre en évidence de différence absolue de performance entre les différents critères retenus (BREIMAN *et al.*, 1984; MURTHY, 1998), si ce n'est une sensibilité plus ou moins élevée aux attributs† multivalués, le gain d'information étant particulièrement concerné tandis que la longueur de description minimale est la plus robuste à ce problème (MURTHY, 1998). Le choix du critère de partition présente donc une importance secondaire pour les performances finales du classificateur. En pratique, cette indifférence relative quant au choix de ce critère se révèle très utile car la gamme de critères disponibles pour chaque algorithme est très étroite et intimement liée à celui-ci.

I.3.4. Taille optimale

La plupart des critères de partition† décrits ci-dessus, et plus particulièrement ceux basés sur la notion d'impureté† (§ I.3.3.1), ne permettent pas d'arrêter la partition prématurément, c'est-à-dire avant l'obtention de nœuds purs, toute partition quelle qu'elle soit entraînant une amélioration du critère du fait de la propriété de convexité† liée à cette notion.

Cette particularité devient rapidement gênante lorsque le jeu de données d'apprentissage est bruité ou que certains facteurs influents du concept† sont inconnus, ce qui s'avère être le cas dans la plupart des situations réelles, car elle entraîne un risque de sur-apprentissage (MINGERS, 1989a). L'algorithme d'induction continue alors la construction de l'estimateur au delà du concept sous-jacent et modélise les variations aléatoires de l'échantillon, en se basant sur des partitions non pertinentes (QUINLAN, 1986; BRESLOW et AHA, 1997).

Diverses stratégies ont dès lors été développées pour juger de la pertinence d'une partition et permettre l'élimination des nœuds excédentaires. Deux voies se distinguent à ce stade, d'une part les méthodes basées sur une règle d'arrêt[45] pendant la croissance de l'arbre, et d'autre part le recours à une sélection du meilleur sous-arbre, postérieure à la construction d'un modèle complet (élagage[46]†).

---

[45] en anglais : *stopping rule*, *pre-pruning*.
[46] en anglais : *pruning*, *post-pruning*.

I.3.4.1. Règles d'arrêt

La plupart des règles d'arrêt sont basées sur un seuil minimal à satisfaire pour valider une nouvelle partition d'un nœud. Sans cette validation, l'algorithme s'arrête pour la branche† concernée, le nœud devient terminal et est transformé en feuille†.

Parmi les règles les plus simples, on retrouve la fixation d'un effectif minimal pour le nœud candidat à la division (FRIEDMAN, 1977). Cette règle est peu robuste mais continue à être utilisée dans bon nombre d'algorithmes comme garde-fou en complément aux autres méthodes. Sa fixation reste arbitraire et largement dépendante du contexte de l'apprentissage (importance du bruit dans l'échantillon, effectif total, etc.). La valeur de ce minimum est le plus souvent fixée entre 5 et 10. Cependant elle présente le risque d'une élimination systématique des singularités du concept† à petite échelle (ESPOSITO, MALERBA et SEMERARO, 1997).

Le seuil peut également être fixé sur le critère utilisé pour la sélection de l'attribut†. Après avoir choisi la meilleure partition du nœud candidat, cette dernière n'est validée que si la valeur correspondante du critère de partition† est supérieure à un seuil prédéterminé (GLESER et COLLEN, 1972; QUINLAN, 1986). De nouveau se pose le problème de la fixation de ce seuil, d'autant que la valeur de la plupart des critères de partition† dépend de l'effectif du nœud. Cet inconvénient peut toutefois être évité en utilisant un critère indépendant de l'effectif et possédant une signification statistique classique, comme la statistique du test de permutation (LI et DUBES, 1986) ou la probabilité associée à un dérivé du test exact de Fisher (MARTIN, 1997).

Enfin, la règle d'arrêt peut être basée sur la signification statistique d'un critère indépendant de celui utilisé pour la partition, par exemple par un test χ^2 réalisé sur des partitions sélectionnées par le gain d'information (QUINLAN, 1986) ou par l'information mutuelle (TALMON, 1986). Ces tests présentent toutefois le même inconvénient que la fixation d'un effectif minimal concernant les particularités locales du concept†, les nœuds de faible effectif présentant la plupart du temps une faible signification statistique (ESPOSITO *et al.*, 1997).

I.3.4.2. Elagage

L'élagage† constitue une étape supplémentaire dans le processus de construction de l'estimateur, qui consiste à extraire au départ d'un arbre complètement développé T_{max} le meilleur sous-arbre T', obtenu par suppression d'une ou plusieurs branches† de T_{max} choisies par l'intermédiaire d'une recherche le plus souvent heuristique. Chaque branche élaguée est alors remplacée par le nœud duquel elle est issue, transformé par cette procédure en une feuille† terminale (BRESLOW et AHA, 1997) (Figure 7).

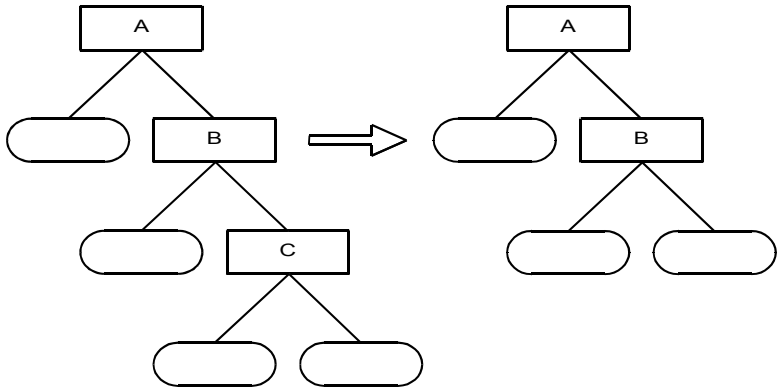


Figure 7. Elagage du sous-arbre issu du noeud C, remplacé par une feuille.

La suppression de sous-arbres du modèle complet a évidemment pour effet d'augmenter le taux d'erreur en resubstitution†, l'arbre ayant été indirectement construit de manière à minimiser ce paramètre. Néanmoins, en présence de sur-apprentissage dans un domaine bruité ou mal caractérisé, l'élagage† des branches† les moins pertinentes permet d'augmenter la précision en généralisation du modèle (BREIMAN *et al.*, 1984; QUINLAN, 1987; MINGERS, 1989a).

Le choix des branches† à élaguer repose pour la plupart sur des estimations de l'erreur de prédiction, basées sur un échantillon indépendant, une validation croisée ou une correction de l'erreur de resubstitution†. Lorsque la méthode nécessite l'utilisation d'un échantillon indépendant de celui utilisé lors de la phase de construction de l'arbre, l'échantillon d'apprentissage† est alors divisé

aléatoirement en deux sous-échantillons, le jeu dit de construction[47] et celui d'élagage[48].

Les paragraphes suivants décrivent succinctement les principales méthodes d'élagage† rencontrées dans la littérature. Etant issues de publications anglophones, la plupart n'ont pas de traduction officielle en français. Nous utiliserons donc ici la terminologie anglaise afin de relier plus efficacement ce document avec la littérature existante. Les traductions françaises livrées sont purement personnelles et uniquement destinées à améliorer le confort de lecture.

a) Minimal cost-complexity pruning (MCCP)[49]

La première méthode d'élagage†, dite du coût-complexité minimal, a été développée par BREIMAN *et al.*, 1984 pour l'algorithme CART. Elle comprend deux étapes, la première générant une série T_i de sous-arbres de T_{max} en fonction d'un certain paramètre α, la seconde sélectionnant dans cette suite le meilleur sous-arbre de la séquence en utilisant une estimation du taux d'erreur en généralisation†.

Chaque membre T_{i+1} de la suite de sous-arbres est obtenu au départ de l'élément T_i par élagage† des branches† présentant l'augmentation minimale du taux d'erreur en resubstitution† (coût) par feuille† élaguée (complexité), cette suite se poursuivant du modèle complet T_{max} jusqu'au nœud racine†.

Le paramètre

$$\alpha = \frac{\mathrm{r}(t) - \mathrm{r}(T_t)}{\left|\Phi(T_t)\right| - 1},$$

mesurant cette augmentation porte le nom de paramètre de coût-complexité, où $\mathrm{r}(t)$ et $\mathrm{r}(T_t)$ représentent le taux d'erreur en resubstitution† respectivement du nœud t et du sous-arbre T_t issu du nœud t, et $\Phi(T_t)$ désigne l'ensemble des feuilles† de ce sous-arbre T_t. Le résultat de cette première étape est un ensemble de sous-arbres

[47] en anglais : *growing set*.
[48] en anglais : *pruning set*.
[49] en français : *coût-complexité minimal*.

emboîtés $T_{max}(\alpha)$, issus de T_{max} et caractérisés chacun par une valeur de α croissante.

Au cours de la seconde étape, le meilleur sous-arbre de $T_{max}(\alpha)$ est sélectionné sur base de sa précision en généralisation, estimée sur base d'un échantillon indépendant ou par une procédure de validation croisée.

Lors de l'estimation du paramètre α par validation croisée, l'échantillon d'apprentissage† E est divisé en k sous-échantillons E_i d'effectifs approximativement égaux. En se basant sur les sous-échantillons $E - E_i$, k ensembles d'arbres $T^1(\alpha)$, $T^2(\alpha)$, …, $T^k(\alpha)$ peuvent être construits selon la procédure décrite ci-dessus, dont la précision en généralisation de chacun des membres peut être prédite au moyen du sous-échantillon E_i complémentaire.

La précision de chaque membre de la séquence globale $T_{max}(\alpha_i)$ est ensuite estimée par la moyenne des taux d'erreurs des arbres $T^1(\alpha')$, $T^2(\alpha^2)$, …, $T^k(\alpha^k)$ les plus proches de $T_{max}(\alpha_i)$. Pour définir cette proximité, BREIMAN *et al.*, 1984 utilisent la valeur du paramètre α. Pour chaque arbre $T_{max}(\alpha_i)$, on définit α_i' comme la moyenne géométrique $\sqrt{(\alpha_i \alpha_{i+1})}$ de l'intervalle $[\alpha_i, \alpha_{i+1}[$. L'arbre de la séquence $T^1(\alpha)$ le plus proche de $T_{max}(\alpha_i)$ est défini comme celui présentant la valeur du paramètre α immédiatement inférieure à α_i'.

Une fois les taux d'erreurs des membres de $T_{max}(\alpha)$ connus, deux règles de sélection coexistent. L'arbre sélectionné peut être soit celui présentant le nombre d'erreurs de classification minimal (règle 0-SE), soit le plus petit arbre de $T_{max}(\alpha)$ présentant un taux d'erreur n'excédant pas le taux d'erreur minimal sur $T_{max}(\alpha)$ augmenté d'une fois son erreur standard (règle 1-SE), cette dernière valeur étant calculée par

$$\sigma_r = \sqrt{\frac{\rho(1-\rho)}{N}},$$

où ρ représente le taux d'erreur théorique du prédicteur, estimé ici par son taux d'erreur observé sur un échantillon indépendant ou par validation croisée, et N est l'effectif total de l'échantillon ayant servi à

établir cette estimation. L'utilisation de cette formule est toutefois sujette à caution dans le cas la validation croisée, l'hypothèse d'indépendance qui la sous-tend n'étant pas respectée pour les k sous-échantillons utilisés (GUEGUEN et NAKACHE, 1988; ESPOSITO *et al.*, 1997).

b) Reduced error pruning (REP)[50]

Cette méthode proposée par QUINLAN, 1987 est l'une des plus simples. Elle consiste à évaluer directement les performances des différents sous-arbres de T_{max} sur base d'un échantillon indépendant (jeu d'élagage). Pour chaque nœud t de T_{max}, le nombre d'erreurs de classification du jeu d'élagage du sous-arbre T_t issu de ce nœud est comparé aux erreurs commises si le nœud t était transformé en feuille† et la classe attribuée à la majorité simple des données observées. Le sous-arbre T_t est supprimé si la différence entre ces deux erreurs est à l'avantage du nœud t, si toutefois ce sous-arbre T_t ne contient aucun sous-arbre présentant une erreur inférieure. L'algorithme parcourt donc l'arbre des feuilles à la racine† (approche ascendante).

Elle présente l'avantage d'avoir une complexité algorithmique linéaire, mais montre une tendance à élaguer de manière excessive. Cet inconvénient est lié à l'utilisation exclusive d'un échantillon d'élagage indépendant, qui néglige l'information contenue dans le jeu d'apprentissage. Ce problème est d'autant plus important que l'échantillon d'élagage est souvent d'une taille nettement inférieure au précédent (ESPOSITO *et al.*, 1997). Cela peut par exemple conduire à la suppression de sous-arbres traitant des cas rares non représentés dans l'échantillon d'élagage. Elle est cependant la seule méthode garantissant l'obtention d'un arbre optimal pour l'échantillon d'élagage (QUINLAN, 1987).

c) Pessimistic error pruning (PEP)[51]

Afin de pallier l'inconvénient de la méthode précédente, QUINLAN, 1987 propose le recours à un seul échantillon pour la construction et l'élagage† de l'arbre. Néanmoins, comme nous l'avons vu au § I.3.3, l'utilisation du taux d'erreur en resubstitution† comme estimation des

[50] en français : *erreur réduite*.
[51] en français : *erreur pessimiste*.

performances en généralisation du modèle conduit à une valeur largement optimiste de ces performances et ne peut dès lors pas être utilisée telle quelle.

QUINLAN, 1987 introduit donc dans cette estimation une correction de continuité basée sur la distribution binomiale qui appliquée à chaque feuille† des sous-arbres évalués est censée réduire ce biais. L'erreur associée à un nœud *t* devient donc

$$e'(t) = e(t) + \tfrac{1}{2},$$

e(*t*) et n(*t*) représentant respectivement le nombre d'individus mal classés et l'effectif total du nœud *t*.

Sur cette même base, on obtient donc pour le sous-arbre T_t l'erreur suivante :

$$e'(T_t) = \sum_{s \in \Phi_{T_t}} \left[e(s) + \tfrac{1}{2} \right] = \sum_{s \in \Phi_{T_t}} e(s) + \frac{\left| \Phi_{T_t} \right|}{2},$$

Φ_{Tt} désignant l'ensemble des feuilles† du sous-arbre T_t.

Chaque nœud *t* est évalué selon cette méthode, en partant du nœud racine†. Le sous-arbre T_t issu de ce nœud est élagué si l'erreur estimée au nœud *t* est inférieure ou égale à celle du sous-arbre T_t augmentée d'une valeur égale à une fois son erreur standard, ceci afin de compenser le caractère optimiste du critère malgré la correction appliquée. L'erreur standard est calculée par la formule :

$$SE(e'(T_t)) = \sqrt{\frac{e'(T_t) \cdot \left(n(t) - e'(T_t) \right)}{n(t)}},$$

basée sur l'hypothèse d'une distribution binomiale de l'erreur.

Les principaux avantages de cette procédure sont, d'une part, sa rapidité d'exécution, liée à son approche descendante (les nœuds d'un sous-arbre élagué ne doivent donc plus être évalués) et à sa complexité algorithmique linéaire, et d'autre part, l'absence de recours à un jeu de données indépendant de l'échantillon d'apprentissage† (QUINLAN,

1987). Toutefois, comme le signalent MINGERS, 1989a et ESPOSITO *et al.*, 1997, cette correction ne repose sur aucune base théorique solide, étant employée totalement hors de son contexte statistique qui relève de l'approximation d'une distribution binomiale par une normale. L'utilisation de cette correction revient en fait à introduire dans la mesure une notion de coût lié à la complexité du modèle, en affectant une pénalité arbitraire de ½ à chaque feuille†.

d) Minimum error pruning (MEP)[52]

Basée à l'origine sur un échantillon d'apprentissage† unique tout comme l'erreur pessimiste de QUINLAN, 1987, la méthode proposée par NIBLETT et BRATKO, 1987 apporte également une correction à l'erreur observée, calculant ainsi un taux d'erreur attendu[53] destiné à estimer au mieux l'erreur en généralisation†.

MINGERS, 1989a a montré que la version originale de ce critère, qui supposait une distribution équiprobable des classes, était fortement sensible au nombre de classes du concept†. Dans sa version améliorée (CESTNIK et BRATKO, 1991), la correction se base sur la connaissance des probabilités *a priori* d'appartenance aux différentes classes. La probabilité qu'une observation du nœud t appartienne à la classe i se calcule comme suit :

$$\mathrm{p}_i(t) = \frac{\mathrm{n}_i(t) + m\pi_i}{\mathrm{n}(t) + m},$$

où π_i est la probabilité *a priori* de la classe i et m est un facteur de pondération déterminant l'impact de cette probabilité dans l'estimation de la probabilité *a posteriori* $\mathrm{p}_i(t)$.

L'affectation des classes aux différents nœuds s'effectuant à la majorité simple, le taux d'erreur attendu se calcule selon la formule suivante :

$$\mathrm{EER}(t) = \min_i \left[1 - \mathrm{p}_i(t)\right].$$

[52] en français : *erreur minimale*.
[53] en anglais : *expected error rate* (EER).

La procédure d'élagage† consiste à calculer pour chaque nœud interne *t*, partant des feuilles† vers la racine† de l'arbre (approche ascendante), le taux d'erreur attendu. Cette valeur est ensuite comparée au taux d'erreur attendu du sous-arbre T_t, calculé par la somme pondérée des erreurs des nœuds fils de *t*, la pondération étant donnée par la probabilité d'une observation appartenant à *t* atteigne chaque nœud fils.

Lors de l'application de cette méthode, les différentes probabilités nécessaires à son exécution sont estimées sur base des fréquences observées dans l'échantillon d'apprentissage†.

La sévérité de l'élagage† dépend de la valeur du paramètre *m*, une valeur élevée du paramètre entraînant généralement un élagage plus drastique, sans toutefois garantir la monotonicité de la relation entre *m* et la taille finale de l'arbre. Le problème de la fixation de *m* constitue donc un point critique de la méthode. CESTNIK et BRATKO, 1991 proposent de recourir à un expert connaissant le domaine étudié, tandis que ESPOSITO *et al.*, 1997 utilisent un jeu de données indépendant pour estimer la précision des arbres obtenus pour différentes valeurs de *m* afin d'appuyer leur choix final.

e) Critical value pruning (CVP)[54]

Proposée par MINGERS, 1987, cette méthode se base sur la fixation d'un seuil, appelé valeur critique, sur le critère de partition†. Si pour un nœud donné le critère n'atteint pas la valeur critique, le sous-arbre qui en est issu est élagué. Elle rejoint en cela les règles d'arrêt décrites précédemment. Toutefois, l'effet d'horizon caractéristique de ces méthodes est évité grâce à l'introduction d'une seconde condition : un sous-arbre ne peut être élagué que si aucun de ses nœuds fils n'est lui-même significatif.

Le degré d'élagage† est évidemment fortement dépendant de la valeur critique adoptée, un seuil plus élevé entraînant un élagage plus sévère. MINGERS, 1987 recommande d'établir une suite d'arbres élagués au départ de l'arbre complet T_{max} grâce à des valeurs de seuil croissantes

[54] en français : *valeur critique*.

et de sélectionner ensuite l'arbre avec les meilleures performances globales estimées sur un échantillon indépendant.

ESPOSITO *et al.*, 1997 font état de blocages prématurés potentiels de la procédure, liés notamment à l'utilisation du gain d'information relatif. De plus, la séquence d'arbres générée par cette méthode ne présente pas de garantie de contenir l'arbre présentant l'erreur minimale sur l'échantillon d'élagage†.

f) Error based pruning (EBP)

Cette méthode est celle implémentée par QUINLAN, 1993 pour son algorithme C4.5 et peut être considérée comme une amélioration de l'erreur pessimiste. Elle se base également sur un seul échantillon pour la construction et l'élagage† de l'arbre mais utilise cette fois une stratégie ascendante pour l'exploration des nœuds, partant des feuilles† jusqu'à la racine† de l'arbre. La principale innovation consiste toutefois en l'introduction d'une nouvelle opération, la greffe[55] d'une branche† d'un sous-arbre à la place de celui-ci, qui entre en concurrence avec l'élagage† pur et simple de ce sous-arbre.

L'estimation du taux d'erreur permettant de statuer sur la signification des branches† de l'arbre complet se base sur le calcul d'une limite de confiance supérieure LCS_α autour du taux d'erreur en resubstitution†, telle que

$$\mathrm{P}\left(\frac{\mathrm{e}(t)}{\mathrm{n}(t)} \le LCS_\alpha\right) = \alpha\,,$$

α représentant le niveau de confiance retenu sur l'intervalle.

En posant l'hypothèse d'une distribution binomiale E des erreurs au nœud t, présentant une probabilité de succès égale à p sur $\mathrm{n}(t)$ essais, la valeur de LCS_α peut être calculée comme étant la valeur de p pour laquelle on observe l'égalité

$$\mathrm{P}\big(E \le \mathrm{e}(t)\big) = \alpha$$

[55] en anglais : *grafting*.

Pour chaque nœud interne t, trois taux d'erreurs sont ensuite estimés et comparés : le taux d'erreur du nœud t, celui du sous-arbre T_t issu de t et enfin celui du sous-arbre $T_{t'}$ issu du nœud t', fils de t présentant l'effectif le plus important. En fonction du résultat de cette comparaison, le sous-arbre T_t est soit conservé, soit élagué, soit encore remplacé par le sous-arbre $T_{t'}$, de manière à préserver la solution offrant un taux d'erreur estimé minimal.

Un des principaux griefs à l'encontre de la méthode concerne la validité des hypothèses statistiques formulées concernant la distribution de l'erreur. L'arbre T_{max} ayant été construit pour ajuster au mieux les données, les individus atteignant les différents nœuds peuvent difficilement être considérés comme faisant partie d'un échantillon aléatoire et l'hypothèse d'une distribution binomiale de l'erreur peut également être remise en cause (ESPOSITO *et al.*, 1997).

I.3.4.3. Synthèse

Les méthodes basées sur les règles d'arrêt offrent des performances très inégales car elles souffrent d'un effet d'horizon (BREIMAN *et al.*, 1984; QUINLAN, 1993). Elles peuvent en effet empêcher la croissance d'un nœud peu informatif en soi, mais dont les ramifications subséquentes apportent une information pertinente à l'estimateur. C'est pourquoi cette approche a été largement délaissée au profit des procédures d'élagage†.

Il faut toutefois noter que les règles d'arrêt ont une bien meilleure efficacité algorithmique que les méthodes d'élagage† *a posteriori* (MARTIN et HIRSCHBERG, 1995), ce qui peut relancer leur intérêt dans des problèmes présentant un effectif et/ou une dimensionnalité élevée. Leur effet d'horizon peut d'ailleurs être réduit par certaines modifications de l'algorithme glouton† de base, comme la recherche en avant[56] ou la construction inductive d'attributs[57] (§ I.4.3).

Comparant les méthodes du coût-complexité minimal, de l'erreur réduite et de l'erreur pessimiste sur six jeux de données réelles et artificielles, QUINLAN, 1987 observe des performances en généralisation

[56] en anglais : *lookahead search*.
[57] en anglais : *inductive feature construction*.

très semblables entre ces trois méthodes, tout en remarquant la taille plus réduite des sous-arbres obtenus par le critère du coût-complexité.

MINGERS, 1989a étend cette comparaison aux cinq premières méthodes décrites ci avant, sur cinq jeux d'exemples (quatre réels et un artificiel), en combinaison avec quatre critères de partition†. La conclusion la plus importante de cette étude empirique est l'absence d'interaction entre les choix opérés lors de la phase de construction de l'arbre (critères de partition) et celle d'élagage†, qui confirme donc la validité d'une réflexion séparée concernant leur optimisation. Il constate également des différences en ce qui concerne la taille finale de l'arbre élagué et ses performances en généralisation entre les algorithmes d'élagage. En effet, les méthodes ne requérant pas un échantillon indépendant (l'erreur pessimiste et l'erreur minimale) ont tendance à conserver une taille finale plus importante tout en ayant des performances en généralisation significativement inférieures aux trois autres, tandis que l'erreur réduite et le coût-complexité minimal se distinguent à l'autre bout de l'échelle. Il convient toutefois de noter que c'est la version originale de l'erreur minimale qui a été utilisée et que de plus, par construction méthodologique de l'expérience, ces deux méthodes ont été testées sur un échantillon global d'effectif plus réduit (uniquement le *growing set*) que les trois autres nécessitant l'utilisation d'un échantillon indépendant (*growing set* + *pruning set*).

ESPOSITO *et al.*, 1997 corrigent ces biais dans leur propre expérience comparative, reprenant les six méthodes d'élagage† décrites ci-dessus (dont 4 variantes du coût complexité minimal, en fonction de l'utilisation d'un échantillon indépendant ou de la validation croisée, ainsi que de la règle d'une erreur standard ou non) testées sur 15 jeux de données. Sur base de leurs performances empiriques, aucune différence globale entre les méthodes utilisant ou non un échantillon indépendant lors de la phase d'élagage n'apparaît cette fois clairement. Individuellement, parmi les meilleures méthodes pour ce même critère, on retrouve essentiellement *l'error based pruning* de QUINLAN, 1993 et le coût-complexité minimal de BREIMAN *et al.*, 1984 utilisant la validation croisée, confirmant en partie les conclusions de MINGERS, 1989a concernant ce dernier algorithme.

En pratique, le choix des méthodes d'élagage†, tout comme celui des critères de partition†, reste toutefois limité car intimement lié à l'algorithme choisi. Les deux principaux algorithmes ayant été introduit avec succès hors des laboratoires de recherche en apprentissage automatique, à savoir CART (BREIMAN *et al.*, 1984) et C4.5 (QUINLAN, 1993), ont imposé leurs méthodes d'élagage, heureusement classées parmi les plus efficaces par les différentes études comparatives que nous venons de citer (*MCCP* et *EBP*).

Le tableau 2 résume les principales caractéristiques des six algorithmes d'élagage abordés dans ce document.

Tableau 2. Caractéristiques des principaux algorithmes d'élagage (ESPOSITO *et al.*, 1997)

Méthode	Stratégie	Echantillon d'élagage	Comportement
Minimal cost-complexity (BREIMAN *et al.*, 1984)	ascendante	oui non (VC)	sur-élagage (1SE), bonne précision (sauf 1SE)
Reduced error (QUINLAN, 1987)	ascendante	oui	sur-élagage
Pessimistic error (QUINLAN, 1987)	descendante	non	bonne précision
Minimum error (CESTNIK et BRATKO, 1991)	ascendante	oui	sous-élagage
Critical value (MINGERS, 1987)	ascendante	oui	sous-élagage
Error-based (QUINLAN, 1993)	ascendante	non	sous-élagage, bonne précision

I.3.5. Prédiction finale

Une fois l'estimateur construit, l'affectation d'un nouvel individu à une classe peut être aisément réalisée en deux étapes, dont la seconde connaît quelques variantes.

Lors de la première étape, l'individu est introduit à la racine† de l'arbre et circule en direction de ses feuilles† en suivant le chemin correspondant aux résultats des tests se présentant sur son passage, en fonction de ses propres caractéristiques.

La seconde étape est formée par l'affectation de classe proprement dite. Quand il atteint une feuille†, la règle générale d'affectation est un vote à la majorité simple réalisé sur les individus de l'échantillon d'apprentissage† ayant atteint cette même feuille (BREIMAN *et al.*, 1984; QUINLAN, 1986). En cas d'ex aequo, un tirage aléatoire peut être réalisé pour les départager.

D'autres règles, d'utilisation plus marginale, ont été développées. Ainsi, BUTTREY et KARO, 2002 proposent d'utiliser la méthode des plus proches voisins localement sur le sous-échantillon appartenant à la feuille† d'arrivée (algorithme *knnTree*), dans l'espoir de concilier le meilleur des deux estimateurs, d'une part en utilisant l'information résiduelle qui n'aurait pu être extraite à la suite des contraintes appliquées aux différentes partitions et d'autre part en limitant le temps de calcul des plus proches voisins grâce à une diminution drastique de l'effectif sur lequel il est appliqué. L'amélioration des performances en prédiction apportée par cet algorithme est toutefois mitigée, l'erreur de prédiction se situant généralement entre celle des deux prédicteurs isolés.

Certaines méthodes de prédiction proposent d'utiliser non seulement l'information contenue dans les feuilles†, mais également celle des nœuds intermédiaires qui jalonnent le parcours de l'individu à reclasser. Ces méthodes, dites de lissage[58] (CHOU, 1991; BUNTINE, 1992) ou de contraction[59] (HASTIE et PREGIBON, 1990) de l'arbre, utilisent une combinaison linéaire de la distribution observée des classes d'un nœud et de celle de ses parents comme estimation de la distribution de probabilité des populations associée à ce nœud. Ce lissage présente un effet similaire à l'élagage† ; en diminuant le poids des dernières partitions, il limite l'effet du bruit sur l'estimation finale.

[58] en anglais : *smoothing*.
[59] en anglais : *shrinking*.

I.4. POINTS CRITIQUES DES MÉTHODES DE SEGMENTATION

I.4.1. Introduction

Les défis de l'extraction de connaissance à partir de bases de données[60] ne sont pas uniquement liés à la taille de ces dernières, mais également à leur nature. Les données réelles recèlent une série d'écueils que doivent contourner avec succès les algorithmes d'apprentissage pour garantir le maintien de leurs performances en prédiction dans ce contexte.

Le bruit, inévitable dans les grandes collections de données, doit être maîtrisé, ce qui est un des objectifs de la phase d'élagage† (§ I.3.4). Le concept† fondamental peut être incomplètement représenté par les individus présents et/ou par les variables mesurées. Il est également susceptible d'être noyé dans un amas d'attributs† non pertinents, d'où la nécessité de critères de sélection performants (§ I.3.3).

Si ces deux problèmes sont partiellement pris en charge par les principes mêmes de la génération d'arbres de décision, d'autres nécessitent certaines adaptions des algorithmes de base, notamment la présence de données manquantes (§ I.4.2), d'interactions entre attributs (§ I.4.3) et l'instabilité liée à l'échantillonnage (§ I.4.4), à laquelle sont sensibles les techniques de segmentation.

I.4.2. Gestion des données manquantes

De nombreux algorithmes de classement† multivariés gèrent la présence de données manquantes en ignorant les individus correspondants, perdant ainsi l'ensemble de leur information, ce qui a des conséquences marginales sur la précision de l'estimateur à condition de disposer d'un échantillon d'effectif suffisamment important et présentant un taux faible de valeurs manquantes.

Dans le cas contraire, il existe divers mécanismes permettant de récupérer partiellement ces individus, certains d'usage général et d'autres spécifiques aux méthodes de segmentation par arbre.

---

[60] en anglais : *knowledge discovery in database* (KDDB), *data mining*.

On doit cependant distinguer le traitement des valeurs manquantes dans l'échantillon d'apprentissage† de celles de l'échantillon test†.

En ce qui concerne l'étape d'apprentissage, FRIEDMAN, 1977 suggère d'ignorer les individus présentant des valeurs manquantes, ou de les substituer par la valeur moyenne de l'attribut† en question dans le sous-échantillon local. QUINLAN, 1986 teste différentes méthodes de réestimation de la valeur de l'attribut manquant sur base du contexte (classe modale, plus grande vraisemblance, arbre de décision†) et constate qu'aucune ne livre des résultats satisfaisants.

Une stratégie alternative consiste à considérer la valeur manquante comme une valeur supplémentaire potentielle de l'attribut†, avec toutefois un risque non négligeable d'instabilité et de fragmentation de la partition si le nombre de valeurs manquantes reste faible.

Une troisième solution proposée par QUINLAN, 1986 consiste à répartir les individus présentant une valeur manquante entre les différentes valeurs disponibles lors du calcul de la valeur du critère de partition† pour un attribut† donné, proportionnellement à leur répartition de classe. Une fois l'attribut sélectionné, ces individus peuvent soit être éliminés des sous-échantillons résultants, l'information subséquente étant donc perdue, soit être transmis de manière fractionnelle, cette dernière approche offrant de bien meilleures performances finales (QUINLAN, 1989).

Lors de la phase de prédiction, la classification peut également être stoppée au premier nœud comprenant une valeur manquante, utilisant dès lors l'information intermédiaire offerte par ce nœud pour établir l'appartenance de classe. Mais cette méthode offre des performances nettement inférieures aux suivantes.

Un individu atteignant un nœud reposant sur un attribut† pour lequel il présente une valeur manquante peut être propagé sous forme de fractions proportionnelles à la distribution des valeurs de l'attribut à ce nœud, et donc poursuivre plusieurs chemins concurrents. L'attribution finale de la classe est obtenue par un vote à la majorité pondérée des étiquettes des feuilles† atteintes (QUINLAN, 1986). Cette méthode offre des performances intéressantes et se montre assez

robuste quant à la qualité des prédictions qu'elle fournit (QUINLAN, 1989).

L'attribut† manquant peut également être réestimé, avec de bien meilleurs résultats que lors de l'apprentissage, bien que ceux-ci soient variables en fonction des domaines traités. Une variante de cette procédure consiste en la définition de partitions suppléantes[61] (BREIMAN *et al.*, 1984). Lors de la construction de l'arbre, une fois la partition d'un nœud sélectionnée, l'algorithme recherche parmi les autres partitions celles qui la prédisent au mieux, suivant un critère semblable à celui utilisé pour la construction de l'arbre. Ces partitions alternatives sont ensuite ordonnées en fonction de ce critère et stockées. Lors de la recherche de classe d'un nouvel individu, si celui-ci présente une valeur manquante pour la variable de partition, il est propagé en fonction des résultats de la première partition alternative, si cette dernière est manquante, en fonction de la seconde, etc. Si toutes les partitions retenues sont manquantes, l'individu est dirigé vers la branche† majoritaire.

I.4.3. Interactions entre attributs

La présence d'interactions dans le concept† augmente considérablement la difficulté d'apprentissage pour les algorithmes de segmentation classiques. Cette complexité est causée par l'approche gloutonne† retenue dans la phase d'exploration des partitions qui est limitée aux cas unidimensionnels et évalue donc l'effet de chaque attribut† individuellement. Pourtant, la structure hiérarchique des arbres de décision, autorisant les ajustements locaux du modèle, est idéale pour la représentation de ces interactions. Un des premiers algorithmes de segmentation porte d'ailleurs le nom de *Automatic Interaction Detector* (AID, MORGAN et SONQUIST, 1963; SONQUIST et MORGAN, 1964).

On fait donc face à une situation paradoxale dans laquelle un estimateur est parfaitement conçu pour représenter les interactions, mais dont la méthode de génération n'est pas optimale pour la détection de ces mêmes interactions. Non optimalité ne signifie

[61] en anglais : *surrogate splits*.

toutefois pas incapacité totale. La complexité d'apprentissage des interactions est liée à leur nature. Certaines interactions laissent une marque dans l'espace unidimensionnel et peuvent être détectées, tandis que d'autres seront la plupart du temps ignorées (figure 8).

Figure 8. Deux structures d'interactions booléennes avec des complexités d'apprentissage différentes. Le concept A OR B (OU classique, au moins une condition vraie) présente un déséquilibre de classes détectable dans chaque espace unidimensionnel, tandis que l'information marginale apportée par chaque attribut séparé A ou B est nulle pour le concept A XOR B (OU exclusif, une et une seule condition vraie).

Les effets de l'interaction sur la difficulté d'apprentissage d'un concept† (complexité du concept), sont à mettre en relation avec une autre notion, celle de dispersion (ou à l'inverse de concentration) du concept. La concentration d'un concept augmente lorsque les exemples d'une même classe se regroupent de manière compacte dans l'espace des attributs†. La majorité des individus sont alors entourés de membres appartenant à la même classe. A l'inverse, un concept est d'autant plus dispersé que les individus d'une même classe sont éclatés en poches éloignées les unes des autres. Il est évident que plus un concept est concentré, plus il est aisé à synthétiser et donc à apprendre. L'un des effets principaux de l'interaction est d'augmenter la dispersion des concepts, en fragmentant leurs réponses selon plusieurs axes, ce qui a pour effet immédiat d'augmenter leur difficulté d'apprentissage (RENDELL et CHO, 1990; RENDELL et SESHU, 1990).

Comme signalé plus haut, cet effet est fonction de la nature de l'interaction, dont la complexité interne varie selon deux axes (PÉREZ et RENDELL, 1996b):

- le nombre de variables impliquées (ordre de l'interaction),

- le nombre de connexions logiques nécessaires pour la représenter,

chacun participant à l'augmentation de la dispersion du concept† sous-jacent dans l'espace des attributs†.

Même lorsqu'elle est correctement détectée, la présence d'interactions d'ordre élevé présente plusieurs effets néfastes sur les méthodes de partition récursive. Elle entraîne notamment la répétition d'attributs† (un même attribut est testé à plusieurs reprises au cours d'un même chemin), la réplication de sous-arbres identiques (chaque pic du concept† devant être appris individuellement certaines informations sont redondantes dans l'arbre), et enfin la fragmentation des données, qui sont progressivement éclatées en sous-échantillons d'effectifs de plus en plus réduits comme conséquence des deux autres problèmes. Ce qui conduit à l'obtention d'arbres de grandes tailles, instables et de performances médiocres en prédiction (SETIONO et LIU, 1998).

Concernant la détection des interactions, PÉREZ et RENDELL, 1996b proposent de modifier l'espace de recherche des partitions en augmentant la profondeur de cette recherche. Au lieu de tester un attribut† à la fois, l'exploration porte sur des partitions basées sur un enchaînement de plusieurs attributs, d'où le nom de recherche en avant[62] donné à cette méthode. Chaque partition potentielle est ainsi combinée à toutes les partitions subséquentes possibles sur deux ou plusieurs niveaux et c'est la qualité prédictive de l'ensemble qui est évaluée. Cette recherche entraîne toutefois un coût en temps de calcul rapidement prohibitif qui limite la détection aux interactions de deux facteurs au maximum.

NAZAR et BRAMER, 1998 et BREMNER et TAPLIN, 2002 recherchent quant à eux de nouveaux critères de sélection adaptés à la présence d'interactions, mais la qualité de leurs résultats est établie sur une gamme très étroite de problèmes.

De plus, ces approches concernant la détection des interactions ne résolvent pas les problèmes de structure liés à la répétition, la réplication et la fragmentation.

---

[62] en anglais : *lookahead search*.

La solution la plus évidente à ces problèmes consiste en un changement de représentation du concept† par construction de nouveaux attributs synthétiques[63] au départ des variables brutes, de manière à éliminer les interactions. Cette construction peut être directement basée sur les résultats du processus d'induction[64] en cours et donc employer l'arbre généré comme outil de recherche de nouveaux attributs (algorithmes *FRINGE*, PAGALLO et HAUSSLER, 1990, *CITRE*, MATHEUS et RENDELL, 1989 et *LFC*, RAGAVAN et RENDELL, 1993). Elle peut également utiliser lors de son exploration des méthodes indépendantes telles que les réseaux neuronaux (SETIONO et LIU, 1998), les algorithmes génétiques (FREITAS, 2001), des projections multidimensionnelles (algorithme *MRP*, PÉREZ et RENDELL, 1995; 1996a; PÉREZ, VILALTA et RENDELL, 1996) ou d'autres algorithmes de recherche gloutons† (algorithme *XofN*, ZHENG, 2000). Les premières méthodes souffrent toutefois des problèmes de détection des interactions des algorithmes de segmentation par arbres et sont donc moins efficaces que celles basées sur des procédures inductives indépendantes, qui quant à elles possèdent un coût additionnel en puissance de calcul non négligeable (NAZAR et BRAMER, 1998).

I.4.4. Stabilité des résultats

Comme d'autres méthodes d'extraction d'informations à partir d'un échantillon de données, telles que les réseaux neuronaux et la régression pas à pas, la segmentation par arbres de décision est malheureusement un estimateur instable.

Un estimateur est dit *instable* lorsque qu'un changement de faible amplitude dans les données de base peut entraîner des modifications importantes de la prédiction finale (BREIMAN, 1996b). Cette instabilité se marque par des transformations de la structure de l'arbre (ajout, suppression, inversion de nœuds), qui peuvent conduire à des bouleversements dans les attributions finales de classe (RUEY-HSIA, 2001).

L'origine de cette instabilité est liée à la procédure de sélection des partitions. Pour rappel, lors de chaque étape, l'ensemble des partitions

[63] en anglais : *feature construction*.
[64] en anglais : *constructive induction*.

disponibles sont évaluées via un critère reflétant leur valeur informative respective (§ I.3.2 et I.3.3), calculé sur base des données du nœud à diviser. La partition présentant la valeur la plus élevée de ce critère est retenue et la procédure se poursuit pour chacun des sous-ensembles obtenus. Si pour ce nœud plusieurs partitions ont une valeur de sélection proche du maximum, une modification légère des données peut suffire à renverser leur ordre de préférence, et donc favoriser une partition alternative.

Cette instabilité est favorisée par certains critères de partition† (§ I.3.3.4), mais surtout par des effectifs faibles. Chaque individu y exerce un poids relatif important dans l'évaluation du critère, de plus ces échantillons sont sensibles aux variations aléatoires. C'est pourquoi l'instabilité se marque surtout, mais pas exclusivement, dans les niveaux inférieurs des arbres de décision. L'élagage† peut y apporter une solution partielle… ou augmenter encore la variabilité des solutions retenues, selon que l'instabilité est liée à la présence de bruit ou au manque d'effectif pour confirmer une partition informative : la structure de l'arbre ayant été altérée, la séquence d'élagage est également modifiée et peut conduire, même au départ de structures valides logiquement équivalentes, à des estimateurs complètement différents (RUEY-HSIA, 2001) (Figure 9).

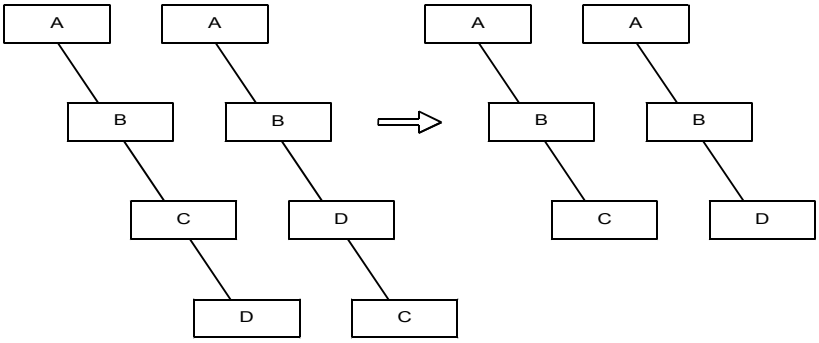


Figure 9. Effet de l'élagage sur deux chemins hiérarchiques logiquement équivalents mais structurellement distincts, livrant alors deux structures logiques différentes (d'après RUEY-HSIA, 2001**).**

Plusieurs voies ont été explorées pour résoudre ce problème. La première consiste en une modification de la procédure de sélection des partitions par construction de nouveaux attributs† composites. Après

avoir évalué les partitions disponibles, une étape additionnelle est insérée afin de sélectionner les alternatives les plus proches de la meilleure partition, la proximité étant mesurée par la quantité de données devant être modifiées pour échanger deux partitions. Un ensemble de nouvelles partitions sont créées en combinant les partitions appartenant à ce peloton de tête au moyen d'opérateurs logiques et leur pertinence est à leur tour jugée par le même critère. La meilleure partition est sélectionnée parmi l'ensemble des partitions examinées, originelles et construites (RUEY-HSIA, 2001). Cette procédure livre des arbres stables, compacts et présentant une précision en prédiction souvent meilleure que les algorithmes de base, mais au prix d'un coût très important en temps de calcul lié à la recherche des groupes de partitions de qualité équivalente et de leurs combinaisons.

Une autre piste de recherche prometteuse est formée par les procédures d'agrégation d'arbres de décision. La réponse à l'instabilité fournie par ces méthodes consiste à générer un faisceau de plusieurs arbres et à utiliser une procédure capable de synthétiser leurs résultats en une prédiction unique. Les divers algorithmes appartenant à cette catégorie se distinguent d'une part par la façon dont ils génèrent des ensembles d'arbres distincts, et d'autre part par leur procédure d'agrégation des prédictions individuelles.

Les techniques d'agrégation sur *bootstrap*[65] ou *bagging*†, développées par BREIMAN, 1996a, marquent une étape importante dans ce domaine, mais des travaux antérieurs abordent des problèmes similaires, liés à la synthèse des résultats d'arbres multiples (SHLIEN, 1990; BUNTINE, 1992; SHLIEN, 1992).

Le principe du *bagging*† est simple : pour construire un estimateur synthétique représentatif d'une population ξ, une solution consiste à utiliser k échantillons indépendants E_1, E_2, …, E_k issus de cette population pour entraîner k estimateurs $\{\varphi(\mathbf{x}, E_i)\}$. La prédiction globale de cette séquence d'estimateur est alors obtenue par l'espérance mathématique de la variable cible, estimée par la moyenne des prédictions observées dans le cas d'une variable cible numérique

[65] en anglais : *bootstrap aggregating*.

(régression), ou par vote à la majorité simple pour les problèmes de classification.

Cependant, dans la réalité, on ne dispose pas de la population ξ mais uniquement d'un échantillon E d'effectif n qui en est issu. L'instabilité augmentant avec la diminution de l'effectif, il n'est pas judicieux de diviser ce dernier en k sous-ensembles indépendants. Le processus d'échantillonnage initial peut toutefois être imité par une succession d'échantillons *bootstrap* de E, c'est à dire par la répétition de k tirages aléatoires avec remise d'effectifs égaux à n dans l'échantillon d'apprentissage†. Ces échantillons perturbés servent de base à la construction de k estimateurs dont les résultats sont synthétisés de la manière décrite ci-avant, d'où le nom de ***b**ootstrap **agg**regat**ing*** ou *bagging*† donné à ces méthodes.

En outre, l'utilisation du *bootstrap* autorise une nouvelle estimation de l'erreur réelle de prédiction. Celle-ci ne nécessite plus l'utilisation ni d'un échantillon test†, ni de la validation croisée, mais recourt aux individus non repris dans le tirage avec remise. Cette estimation[66] (OOB†) est à la fois non biaisée et plus précise que les précédentes (WOLPERT et MACREADY, 1999).

C'est également dans cette ligne de recherche que se place l'algorithme de construction de forêts aléatoires† (*Random Forests*) de BREIMAN, 2001 (§ I.5.4).

Les techniques dites de *boosting* (FREUND et SCHAPIRE, 1999) ou *arcing*[67] (BREIMAN, 1998) forment un autre ensemble dédié à l'agrégation de classificateurs multiples. Ces algorithmes introduisent une phase itérative d'apprentissage. La génération des différents arbres n'est donc plus exécutée en parallèle, mais selon un mode séquentiel.

Une pondération est appliquée aux individus de l'échantillon, tous les individus présentant au départ un poids identique. Cette pondération est soit directement utilisée au sein de l'algorithme de génération du classificateur si celui-ci le permet, soit sert de base à un rééchantillonnage du jeu d'apprentissage. A chaque itération, un arbre

[66] en anglais: *out-of-bag error rate*.
[67] ***a**daptively **r**esample and **c**ombine*.

est généré et le poids des individus mal classés est augmenté de manière à forcer l'algorithme d'apprentissage à se concentrer sur ses erreurs. Cette procédure est répétée un nombre prédéterminé d'itérations et la prédiction finale est obtenue par un vote à la majorité établi sur l'ensemble de la séquence d'estimateurs générés, pondéré par une fonction de l'erreur apparente de chacun.

Les différents algorithmes appartenant à cette catégorie (*Adaboost*, FREUND et SCHAPIRE, 1999, *LogitBoost*, FRIEDMAN, HASTIE et TIBSHIRANI, 2000, *arc-x4*, BREIMAN, 1998) se distinguent essentiellement par leur technique de rééchantillonnage, liée notamment à des fonctions de pondération distinctes. Ils partagent par contre les mêmes propriétés générales, à savoir une réduction rapide non seulement de l'erreur apparente, mais également une amélioration des performances en généralisation au fur et à mesure des itérations. Cependant, certaines déviances peuvent parfois être observées dans le processus d'apprentissage, essentiellement sur des échantillons de faible effectif. Ce comportement épisodique peut être attribué à une sensibilité accrue aux données aberrantes, qui corrompent alors l'essentiel du processus d'apprentissage à leur bénéfice (BREIMAN, 1998; BAUER et KOHAVI, 1999; DIETTERICH, 2000).

D'autres méthodes peuvent être utilisées pour assurer la diversité des classificateurs destinés à être agrégés. HO, 1998 utilise des sous-espaces sélectionnés de manière aléatoire et indépendante parmi les attributs† disponibles pour construire les arbres formant ses forêts de décision, avec à la clé une amélioration significative des performances en prédiction. AMIT et GEMAN, 1997 introduisent une perturbation non pas au niveau de l'échantillon d'apprentissage†, mais directement dans le choix des partitions internes, en instaurant à chaque nœud une présélection aléatoire préalable au choix de la partition optimale. DIETTERICH, 2000 choisit quant à lui d'inverser ces deux opérations, pratiquant une sélection aléatoire[68] parmi les vingt meilleures partitions générées, avec un succès similaire. Quant à BREIMAN, 2000, il prouve sur les plans théorique et empirique que la perturbation aléatoire des étiquettes de classes[69] de l'échantillon d'apprentissage† –

[68] en anglais : *randomization*.

[69] en anglais : *output smearing*, *output flipping*.

l'ajout de bruit – conduit à l'obtention d'un ensemble d'estimateurs dont la synthèse présente également une erreur en généralisation† considérablement réduite.

BAUER et KOHAVI, 1999 et DIETTERICH, 2000 confirment dans leurs études respectives les gains de performances liés à l'utilisation des techniques d'agrégation d'arbres. Les méthodes de *bagging*† permettent essentiellement une réduction de la variance des estimateurs, tandis que l'amélioration apportée par les algorithmes de *boosting* s'étend également à une diminution du biais, ce qui leur confèrent des performances supérieures si on se limite toutefois aux données peu bruitées. Dans le cas contraire, les techniques de *bagging* se montrent les plus robustes et les plus performantes. Dans un cas comme dans l'autre, la randomisation des partitions présente des résultats intéressants, proches de ceux du *bagging*.

L'inconvénient de ces méthodes d'agrégation est la perte de lisibilité du modèle fourni, composé d'un grand nombre d'arbres distincts et donc plus difficile à synthétiser et à soumettre à l'expertise humaine. C'est pourquoi d'autres méthodes ont également été proposées pour synthétiser non plus les résultats mais la structure d'un arbre sous forme d'un « consensus » au départ d'un jeu de classificateurs du même type, notamment par SHANNON et BANKS, 1999 et WANG et ZHANG, 2000, avec des performances toutefois inférieures aux méthodes précédentes en terme de qualité de prédiction.

Enfin, TODOROVSKI et DZEROSKI, 2003 empruntent aux méthodes de segmentation leur structure de prédiction pour combiner les résultats de plusieurs classificateurs de natures différentes (arbres de décision, générateur de règles, plus proches voisins et classificateur Bayésien naïf) sous forme de méta arbres de décision[70]. Les feuilles† de ces arbres ne livrent pas directement une affectation de classe mais l'identité du meilleur algorithme de prédiction à utiliser.

[70] en anglais : *meta decision trees*.

I.5. ALGORITHMES DE GÉNÉRATION D'ARBRES DE DÉCISION

I.5.1. Introduction

Les phases de la construction qui viennent d'être décrites sont implémentées dans une grande variété d'algorithmes, dont l'évolution récente ne doit pas occulter des origines plus anciennes.

Nous retracerons ici quelques grandes étapes de cette évolution (§ I.5.2), pour nous intéresser ensuite plus particulièrement à deux d'entre d'elles, d'une part l'algorithme CART (§ I.5.3), qui constitue une référence incontournable dans le domaine de la prédiction par arbre de décision†, et d'autre part l'algorithme Random Forests (§ I.5.4), dont nous évaluerons le comportement en prédiction dans la partie expérimentale, qui forme un des derniers raffinements des méthodes d'agrégation d'arbres.

I.5.2. Bref historique

Bien que la généralisation de l'utilisation des méthodes de segmentation par arbres soit relativement récente, datant du début des années '90, ses origines sont bien plus anciennes.

On peut considérer l'algorithme AID de MORGAN et SONQUIST, 1963 comme le précurseur de cette famille. L'objectif initial de cet algorithme était la prédiction de valeurs quantitatives par une segmentation pas à pas sur base de prédicteurs de nature numérique ou catégorielle, formant dès lors un arbre de régression. Le choix de la partition est réalisé via une recherche exhaustive au travers de toutes les répartitions en deux groupes envisageables sur base des attributs† disponibles. Le critère de choix est la minimisation de la somme des carrés de écarts de la variable dépendante. Une fois la segmentation effectuée, la procédure est répétée indépendamment sur chacun des deux sous-ensembles créés. Ce mode de construction autorisant de manière naturelle la représentation d'interactions au sein du jeu de données étudiés, le nom de *Automatic Interaction Detector* lui fut donné.

Plus tard, KASS, 1980 proposa une modification de cet algorithme permettant la prise en charge de variables dépendantes de nature

catégorielle, qu'il nomma CHAID, pour *Chi-square AID*. Comme son nom l'indique, son principe de segmentation est basé sur un test Khi carré d'indépendance servant à juger de la pertinence d'une partition via la p-valeur qui peut en être extraite. Deux phases utilisant ce test se succèdent alors, la première cherchant à regrouper les catégories d'un même prédicteur en groupes homogènes, la seconde jugeant de la qualité de ces attributs† condensés en vue de la prédiction de la variable cible. La procédure se poursuit ainsi récursivement jusqu'à ce qu'aucun attribut n'offre de partition statistiquement significative. Cet algorithme et ses modifications furent les premiers à connaître une certaine diffusion dans les domaines appliqués, notamment en marketing où ses partitions multivaluées conviennent particulièrement aux problèmes de segmentation des marchés.

Mais le véritable point de départ du succès de ces méthodes furent les travaux de BREIMAN *et al.*, 1984 et QUINLAN, 1986. Les algorithmes CART (dichotomique) et ID3 (multivalué) résultant de ces travaux posèrent les bases théoriques et appliquées de tout un nouveau domaine, devenant des références pour la plupart des études subséquentes sur les arbres de décision (plus exactement, c'est l'algorithme C4.5, QUINLAN, 1993, le successeur d'ID3, qui joua ce rôle dans la branche des algorithmes multivalués).

I.5.3. Méthode CART

L'algorithme CART (*Classification And Regression Tree*), décrit par BREIMAN *et al.*, 1984, est basé sur la recherche de partitions dichotomiques univariées au départ des attributs descripteurs† fournis. Chaque nœud comporte donc un test logique simple, basé sur un attribut unique, qui conduit à deux branches† correspondant aux exemples positifs (vrai) ou négatifs (faux) en fonction des résultats de ce test. Il peut être utilisé aussi bien pour des variables cibles numériques (régression) que qualitatives (classement†).

La recherche des partitions candidates s'effectue en scindant le groupe d'exemples appartenant au nœud en deux sous-ensembles en fonction de leurs valeurs d'attributs†. Pour les attributs numériques ou ordinaux, cette division prend la forme d'un seuil fixé au sein de l'échelle de valeur de ces attributs entre deux modalités (valeur ou

niveau) distinctes. Pour m modalités, l'espace de recherche est donc limité à m-1 possibilités. L'adaptation de cette procédure aux attributs purement qualitatifs est plus délicate ; le coût de recherche de la partition optimale devient rapidement prohibitif, la séparation de m modalités en deux groupes pouvant conduire dans l'absolu à 2^{m-1}-1 partitions candidates. Certaines méthodes de recherche permettent toutefois de réduire drastiquement ce nombre. BREIMAN *et al.*, 1984 démontrent ainsi dans le cas des problèmes à deux classes cibles† qu'en ordonnant les modalités par probabilités d'appartenance croissantes à l'une des classes, la partition optimale intervient nécessairement entre deux modalités successives, ne laissant plus que m-1 alternatives à tester. CHOU, 1991, MOLA et SICILIANO, 1997 et SHIH, 2001 prolongent ces résultats en proposant des solutions adaptées aux problèmes à trois classes ou plus, ainsi qu'à des familles de critères plus larges que celles utilisées à l'origine par la méthode CART.

Chaque partition candidate à la division d'un nœud est alors évaluée par l'intermédiaire d'une valeur différentielle d'impureté† (§ I.3.3.1) utilisant la somme des carrés des écarts (régression), l'indice de Gini ou le critère d'entropie (classement†). La partition optimale retenue est celle présentant une réduction maximale de l'impureté entre le nœud père et ses deux fils. En outre, lors de cette phase, une ou plusieurs partitions suppléantes (§ I.4.2) sont également mémorisées de manière à pallier l'éventuelle absence de certaines informations lors de la phase de prédiction.

Cette procédure est répétée de manière récursive pour chaque nouveau nœud créé jusqu'à l'obtention de nœuds purs ou d'effectifs inférieurs à une limite préétablie, construisant ainsi un arbre dit maximal.

Cet arbre maximal est ensuite élagué par la méthode du coût-complexité minimal (§ I.3.4.2.a) de manière à fournir l'arbre de prédiction final, utilisant pour ce faire soit un échantillon séparé, soit les techniques de validation croisée lors de l'estimation des erreurs de prédiction et de la valeur optimale du critère de coût-complexité nécessaire à cette phase.

Enfin, l'attribution des étiquettes de prédiction aux différentes feuilles† de l'arbre s'effectue par calcul de la moyenne ou par vote à la

majorité simple sur base des individus de l'échantillon d'apprentissage† appartenant à chacune des feuilles, en fonction de la nature numérique ou catégorielle de la variable cible. L'estimateur peut également fournir des probabilités d'appartenance aux différentes classes cibles†, estimées par les fréquences relatives des exemples correspondants.

La prédiction de la valeur (numérique ou qualitative) attachée à un nouvel individu s'effectue très simplement, en propageant cet individu dans l'arbre ainsi construit selon les résultats des tests, éventuellement suppléants, réalisés sur ses valeurs d'attributs† et en lui associant la prédiction fournie par l'étiquette de la feuille† à laquelle il aboutit.

Afin d'illustrer les résultats produits par cet algorithme, l'exemple de QUINLAN, 1986 décrit au paragraphe I.3.2.1 est ici utilisé pour construire un arbre CART (Figure 10).

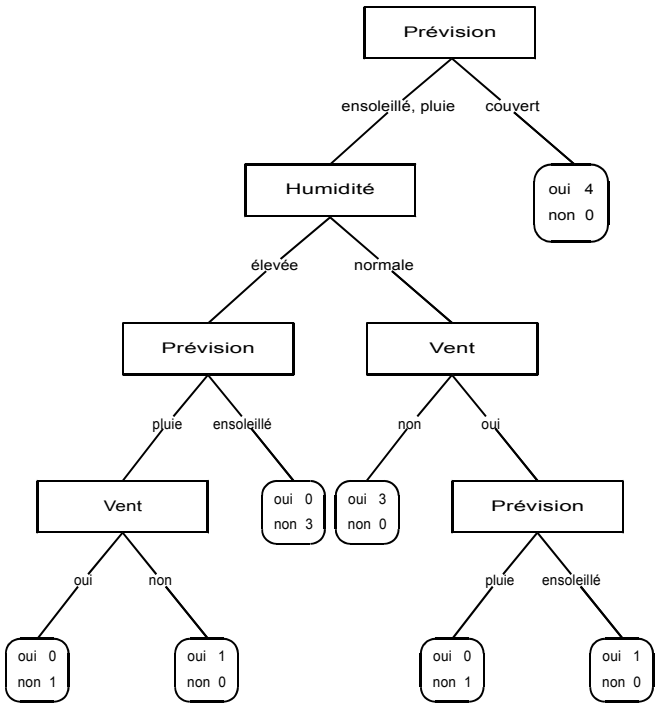


Figure 10. Arbre de décision† CART basé sur l'exemple de QUINLAN, 1986.

La partition dichotomique présente à la fois des avantages et inconvénients par rapport aux partitions multivaluées telles qu'utilisées par l'algorithme ID3 (§ I.3.2). Elle limite la fragmentation de

l'échantillon lorsque plusieurs modalités d'un même attribut† présentent des profils de classement similaires, ces derniers étant alors regroupés dans une même branche† de la partition, mais dans le cas contraire elle nécessite plusieurs tests pour les séparer deux à deux.

Bien qu'il soit connu pour livrer des résultats parfois instables, l'algorithme CART est toujours largement diffusé et sert de référence dans la plupart des études consacrées à l'amélioration des techniques de classement† par arbres de décision.

I.5.4. Random Forests

L'algorithme *Random Forest*s - *Random Input* (*Forest-RI*, BREIMAN, 2001) est l'un des derniers aboutissements de la recherche consacrée à l'agrégation d'arbres randomisés (§ I.4.4). Synthétisant les approches développées respectivement par BREIMAN, 1996a et AMIT et GEMAN, 1997, il génère un jeu d'arbres doublement perturbés au moyen d'une randomisation opérée à la fois au niveau de l'échantillon d'apprentissage† et des partitions internes.

Chaque arbre du jeu est ainsi généré au départ d'un sous-échantillon *bootstrap* du jeu d'apprentissage complet, de manière similaire aux techniques de *bagging*†. Ensuite, l'arbre est construit en utilisant la méthodologie CART, à la différence près qu'à chaque nœud la sélection de la meilleure partition, basée sur l'indice de Gini (§ I.3.3.2), s'effectue non pas sur le set complet de M attributs† mais sur un sous-ensemble sélectionné aléatoirement au sein de celui-ci. La taille F de cette sélection est fixée préalablement à l'exécution de la procédure ($1 \leq F \leq M$). L'arbre est ainsi développé jusqu'à sa taille maximale, sans élagage†. Lors de la phase de prédiction, l'individu à classer est propagé dans chaque arbre de la forêt et étiqueté en fonction des règles CART. La prédiction globale de la forêt est fournie par un vote à la majorité simple des attributions de classe des arbres individuels. Notons que les techniques de *bagging* constituent dès lors un cas particulier de l'algorithme *Forest-RI* pour lequel F = M.

Cet algorithme appartient à la famille plus large des forêts aléatoires†, définie comme suit par BREIMAN, 2001.

« Une forêt aléatoire est un classificateur consistant en une collection de prédicteurs structurés en arbres $[T(\mathbf{x}, \Theta_k), k = 1, \ldots]$ où les $[\Theta_k]$ sont des vecteurs aléatoires de distributions identiques et où chaque arbre fournit un vote unitaire pour la classe la plus populaire pour chaque entrée **x**. »

Le principal avantage de cette structure est qu'elle permet d'éviter le danger que représente le sur-apprentissage[71] pour toute méthode de prédiction basée sur l'induction. BREIMAN, 2001 démontre que lorsque le nombre d'arbres impliqués dans la forêt de prédiction augmente, le taux d'erreur en généralisation† converge vers une valeur limite, dont une borne supérieure peut être estimée sur base des caractéristiques intrinsèques de la forêt.

Si on pose la fonction marginale d'une forêt aléatoire† $T(\mathbf{X}, \Theta)$ suivante

$$mr(\mathbf{X}, Y) = \mathrm{P}_\Theta\left(T(\mathbf{X}, \Theta) = Y\right) - \max_{j \neq Y} \mathrm{P}_\Theta\left(T(\mathbf{X}, \Theta) = j\right),$$

qui représente le degré de confiance du classement† établi par les arbres de cette forêt sur la population (**X**, Y), mesuré par la différence de probabilité entre la prédiction de la classe correcte Y et la meilleure classe erronée $j \neq Y$, on peut définir la valeur de prédiction[72] d'un jeu d'arbres {T(**x**, Θ)} par l'espérance mathématique de cette fonction

$$s = E_{\mathbf{X},Y}\left[mr(\mathbf{X}, Y)\right].$$

La dépendance entre arbres d'une forêt ρ(Θ, Θ') est quant à elle mesurée par la corrélation entre leur fonction marginale brute, évaluée pour des valeurs de paramètres Θ, Θ' fixées et distinctes.

Moyennant ces définitions, une limite supérieure à l'erreur en généralisation† (TEG) de toute forêt aléatoire† est donnée par la relation

$$TEG \leq \bar{\rho}\left(1 - s^2\right) / s^2 \ .$$

---

[71] en anglais : *overfitting*.

[72] en anglais : *strength*.

Cette estimation, bien qu'assez lâche en pratique, permet d'identifier deux éléments clés conditionnant la qualité de prédiction des forêts aléatoires†. Il apparaît sans surprise que le TEG d'une forêt est d'autant plus bas que la valeur de prédiction des arbres qui la constituent est grande. Mais de plus cette valeur varie proportionnellement au degré de dépendance qu'ils entretiennent entre eux. La clé de l'amélioration des forêts de prédiction consiste donc à produire un jeu d'arbres peu corrélés, tout en préservant autant que possible leur qualité individuelle.

Cet objectif est atteint dans l'algorithme *Forest-RI* grâce à la double randomisation. Les techniques de *bagging*† ayant déjà prouvé leur efficacité (BREIMAN, 1996a; BAUER et KOHAVI, 1999; DIETTERICH, 2000), la perturbation supplémentaire opérée au niveau du choix des partitions optimales à l'intérieur même de l'arbre a pour but de réduire la corrélation entre deux arbres d'une même forêt tout en maintenant autant que possible leur valeur de prédiction individuelle. Les procédures de sélection sont maintenues sur le sous-ensemble d'attributs† échantillonnés à chaque nœud et chaque arbre est développé jusqu'à sa taille maximale.

Les résultats de ces considérations théoriques ont été vérifiés empiriquement par BREIMAN, 2001 sur une série de 20 jeux de données (16 réels et 4 artificiels). Il apparaît en outre dans cette expérience que le nombre F d'attributs† présélectionnés aléatoirement a peu d'influence sur le TEG final. Lorsque F croît, la valeur de prédiction individuelle des arbres se stabilise rapidement après une légère augmentation, tandis que leur corrélation continue de croître lentement. Le TEG atteint donc rapidement un minimum avant de se dégrader très progressivement, tout en restant dans une fourchette de valeurs très étroite. Plus surprenant encore, les forêts construites sur ces données avec une valeur $F = 1$, correspondant à une sélection complètement aléatoire d'un attribut à chaque nœud, présentent un TEG à peine supérieur aux autres (moins de 1%), ce qui signifie que même une forêt composée d'arbres construits "au hasard" conserverait une excellente valeur prédictive.

BREIMAN, 2001 teste également l'effet de l'addition de bruit sur les prédictions en généralisation de *Forest-RI* sur 9 jeux de données réels.

Contrairement aux techniques de *boosting*, les performances de la forêt aléatoire† ne sont que peu dégradées par l'adjonction de 5% de bruit aléatoire, ce qui confirme les résultats déjà obtenus avec les techniques de *bagging*† simple (DIETTERICH, 2000).

En outre, son comportement en présence de concepts† très dilués (présence de nombreux attributs† peu informatifs individuellement) semble prometteur, affichant un résultat proche du taux d'erreur bayésien, bien que cette propriété n'ait été vérifiée que sur un exemple artificiel unique, toujours par BREIMAN, 2001. Ce dernier constat est toutefois particulièrement intéressant et laisse entrevoir une possibilité d'utilisation fructueuse de cet algorithme dans les cas de concepts à forte structure d'interaction, caractérisés par une dilution semblable.

Parallèlement à ses excellents résultats en prédiction, la structure de l'algorithme *Forest-RI* lui permet de livrer des renseignements complémentaires concernant l'estimateur qu'il construit. L'utilisation d'échantillons *bootstrap* autorise notamment le calcul du taux d'erreur *out-of-bag* (OOB†) (§ I.4.4), qui fournit une estimation non biaisée du taux d'erreur en généralisation† sans avoir recours à un échantillon test† supplémentaire. Deux autres informations peuvent également être calculées sur demande, une mesure de l'importance des différents attributs† dans le prédicteur final, et une mesure de proximité des individus classés.

L'importance des variables dans le processus de classement† est une notion difficile à définir, celle-ci pouvant être liée à des interactions complexes dans la structure du concept†. Ce paramètre peut être estimé par quatre mesures distinctes intégrées dans l'algorithme *Forest-RI*.

La première méthode consiste à calculer l'augmentation du taux d'erreur OOB† lorsque les modalités de la variable étudiée sont permutées aléatoirement sur les données OOB, les autres variables restant inchangées. Les deuxième et troisième méthodes sont toutes deux basées sur les fonctions marginales des arbres générés. La deuxième mesurent la réduction de la valeur marginale brute liée à la permutation des modalités, tandis que la troisième établit la différence entre le nombre de valeurs marginales brutes qui ont diminué et celles qui ont augmenté. La quatrième mesure ne requiert pas d'altération du jeu de données et est évaluée par la décroissance moyenne du critère de

Gini dans la forêt directement liée à l'utilisation de la variable en question. Cette dernière évaluation est plus rapide à obtenir mais moins fiable que les précédentes. Ces estimations permettent une meilleure compréhension du concept† recherché et peuvent conduire à des présélections de variables, et donc à une réduction dimensionnelle et une simplification des problèmes traités.

La procédure fournit également une matrice de proximité des individus. La proximité entre deux individus est calculée par la fraction d'arbres générés dans lesquels ces derniers aboutissent dans une même feuille†, postulant que deux individus proches devraient suivre un cheminement identique dans l'arbre. Cette mesure peut être utile lors d'une recherche de structure au sein du jeu de données, et ouvre la voie à l'utilisation des forêts aléatoires† en classification non supervisée.

Enfin, la dernière version de l'algorithme (v4.0, BREIMAN, 2003) intègre une gestion des données manquantes à la fois en phase d'apprentissage et de prédiction par remplacement des informations manquantes au moyen de deux procédures distinctes, au choix de l'utilisateur. La plus rapide utilise la médiane ou le mode de la colonne concernée, respectivement pour des variables de nature numérique et catégorielle, tandis que la seconde implique un processus itératif utilisant les proximités entre individus pour établir sa valeur de réestimation.

I.5.5. Exemple d'application – données Soybean

Afin d'illustrer les deux algorithmes décrits ci-avant, nous allons les appliquer successivement sur un jeu de données traitant d'un problème de diagnostic phytosanitaire sur le Soja (*Glycine max* L., *Fabaceae*, *Papilionoideae*).

I.5.5.1. Matériel

Ces données sont rassemblées dans une base de données intitulée Soybean, contenant un ensemble de 35 caractéristiques phénologiques observées sur des plants de Soja et une variable cible codant la maladie diagnostiquée (19 classes). Ce jeu de données est composé de 683 observations, dont 562 complètes. Ces données proviennent du dépôt en ligne de l'*UCI Repository Of Machine Learning Databases* (BLAKE et

MERZ, 1998) et ont été choisies pour leur typicité dans le cadre des problèmes de classement† et leur caractère agronomique.

Les analyses et graphiques ont été réalisés dans l'environnement de programmation statistique R version 1.9.1 sous Windows XP Professionnel SP2. Les procédures correspondant aux implémentations des méthodes *CART* et *Forest-RI* dans l'environnement R sont respectivement les fonctions `rpart` et `RandomForest`.

Un certain nombre d'observations du jeu de données original étant fortement incomplètes, seules les observations ne présentant pas de donnés manquantes ont été utilisées ici afin de simplifier la présentation et la comparaison des méthodes introduites. L'effectif total du jeu de données utilisé s'élève donc à 562 individus.

I.5.5.2. Méthode

Pour garantir une estimation non biaisée des résultats de classement† obtenus par chacune des méthodes présentées, le jeu de données initial a préalablement été scindé en deux parties par un tirage aléatoire sans remise des individus. Un jeu de données indépendant de 150 observations a donc été réservé pour l'estimation de l'erreur de classement† (échantillon test†), les 412 observations restantes étant utilisées pour la construction de l'estimateur (échantillon d'apprentissage†).

Les deux méthodes ont alors été appliquées sur le même jeu d'apprentissage avec leurs paramètres par défaut (fonctions `rpart` et `RandomForest`). Sur base des estimateurs ainsi obtenus, une prédiction des classes de pathologie a été effectuée sur le jeu de test, prédiction confrontée dans une table à la classe réelle extraite de ce même jeu de données.

La procédure d'estimation décrite dans le paragraphe précédent a ensuite été répétée 20 fois, avec une sélection d'échantillons d'apprentissage† et de test indépendante pour chaque répétition. Cette fois, seuls les taux d'erreurs observés sur l'échantillon test† ont été enregistrés et servent de base à une analyse comparative de la performance en prédiction des deux algorithmes réalisée par un test *t*

d'égalité de moyennes, échantillons associés par paires et un test *F* d'égalité des variances.

I.5.5.3. Résultats

L'application de la fonction `rpart` sur le jeu d'apprentissage livre un estimateur qui peut être représenté sous la forme d'une suite hiérarchique de règles ou par un graphe arborescent comme celui de la figure 11.

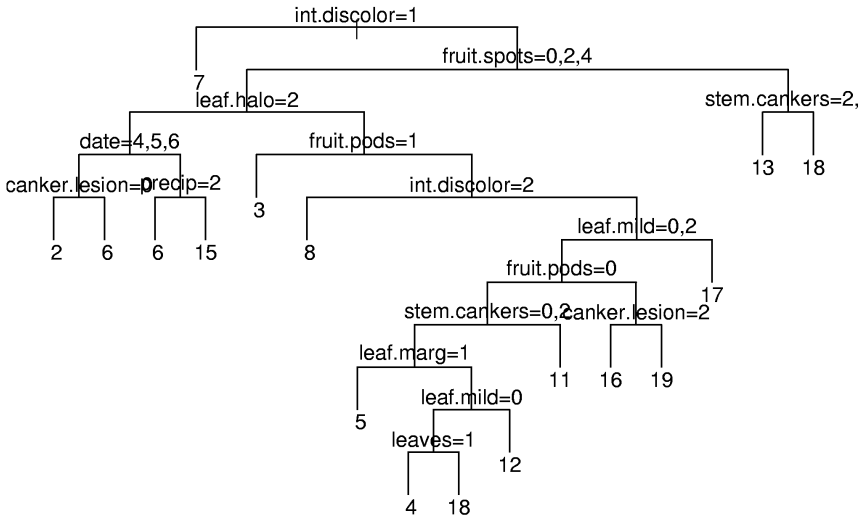


Figure 11. Représentation graphique d'un arbre CART construit par la fonction `rpart` sur le jeu de données Soybean.

L'estimateur ainsi obtenu nous permet de classer les individus du groupe test et de comparer les prédictions avec les pathologies réellement diagnostiquées, telles que rassemblées dans le tableau 3. Le nombre de prédictions correctes pour cet exemple se calcule aisément en sommant la diagonale de ce tableau et est égal à 118, ce qui équivaut à un taux d'erreur estimé de 21,3%.

Tableau 3. Table de comparaison des diagnostics réels et des prédictions correspondantes obtenues par la fonction `rpart`.

Diagnostic réel \ Prédiction (numéro de classe)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
1	0																			0
2		21				2														23
3			7	1																8
4				4		2														6
5					1										1					2
6						20							1		4					25
7						3	14								1					18
8								6												6
9									0											0
10										0										0
11											4									4
12		1		4		1						0								6
13		4				5							15							24
14														0						0
15		2													5					7
16																4				4
17																	7			7
18																		2		2
19																			8	8
	0	28	7	9	1	33	14	6	0	0	4	0	16	0	11	4	7	2	8	

La méthode `RandomForest` délivrant un assemblage de plusieurs arbres de décisions, elle perd en intelligibilité ce qu'elle gagne en précision et se prête peu à une représentation graphique simple.

Néanmoins ses prédictions sont de même nature que celles de la méthode précédente et permettent une confrontation similaire avec les valeurs réelles, dont les résultats sont rassemblés dans le tableau 4. Le nombre de prédictions correctes pour cet exemple est égal à 139, ce qui équivaut à un taux d'erreur estimé de 7,3%.

Tableau 4. Table de comparaison des diagnostics réels et des prédictions correspondantes obtenues par la fonction `RandomForest`.

Diagnostic réel \ Prédiction (numéro de classe)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
1	0																			0
2		22				1														23
3			8																	8
4				6																6
5					2															2
6						25														25
7							18													18
8								6												6
9									0											0
10										0										0
11											4									4
12												6								6
13		6				1							17							24
14														0						0
15		1				2									4					7
16																4				4
17																	7			7
18																		2		2
19																			8	8
	0	29	8	6	2	29	18	6	0	0	4	6	17	0	4	4	7	2	8	

Après avoir répété vingt fois la procédure décrite ci-dessus, nous avons obtenu deux séries appariées de taux d'erreur permettant d'estimer le taux d'erreur moyen et sa variabilité. Les résultats de cette simulation sont résumés à la figure 12.

L'algorithme *Forest-RI* semble donc présenter des résultats à la fois plus fiables et plus stables sur cet exemple, hypothèse confirmée par l'analyse statistique des résultats.

Le taux d'erreur moyen de la méthode *CART* est estimé à 18,0%, avec un écart-type de 3,28%, les paramètres équivalents pour la méthode *Random Forest* étant évalué respectivement à 7,6% et 1,81%. La différence observée est donc égale à 10,4%, avec un intervalle de confiance allant de 9,0% à 11,8% ($t_{19} = 15{,}68$, $P < 0{,}001$). Le rapport des variances estimé est quant à lui égal à 3,29, avec un intervalle de confiance de 1,30 à 8,30 ($F_{19,\ 19} = 3{,}29$, $P = 0{,}013$)

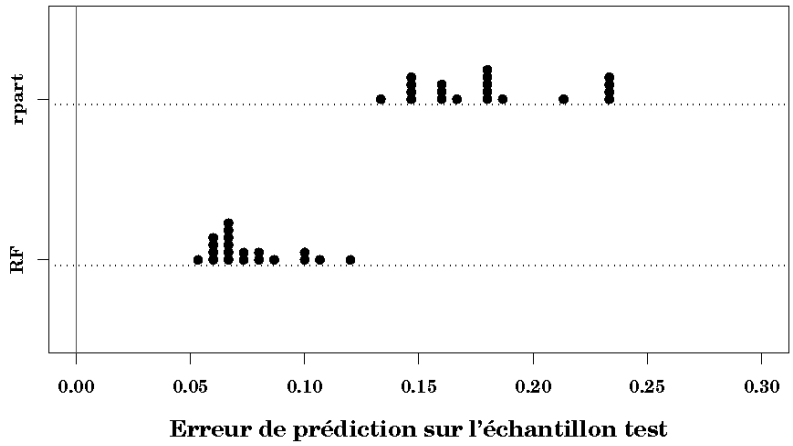


Figure 12. Distribution du taux d'erreur en généralisation observé des estimateurs `rpart` et `RandomForest` (données Soybean, n_{train} = 412, n_{test} = 150, 20 répétitions).

I.5.5.4. Conclusions

L'avantage de la méthode *Forest-RI* sur la méthode standard *CART* apparaît clairement sur cet exemple. Néanmoins, aucune généralisation ne peut être tirée de cette expérience au cadre limité. Une expérimentation explorant l'espace des paramètres tant de la méthode que des données permettrait de délimiter une zone optimale d'utilité de la méthode *Random Forest*, et éventuellement des situations dans lesquelles son utilisation est à éviter.

CHAPITRE II. EXPÉRIMENTATION

II.1. INTRODUCTION

Comme nous l'avons vu au cours du chapitre précédent (§ I.4.3), l'un des principaux problèmes liés à l'apprentissage par arbres de décision reste le traitement des structures d'interactions complexes, pour lesquelles aucune solution universelle n'a encore pu être proposée. Cet état de fait est d'autant plus gênant lorsque l'on envisage l'utilisation de ces méthodes dans le domaine agronomique, dont les phénomènes biologiques et environnementaux sont connus pour être le siège de telles interactions.

L'utilisation des méthodes classiques de génération d'arbres de décision sur des concepts† à forte structure d'interaction entraîne une série de problèmes avérés tels que la réplication de sous-arbres et la répétition d'attributs†. Ces phénomènes sont à l'origine d'une fragmentation des données de l'échantillon d'apprentissage† et conduisent à une instabilité accrue des structures induites, qui aboutit finalement à une réduction de l'efficacité des arbres de décisions et une augmentation de leur erreur en généralisation† (§ I.4.3).

Or, une gamme complète d'algorithmes d'apprentissage, à savoir les forêts aléatoires† telles que définies par BREIMAN, 2001, exploite positivement l'instabilité structurelle des arbres de décision de manière à améliorer drastiquement la qualité des prédictions fournies. De plus, l'algorithme *Forest-RI*, membre de cette famille, a été testé avec succès par ce même chercheur sur un concept† artificiel très dispersé, autre caractéristique liée à la présence de fortes interactions (§ I.5.4).

Néanmoins, l'information dont on dispose sur le comportement de prédiction des forêts aléatoires† reste limitée à un panel restreint de jeux de données principalement issus de l'*UCI Repository of machine learning databases* (BLAKE et MERZ, 1998), certes couramment utilisés dans le domaine de l'apprentissage automatique[73], mais également connus pour être épurés et idéalement formatés pour les algorithmes de cette branche (SEGAL, 2004). De plus, la structure interne et les caractéristiques des concepts† théoriques qui sous-tendent ces exemples demeurent indéfinies. Il est donc délicat de généraliser les conclusions tirées sur base de ces exemples ponctuels.

Comme signalé dans l'introduction générale, la présente étude a pour objectif de combler cette dernière lacune afin d'évaluer les possibilités d'utilisation des méthodes d'apprentissage par forêts aléatoires† sur des concepts† présentant des structures d'interaction marquées, ouvrant dès lors une voie pour la diffusion de leur usage en agronomie. Pour atteindre ce but, une campagne de simulation a été menée, analysant les performances des algorithmes en explorant à la fois la gamme de leurs paramètres internes et la diversité des jeux de données, au travers de caractéristiques telles que le degré de complexité des concepts sous-jacents, le taux de contamination de ces derniers par un bruit de fond et leur dilution dans un ensemble d'informations non pertinentes.

Au cours de ce chapitre, nous décrirons le plan adopté pour conduire cet ensemble de simulations. Nous aborderons la délimitation de l'espace exploré au cours de celle-ci, d'abord au travers des paramètres internes des algorithmes de traitement des données (§ II.2), ensuite des caractéristiques des données elles-mêmes (§ II.3). Après avoir argumenté la sélection du logiciel ayant servi de plate-forme à la simulation (§ II.4), nous terminerons par une présentation générale de l'algorithme de simulation (§ II.5).

II.2. PARAMÈTRES DES ALGORITHMES

Parmi les algorithmes de génération de forêts aléatoires†, l'algorithme *Forest-RI* se démarque par sa double randomisation qui lui

[73] en anglais : *machine learning*.

permet d'améliorer ses qualités de prédiction par rapport aux techniques plus simples telles que le *bagging*† (§ I.5.4). De plus, la variabilité introduite au niveau de la sélection des partitions internes constitue un atout dans l'exploration des chemins de décision multiples créés par les structures d'interaction. Enfin, comme déjà mentionné, il a montré une excellente capacité de prédiction sur un exemple artificiel basé sur un concept† présentant des caractères communs avec les structures d'interaction complexes (BREIMAN, 2001). C'est donc cette forme de forêt de prédiction, distribuée sous le nom générique de *Random Forests* (BREIMAN, 2002; 2003) qui retiendra notre attention au cours de cette expérience.

Cette implémentation présente l'avantage de ne dépendre que d'un nombre très limité de paramètres pour s'exécuter, ce qui rend leur exploration aisée. Les deux arguments principaux de la méthode sont le nombre d'attributs† présélectionnés aléatoirement lors de chaque partition interne (§ II.2.1) et le nombre d'arbres formant la forêt globale de prédiction (§ II.2.2). En outre, une méthode de classement† alternative a été désignée pour servir de référence lors de la comparaison des performances en prédiction de la méthode *Random Forests* (§ II.2.3), ce qui porte le nombre total de méthodes testées à onze (2 présélections d'attributs x 5 nombres d'arbres + 1 méthode CART).

II.2.1. Nombre d'attributs présélectionnés

La présélection aléatoire d'un sous-ensemble de F attributs† du jeu d'apprentissage global à chaque nouvelle partition au sein des arbres formant la forêt constitue un des caractères distinctifs principaux de l'algorithme *Random Forests*. C'est cette procédure qui assure, avec le tirage d'un échantillon *bootstrap* distinct pour chaque arbre généré, le maintien d'une corrélation faible entre ces derniers et conduit à la réduction de variance observée sur les prédictions en généralisation de l'estimateur agrégé.

La méthode affiche toutefois une sensibilité peu marquée à ce paramètre, le taux d'erreur en généralisation† (TEG) atteignant rapidement un plateau lorsque la valeur de F augmente. L'amélioration

du TEG observée par BREIMAN, 2001 entre la valeur $F = 1$ et le plateau reste elle-même étonnamment faible.

Cette dernière constatation peut toutefois s'expliquer par la nature des données utilisées dans l'étude de BREIMAN, 2001. Les jeux d'apprentissage employés, issus principalement de l'*UCI Repository of machine learning databases* (BLAKE et MERZ, 1998), sont formés de concepts† décrits dans leur intégralité par les attributs† fournis, lesquels ont été soigneusement sélectionnés pour leur pertinence. Dès lors, il semble normal que n'importe lequel d'entre eux, sélectionné au hasard, apporte une contribution significative à la connaissance du concept sous-jacent et induise une amélioration de la prédiction associée.

C'est afin de vérifier cette dernière hypothèse que, malgré la faible sensibilité globale affichée par ce paramètre, nous l'avons inclus dans cette étude. Nous avons sélectionné deux valeurs pour ce dernier, $F = 1$ représentant la sélection purement aléatoire d'attributs†, et $F = \text{int}(\log_2 M + 1)$, tel que recommandé par l'étude originelle de BREIMAN, 2001 (M représentant le nombre total d'attributs du jeu d'apprentissage), codées par RND (F = 1) et BEST ($F = \text{int}(\log_2 M + 1)$) lors de l'analyse et l'interprétation des résultats.

II.2.2. Nombre d'arbres agrégés

Le second paramètre étudié est le nombre d'arbres agrégés pour former l'estimateur final.

BREIMAN, 1996a montre une diminution puis une stabilisation rapide du TEG lors d'une augmentation de cette valeur au cours d'expériences menées sur les techniques de *bagging*†. On peut raisonnablement s'attendre à un comportement similaire des forêts aléatoires† sans pouvoir toutefois préciser un seuil minimal pour cet effectif, étant donné l'inconnue formée par l'introduction de la randomisation des attributs† dans le processus de construction des arbres.

L'augmentation du nombre d'arbres constituant la forêt a également pour effet de faire baisser la variabilité du prédicteur global, qu'il est intéressant de mesurer au travers de la variance du TEG afin de modéliser le gain de stabilité associé.

Cinq valeurs ont donc été fixées pour la taille de la forêt aléatoire†, de manière à couvrir selon une échelle logarithmique les ordres de grandeur jouxtant la valeur 100, utilisée par BREIMAN, 2001, à savoir 10, 50, 100, 500 et 1000 arbres.

II.2.3. Méthode de référence

Afin de fournir une base de comparaison aux performances de l'algorithme *Random Forests*, il convient de tester en parallèle un second algorithme fournissant des prédictions de même nature. L'algorithme *CART* (BREIMAN *et al.*, 1984) a été choisi pour assumer cette tâche, tant pour sa structure interne proche du précédent, qui permet d'appréhender directement le bénéfice lié à l'agrégation et la randomisation des prédicteurs individuels, que pour son caractère de pionnier dans le domaine du classement† par arbres de décision, qui en fait un algorithme universellement reconnu et largement diffusé.

Le choix du critère de partition† associé s'est porté sur l'indice de Gini, également utilisé par l'algorithme *Random Forests*, afin de limiter les éventuelles divergences de résultats liées à la méthode de sélection des partitions. En outre, chaque arbre généré par la méthode *CART* a été élagué par l'intermédiaire d'un critère de coût complexité dont la valeur minimale a été déterminée par validation croisée en 10 sous-ensembles (méthode MCCP, 10-*fold* CV, 0-SE). On suit ainsi le mode opératoire le plus couramment utilisé pour cette méthode, réputé garantir un meilleur taux d'erreur en généralisation† pour les estimateurs générés par cet algorithme (§ I.3.4.2).

II.3. DONNÉES SIMULÉES

Les différentes variantes des algorithmes décrites ci-dessus ont été testées sur des jeux de données artificiels spécialement créés dans ce but. Ces derniers présentent l'avantage par rapport aux données réelles de permettre une connaissance et une maîtrise complètes de leurs caractéristiques sous-jacentes et autorisent par là une étude systématique de celles-ci.

Parmi ces caractéristiques, trois ont plus particulièrement retenu notre attention par l'influence essentielle qu'elles exercent sur les

qualités de prédiction de tout estimateur inductif. Dans les paragraphes suivants, nous définirons une mesure destinée à évaluer le degré de complexité en apprentissage d'un concept†, en relation directe avec la notion d'interaction (§ II.3.1) et, tout en explicitant le choix de la nature binaire des concepts étudiés, nous détaillerons son implication dans leur sélection finale (§ II.3.2). La présence à divers niveaux d'un bruit de fond aléatoire (§ II.3.3) et de variables n'apportant aucune information sur le concept (attributs non pertinents ou parasites[74]†, § II.3.4) a également été simulée afin d'évaluer leurs effets conjugués sur la qualité des estimateurs fournis.

II.3.1. Complexité d'apprentissage

L'étude de l'influence des structures d'interactions sur la qualité de prédiction des forêts aléatoires† nécessite un moyen de caractériser objectivement ces interactions et de pouvoir les classer selon leurs effets potentiels sur l'apprentissage du concept† auquel elles appartiennent, ce qui implique la définition d'une mesure formelle de complexité interne du concept.

PÉREZ et RENDELL, 1996b citent deux facteurs principaux influençant la complexité liée à une structure d'interaction, le nombre de variables impliquées (ordre de l'interaction) et le nombre de connexions logiques nécessaires pour la représenter. Toutefois, ces notions ne traduisent qu'indirectement les effets de l'interaction sur la difficulté de l'apprentissage automatique, qu'une mesure de complexité liée à ce domaine devrait idéalement refléter. Ces effets peuvent par contre être avantageusement reliés à la notion de dispersion d'un concept†, déjà abordée au paragraphe I.4.3 (RENDELL et CHO, 1990; RENDELL et SESHU, 1990).

Etant donné que la plupart des méthodes de classement†, qu'elles soient paramétriques ou non, s'acquittent de cette tâche en cherchant à identifier des régions de l'espace des attributs† présentant une homogénéité de classe (§ I.2 et I.3), on conçoit aisément que la dispersion de ces régions, notion englobant à la fois leur multiplication et leur éloignement, augmente la difficulté d'apprentissage du concept†

[74] en anglais : *irrelevant attributes*.

sous-jacent. En effet, chaque région doit alors faire l'objet d'un apprentissage séparé, multipliant d'autant le travail de l'algorithme.

Pour décrire cette dispersion, RENDELL et SESHU, 1990 ont développé la notion de *déviation* autour d'un point quelconque d'un concept†, définie par la moyenne des différences absolues de classe entre ce point et ses voisins immédiats (le voisinage d'un point étant défini dans l'étude originale par les régions ne différant de ce point que par la valeur d'un seul attribut†). L'espérance mathématique de la déviation ponctuelle sur l'ensemble du concept forme alors la *variation moyenne* de ce concept[75], qui peut s'interpréter comme le taux d'erreur attendu d'un classement† exhaustif par les plus proches voisins.

Pour un concept† booléen à p dimensions, la variation s'exprime aisément par

$$V_p = \frac{1}{p2^p} \sum_{i=1}^{p} \sum_{voisin(x,y,i)} \delta(x,y),$$

où δ(x,y) est égal à 0 si les exemples voisins x et y appartiennent à la même classe, et 1 sinon (PÉREZ et RENDELL, 1996b). Elle s'évalue sur une échelle continue comprise entre 0 et 1, la première correspondant à un concept† « plat », de classe uniforme sur l'ensemble du domaine et d'apprentissage trivial, tandis que la valeur maximale est associé à un concept pour lequel le voisinage de chaque point appartient de manière systématique à une classe différente de ce dernier, qui nécessite donc que chaque point soit évalué de manière individuelle.

Cette mesure prend efficacement en compte la « rugosité » du concept† mais, étant formulée par une valeur moyenne sur l'espace des attributs†, elle n'intègre pas l'accroissement de complexité lié à la dimensionnalité du problème. En effet, pour une même valeur de variation moyenne, un concept s'étendant sur un nombre plus élevé d'attributs voit la taille de son espace interne augmenter de manière exponentielle, ce qui rend son apprentissage d'autant plus ardu. C'est pourquoi nous avons adapté la formule de PÉREZ et RENDELL, 1996b afin de définir notre propre mesure de complexité prenant en compte à

[75] en anglais : *average concept variation*.

la fois la répartition moyenne du concept au sein de son espace et la dimension de ce dernier.

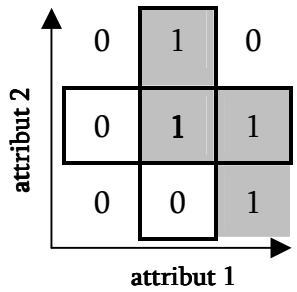


Figure 13. Exemple de calcul de la déviation locale autour d'un point pour une classe binaire et un espace à deux dimensions. Le voisinage du point central, correspondant aux points ne différant du point central que par la valeur d'un seul attribut, est entouré. Sur cet exemple, la somme des différences locales de classe est égale à (0+0+1+1) = 2, sur un maximum de $2^p = 4$. La déviation autour du point est donc égale à 0,5.

Deux corrections ont été testées sur des concepts† de tailles et de variations moyennes diverses. La première consiste à multiplier la variation par le nombre d'attributs† intervenant dans le concept (correction multiplicative linéaire), tandis que la seconde utilise comme facteur multiplicatif l'exponentielle en base 2 de ce nombre, reflétant ainsi l'augmentation proportionnelle de la taille de l'espace des attributs† (correction multiplicative exponentielle). Il est rapidement apparu que le terme exponentiel entraînait une surpondération importante du facteur dimensionnel dans l'évaluation de la complexité des concepts. C'est donc la correction linéaire qui a finalement été retenue.

La mesure de complexité utilisée pour décrire les concepts† binaires simulés dans cette étude s'écrit donc

$$C_p = \frac{1}{2^p} \sum_{i=1}^{p} \sum_{voisin(x,y,i)} \delta(x,y),$$

suivant une notation identique à la formule de variation moyenne précédemment décrite.

II.3.2. Nature des concepts étudiés

La diversité globale des concepts† est essentiellement liée à trois ensembles de caractères, dont les combinaisons délimitent l'espace des concepts possibles. Le premier est formé par la nature et les qualités des attributs descripteurs†, qui peuvent être numériques, ordonnés, qualitatifs, binaires, etc. et présenter des gammes de valeurs illimitées (à l'exception évidente du type binaire). Le deuxième est basé sur des critères identiques concernant cette fois la variable cible, qui offre bien évidemment une diversité similaire. Enfin, le dernier axe englobe les opérateurs de liaison, qui organisent les relations existant entre les attributs et la cible† du concept et cristallisent l'essence même du concept, qu'ils soient arithmétiques, logiques, relationnels, etc.

Dans un souci de simplification et d'efficacité, il est nécessaire de restreindre cet espace hétérogène préalablement à la simulation, sous peine de disperser inutilement les ressources engagées, en veillant toutefois à préserver au maximum les capacités de généralisation des conclusions tirées au sein du sous-ensemble de concepts† sélectionnés.

Dans cette optique, nous avons décidé de limiter nos simulations à des concepts† de nature purement booléenne, tant au niveau des attributs descripteurs† que des variables cibles. Limité à une échelle de valeur à deux niveaux, ce type de variable évite les choix nécessairement arbitraires de plages de valeurs qui accompagnent les autres formes d'attributs†. Les concepts définis sur leur base sont aisés à générer de manière automatisée et peuvent être représentés par une écriture synthétique claire et compacte, tout en préservant une complexité interne potentiellement élevée, cette dernière caractéristique pouvant par ailleurs être évaluée par une procédure simple (§ II.3.1).

Cette restriction imposée quant à la nature des attributs† a peu d'influence sur les processus de construction et d'estimation des algorithmes étudiés. En effet, la structure interne des arbres générés est basée sur une partition dichotomique fondée sur un test logique univarié, nécessitant donc une binarisation préalable des attributs testés, via des techniques standardisées (§ I.5.3). L'utilisation d'attributs binaires court-circuite uniquement cette phase de prétraitement, sans

préjudice pour la comparaison finale des résultats puisque cette phase est commune à toutes les méthodes testées.

La limitation implicite sur la forme générale des concepts† liée à un espace binaire est au premier abord plus sérieuse. Les concepts basés sur des attributs† binaires sont par essence linéairement séparables, qui plus est par des hyperplans parallèles aux axes, générant ainsi un cas idéal pour l'apprentissage par les algorithmes testés, minoritaire dans l'espace général des concepts. Toutefois, les concepts non linéairement séparables ou séparables par des hyperplans non parallèles aux axes sont estimés par les arbres de décision dichotomiques par une succession de tests univariés hiérarchisés utilisant de manière répétitive les mêmes variables, formant ainsi une structure de séparation cloisonnée en hyperrectangles imbriqués approchant la limite réelle de classe (voir par exemple la figure 14). Ce type de comportement peut également être observé en présence d'interactions entre attributs, ce qui permet d'émuler la prise en charge des concepts de forme quelconque par la définition de structures d'interaction binaires appropriées.

La gamme des opérateurs employés découle des choix précédents et emprunte aux domaines arithmétiques et logiques pour leur représentation synthétique, bien que fondamentalement tous puissent être exprimés sous forme disjonctive en recourant uniquement aux opérateurs booléens de base (AIZENSTEIN, 1993).

Moyennant ces considérations générales, la sélection des concepts† particuliers a été intimement liée à la recherche d'une certaine représentativité de ceux-ci le long d'une échelle de complexité.

Nous avons tout d'abord retenu une série d'opérations logiques de base en nous inspirant des travaux de PÉREZ et RENDELL, 1995. Ces opérations ont l'avantage de fournir des concepts† traduisant des formes logiques complexes et variées, le tout sous une écriture synthétique simple et indépendante du nombre d'attributs† booléens choisis. Grâce à cette propriété, pour chaque famille d'opération, nous avons ensuite sélectionné plusieurs tailles d'espace booléen, afin de couvrir à la fois les échelles de variation et de dimensionnalité, toutes deux impliquées dans la définition de la complexité. Nous nous sommes toutefois limités à un nombre maximal de 10 attributs pertinents, délimitant un espace

de 1024 exemples distincts, suivant des contraintes de temps de calcul liées à l'introduction subséquente des variables parasites† (§ II.3.4).

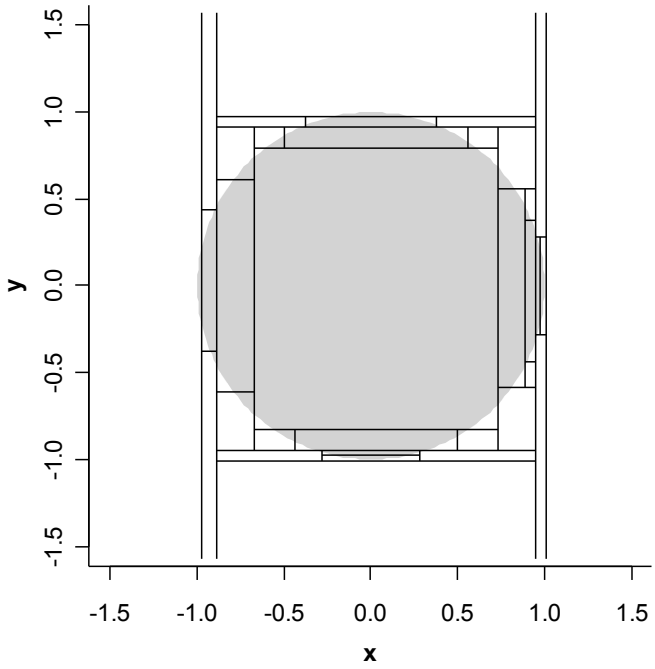


Figure 14. Exemple d'approximation d'un concept non linéaire par un arbre dichotomique. Le concept est défini par l'appartenance au cercle de rayon unitaire (en gris), basé sur la connaissance des coordonnées euclidiennes x et y ; les zones rectangulaires illustrent les sous-espaces construits par la procédure de partition dichotomique de l'arbre pour apprendre ce concept.

Chaque concept† est donc défini par une opération logique et/ou arithmétique portant sur les valeurs de ses attributs descripteurs† booléens. Le résultat de cette opération livre directement la classe à laquelle appartient l'individu décrit. La variable de classe étant également de nature booléenne, seuls deux résultats sont possibles, VRAI ou FAUX. Un nom informel a été attribué à chaque famille d'opération de manière à faciliter leur identification lors des discussions futures. Les familles de concepts booléens retenues sont les suivantes, en adoptant la convention de notation VRAI = 1 et FAUX = 0 et où $x_1, \ldots, x_p$ représentent les attributs descripteurs† booléens :

- **Parité**

La cible† est VRAI si le nombre d'attributs descripteurs† VRAI est impair, et FAUX sinon.

Ce concept† correspond au cas extrême décrit lors de la définition de la variation, pour lequel chaque point du concept est entouré de voisins de classe opposée à la sienne. La variation moyenne est donc égale à 1 et la complexité des concepts générés au nombre de variables pertinentes *p*.

$$impair(x_1, x_2, \ldots, x_p) = \left(\sum_{i=1}^{p} x_i \right) \bmod 2$$

En prenant comme exemple un concept† à quatre dimensions, l'individu (0, 0, 1, 0) présente un seul attribut VRAI, nombre impair, et appartient donc à la classe VRAI (reste = 1), tandis que l'individu (1, 0, 1, 0) en compte deux, nombre pair et fait donc partie de la classe FAUX (reste = 0).

- **Majorité**

La cible† est VRAI si le nombre d'attributs descripteurs† VRAI est strictement supérieur au nombre d'attributs FAUX, et FAUX sinon.

$$majorité\ (x_1, x_2, \ldots, x_p) = \begin{cases} 1 & \text{si } \sum_{i=1}^{p} x_i > p/2 \\ 0 & \text{sinon} \end{cases}$$

Ainsi, l'individu (0, 1, 1, 1, 0) compte une majorité d'attributs VRAI (trois sur cinq), ce qui le classe dans la cible VRAI, tandis que l'exemple (1, 0, 1, 0, 0) n'en compte que deux sur cinq, rejoignant donc la classe FAUX.

- **Multiplexeur**

En fonction des combinaisons de valeurs des deux premiers attributs†, la cible† correspond soit à une conjonction des attributs suivants (symbolisée par $\wedge$, VRAI si toutes les arguments sont VRAI), soit à une disjonction (symbolisée par $\vee$, VRAI si au moins un des arguments est VRAI) soit à une disjonction exclusive (symbolisée par

XOR, VRAI si un et un seul argument est VRAI) ou encore à la négation de cette dernière (symbolisée par !, VRAI si l'argument est FAUX).

$$multiplex(x_1, x_2, \ldots, x_p) = \begin{cases} \wedge(x_3, \ldots, x_p) & \text{si } x_1 = 0 \text{ et } x_2 = 0 \\ \vee(x_3, \ldots, x_p) & \text{si } x_1 = 0 \text{ et } x_2 = 1 \\ XOR(x_3, \ldots, x_p) & \text{si } x_1 = 1 \text{ et } x_2 = 0 \\ !XOR(x_3, \ldots, x_p) & \text{si } x_1 = 1 \text{ et } x_2 = 1 \end{cases}$$

Par exemple, l'individu (0, 1, 1, 0) correspond à une disjonction sur ses deux derniers attributs ($x_1 = 0$ et $x_2 = 1$), opération dont le résultat est VRAI puisqu'au moins un de ceux-ci est VRAI. L'individu (1, 1, 1, 1) entraîne quant à lui l'évaluation d'une disjonction exclusive sur ses deux derniers attributs ($x_1 = 1$ et $x_2 = 0$), dont le résultat est FAUX, plus d'un de ceux-ci présentant une valeur VRAI.

Ces trois premiers concepts† garantissent en outre une distribution équilibrée de la variable cible entre les exemples positifs et négatifs.

- **Déséquilibre**

La cible† est VRAI si le nombre d'attributs† VRAI dans la première moitié de la liste des attributs est strictement supérieur au nombre d'attributs VRAI dans la seconde, FAUX sinon.

$$deséquilibre(x_1, x_2, \ldots, x_p) = \begin{cases} 1 & \text{si } \sum_{i=1}^{p/2} x_i > \sum_{i=p/2+1}^{p} x_i \\ 0 & \text{sinon} \end{cases}$$

L'individu (1, 1, 1, 0) appartient à la classe VRAI, le nombre d'attributs VRAI dans la première moitié de sa description étant strictement supérieur à celui de la seconde (deux contre un), tandis que les exemples (0, 1, 1, 0) et (0, 0, 1, 0) sont classés dans la cible FAUX (égalité à un contre un et infériorité à zéro contre un).

- **Equilibre**

La cible† est VRAI si le nombre d'attributs† VRAI dans la première moitié de la liste des attributs est égale au nombre d'attributs VRAI dans la seconde, FAUX sinon.

$$\textit{équilibre}(x_1, x_2, \ldots, x_p) = \begin{cases} 1 & \text{si } \sum_{i=1}^{p/2} x_i = \sum_{i=p/2+1}^{p} x_i \\ 0 & \text{sinon} \end{cases}$$

Si on reprend les mêmes exemples que la famille précédente, les individus (1, 1, 1, 0) et (0, 0, 1, 0) appartiennent tous deux cette fois à la classe FAUX, tandis que l'exemple (0, 1, 1, 0) est classé dans la cible VRAI.

- **Compteur**

La cible† est VRAI si le nombre d'attributs descripteurs† VRAI est compris entre deux valeurs i et j (i ≤ j), et FAUX sinon.

$$\textit{compteur}(x_1, x_2, \ldots, x_p, i, j) = \begin{cases} 1 & \text{si } i \le \sum_{i=1}^{p} x_i \le j \\ 0 & \text{sinon} \end{cases}$$

Concernant la fixation des paramètres *i* et *j*, une exploration systématique de ces derniers a montré que, pour un nombre *p* de variables, la variation associée était maximale lorsque *i* et *j* sont confondus et compris entre *p*/2 – 1 et p/2 + 1, à l'exclusion des bornes de cet intervalle. Les valeurs de ces paramètres ont donc été fixées suivant cette règle.

Les exemples (0, 0, 1, 0) et (0, 1, 1, 0) sont respectivement classés selon cette règle dans les cibles FAUX (nombre d'attributs VRAI égal à un) et VRAI (nombre d'attributs VRAI égal à deux), le paramètre $i = j$ appartenant à]1, 3[, et donc étant égal à 2.

- **Conjonction**

La cible† est VRAI si tous les attributs† de la première moitié de la liste des attributs sont VRAI et qu'au moins un attribut de la seconde est FAUX, FAUX sinon.

$$\textit{conj}(x_1, x_2, \ldots, x_p) = \wedge(x_1, \ldots, x_{p/2}) \wedge !\wedge(x_{p/2+1}, \ldots, x_p)$$

Les exemples (0, 0, 1, 0), (0, 1, 1, 0), (1, 1, 1, 1), etc. appartiennent tous à la classe FAUX, tandis que seuls les individus (1, 1, 1, 0), (1, 1, 0, 1) et (1, 1, 0, 0) font partie de la classe VRAI.

Ce dernier concept† a été ajouté afin d'observer l'effet associé à une distribution fortement déséquilibrée de la variable cible, les exemples positifs étant nettement plus rares que leurs opposés.

Pour chaque famille, plusieurs concepts† ont été générés en faisant varier le nombre de attributs sur lesquels ils sont basés, de façon à couvrir de manière optimale à la fois la gamme des valeurs de variation et de nombre d'attributs. Les vingt-trois concepts retenus et leurs principales caractéristiques sont repris en détails dans le tableau 5.

Tableau 5. Caractéristiques générales des concepts générés (p = nombre d'attributs, C_p = complexité, V_p = variation moyenne, taille = nombre total d'exemples, - = nombre d'exemples négatifs, + = nombre d'exemples positifs et %+ = proportion d'exemples positifs).

Concept	p	C_p	V_p	Taille	-	+	%+
impair	4	4	1	16	8	8	50,0%
impair	6	6	1	64	32	32	50,0%
impair	8	8	1	256	128	128	50,0%
impair	10	10	1	1024	512	512	50,0%
majorité	5	1,875	0,375	32	16	16	50,0%
majorité	7	2,188	0,313	128	64	64	50,0%
majorité	9	2,461	0,273	512	256	256	50,0%
multiplexeur	4	3	0,75	16	8	8	50,0%
multiplexeur	6	2,5	0,417	64	32	32	50,0%
multiplexeur	8	1,75	0,219	256	128	128	50,0%
déséquilibre	6	1,875	0,313	64	42	22	34,4%
déséquilibre	8	2,188	0,273	256	163	93	36,3%
déséquilibre	10	2,461	0,246	1024	638	386	37,7%
équilibre	6	3,75	0,625	64	44	20	31,3%
équilibre	8	4,375	0,547	256	186	70	27,3%
équilibre	10	4,922	0,492	1024	772	252	24,6%
compteur	4	3	0,75	16	10	6	37,5%
compteur	7	3,828	0,547	128	93	35	27,3%
compteur	10	4,922	0,492	1024	772	252	24,6%
conjonction	4	1	0,25	16	13	3	18,8%
conjonction	6	0,75	0,125	64	57	7	10,9%
conjonction	8	0,5	0,063	256	241	15	5,9%
conjonction	10	0,313	0,031	1024	993	31	3,0%

II.3.3. Bruit de fond

Les données générées jusqu'ici correspondent à des concepts† purs, c'est-à-dire définis sans aucune ambiguïté ni erreur de mesure. Or les jeux de données réels sont souvent entachés de tels imperfections dans la détermination des attributs† ou de la variable cible, *a fortiori* lorsque l'on s'intéresse au domaine biologique, reconnu pour sa variabilité.

En plus de ces jeux de données purs, des échantillons perturbés ont donc été créés afin d'étudier la robustesse des performances en prédiction des forêts aléatoires† face à cette adjonction de bruit de fond dans les jeux d'apprentissage, les rapprochant ainsi des conditions réelles d'utilisation. Nous avons uniquement considéré le cas d'une perturbation aléatoire de la variable cible, étant donné que cette opération menée sur les attributs descripteurs† conduit a un effet strictement identique en modifiant la classe finale théorique du concept† correspondant.

Pour simuler cette erreur, une fraction variable des exemples de chaque échantillon a été sélectionnée au hasard et a vu sa valeur cible théorique remplacée par un tirage aléatoire d'une variable de Bernoulli équiprobable.

En plus du concept† de base, des échantillons correspondant à quatre taux d'erreurs aléatoires moyens ont ainsi été étudiés, à savoir 5, 10, 25 et 50% d'exemples perturbés, disposés selon une progression géométrique approximative. La dernière valeur de cette série a été choisie de manière à ménager des échantillons composés pour moitié d'un concept aléatoire mélangé au concept théorique de base.

II.3.4. Variables parasites

Lors de l'utilisation en routine de méthodes de classement†, il est rare que les attributs† de base du concept† aient été identifiés avec exactitude au préalable. On dispose le plus souvent d'un ensemble de mesures potentiellement pertinentes eu égard à la cible† du classement, sans aucune garantie concernant cette qualité. Le concept recherché est donc plus ou moins noyé dans un amas d'attributs de pertinences variables et la méthode utilisée doit donc être capable de distinguer avec discernement les véritables relations des concordances fortuites sur base des données fournies.

Or, la plupart des études concernant les arbres de décisions et leurs dérivés ont été réalisées au moyen de données réelles ou artificielles soigneusement sélectionnées, ne retenant ou ne générant que les attributs† utiles à la compréhension du concept†. On dispose donc de peu d'informations sur la faculté de discernement de ces algorithmes, malgré certaines expériences annexes prometteuses de BREIMAN, 2001

sur le calcul des importances de variables livrées par l'algorithme *Random Forest* sur certains exemples tirés de l'*UCI Repository of machine learning databases* (BLAKE et MERZ, 1998).

Dans la présente simulation, cette faculté a été éprouvée sur les algorithmes étudiés de manière systématique via l'introduction dans les échantillons d'apprentissage† d'attributs parasites†, sans aucun rapport avec les concepts† sous-jacents. Ces attributs supplémentaires sont construits par l'intermédiaire de tirages aléatoires équiprobables de valeurs booléennes et intégrés aux jeux de données avant leur injection dans les procédures d'apprentissage.

Cinq niveaux de pollution par des variables non pertinentes ont été testés, en proportion avec le nombre d'attributs† de base du concept† (0, 25, 50, 100 et 200% de p, arrondi à l'unité la plus proche), les quatre taux supérieurs à 0% suivant une progression géométrique de raison égale à deux.

II.4. CHOIX DE LA PLATE-FORME LOGICIELLE

II.4.1. Critères de sélection

L'implémentation physique de la simulation nécessite un environnement de programmation adapté aux différentes phases de son exécution que sont la génération des jeux des données, la construction des estimateurs et le stockage des résultats.

Pour faciliter la génération des concepts† et des échantillons associés, le langage doit posséder une structure de données aisée à manipuler, des outils de recombinaison de ces structures (extraction, agglomération) et être doté de générateurs pseudo-aléatoires performants et paramétrables afin d'assurer la reproductibilité des résultats.

La construction des estimateurs suppose en outre l'existence au sein de ce langage de fonctions traduisant les algorithmes *CART* et *Random Forest*, dont l'exécution doit pouvoir être modulée par la définition de paramètres adéquats et/ou par modification de leur code source.

L'importation et l'exportation des données et résultats étant assurée par des fichiers ASCII bruts, dont la gestion est universellement répandue dans les logiciels, ce critère se révèle peu discriminant. Toutefois, l'utilisation de deux machines fonctionnant sous des systèmes d'exploitation différents (Unix Solaris et Microsoft Windows XP Pro) pour la simulation et l'analyse des résultats implique le recours à un logiciel disponible sur ces deux plates-formes.

Enfin, l'environnement de travail doit offrir une interface et un langage de programmation souple et concis, autorisant une mise en œuvre efficace et élégante de la procédure de simulation en elle-même, reliant ainsi les différentes phases dans un code unique, paramétrable et réutilisable.

II.4.2. L'environnement R

Se basant sur ce cahier des charges technique, auquel s'ajoute une certaine préférence philosophique personnelle pour les logiciels issus du domaine libre, le choix final s'est porté sans hésitation sur l'environnement de programmation statistique R (IHAKA et GENTLEMAN, 1996).

Le projet R consiste en une implémentation libre du langage S, développé depuis les années septante dans les laboratoires Bell par John Chambers et son équipe et distribué depuis 1993 sous licence commerciale exclusive par Insightful Corp. Initié dans les années nonante par Robert Gentleman et Ross Ihaka (Université d'Auckland, Nouvelle-Zélande), auxquels sont venus s'ajouter un noyau de chercheurs du monde entier en 1997, il constitue aujourd'hui un langage et un environnement de programmation intégré d'analyse statistique.

L'objectif de ce projet est de fournir un environnement interactif d'analyse de données, doté d'outils graphiques performants et permettant une adaptation aisée aux besoins des utilisateurs, depuis l'exécution de tâches routinières jusqu'au développement d'applications entières.

Le choix s'est donc porté sur une architecture fonctionnelle orientée-objet, structure alliant la facilité d'utilisation à la souplesse et

la puissance de la programmation. La plupart des fonctions disponibles sont en outre directement écrites en langage R, dont le code source est accessible par tout utilisateur au moyen d'un simple éditeur ASCII, ce qui rend leur manipulation et leur personnalisation aisées.

De plus, l'adoption d'une licence libre de type GNU/GPL (*General Public License*) a favorisé son développement et permis son port vers de nombreux systèmes informatiques (Unix, Linux, Macintosh, Windows, etc.). Projet dynamique, R est en constante évolution et bénéficie de fréquentes mises à jour ainsi que d'une très large bibliothèque de fonctions spécialisées (reprenant notamment une implémentation des algorithmes *CART* et *Random Forest*), disponibles gratuitement sur le site du CRAN (*Comprehensive R Archive Network*, http://cran.r-project.org/).

Avant tout développé par et pour des scientifiques, il est aujourd'hui largement diffusé dans la communauté académique et sert de support à de nombreuses recherches et publications.

II.4.3. Environnement matériel

Les simulations ont été réalisées sur une machine Sun Ultra 1 modèle 170 équipée d'un processeur Ultra Sparc à 167 Mhz et de 64 Mb de mémoire centrale tournant sous le système d'exploitation Solaris 2.5.1.

Elles ont été exécutées sous environnement R version 1.7.0, au moyen des librairies `randomForest 3.9-6` (BREIMAN et ADELE, 2003) et `rpart 3.1-12` (THERNEAU et ATKINSON, 1997; 2003), correspondant aux implémentations locales respectives des algorithmes *Random Forest* et *CART*.

II.5. ALGORITHME DE SIMULATION

Le noyau de la simulation informatique à la base de ce travail est formé par un ensemble de scripts et de fonctions R qui automatisent et rendent parfaitement reproductible l'ensemble de l'expérimentation.

L'exécution de l'ensemble des algorithmes d'apprentissage sur toutes les variantes de jeux de données issus d'un concept† défini est sous le contrôle de la fonction `concept.test`, qui accepte en arguments le nom de la fonction génératrice du concept et le nombre d'attributs† de base de ce dernier, ainsi que la liste des valeurs des paramètres des jeux d'apprentissage à tester (`ntrain` = effectif de l'échantillon d'apprentissage†, `irr` = taux de variables non pertinentes et `noise` = niveau de bruit).

Après avoir généré une liste de graines[76] aléatoires, cette fonction générale fait appel à une autre fonction (`rfsimul`) pour la génération des vingt échantillons aléatoires et indépendants nécessaires à la comparaison des onze méthodes testées, ces vingt échantillons partageant les mêmes paramètres `ntrain`, `irr` et `noise`. Chaque appel de la fonction `rfsimul` faisant référence à une graine clairement identifiée, il est aisé de reproduire à l'identique les tirages effectués et les estimateurs qui en découlent. Chaque estimateur est ensuite validé par comparaison avec le concept† de base non perturbé et fournit une matrice de confusion qui est stockée en vue de l'analyse finale de leurs performances respectives.

La conjonction de ces deux fonctions permet de générer l'ensemble des combinaisons des facteurs du plan expérimental décrit ci-avant pour un concept† donné. Le pseudo-code correspondant aux fonctions `concept.test` et `rfsimul` est donné pour information à la figure 15, les différentes étapes de cette procédure étant décrites de manière plus approfondie dans les paragraphes suivants.

[76] point de départ des générateurs pseudo-aléatoires, en anglais : *seed*.

```
Pour chaque effectif d'apprentissage
   Pour chaque taux de variables non pertinentes
       Pour chaque niveau de bruit
              Génération du concept de base
              Pour 20 répétitions
                     Tirage échantillon d'apprentissage pur
                     Ajout des attributs non pertinents
                     Ajout du bruit aléatoire
                     Constitution de l'échantillon test
                     Pour chaque algorithme d'apprentissage
                            Construction de l'estimateur
                            Validation de l'estimateur
                            Stockage matrice de confusion
                     Suivant
              Suivant
              Enregistrement des résultats dans un fichier
       Suivant
   Suivant
Suivant
```

Figure 15: pseudo-code des fonctions `concept.test` et `rfsimul`.

II.5.1. Génération des concepts de base

La génération de chaque concept† de base est réalisée au moyen de fonctions `R` dédiées à chacune des familles de concept décrites au paragraphe II.3.2 et spécialement créées dans ce but. Ces fonctions reçoivent en argument une matrice de variables booléennes représentant les attributs† de ce concept et délivrent en retour une variable booléenne correspondant à la cible† du concept visé pour les individus décrits par la matrice booléenne.

Lors de l'exécution du programme, la fonction génératrice du concept† étudié est exécutée sur une matrice à p variables représentant l'entièreté de l'espace des attributs† pertinents, et fournit donc une image exhaustive du concept qui servira de base au tirage des échantillons d'apprentissage†.

Pour des raisons techniques liées à l'utilisation des fonctions de construction des estimateurs par arbres, les variables booléennes ont été codées en interne par des facteurs à deux modalités, 0 pour FAUX et 1 pour VRAI.

II.5.2. Constitution des échantillons d'apprentissage et de validation

Lors de chacune des vingt répétitions exécutées pour une même combinaison d'effectif d'échantillon, de taux de variables parasites† et de niveau de bruit, un échantillon d'effectif `ntrain` est constitué par tirage aléatoire équiprobable avec remise au sein du concept† de base généré.

En présence d'attributs parasites†, ceux-ci sont créés au moyen de variables booléennes aléatoires équiprobables et ajoutés à l'échantillon d'apprentissage†.

Le nombre *b* d'individus touchés par le bruit aléatoire est ensuite déterminé par tirage d'une variable binomiale de paramètre *n* égal à l'effectif de l'échantillon, avec une probabilité de succès correspondant au niveau de bruit de l'itération en cours. La sélection de ces *b* individus s'effectue ensuite par un tirage aléatoire équiprobable sans remise au sein de l'échantillon d'apprentissage† et les valeurs cibles correspondantes sont remplacées par une réalisation aléatoire d'une variable de Bernoulli équiprobable. Un exemple fictif de résultat est présenté au tableau 6.

Une fois l'échantillon d'apprentissage† formé, un échantillon test† est également constitué, soit en reprenant de manière exhaustive l'espace global des attributs† (pertinents ou non) si la dimension *M* de cet espace est inférieure ou égale à 10, soit par tirage d'un échantillon de 1024 (= 2^{10}) individus si cet espace est de taille supérieure, ceci afin de limiter les temps de calculs nécessaires à la validation des concepts† les plus larges, qui contiennent jusqu'au 2^{30} individus (10 attributs descripteurs + 200% attributs parasites = 30 dimensions).

Tableau 6. Exemple d'échantillon d'apprentissage de 10 individus correspondant au concept *impair* à 4 facteurs, avec un taux de variables parasites égal à 50% (colonnes x5 et x6) et un bruit de fond moyen de 10% (en gras, les valeurs de y remplacées par le bruit équiprobable)

x1	x2	x3	x4	*x5*	*x6*	y
1	1	0	1	*1*	*1*	**1**
1	0	0	0	*0*	*1*	1
0	0	0	0	*1*	*1*	0
0	0	1	1	*1*	*1*	0
0	1	0	0	*0*	*1*	1
1	1	1	0	*1*	*1*	**0**
1	1	0	1	*0*	*0*	1
1	1	0	1	*0*	*0*	1
1	1	1	0	*1*	*1*	1
1	1	1	0	*0*	*0*	1

II.5.3. Génération des estimateurs

L'échantillon d'apprentissage† nouvellement formé va ensuite servir de base à la comparaison des 11 algorithmes testés, en débutant par la méthode des forêts aléatoires†.

Une double boucle est initiée, la première portant sur le nombre d'arbres constituant la forêt (`nTree`), la seconde sur le nombre d'attributs† présélectionnés de manière aléatoire à chaque nœud (`nPres`). Pour chaque combinaison de ces paramètres, un estimateur est construit grâce à la fonction `randomForest` et validé sur l'échantillon test†. La matrice de confusion 2x2 résultant de cette validation est ensuite stockée dans une table, accompagnée du numéro de l'itération portant sur les échantillons (`run`) et des paramètres `nTree` et `nPress` correspondants.

Un estimateur CART est également généré (fonction `rpart`) et élagué ensuite par le critère du coût-complexité minimal, déterminé par validation croisée. Ce dernier est validé et ses résultats sauvegardés

selon la même procédure que les forêts aléatoires†, les facteurs `nTree` et `nPress` étant fixés arbitrairement à zéro pour cet algorithme.

II.5.4. Enregistrement des résultats

Les tables contenant les résultats de validation des 11 algorithmes sur les 20 échantillons correspondant à une même combinaison d'effectif d'apprentissage, de taux d'attributs parasites† et de niveau de bruit sont compilées et sauvegardées dans un fichier ASCII dont le nom est généré automatiquement sur base de ces derniers paramètres et du nom du concept† testé.

Parallèlement, un fichier ASCII nommé `seed.log` est complété par les graines des générateurs aléatoires utilisées pour chaque itération, dans un souci de reproductibilité des expériences.

Une fois la simulation clôturée pour un concept† donné, un script compile les différents fichiers résultats en un fichier ASCII unique, ajoutant au passage les caractéristiques de chaque lot d'échantillon (`ntrain`, `irr` et `noise`) et calculant le taux d'erreur associé à chaque estimateur généré sur base de sa matrice de confusion.

Pour un concept† donné, nous obtenons donc un fichier comportant 13 colonnes (huit pour les descripteurs, quatre pour les éléments de la matrice de confusion, une pour le taux d'erreur en généralisation†) et 27.500 lignes (Tableau 7).

Tableau 7. Description des données de simulation correspondant à un concept particulier.

Facteurs	Effectif	Niveaux
Paramètres des échantillons		
Effectif d'apprentissage (`ntrain`)	5	50, 100, 500, 1000, 5000
Variables parasites (`irr`)	5	0, 25, 50, 100, 200%
Niveau de bruit (`noise`)	5	0, 5, 10, 25, 50%
	125	
Paramètres des algorithmes		
Présélection d'attributs (`nPres`)	2	1, int($\log_2 M + 1$)
Effectif de la forêt (`nTree`)	5	10, 50, 100, 500, 1000
	10	
+ CART	**11**	
Echantillonnage	**20**	
Total par concept	125 x 11 x 20 = **27.500 observations**	

CHAPITRE III. METHODES ANALYTIQUES

III.1. INTRODUCTION

La simulation décrite ci-avant produit un ensemble d'observations correspondant aux taux d'erreur en généralisation† issus de la combinaison exhaustive des concepts†, échantillons et algorithmes de prédiction étudiés, soit un total de 632.500 valeurs observées (23 concepts x 27.500 observations par concept, tableau 7).

Etant donné la nature factorielle de l'expérience ainsi conduite, l'analyse des effets des différents paramètres sur les performances en prédiction des méthodes étudiées peut être décrite par un modèle global d'analyse de la variance reprenant l'ensemble de ces paramètres et leurs interactions. Toutefois, l'interprétation pratique d'un tel modèle unique s'avère extrêmement délicate et complexe, notamment par la présence de termes d'interaction fixes d'ordres élevés. De plus, cette analyse masque l'appartenance des facteurs à trois domaines bien distincts, le premier caractérisant l'algorithme utilisé, le second le concept† à apprendre et le troisième l'échantillon mis à disposition pour atteindre ce but.

Pour ces différentes raisons, l'analyse des résultats de l'expérience a été scindée en deux phases, la première s'intéressant principalement aux paramètres d'exécution des algorithmes des forêts aléatoires†, dans le but d'en extraire la ou les combinaison(s) conduisant aux meilleurs résultats en prédiction (§ III.2). La seconde phase, s'appuyant sur les résultats de la première, étudie les effets des caractéristiques du concept† et de l'échantillon sur les performances du(des) meilleur(s)

algorithme(s) sélectionnés (§ III.3), afin de valider la fiabilité de ces méthodes dans des conditions se rapprochant du domaine agronomiques (effectifs faibles à modérés, présence d'interactions et d'une forte variabilité).

III.2. PARAMÈTRES DES ALGORITHMES RANDOM FORESTS

L'objectif de cette première étape analytique est de déterminer avec précision l'influence du nombre d'arbres constituant la forêt de prédiction et celle du nombre d'attributs† présélectionnés lors de la construction des partitions au sein de chacun de ces arbres sur les performances en prédiction de la méthode *Random Forest*.

Sur cette base, nous rechercherons la ou les combinaisons de paramètres offrant des performances optimales sur le domaine expérimental considéré, ce qui d'une part permettra de formuler des recommandations aux futurs utilisateurs de cette méthode, et servira d'autre part de point de départ à une analyse consacrée cette fois à l'influence des paramètres de l'échantillon.

III.2.1. Structuration des données

Les concepts† étudiés présentant par construction des caractéristiques volontairement distinctes, les données seront scindées selon ce critère préalablement à l'analyse, afin d'éviter l'introduction d'interactions triviales dans le modèle et de permettre une interprétation nuancée des résultats en fonction des connaissances acquises sur ces concepts. L'analyse sera donc répétée de manière indépendante sur chacun des fichiers de données brutes issus de la simulation, correspondant aux différents concepts.

Si le point central de cette première étape reste l'étude des paramètres des algorithmes de prédiction, il convient de garder à l'esprit les éventuelles interactions de ces facteurs avec les caractéristiques de l'échantillon. Néanmoins, intégrer l'ensemble de ces dernières dans l'analyse des résultats nous ramène aux problèmes d'interprétation du modèle posés par une analyse globale déjà exposés dans l'introduction de ce chapitre, une solution alternative doit donc être envisagée.

Parmi les paramètres de l'échantillon, on peut distinguer deux groupes selon le degré de connaissance *a priori* dont peut en disposer un utilisateur en conditions réelles : d'une part les caractères connus, dont fait partie l'effectif de l'échantillon, et d'autre part les paramètres estimés voire totalement inconnus, tels le niveau de bruit, le taux de présence de variables parasites† ou les caractéristiques du concept† sous-jacent. Les premiers peuvent être intégrés avec exactitude dans tout processus de décision relatif aux méthodes de prédiction à appliquer, leur influence doit donc être évaluée avec attention. Quant aux seconds, leur connaissance étant au mieux approximative, il convient essentiellement de vérifier la validité des conclusions concernant les algorithmes tout au long d'une plage de valeurs la plus étendue possible de ces facteurs.

L'effectif de l'échantillon sera donc intégré au modèle d'analyse concernant les paramètres des algorithmes RF, tandis que le taux de présence de variables non pertinentes et le niveau de bruit serviront de base à la décomposition des résultats.

Ces deux derniers facteurs offrant 25 combinaisons factorielles distinctes, seul un sous-ensemble de celles-ci sera toutefois envisagé afin de ne pas alourdir l'interprétation, l'analyse fine de ces paramètres faisant l'objet de la seconde phase analytique. Trois cas ont donc été retenus sur une échelle représentant le niveau général de perturbation de l'échantillon, codés *Low* (`irr` = 0%, `noise` = 0%), *Medium* (`irr` = 50%, `noise` = 10%) et *High* (`irr` = 200%, `noise` = 50%), correspondant aux extrémités et au point central de la diagonale du domaine expérimental de ces deux facteurs (Figure 16).

Au total, ce seront donc 23 concepts† x 3 niveaux de perturbation = 69 jeux de données qui seront analysés séparément.

L'analyse portant sur les paramètres internes des forêts aléatoires†, les résultats concernant l'algorithme *CART* ont été volontairement exclus de cette première phase.

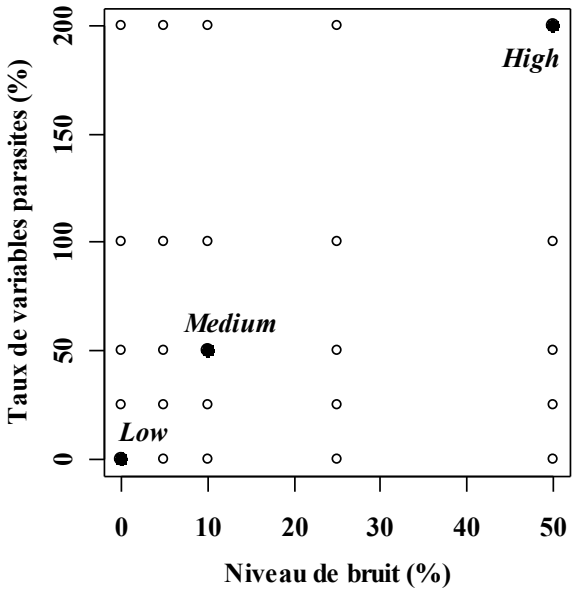


Figure 16. Position des trois cas de figure sélectionnés pour la phase 1 de l'analyse sur le domaine expérimental des facteurs noise et irr.

III.2.2. Analyse graphique exploratoire

Dans le but de faciliter l'interprétation des tests statistiques découlant de l'analyse des données ainsi préparées, des représentations graphiques ont été générées pour chaque concept† en respectant la structure définie au paragraphe précédent, utilisant pour cela les outils graphiques mis à la disposition par le logiciel R, et plus particulièrement la bibliothèque `lattice` consacrée aux graphes conditionnels (SARKAR, 2004).

Chaque graphe est ainsi scindé en quinze panneaux selon les combinaisons des facteurs liés aux échantillons (trois niveaux de perturbation et cinq effectifs d'apprentissage), chaque panneau présentant l'évolution de la moyenne du taux d'erreur en généralisation† en fonction du nombre d'arbres constituant la forêt (`nTree`) pour les deux méthodes de présélection d'attributs† (`nPres` RND et BEST), représentées par des séries de données distinctes, auxquelles s'ajoute une ligne de référence pointillée équivalant au taux

d'erreur de base† du concept† (taux d'erreur lié à une prédiction par vote à la majorité simple sur l'entièreté du domaine du concept).

III.2.3. Analyse de la variance

Le modèle analytique choisi correspondant aux soixante-neuf jeux de données décrits au paragraphe III.2.1 est une analyse de la variance à quatre facteurs, modèle mixte partiellement hiérarchisé avec *a*, *b* et *c* fixe et *d* aléatoire hiérarchisé à *a* (Figure 17).

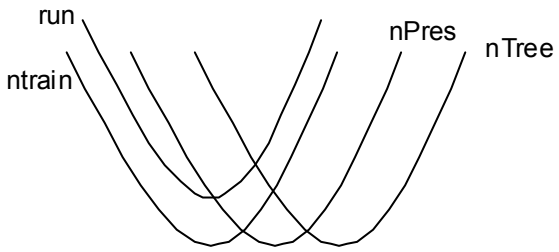


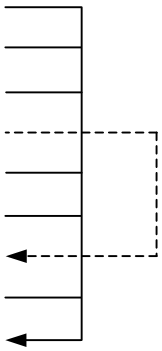
Figure 17. Modèle d'analyse de la variance, phase 1 (paramètres des algorithmes RF); nPres = présélection des attributs, nTree = effectif de la forêt de décision, ntrain = effectif d'apprentissage, run = identifiant de l'échantillon aléatoire d'apprentissage.

Le tableau d'analyse de la variance correspondant est repris ci-dessous, le terme d'erreur résultant de l'agrégation des interactions aléatoires `nPres:run` et `nTree:run` toutes deux représentatives de la variation résiduelle, de par la structure de la simulation.

Ce modèle est appliqué à chacun des sous-ensembles de données issus de la combinaison des concepts† et des niveaux de perturbation. Les éventuelles interactions détectées seront décomposées et interprétées avec l'assistance des représentations graphiques décrites au paragraphe précédent. Une fois les interactions résolues, les effets des facteurs significatifs sont évalués au travers d'une structuration des moyennes par la méthode de Tukey (TUKEY, 1952).

Tableau 8. Tableau d'analyse de la variance, phase 1 (paramètres des algorithmes RF).

Sources de variation	Degrés de liberté
nPres	1
nTree	4
nPres:nTree	4
ntrain	4
nPres:ntrain	4
nTree:ntrain	16
run(ntrain)	95
nPres:nTree:ntrain	16
Erreur	855
Total	999



Etant donné la taille importante des échantillons et le caractère parfaitement contrôlé de l'expérimentation, l'interprétation des résultats est basée sur un risque de première espèce fixé à 0,1%, mieux adapté à ces conditions particulières liées aux simulations informatiques.

L'ensemble de ces analyses ont été réalisées dans l'environnement R, version 2.0.1 (IHAKA et GENTLEMAN, 1996).

III.3. PARAMÈTRES DES ÉCHANTILLONS D'APPRENTISSAGE

L'étude des paramètres des algorithmes *Random Forest* conduit à la sélection de la combinaison de ces derniers offrant les taux d'erreur les plus bas sur le domaine étudié. Le comportement de cette version optimalisée des méthodes *RF* face aux variations des caractéristiques des échantillons d'apprentissage† va à présent être décortiqué, au travers de l'étude des effets de leur effectif, de l'adjonction de bruit et de variables non pertinentes et des caractéristiques du concept† à apprendre (complexité, taille de la cible†).

L'objectif de cette seconde phase de l'analyse est de définir les conditions au sein desquelles le gain de précision que les forêts aléatoires† apportent par rapport aux techniques de segmentation classiques, représentées par l'algorithme *CART* justifie leur utilisation

préférentielle, mais également de rechercher les éventuelles zones d'exclusion correspondant à des performances dégradées par rapport à cette même technique de référence.

III.3.1. Structuration des données

La distinction entre les caractéristiques de l'échantillon selon le degré de connaissance de l'utilisateur qui a été relevée au cours de la première phase est plus que jamais d'actualité et va influencer le traitement analytique réservé à ces paramètres.

Lors de travaux de classement† en conditions réelles, les propriétés exactes du véritable concept† sont évidemment inconnues, mais certaines, comme la taille des classes cibles, peuvent être estimées sur base de l'échantillon d'apprentissage†. La complexité peut quant à elle faire éventuellement l'objet d'hypothèses basées sur des considérations théoriques ou des expériences similaires antérieures. L'influence de ces caractéristiques sur le comportement des algorithmes de classement par arbres doit donc être intégrée dans le processus de sélection de ces algorithmes. Mais ces critères ne peuvent pas être fixés librement *a priori* et donc faire l'objet d'une expérimentation factorielle systématique comme les autres paramètres de la simulation. C'est pourquoi les concepts choisis pour cette étude, formant un panel de cas variés, feront chacun l'objet d'une analyse séparée, mais dont l'interprétation sera appuyée par leurs propriétés respectives.

En l'absence de connaissance du véritable concept† sous-jacent, les niveaux de perturbation de l'échantillon, représentés à la fois par les pollutions par bruit de fond et par adjonction de variables parasites†, ne peuvent également qu'être supposés ou grossièrement estimés. De plus, ces paramètres font partie intégrante du jeu de données et le degré d'efficacité de leur modification éventuelle par certains prétraitements (lissage, sélection de variables) est imprévisible. Malgré leur nature numérique, ces deux facteurs et leur combinaison présentent donc un intérêt d'abord qualitatif, générant une grille de situations dans lesquelles les algorithmes *RF* et *CART* pourront être aisément comparés.

L'effectif d'apprentissage est au contraire une caractéristique sous la dépendance directe de l'utilisateur. Non seulement cette valeur lui est

connue avec précision, mais elle peut être modifiée par l'adjonction ou le retrait d'individus dans le cas où les observations proviennent d'une expérience ou d'un échantillonnage planifiés, ou sont extraites d'une base de données de dimension largement supérieure aux besoins du classement†. Ces individus supplémentaires ont un coût, d'abord informatique, bien que ce dernier soit de plus en plus négligeable à l'heure actuelle, mais également matériel, lié au temps et aux moyens nécessaires pour effectuer les nouvelles observations. Il est donc essentiel de connaître la forme du modèle général liant l'évolution du taux d'erreur en prédiction à l'augmentation de l'effectif d'apprentissage pour mettre en balance les coûts et les bénéfices attendus d'une telle opération.

III.3.2. Analyse graphique exploratoire

En fonction de ces considérations concernant l'intérêt analytique des différents paramètres, une série de représentations graphiques ont été générés sur base des mêmes données, offrant une bonne perspective sur les résultats de la simulation concernant l'influence de l'échantillon. Ces représentation s'appuient sur les mêmes outils graphiques que la phase 1, à savoir les graphes conditionnels et la bibliothèque de fonctions R qui leur est dédiée, `lattice` (SARKAR, 2004).

Ces graphiques ont pour objectif de mettre en évidence la relation existant entre l'effectif d'apprentissage et le taux d'erreur des deux algorithmes testés, conditionnellement aux autres caractéristiques de l'échantillon. Chaque concept† généré fait l'objet d'un graphe conditionnel formé d'une grille de 25 panneaux, les lignes correspondant aux cinq niveaux de bruit (`noise`, 0, 5,10, 25 et 50%) et les colonnes aux cinq taux de présence de variables parasites† (`irr`, 0, 25, 50, 100 et 200%). Chaque panneau présente l'évolution du taux d'erreur moyen observé en fonction de l'effectif d'apprentissage (`ntrain`, selon une échelle logarithmique), et cela pour les algorithmes RF et CART, représentés par des séries de données distinctes. Comme précédemment, une ligne de référence pointillée équivalant au taux d'erreur de base† du concept est également présente dans chaque panneau.

III.3.3. Analyse de la variance

Le modèle d'analyse de la variance initial appliqué aux vingt-trois concepts† étudiés correspond à un modèle mixte à cinq facteurs, quatre fixes croisés et un aléatoire, ce dernier, formé par les échantillons d'apprentissage† (`run`), étant hiérarchisé aux paramètres des échantillons (`ntrain`, `noise` et `irr`) mais croisé avec les algorithmes (`algo`).

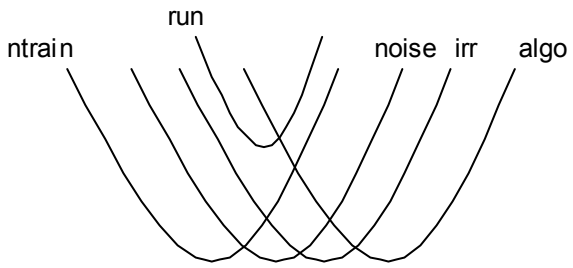


Figure 18. Modèle d'analyse de la variance, phase 2 (paramètres des échantillons); ntrain = effectif d'apprentissage, noise = taux de bruit, irr = taux de variables parasites, algo = algorithme de classement, run = identifiant de l'échantillon aléatoire d'apprentissage.

Le tableau d'analyse de la variance correspondant est repris ci-dessous au tableau 9.

Ce modèle est appliqué à chaque concept† indépendamment. Une fois les interactions décomposées conformément aux résultats de l'analyse de la variance et à la structuration des facteurs définies au paragraphe III.3.1, le comportement des algorithmes *RF* et *CART* sera étudié au travers de la modélisation de la relation liant l'effectif d'apprentissage et le taux d'erreur du prédicteur correspondant.

Pour les mêmes raisons liées à l'origine des données de cette analyse statistique déjà exposées lors de la première phase analytique, le risque alpha utilisé lors de l'interprétation des résultats est fixé à un pour mille. L'ensemble des analyses de cette phase ont également été réalisées dans l'environnement R, version 2.0.1 (IHAKA et GENTLEMAN, 1996).

Tableau 9. Tableau d'analyse de la variance, phase 2 (paramètres des échantillons).

Sources de variation	Degrés de liberté
algo	1
ntrain	4
irr	4
noise	4
algo:ntrain	4
algo:irr	4
algo:noise	4
ntrain:irr	16
ntrain:noise	16
irr:noise	16
algo:ntrain:irr	16
algo:ntrain:noise	16
algo:irr:noise	16
ntrain:irr:noise	64
algo:ntrain:irr:noise	64
run(ntrain irr noise)	2375
algo:run	2375
Total	4999

CHAPITRE IV. INTERPRETATION DES RESULTATS

IV.1. INTRODUCTION

Les résultats des analyses décrites au Chapitre III sont ici exposés et interprétés.

La première étape analytique, portant sur les paramètres des algorithmes *Random Forests*, doit permettre de mettre en évidence d'une part, l'effet du mode de présélection des attributs† lors de la construction des arbres et d'autre part, celui de la taille des forêts aléatoires†, ce qui conduit à sélectionner une combinaison optimale de paramètres, robuste quant à ses performances sur la gamme de situations testées (§ IV.2).

Le comportement de cette forme optimale de l'algorithme *Random Forest* face aux modifications des caractéristiques de l'échantillon est ensuite décortiqué et comparé à celui de l'algorithme *CART*. L'étude conjointe de l'influence de l'effectif d'apprentissage, du niveau de bruit, des variables parasites† et du concept† de base est ainsi réalisée. L'objectif de cette seconde étape est de définir les limites des conditions d'utilisation des méthodes *RF* afin de vérifier leur applicabilité dans les problèmes de type agronomique (§ IV.3).

IV.2. PARAMÈTRES DES ALGORITHMES RANDOM FORESTS

Notre stratégie globale d'interprétation est la suivante : d'abord dégager dans la mesure du possible les tendances communes aux différents concepts† testés (§ IV.2.1 et IV.2.2), pour ensuite s'intéresser

aux exceptions à cette tendance en tentant de les expliquer par les caractéristiques propres des concepts concernés (§ IV.2.3). Nous conclurons enfin en tentant d'extraire de ces résultats une recommandation générale quant aux valeurs de ces paramètres à adopter lors de l'utilisation des forêts aléatoires† (§ IV.2.4).

IV.2.1. Analyse globale et influence de la présélection des attributs

Dans ce but, nous présentons à la figure 19 une représentation graphique du comportement général des algorithmes *Random Forest* suite à la modification de leurs paramètres `nPres` et `nTree`. Cette synthèse est basée sur une moyenne du taux d'erreur observé sur l'ensemble des concepts†, la structure générale du graphique restant fidèle à la description du paragraphe III.2.2. Cette représentation a une valeur essentiellement qualitative et synthétique et a été créée dans le but d'alléger la présentation visuelle des résultats.

Les premières constatations, triviales, concernent les paramètres de l'échantillon et confirment l'influence conjuguée du degré de perturbation de l'échantillon d'apprentissage† et de son effectif sur les performances des estimateurs. Plus un échantillon est perturbé par rapport au concept† qu'il représente, plus son effectif doit être élevé pour garantir un taux d'erreur donné. Ces considérations seront affinées lors de la seconde phase de l'analyse.

Concernant les paramètres des algorithmes, on remarque un avantage non négligeable et quasi systématique de la méthode de présélection multiple d'attributs† par rapport à la présélection unique (facteur `nPres`). Cet avantage apparaît plus nettement encore lorsque l'on s'intéresse aux concepts† individuels, la présélection unique offrant dans de nombreux cas une performance juste équivalente au vote à la majorité simple dans l'échantillon (taux d'erreur de base† du concept, représenté par une ligne horizontale pointillée). De plus, cette méthode reste relativement insensible à l'augmentation de l'effectif de l'échantillon d'apprentissage†, contrairement à la présélection multiple qui profite de cette augmentation pour améliorer ses prédictions.

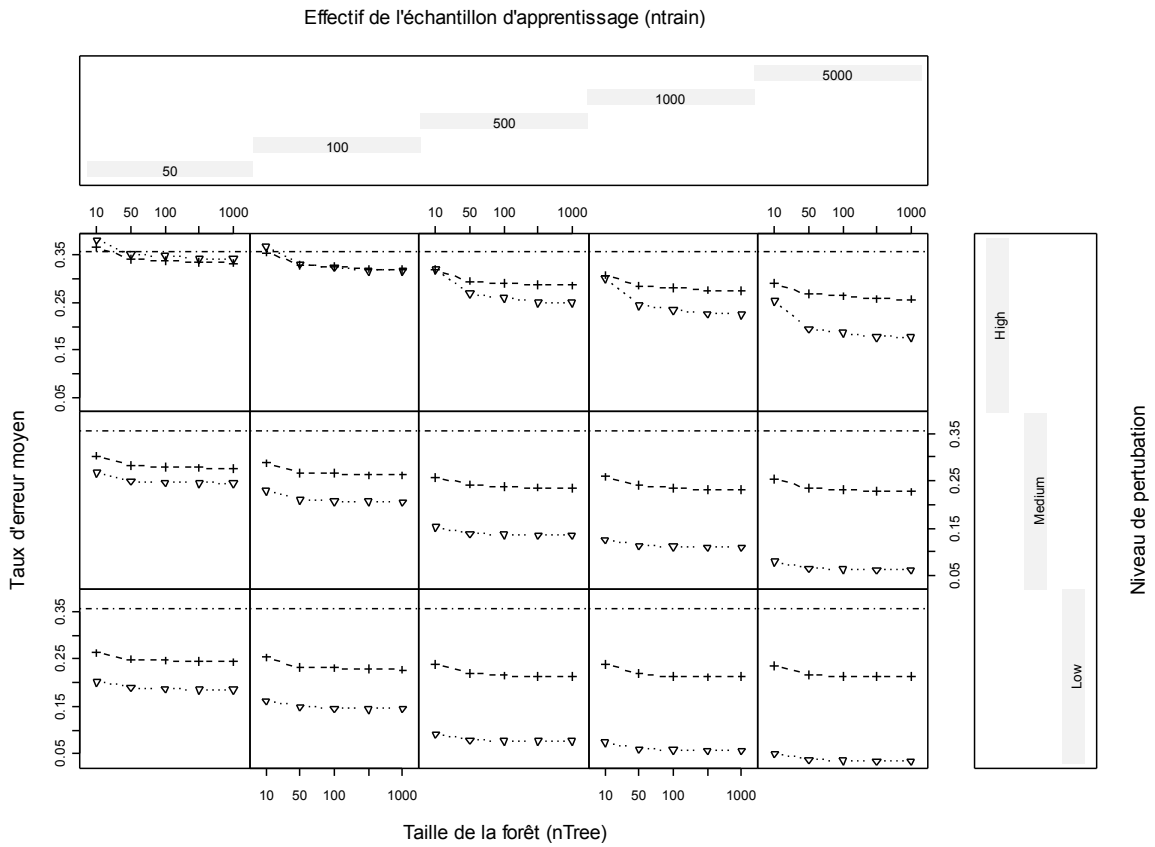


Figure 19. Performances moyennes des algorithmes RF (−+− nPres = RND, ··∇·· nPres = BEST, lignes = niveaux de perturbation, colonnes = effectifs d'apprentissage).

Le nombre d'arbres à la base de la forêt de décision ne paraît quant à lui pas présenter une influence déterminante sur la qualité des estimations, l'amplitude des taux d'erreur observés tout au long de la plage de valeurs de ce facteur restant faible. On remarque toutefois une légère amélioration des performances lorsque l'effectif de la forêt augmente, mais celles-ci atteignent rapidement un plateau. Cependant la représentation graphique de la figure 19 ne permet pas de préciser davantage les limites de ce plateau.

L'analyse de la variance confirme et affine ces observations. Suivant le même souci de synthèse, le tableau 10 reprend les tendances générales de cette analyse, représentées par les valeurs observées des tests de Fisher et les p-valeurs correspondant aux différents termes du modèle d'analyse de la variance décrit au paragraphe III.2.3. Ces valeurs apparaissent sous une forme condensée par le calcul des moyennes et médianes des ces valeurs au travers des différents concepts† simulés.

Les deux interactions présentant les valeurs du test F les plus élevées quel que soit le degré de perturbation de l'échantillon font toutes deux intervenir le mode de présélection des attributs† (`nPres:nTree` et `nPres:ntrain`). Celles-ci sont liées à la différence de sensibilité des deux méthodes à l'évolution des deux paramètres d'effectifs (échantillonnage et forêt) déjà constatée graphiquement, la présélection complètement aléatoire (RND) présentant un taux d'erreur peu variable lors de l'augmentation de ces facteurs, au contraire de la présélection multiple associée à un critère d'évaluation classique (BEST). Ces considérations, conjuguées à l'effet marqué affiché par le facteur `nPres` lui-même, largement en défaveur de la sélection aléatoire, confirme le caractère sous-optimal de cette dernière méthode.

Tableau 10. Synthèse des résultats de l'analyse de la variance sur les 23 concepts testés (phase 1 – paramètres des algorithmes RF).

Pertub.	Sources de variation	Valeurs moyennes		Valeurs médianes	
		F	P-valeur	F	P-valeur
Low	nPres	13066,63	3,58%	6804,72	0,00%
	nTree	57,65	11,28%	24,83	0,00%
	nPres:nTree	31,58	6,90%	16,72	0,00%
	ntrain	514,84	0,70%	105,36	0,00%
	nPres:ntrain	1530,45	2,50%	157,81	0,00%
	nTree:ntrain	4,80	39,34%	1,17	28,80%
	nPres:nTree:ntrain	2,83	38,41%	1,52	8,73%
Medium	nPres	10703,56	0,00%	4641,73	0,00%
	nTree	75,96	7,70%	22,93	0,00%
	nPres:nTree	33,43	6,47%	20,20	0,00%
	ntrain	328,55	0,02%	93,92	0,00%
	nPres:ntrain	1723,30	0,00%	298,43	0,00%
	nTree:ntrain	2,67	25,30%	1,48	9,86%
	nPres:nTree:ntrain	3,49	22,71%	1,68	4,45%
High	nPres	1919,38	0,03%	1341,21	0,00%
	nTree	174,10	4,50%	157,54	0,00%
	nPres:nTree	44,45	20,18%	22,87	0,00%
	ntrain	90,52	5,36%	59,83	0,00%
	nPres:ntrain	315,47	3,37%	75,54	0,00%
	nTree:ntrain	3,62	35,16%	1,32	17,99%
	nPres:nTree:ntrain	5,66	13,09%	3,14	0,00%

IV.2.2. Influence de la taille des forêts aléatoires

Dès lors, la résolution des interactions citées ci-dessus est réalisée par la décomposition de l'analyse selon le facteur `nPres`, la sélection multiple d'attributs† faisant seule l'objet d'une interprétation plus poussée, ce qui conduit à un modèle d'analyse de la variance simplifié portant uniquement sur les facteurs `nTree` et `ntrain` dont les résultats sont synthétisés au tableau 11, construit selon une procédure identique au tableau précédent.

Tableau 11. Synthèse des résultats de l'analyse de la variance sur les 23 concepts testés, restreinte à la préselection multiple (`nPres` > 1, phase 1 – paramètres des algorithmes RF).

Pertub.	Sources de variation	Valeurs moyennes		Valeurs médianes	
		F	P-valeur	F	P-valeur
Low	nTree	63,48	4,28%	38,99	0,00%
	ntrain	689,19	0,00%	309,52	0,00%
	nTree:ntrain	5,35	18,90%	2,31	0,30%
Medium	nTree	107,07	5,41%	85,98	0,00%
	ntrain	552,76	0,00%	347,19	0,00%
	nTree:ntrain	6,02	5,57%	3,90	0,00%
High	nTree	419,75	2,72%	398,39	0,00%
	ntrain	125,93	5,55%	61,42	0,00%
	nTree:ntrain	8,60	4,06%	5,16	0,00%

L'influence mutuelle de ces deux derniers facteurs s'affirme alors plus nettement comme le montre les p-valeurs associées à leur interaction, mais la valeur du test F reste toutefois d'un ordre de grandeur nettement inférieur à celles des facteurs principaux.

La figure 20 permet de mieux interpréter l'effet conjugué des tailles de l'échantillon d'apprentissage† (`ntrain`) et de la forêt de décision (`nTree`) sur les prédictions des estimateurs RF. Ce graphique présente l'évolution pour l'ensemble des concepts† du taux d'erreur moyen en fonction de l'effectif de l'échantillon d'apprentissage, exprimé selon une échelle logarithmique, et cela pour les cinq tailles de forêts générées (représentées par différentes séries de données) et les trois niveaux de perturbation de l'échantillon (un par panneau).

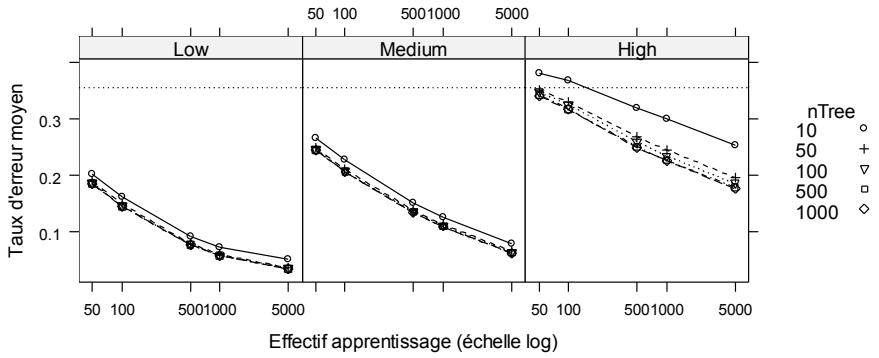


Figure 20. Evolution du taux d'erreur moyen en fonction de l'effectif d'apprentissage (`ntrain`), pour les différentes tailles de forêts aléatoires testées (`nTree`, correspondance des symboles avec les tailles à droite du graphique), sur les 23 concepts simulés (méthode de présélection BEST).

On constate aisément que l'interaction détectée par le test statistique reste en pratique négligeable, l'évolution du taux d'erreur suivant des pentes presque parallèles quel que soit le nombre d'arbres dans les forêts aléatoires† générées. Cette divergence entre les conclusions statistique et pratique est liée à l'extrême sensibilité des tests statistiques dans les conditions de simulation (variabilité faible et degrés de liberté élevés).

Ce graphique confirme l'influence relativement limitée de la taille des forêts de décision sur la qualité de la prédiction finale, influence qui augmente toutefois avec le degré de perturbation de l'échantillon, ce qui n'apparaissait pas clairement sur les représentations précédentes. Ainsi, si aux niveaux de perturbation les plus faibles seules les forêts de 10 arbres semblent se détacher visuellement du lot et présenter des

performances inférieures aux autres, au niveau le plus perturbé seules les forêts de 500 et 1000 arbres sont encore indistinctes.

Cette dernière observation demande à être confirmée par une structuration de moyenne effectuée sur le facteur `nTree` par la méthode de Tukey[77], appliquée avec un risque de première espèce égal à 1‰ à chaque concept† et pour chaque niveau de perturbation, et dont les résultats sont repris dans le tableau 12 au travers des occurrences de différences non significatives entre les niveaux correspondants sur les 69 jeux de données testés.

Si l'on s'intéresse globalement à toutes les situations testées, seules les forêts comportant respectivement 500 et 1000 arbres assurent un taux d'erreur minimal et ce de manière strictement équivalente (100% d'absence de différences significatives) . Les performances se dégradent ensuite avec la réduction de l'effectif de la forêt, les forêts comportant une centaine d'arbres talonnant toutefois de près ces résultats (65 absences de différences significatives avec les forêts de 1000 arbres, soit 94,2% des situations étudiées).

La dépendance du facteur `nTree` par rapport au degré de perturbation de l'échantillon apparaît clairement si l'on décompose les résultats ci-dessus selon ce dernier facteur. Pour les niveaux de perturbation faible et moyen, l'équivalence des performances s'étend à toutes les tailles de forêts, à l'exception toutefois de la plus faible (10 arbres) qui montre des résultats moins bons que la moyenne dans plus de la moitié des cas (taux d'erreur globaux en moyenne 1,5% et 1,8% plus élevés, respectivement pour les perturbations faibles et moyennes). L'éclatement des performances de prédiction intervient essentiellement dans les échantillons les plus perturbés, qui dégradent les résultats des effectifs intermédiaires de 50 et, dans une moindre mesure, de 100 arbres par rapport aux tailles plus importantes (augmentation du taux d'erreur moyen de 0,8%, 1,6% et 6,2% par rapport aux forêts de 500 et 1000 arbres, respectivement pour les forêts de 100, 50 et 10 arbres).

[77] En anglais: *Honestly Significant Difference, Tukey's HSD*.

Tableau 12. Résultats de la structuration des moyennes du facteur nTree par la méthode de Tukey (alpha = 0.001) : occurences des différences non significatives sur les 23 concepts testés pour les trois niveaux de perturbation (69 jeux de données au total).

Tous

nTree	10	50	100	500
50	25			
100	22	67		
500	22	53	67	
1000	23	53	65	69

Low

nTree	10	50	100	500
50	12			
100	10	23		
500	10	23	23	
1000	11	23	23	23

Medium

nTree	10	50	100	500
50	10			
100	9	23		
500	9	20	23	
1000	9	20	23	23

High

nTree	10	50	100	500
50	3			
100	3	21		
500	3	10	21	
1000	3	10	19	23

IV.2.3. Cas particuliers

Certains concepts† se démarquent quelque peu de ce schéma général, appartenant aux trois dernières familles simulées (*équilibre*, *compteur* et *conjonction*) et ce d'autant plus que leur dimensionnalité est élevée. Chez ces concepts, dans certaines conditions, la présélection multiple d'attributs† (BEST) affiche des performances inférieures à la présélection aléatoire (RND), alors que le taux d'erreur de cette dernière ne s'écarte pas du taux d'erreur de base† du concept.

Ce comportement apparaît lorsque trois conditions sont réunies : un échantillon fortement perturbé, un effectif d'apprentissage faible et une forêt de taille peu importante. En outre, les concepts† touchés, bien que qu'appartenant à une large amplitude de degrés de complexité, présentent tous des cibles† positives de taille réduite (min. 3,0%, max. 31,3%, tableau 5) . Cette caractéristique les rend naturellement plus sensibles aux perturbations de nature aléatoire équiprobable introduites par la simulation (§ II.3.3 et II.3.4), et nécessite en outre un effectif d'apprentissage plus élevé que la moyenne afin d'assurer la présence d'un nombre suffisant d'exemples des deux classes cibles†.

Dans cet environnement d'apprentissage extrême, aucune des méthodes testées ne parvient à descendre sous le taux d'erreur de base† du concept† et donc à établir une performance meilleure que le choix systématique de la classe majoritaire du concept. Mais les forêts aléatoires† utilisant la présélection multiple d'attributs† offrent des prédictions de qualité encore inférieure à ce classificateur trivial. Comment expliquer ce comportement ?

Si on se réfère à l'inéquation théorique limitant le taux d'erreur d'une forêt aléatoire† établie par BREIMAN, 2001 (§ I.5.4), on se rappelle que ce dernier dépend à la fois de la valeur prédictive individuelle des arbres de la forêt mais également de leur corrélation.

Dans le cas de la présélection aléatoire d'attributs† (RND), la valeur prédictive de chaque arbre est faible mais peu sensible aux perturbations de l'échantillon puisque leur structure est essentiellement liée au hasard. Cette méthode favorise l'apparition de feuilles† terminales dont la répartition des classes suit grossièrement celle de l'échantillon d'apprentissage†. Les différents arbres de la forêt

étant peu corrélés, du fait de l'indépendance des sélections d'attributs d'un arbre à l'autre, les prédictions globales de la forêt s'accordent avec un vote à la classe majoritaire.

La présélection multiple (BEST) reste quant à elle assortie d'un processus de sélection finale de l'attribut† déterministe, potentiellement influencé par les variations aléatoires de l'échantillon d'apprentissage† liées au bruit de fond ou aux variables parasites†, introduisant des biais locaux dans les prédictions. La valeur prédictive individuelle des arbres diminue tandis que leur corrélation mutuelle augmente, conduisant à un taux d'erreur qui finit par dépasser le taux d'erreur de base† du concept† pur.

Les contre-performances associées à la présélection multiple d'attributs† seraient donc liées à sa sensibilité supérieure aux perturbations, en raison de la subsistance de processus déterministes dans son mode de construction. Pour corroborer cette hypothèse, on remarque que c'est dans les situations particulières évoquées ci-dessus que la taille de la forêt présente la plus grande influence, l'augmentation du nombre d'arbres permettant alors de gommer partiellement les biais locaux par un effet de moyenne.

IV.2.4. Conclusions

Les analyses précédentes montrent un effet très net du mode de présélection des attributs†, la présélection complètement aléatoire offrant des performances le plus souvent médiocres, généralement en retrait de la présélection multiple assortie d'un critère de sélection classique et profitant peu de l'apport d'information accompagnant une augmentation de l'effectif d'apprentissage. Cela va à l'encontre des observations faites par BREIMAN, 2001 et confirme l'hypothèse selon laquelle ces dernières étaient liées aux qualités particulières des jeux de données utilisés, notamment l'absence de variables non pertinentes.

La présélection multiple d'attributs† s'impose donc comme méthode de référence pour les utilisations ultérieures de la méthode, moyennant certaines précautions concernant les caractéristiques de l'échantillon d'apprentissage† qui pourront être précisées lors de la seconde phase de l'analyse des résultats de la simulation qui leur est consacrée.

L'influence de la taille de la forêt de décision est conforme aux attentes et présente une courbe asymptotique similaire à celle déjà observée pour les méthodes de *bagging*†, dont les forêts aléatoires† dérivent (§ II.2.2), du moins lorsque la sélection multiple d'attributs† est utilisée pour construire les arbres. Le plateau est rapidement atteint lorsque les échantillons sont peu ou modérément perturbés (dès 50 arbres), mais lorsque le degré de perturbation atteint des niveaux élevés, au moins 100 voire 500 arbres sont nécessaires pour garantir un taux d'erreur minimal. Etant donné le coût linéaire et donc limité en temps de calcul de ce paramètre, la valeur de 500 arbres assure une certaine robustesse des résultats par rapport aux caractéristiques de l'échantillon d'apprentissage† utilisé.

Finalement, on peut donc retenir la présélection multiple d'attributs†, accompagnée de la génération d'une forêt de 500 arbres, comme étant la combinaison de paramètres assurant un comportement de prédiction optimal des forêts aléatoires† qui en découlent, combinaison qui sera utilisée lors de l'étude des facteurs dépendant de l'échantillon d'apprentissage†.

IV.3. PARAMÈTRES DES ÉCHANTILLONS D'APPRENTISSAGE

Au cours de cette seconde phase, nous interpréterons tout d'abord les effets globaux des paramètres de l'échantillon sur le taux d'erreur en prédiction des classificateurs (§ IV.3.1). Nous préciserons ensuite la nature de ces effets en modélisant la relation unissant l'effectif d'apprentissage et le taux d'erreur en prédiction (§ IV.3.2), avant de nous intéresser à l'influence potentielle des caractéristiques du concept† sous-jacent sur les paramètres de cette relation (§ IV.3.3). Enfin, nous détaillerons les exceptions aux conclusions tirées dans les paragraphes précédents (§ IV.3.4) avant de reprendre synthétiquement l'ensemble des conclusions tirées de ces analyses (§ IV.3.5).

IV.3.1. Analyse globale

Le modèle d'analyse de la variance décrit au paragraphe III.3.3 a été appliqué indépendamment aux vingt-trois concepts† simulés. Les résultats de cette inférence sont synthétisés dans le tableau 13,

construit par le même procédé que les tableaux d'analyse de la variance présentés au cours de l'étude des paramètres des algorithmes *RF*.

Tableau 13. Synthèse des résultats de l'analyse de la variance globale sur les 23 concepts testés (phase 2 – paramètres des échantillons).

Sources de variation	Valeurs moyennes		Valeurs médianes	
	F	P-valeur	F	P-valeur
algo	9733,39	0,52%	6076,70	0,00%
ntrain	5820,99	0,00%	4981,21	0,00%
irr	1747,76	0,00%	130,89	0,00%
noise	735,12	0,00%	732,00	0,00%
algo:ntrain	1062,08	0,00%	342,75	0,00%
algo:irr	518,86	0,00%	57,76	0,00%
ntrain:irr	220,43	0,01%	13,23	0,00%
algo:noise	128,66	0,00%	106,42	0,00%
ntrain:noise	31,97	0,02%	12,79	0,00%
irr:noise	17,91	1,87%	6,19	0,00%
algo:ntrain:irr	225,89	0,20%	15,32	0,00%
algo:ntrain:noise	17,49	3,75%	14,54	0,00%
algo:irr:noise	12,06	0,47%	3,91	0,00%
ntrain:irr:noise	5,40	0,07%	2,89	0,00%
algo:ntrain:irr:noise	8,71	2,21%	2,98	0,00%

L'examen des p-valeurs médianes nous révèle un fort degré d'intrication des effets des différents paramètres. En effet, l'ensemble des interactions, y compris celle faisant intervenir les quatre paramètres simultanément, sont caractérisées par un effet très hautement significatif dans au moins 50% des analyses, tout comme d'ailleurs les effets principaux.

Lorsque l'on s'intéresse cette fois aux valeurs médianes observées des tests F, on constate que les cinq plus grandes valeurs correspondent à des interactions qui font intervenir le facteur algorithme (algo:ntrain, algo:noise, algo:irr, algo:ntrain:irr, algo:ntrain:noise), associé de manières diverses aux facteurs dépendant de l'échantillon. Cela suggère une différence importante de comportement des deux algorithmes testés face aux variations de ces paramètres.

Pour vérifier analytiquement cette hypothèse, le modèle d'analyse global a été décomposé selon le facteur algorithme et réestimé. Les résultats de cette décomposition sont repris dans le tableau 14.

La divergence de sensibilité envers les paramètres de l'échantillon entre les algorithmes *RF* et *CART* apparaît très nettement par un simple examen des p-valeurs moyennes et médianes. Si on se réfère plus particulièrement à ces dernières, les différents paramètres de l'échantillon conservent un fort niveau d'interdépendance chez l'algorithme *RF*, l'ensemble des interactions affichant un niveau de signification très élevé, tandis seule l'interaction entre l'effectif d'apprentissage et le niveau du bruit de fond conserve un effet significatif dans plus de la moitié des cas avec l'algorithme *CART*. De plus, chez ce dernier, le taux de variables parasites† montre également un effet de moindre importance sur le taux d'erreur moyen, qui se traduit par une valeur observée du test F près de soixante fois plus faible que son homologue et une p-valeur médiane juste inférieure au seuil de signification de 1‰ fixé pour cette analyse.

Ces premières interprétations analytiques sont utilement éclairées par la figure 21, qui présente sous une forme synthétique les résultats de la simulation concernant les paramètres des échantillons. Le graphique suit une conformation identique à celle décrite au paragraphe III.3.2, mettant en évidence la relation entre l'effectif d'apprentissage et le taux d'erreur, mais il est basé sur une moyenne calculée au travers des 23 concepts†.

Cette figure confirme visuellement la divergence de comportement moyen de l'algorithme *Random Forest* et de la méthode *CART*.

Tableau 14. Synthèse des résultats de l'analyse de la variance sur les 23 concepts testés, décomposée selon l'algorithme utilisé (phase 2 – paramètres des échantillons).

Algo.	Sources de variation	Valeurs moyennes		Valeurs médianes	
		F	P-valeur	F	P-valeur
RF.BEST.500	ntrain	5826,54	0,00%	4624,33	0,00%
	irr	2269,74	0,00%	293,13	0,00%
	noise	1034,20	0,00%	1109,38	0,00%
	ntrain:irr	357,30	0,00%	45,75	0,00%
	ntrain:noise	37,94	0,08%	31,38	0,00%
	irr:noise	16,45	0,00%	10,53	0,00%
	ntrain:irr:noise	9,88	0,00%	6,24	0,00%
CART	ntrain	2013,74	3,83%	1413,93	0,00%
	irr	350,76	22,99%	5,13	0,04%
	noise	120,13	5,87%	66,64	0,00%
	ntrain:irr	97,54	33,24%	1,40	13,26%
	ntrain:noise	17,31	4,21%	7,02	0,00%
	irr:noise	14,51	22,13%	1,41	12,51%
	ntrain:irr:noise	6,00	33,03%	1,20	13,56%

Le taux d'erreur moyen associé à l'algorithme *Random Forest* montre une évolution suivant une courbe décroissante approximativement linéaire en fonction du logarithme de l'effectif d'apprentissage, ce qui traduit l'existence probable d'un modèle général de type décroissance logarithmique entre ces deux variables brutes. Cette tendance linéaire subit toutefois une distorsion asymptotique inévitable à l'approche de l'axe des abscisses, qui correspond aux cas d'apprentissage les plus favorables (effectif élevé, niveau de perturbation faible) auxquels sont associés une erreur quasi nulle.

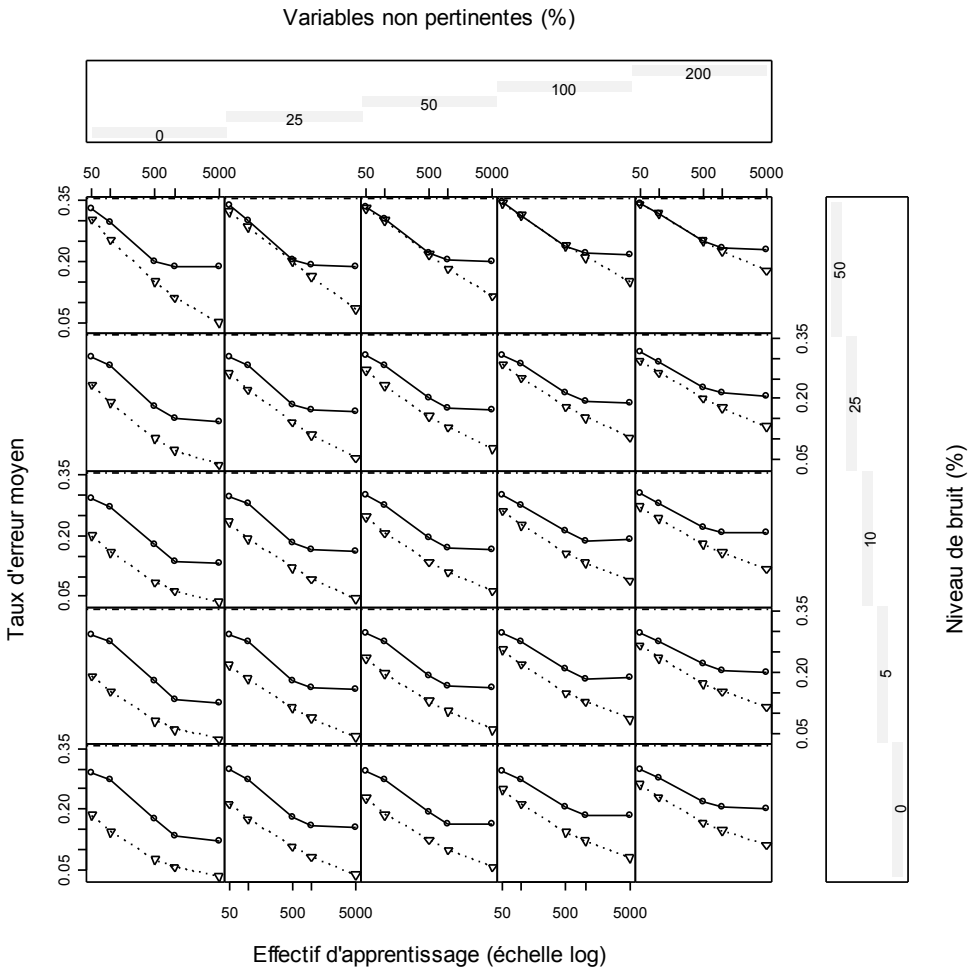


Figure 21. Evolution du taux d'erreur moyen en fonction de l'effectif d'apprentissage(−o− CART, ··∇·· RF, lignes = niveaux de bruit (en %), colonnes = taux de variables parasites (en %)).

Les courbes décrivant l'algorithme *CART* présentent d'abord une évolution grossièrement parallèle à celles de l'algorithme *RF*, bien que décalée verticalement vers les valeurs d'erreur plus élevées. Mais la relation atteint un plateau marqué aux environs des effectifs de 500 et 1000 individus qui est situé à un niveau nettement supérieur au taux d'erreur minimal. Contrairement à la méthode *RF*, cette asymptote n'est donc pas liée au caractère borné de la mesure mais pourrait traduire une caractéristique interne propre à la méthode.

Les forêts aléatoires† affichent sans surprise des performances moyennes significativement supérieures à la méthode *CART*, ce qui est confirmé par l'analyse de la variance globale (Tableau 13), le facteur algorithme correspondant à la p-valeur et au F observé respectivement la plus faible et le plus élevé du tableau des valeurs médianes. Mais cette interprétation doit d'être minutieusement nuancée en fonction des nombreuses interactions dans lesquelles ce facteur est impliqué, ce qui sera l'objet de la suite de ce paragraphe, mais également en fonction des concepts† et de leurs caractéristiques individuelles (§ IV.3.3)

On peut noter que l'écart de taux d'erreur observé en faveur de l'algorithme *RF* se réduit avec le niveau général de perturbation de l'échantillon, jusqu'à ce que les deux courbes se fondent l'une dans l'autre dans les cas extrêmes, alors que les méthodes basées sur l'agrégation d'arbres de type *bagging*† sont réputées moins sensibles au bruit que les formes classiques à arbre unique (DIETTERICH, 2000, BREIMAN, 2001). Toutefois, même dans ces situations, la méthode *RF* tire un avantage constant de l'augmentation de l'effectif qui lui permet de surpasser la méthode *CART*, bloquée par son asymptote, en profitant du surcroît d'information présent dans les grands échantillons (effectif > 1000).

La confirmation de ces observations passe à présent par la modélisation de la relation entre l'effectif d'apprentissage et le taux d'erreur des classificateurs correspondants.

IV.3.2. Effet de l'effectif d'apprentissage

La figure 21 montre une relation de type approximativement linéaire entre le logarithme de l'effectif d'apprentissage et le taux d'erreur associé, accompagnée de phénomènes asymptotiques à des degrés divers selon les situations et les algorithmes. L'examen des graphiques des concepts† individuels ne remet pas en cause ce constat général. Le choix du modèle générique destiné à approcher ces valeurs observées s'est donc naturellement porté vers la régression linéaire segmentée (MUGGEO, 2003).

Ce type de régression consiste à ajuster une ligne brisée aux observations, c'est-à-dire à associer au cours d'un même ajustement

deux modèles linéaires distincts, valables de part et d'autre d'un point de rupture qui constitue également un point de jonction assurant la continuité du modèle (Figure 22).

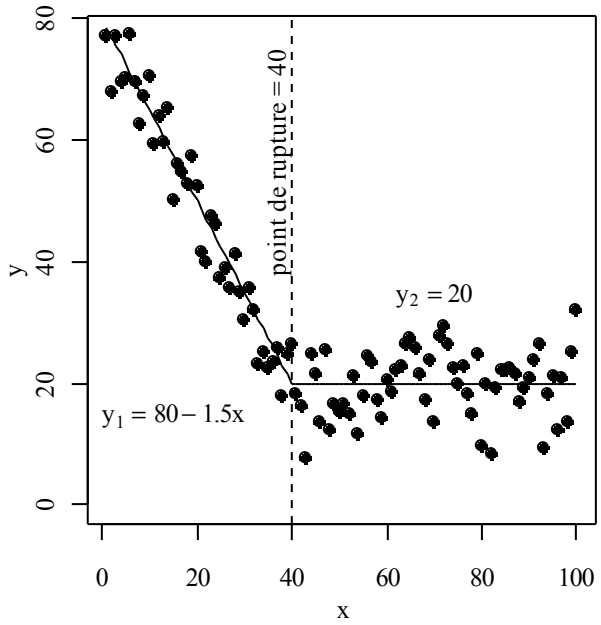


Figure 22. Exemple de régression simple segmentée.

Chacune des 1150 combinaisons individuelles des facteurs algorithme (2), niveau de bruit (5) et variables parasites† (5) de chaque concept† (23) a fait l'objet d'une modélisation visant à établir la relation existant entre le logarithme en base 10 de l'effectif d'apprentissage et le taux d'erreur observé. Deux modèles ont été testés afin de tenir compte de la présence ou non d'une asymptote, une régression simple et une régression segmentée (libraire R `segmented` 0.1-4, MUGGEO, 2004). La régression segmentée a été retenue uniquement si :

- le point de rupture estimé est situé dans le domaine observé de l'effectif d'apprentissage (condition d'existence de la régression segmentée);
- la réduction de l'erreur résiduelle par rapport à la régression simple est significative au seuil de 1‰.

Dans le cas contraire, les données ont été modélisées par une régression simple, dont la signification a elle-même été vérifiée. En cas d'absence de pente significative, une droite horizontale a été ajustée. Les coefficients de chaque modèle ont été enregistrés, ainsi que l'abscisse du point de rupture le cas échéant. Ces coefficients sont utilisés en tant que nouvelles données synthétiques, caractéristiques du comportement de prédiction des deux algorithmes, lors des analyses comparatives subséquentes.

IV.3.2.1. Pente initiale

Etant donné la forme analytique du modèle, l'examen des pentes avant rupture nous renseigne sur le gain initial attendu sur le taux d'erreur pour une augmentation d'un facteur 10 de l'effectif d'apprentissage, indépendamment du phénomène asymptotique qui sera détaillé par la suite. Une analyse de la variance à quatre facteurs croisés fixes a donc été réalisée, faisant intervenir le concept†, l'algorithme, le niveau de bruit et le taux de variables parasites†. Etant donné l'absence de répétition, l'interaction d'ordre 3 a été hypothétiquement considérée comme non significative et utilisée comme variation résiduelle. Le tableau 15 reprend les résultats de cette analyse, décomposés par algorithme, une analyse préalable ayant montré des effets très hautement significatifs de ce facteur ($F_{1,352} = 45{,}95$, $P < 0{,}001$) et de ses interactions ($F_{max} = F_{88,352} = 10{,}58$, $P < 0{,}001$).

Pour interpréter ce tableau, signalons tout d'abord que, bien que de nature fixe et traité comme tel, nous considérons ici le facteur concept† comme étant un facteur de contrôle, équivalent à un bloc. Nous ne nous intéressons donc pas directement à ce facteur et ses interactions, dont tous présentent des effets significatifs, tout en gardant à l'esprit que ces dernières trahissent la diversité des influences factorielles selon les concepts étudiés. L'interprétation suivante concerne donc le comportement moyen des algorithmes, au travers des vingt-trois concepts étudiés.

Tableau 15. Tableau d'analyse de la variance sur les pentes avant rupture de la régression du taux d'erreur en fonction du logarithme en base 10 de l'effectif d'apprentissage.

Algo.	Sources de variation	dl	CM	F	P-valeur
RF.BEST.500	concept	22	0,2120	182,60	0,00%
	irr	4	0,0446	38,39	0,00%
	noise	4	0,0078	6,68	0,00%
	concept:irr	88	0,0170	14,64	0,00%
	concept:noise	88	0,0044	3,78	0,00%
	irr:noise	16	0,0039	3,33	0,00%
CART	concept	22	0,3175	273,41	0,00%
	irr	4	0,0849	73,09	0,00%
	noise	4	0,0015	1,28	27,72%
	concept:irr	88	0,0166	14,29	0,00%
	concept:noise	88	0,0069	5,93	0,00%
	irr:noise	16	0,0006	0,54	92,30%
	Résidus	352	0,0012		

La principale différence entre les algorithmes *RF* et *CART* qui ressort du tableau 15 concerne le facteur représentant le niveau de bruit (`noise`). Alors que ce dernier présente une interaction avec la présence de variables parasites† et un effet global très hautement significatif chez les forêts aléatoires†, il ne semble avoir aucune influence sur le taux de décroissance initial de l'erreur chez la méthode *CART*.

Ce constat est corroboré par l'examen des pentes moyennes, rassemblées dans le tableau 16. Alors que le taux de décroissance de l'erreur de prédiction associé à *RF* diminue globalement avec l'augmentation du niveau de bruit, il reste inchangé chez *CART*, et ce

quelque soit le taux de variables parasites†. Par contre la méthode *CART* apparaît plus sensible aux variations de ce dernier paramètre.

Tableau 16. Valeurs moyennes des pentes avant rupture du modèle linéaire reliant le taux d'erreur au logarithme en base 10 de l'effectif d'apprentissage (erreur standard = 0,007).

Algo.	Niveau de bruit (%)	Taux de variables parasites (%)					
		0	25	50	100	200	Moyenne
RF.500.Best	0	-0,143	-0,143	-0,149	-0,129	-0,107	-0,134
	5	-0,139	-0,138	-0,138	-0,122	-0,101	-0,128
	10	-0,142	-0,151	-0,132	-0,107	-0,116	-0,130
	25	-0,156	-0,138	-0,103	-0,102	-0,090	-0,118
	50	-0,165	-0,119	-0,107	-0,096	-0,087	-0,115
	Moyenne	-0,149	-0,138	-0,126	-0,111	-0,100	-0,125
CART	0	-0,143	-0,134	-0,107	-0,089	-0,079	-0,110
	5	-0,145	-0,123	-0,107	-0,087	-0,081	-0,109
	10	-0,147	-0,128	-0,111	-0,088	-0,092	-0,113
	25	-0,141	-0,126	-0,105	-0,090	-0,075	-0,107
	50	-0,148	-0,149	-0,115	-0,085	-0,084	-0,116
	Moyenne	-0,145	-0,132	-0,109	-0,088	-0,082	-0,111

Mis à part l'absence d'effet du bruit de fond chez la méthode *CART*, l'augmentation du niveau de perturbation, qu'il provienne du bruit de fond ou des variables parasites†, entraîne généralement une réduction du taux de décroissance marginal de l'erreur de prédiction, ce qui s'explique aisément par la diminution de la qualité de l'information fournie par chaque individu supplémentaire en présence de perturbation de ses caractéristiques.

Une exception notable peut toutefois être observée chez les forêts aléatoires† en absence de variables parasites†, où la valeur absolue de la pente moyenne augmente avec le niveau de bruit. La figure 21 nous

offre une explication potentielle à ce phénomène : en l'absence de bruit de fond et de variables parasites†, le taux d'erreur en prédiction des algorithmes *RF* étant déjà très bas, la relation avec l'effectif d'apprentissage subit très tôt l'effet d'asymptote lié à l'abscisse et s'aplatit ; lorsque le bruit de fond augmente, le taux d'erreur en prédiction augmente également et, s'éloignant de l'asymptote, la relation en subit moins les effets et voit sa pente se raidir.

Concernant la comparaison directe des deux algorithmes dans des situations de perturbation identiques, leurs performances en terme de réduction d'erreur sont similaires dans les cas extrêmes (aucune perturbation ou perturbation maximale). La différence se marque principalement dans les valeurs intermédiaires, le plus souvent en faveur des forêts aléatoires† à l'exception de quelques cas présentant un niveau élevé de bruit et un taux de variables parasites† intermédiaires. Pour un échantillon donné, le gain d'information accompagnant une augmentation de l'effectif d'apprentissage est donc généralement plus élevé avec les forêts aléatoires†.

Ces considérations sur le taux de décroissance de l'erreur de prédiction par rapport à l'effectif d'apprentissage doivent à présent être complétées par une étude similaire sur la position verticale du point de départ de cette relation.

IV.3.2.2. Ordonnée à l'origine du domaine

Une indication concernant l'erreur initiale associée à la relation entre l'erreur de prédiction et l'effectif d'apprentissage (en base logarithmique) nous est fournie par l'ordonnée à l'origine de la régression linéaire, qu'elle soit segmentée ou non. Cependant, cette valeur est fortement sensible aux variations de pente, et ce d'autant plus que le point d'abscisse moyenne est éloigné de l'origine.

Dans le cas présent, l'ordonnée à l'origine correspondrait au taux d'erreur associé à un classificateur construit sur base d'un effectif d'apprentissage constitué d'un seul individu ($\log_{10}(x_0) = 0 \Rightarrow x_0 = 1$) . Ce point est très éloigné des bornes du domaine expérimental étudié et n'apporte aucune information exploitable.

Nous avons donc opéré un changement d'origine de la relation, en considérant la valeur minimale observée de l'effectif d'échantillonnage comme nouveau point de référence du modèle. Le terme d'indépendant du modèle devient dès lors le taux d'erreur estimé par celui-ci pour un effectif de 50 individus. Cette valeur a fait l'objet d'une analyse de la variance basée sur le même modèle que la pente avant rupture (§ IV.3.2.1). Le facteur algorithme ($F_{1,352} = 2821$, $P < 0{,}001$) et ses interactions ($F_{max} = F_{4,352} = 236{,}34$, $P < 0{,}001$) présentant comme pour la pente des effets très hautement significatifs, l'analyse a préalablement été scindée selon ce facteur. Les résultats de cette décomposition sont présentés dans le tableau 17. Son interprétation suit la même démarche que le tableau 15.

Si on se réfère aux résultats du tableau 17, les deux algorithmes ne se distinguent guère par la signification des paramètres des échantillons. Tous deux présentent des effets très hautement significatifs à la fois des facteurs bruit de fond et variables parasites†. Par contre, l'interaction entre ces deux paramètres est cette fois non significative chez les deux méthodes. Si on considère la valeur relative du test F comme indicatrice de l'importance de l'effet d'un facteur (le nombre de niveaux étant identique), on remarque que le bruit de fond semble avoir un impact plus marqué sur l'ordonnée à l'origine du domaine que les variables parasites, contrairement à ce qui a pu être observé sur les pentes initiales des modèles linéaires (Tableau 15).

Les valeurs moyennes des ordonnées sur les vingt-trois concepts†, compilées dans le tableau 18 par algorithme, niveau de bruit et taux de variables parasites†, précisent et complètent ces premières interprétations.

Globalement, le taux d'erreur initial des forêts aléatoires† est plus faible que celui de la méthode CART, mais cet écart se réduit avec l'augmentation du degré de perturbation, jusqu'à s'inverser dans les situations extrêmement perturbées (`noise` = 50% et `irr` ≥ 50%), confirmant par là les observations réalisées au départ de la figure 21.

Tableau 17. Tableau d'analyse de la variance sur les ordonnées à l'origine du domaine de la régression du taux d'erreur en fonction du logarithme en base 10 de l'effectif d'apprentissage.

Algo.	Sources de variation	dl	CM	F	P-valeur
RF.BEST.500	concept	22	0,4155	357,87	0,00%
	irr	4	0,0673	57,99	0,00%
	noise	4	0,1926	165,84	0,00%
	concept:irr	88	0,0102	8,80	0,00%
	concept:noise	88	0,0023	1,94	0,00%
	irr:noise	16	0,0010	0,86	61,36%
CART	concept	22	0,5221	2703,80	0,00%
	irr	4	0,0024	12,29	0,00%
	noise	4	0,0202	104,66	0,00%
	concept:irr	88	0,0012	6,42	0,00%
	concept:noise	88	0,0022	11,17	0,00%
	irr:noise	16	0,0003	1,73	3,96%
	Résidus	352	0,0002		

Le taux d'erreur initial des deux algorithmes augmente avec le niveau de bruit, mais cette augmentation est de moindre amplitude chez la méthode *CART* étant donné son point de départ plus élevé (+3,2% contre +10% pour *RF* entre 0 et 50% de bruit). Par contre l'effet des variables parasites† est très contrasté entre les deux méthodes. Alors que l'adjonction de variables parasites† augmente comme on s'y attend l'erreur initiale du classificateur *RF* (+6,2% entre 0 et 200% de variables parasites†), l'erreur initiale des arbres *CART* reste stable voire diminue légèrement (-0,8%) lorsque le taux de variables parasites augmentent.

Tableau 18. Valeurs moyennes des ordonnées à l'origine du domaine du modèle linéaire reliant le taux d'erreur au logarithme en base 10 de l'effectif d'apprentissage (erreur standard = 0,003).

Algo.	Niveau de bruit (%)	Taux de variables parasites (%)					
		0	25	50	100	200	Moyenne
RF.500.Best	0	0,185	0,215	0,229	0,249	0,261	0,228
	5	0,194	0,221	0,237	0,255	0,266	0,235
	10	0,201	0,234	0,245	0,258	0,273	0,242
	25	0,235	0,260	0,266	0,281	0,290	0,266
	50	0,304	0,324	0,332	0,340	0,340	0,328
	Moyenne	0,224	0,251	0,262	0,277	0,286	0,260
CART	0	0,302	0,303	0,289	0,288	0,287	0,294
	5	0,305	0,299	0,293	0,290	0,291	0,296
	10	0,302	0,303	0,299	0,292	0,297	0,299
	25	0,309	0,305	0,299	0,299	0,300	0,302
	50	0,324	0,331	0,321	0,329	0,327	0,326
	Moyenne	0,308	0,308	0,300	0,300	0,300	0,303

IV.3.2.3. Asymptote

Les conditions initiales du modèle linéaire étant à présent définies, deux cas de figure peuvent se présenter : soit le modèle est simple, auquel cas la relation dont les paramètres sont étudiés ci-avant continue ses effets sur toute l'amplitude des tailles d'échantillons d'apprentissage† étudiées, soit il présente une rupture significative sur ce même domaine, et c'est un nouveau modèle linéaire qui décrit la relation effectif d'apprentissage/erreur de prédiction.

Parmi ces cas de rupture, certains vont nous intéresser plus particulièrement car ils traduisent une stagnation de l'erreur malgré l'augmentation de l'effectif d'apprentissage, à savoir les modèles dont la pente après rupture n'est plus significativement différente de zéro

(risque $\alpha = 1‰$), que nous désignerons indifféremment par ruptures asymptotiques† ou plateaux. Ceux-ci ont été systématiquement détectés et les coordonnées des points de rupture correspondants ont été relevées pour analyse. Sont également compris dans les plateaux les modèles linéaires simples non significatifs, le point d'origine du plateau ayant alors respectivement comme abscisse et comme ordonnée la limite inférieure du domaine des effectifs d'apprentissage et la moyenne des taux d'erreur sur ce même domaine.

Nous allons maintenant tenter de caractériser cet effet asymptotique au travers de l'examen de ses occurrences (quelle fréquence ?) et de sa position horizontale (à partir de quel effectif ?) et verticale (quel taux d'erreur limite ?).

a) Occurences

Le tableau 19 reprend les occurrences des modèles avec plateaux pour les différentes combinaisons d'algorithmes, de niveaux de bruit et de taux de variables parasites† au travers des 23 concepts† étudiés. Ce tableau de contingence est analysé au travers de tests Khi carré d'indépendance, en reprenant les conventions standard en matière de risque de première espèce ($\alpha = 5\%$) étant donné la taille raisonnable du problème.

Conformément aux premières observations réalisées sur la figure 21, ce phénomène apparaît nettement plus fréquent chez l'algorithme *CART* ($\chi^2_{1dl} = 69{,}8$, $P < 0{,}001$). L'effet des facteurs de perturbation est également différent selon l'algorithme utilisé ($\chi^2_{49dl} = 96{,}0$, $P < 0{,}001$).

Le niveau de bruit et le taux de variables parasites† ne présentent d'interdépendance chez aucune des deux méthodes (*RF*: $\chi^2_{16dl} = 3{,}83$, $P = 0{,}999$; *CART*: $\chi^2_{16dl} = 3{,}65$, $P = 0{,}999$), mais leurs effets individuels sont par contre distincts en fonction de ces dernières. Ainsi le niveau de bruit n'influence pas significativement l'apparition des plateaux chez le classificateur *CART* ($\chi^2_{4dl} = 2{,}36$, $P = 0{,}669$), mais il réduit celle-ci chez les forêts aléatoires† ($\chi^2_{4dl} = 13{,}82$, $P = 0{,}008$) de manière hautement significative. De même, le taux de variables parasites ne présente pas d'effet significatif chez *CART* ($\chi^2_{4dl} = 0{,}989$, $P = 0{,}912$) mais bien chez *RF* ($\chi^2_{4dl} = 9{,}79$, $P = 0{,}044$).

Tableau 19. Occurrences des ruptures asymptotiques en fonction des algorithmes, du niveau de bruit et du taux de variables parasites (sur un total de 23 concepts pour chaque combinaison et 575 modèles par algorithme).

Algo.	Niveau de bruit (%)	Taux de variables parasites (%)					
		0	25	50	100	200	Total
RF.500.Best	0	13	12	10	10	9	54
	5	14	12	9	11	8	54
	10	14	8	7	10	10	49
	25	12	7	7	7	6	39
	50	9	5	5	3	3	25
	Total	62	44	38	41	36	**221**
CART	0	17	17	17	17	19	87
	5	18	19	19	17	19	92
	10	19	19	18	19	18	93
	25	18	19	18	17	16	88
	50	19	16	18	11	11	75
	Total	91	90	90	81	83	**435**

Globalement, on peut donc conclure que les plateaux sont moins fréquents chez les forêts aléatoires† et généralement associés aux faibles niveaux de perturbation, tandis que ceux-ci sont répartis de manière approximativement uniforme dans les différents situations de perturbation chez les arbres *CART*. Il reste à présent à déterminer leur position lorsqu'ils existent.

b) Position en abscisse

Etant donné le caractère déséquilibré et incomplet du jeu de données constitué par les coordonnées des ruptures asymptotiques†, il n'est pas possible de présenter un tableau d'analyse de la variance complet pour ces variables. Néanmoins, une analyse simplifiée ne retenant que les concepts† et les algorithmes nous permet de conclure à

l'absence d'effet des algorithmes sur la coordonnée horizontale du point d'origine du plateau exprimée en logarithme en base 10 ($F_{1,614} = 0.0004$, $P = 0.984$, type III).

La répartition de ces points d'asymptotes le long de l'axe des abscisses est donnée par le tableau 20 à titre indicatif. L'absence de point de rupture dans le dernier intervalle s'explique aisément par le principe de la régression segmentée. En effet, une rupture de pente située entre 1000 et 5000 individus ne laisserait qu'un seul niveau d'observation pour ajuster le second modèle linéaire, ce qui est insuffisant pour estimer sa pente. De même, les ruptures appartenant au premier intervalle correspondent en fait aux régressions non significatives, dont l'abscisse du point asymptotique a été fixée à 50.

Tableau 20. Répartition des abscisses des points de ruptures asymptotiques.

Algo.	Effectif d'apprentissage					
	50	100	500	1000	5000	Total
RF	25	82	114	0		221
CART	184	37	214	0		435
Total	209	119	328	0		656

c) Position en ordonnée

Pour les raisons déjà évoquées au paragraphe précédent, seule une analyse de la variance simplifiée a pu être réalisée sur la position verticale de l'asymptote. Celle-ci conclut à un effet très hautement significatif de l'algorithme de classement† utilisé ($F_{1,614} = 489{,}4$, $P < 0.001$, type III).

Cette analyse, appuyée par l'examen des taux d'erreur moyens des plateaux détectés du tableau 21, confirme l'hypothèse selon laquelle la grande majorité des phénomènes asymptotiques rencontrés par les forêts aléatoires† sont liés à l'effet de bord engendré par la proximité de l'axe des abscisses. Les arbres *CART* affichent quant à eux une limitation de la progression du taux d'erreur à un niveau bien plus élevé (+12,3% en moyenne par rapport aux forêts aléatoires), ce qui pourrait traduire une limitation interne du processus d'apprentissage.

Tableau 21. Taux d'erreur moyen correspondant aux ruptures asymptotiques, en fonction des algorithmes, du niveau de bruit et du taux de variables parasites.

Algo.	Niveau de bruit (%)	Taux de variables parasites (%)					
		0	25	50	100	200	Moyenne
RF.500.Best	0	0,002	0,002	0,002	0,098	0,138	0,042
	5	0,004	0,004	0,004	0,116	0,129	0,045
	10	0,003	0,002	0,003	0,105	0,139	0,051
	25	0,004	0,004	0,075	0,161	0,186	0,073
	50	0,013	0,008	0,115	0,360	0,351	0,114
	Moyenne	0,005	0,003	0,031	0,134	0,162	**0,059**
CART	0	0,144	0,144	0,183	0,185	0,195	0,171
	5	0,146	0,153	0,173	0,186	0,187	0,169
	10	0,153	0,153	0,163	0,184	0,200	0,170
	25	0,165	0,163	0,189	0,210	0,221	0,188
	50	0,194	0,195	0,213	0,247	0,282	0,219
	Moyenne	0,161	0,161	0,184	0,198	0,211	**0,182**

IV.3.3. Caractéristiques des concepts

A présent que l'influence des paramètres dépendant strictement de l'échantillon a été prospectée, il reste à déterminer les effets éventuels des caractéristiques du concept† que représente celui-ci. Le tableau 5 du paragraphe II.3.2 comprend une série de paramètres auxquels nous nous intéresserons plus particulièrement ici, à savoir la dimensionnalité du concept (*p*), sa variation interne (*variation*), sa complexité (*complex*) et son taux d'erreur de base† (*err.base*), dont les notions sont définies au cours de ce même paragraphe. Plutôt que d'étudier leurs effets sur le taux d'erreur brut, nous avons choisi de nous intéresser aux paramètres des modèles linéaires établis au paragraphe précédent, qui constituent

un condensé synthétique de ces relations et sont porteurs d'une information déjà raffinée.

Au cours d'une première étape exploratoire, nous avons représenté graphiquement les relations existant entre les paramètres des concepts† et ceux des modèles linéaires, à savoir la pente initiale, l'ordonnée à l'origine du modèle et l'ordonnée de l'éventuelle asymptote, en terme de valeur moyenne pour l'ensemble du concept. Ces relations sont présentées à la figure 23 pour les deux algorithmes testés, accompagnées d'une courbe lissée établie par régression robuste localement pondérée[78] (CLEVELAND, 1979) assurant une visualisation plus aisée de ces dernières.

Trois relations attirent plus particulièrement notre attention par leur qualité apparente :

- l'ordonnée de l'asymptote en fonction de la variation ;
- l'ordonnée à l'origine du domaine en fonction de la variation ;
- la pente initiale en fonction de la dimensionnalité du concept†.

[78] En anglais: *robust locally weighted regression, LOWESS*.

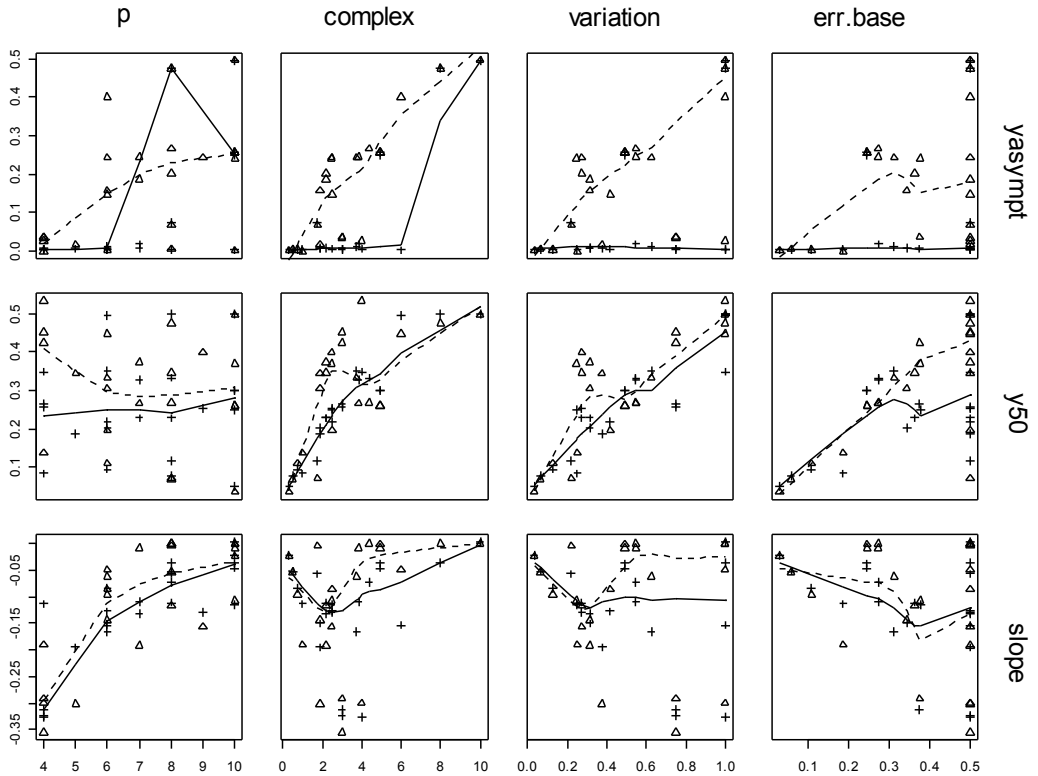


Figure 23. Relations entre les paramètres des concepts et les valeurs moyennes des paramètres des modèles linéaires segmentés (yasympt = ordonnée de l'asymptote, y50 = ordonnée à l'origine du domaine, slope = pente initiale du modèle, + = RF.500.Best, Δ = CART, — = LOWESS RF.500.Best, - - - = LOWESS CART)

IV.3.3.1. Ordonnée de l'asymptote en fonction de la variation

L'examen attentif de la figure 23 montre clairement que les conditions requises pour l'utilisation des techniques de régression classique au sens des moindres carrés ne sont pas vérifiées. On remarque notamment la présence pour chaque algorithme de quelques points s'écartant fortement du comportement global de ses compagnons[79]. La présence de valeurs présentant des résidus aussi extrêmes nous a conduit à confier l'ajustement du modèle à une méthode de régression particulièrement robuste, la régression par les moindres carrés rognés[80] (ROUSSEEUW et LEROY, 1987).

Cet ajustement nous livre les équations suivantes :

RF.500.Best : $yasympt = 0{,}0045 + 0{,}0013 \times variation$;

CART : $yasympt = -0{,}0150 + 0{,}5109 \times variation$.

Même si la méthode numérique utilisée pour la recherche des paramètres ne permet pas le calcul de leurs erreurs standards, ces équations confirment sans équivoque la différence de comportement des deux algorithmes. Alors que la position moyenne des asymptotes est peu influencée par la variation du concept† chez les forêts aléatoires†, ces dernières semblent s'établir préférentiellement à un niveau correspondant à environ la moitié de la valeur de la variation chez les arbres *CART*.

Les exceptions à cette tendance peuvent être aisément expliquées. Les points appartenant à l'algorithme *CART* et situés près de l'axe des abscisses correspondent à des concepts† de faible dimensionnalité, pour lesquels la méthode atteint sans problème un taux d'erreur quasi nul dans la gamme d'effectifs d'échantillonnage testée. Tandis que les individus représentant les forêts aléatoires† et partageant le comportement des arbres *CART* sont en fait des concepts conjuguant une forte variation et une dimensionnalité élevée, conduisant à des modèles pour lesquels l'erreur de base† joue le rôle d'une barrière quasi infranchissable pour chacun des algorithmes.

---

[79] En anglais: *outliers*.

[80] En anglais: *least trimmed squares regression, LTS*.

IV.3.3.2. Ordonnée à l'origine du domaine en fonction de la variation

Les écarts résiduels extrêmes affichés par cette relation étant de moindre importance que la relation précédente, le choix de la méthode de régression s'est reporté sur une régression robuste par repondération itérative des moindres carrés[81] (VENABLES et RIPLEY, 1997), autorisant cette fois le calcul des erreurs standards des paramètres et les tests de signification des modèles ajustés.

Trois modèles concurrents ont été testés et comparés : un modèle linéaire unique pour les deux algorithmes, un modèle à pentes parallèles et ordonnées distinctes et un modèle à paramètres totalement indépendants par algorithme. L'analyse comparée de leur erreur résiduelle montre que le modèle unique doit être préféré aux deux autres ($F_{1,43} = 3{,}94$, $P = 0{,}054$). Les deux algorithmes montrent donc une évolution similaire de leur taux d'erreur moyen pour les faibles effectifs en fonction de la variation du concept†, celui-là augmentant proportionnellement à cette dernière.

On ne peut cependant écarter une influence de l'erreur de base† du concept†, qui apparaît corrélée à l'ordonnée à l'origine mais également à la valeur de la variation, du moins pour les valeurs faibles. L'introduction de cette variable supplémentaire apporte effectivement une contribution significative à la réduction de l'erreur résiduelle ($F_{1,43} = 6{,}29$, $P = 0{,}016$), mais ne remet pas en question le rôle de la variation dans cette relation.

La relation décrivant le lien entre la variation du concept† et l'ordonnée à l'origine moyenne s'écrit donc finalement pour les deux algorithmes (paramètre ± erreur standard) :

$$y50 = 0{,}049 \pm 0{,}026 + 0{,}342 \pm 0{,}041 \times variation + 0{,}192 \pm 0{,}082 \times err.base,$$

l'erreur initiale des modèles croissant proportionnellement à la variation du concept† et à son erreur de base† (écart-type résiduel = 0,054).

[81] En anglais: *iterated re-weighted least squares, IWLS*.

IV.3.3.3. Pente initiale en fonction de la dimensionnalité du concept

La courbe de lissage associée à cette relation suggère au premier abord une modèle de forme non linéaire. Cependant, une transformation logarithmique de l'axe des abscisses corrige cette non linéarité et rend possible l'utilisation de la régression robuste par repondération itérative des moindres carrés déjà utilisée ci-dessus.

Comme précédemment, trois modèles impliquant une influence de plus en plus marquée de l'algorithme de classement† ont été comparés. Il ressort de cette comparaison qu'une fois encore l'algorithme ne joue pas ici de rôle significatif dans la relation ($F_{1,43} = 0{,}49$, $P = 0{,}489$), débouchant donc sur le choix du modèle de régression linéaire unique pour la description du lien unissant dimensionnalité et pente initiale moyenne du modèle. Ce modèle s'écrit alors (paramètre ± erreur standard) :

$$slope = -0{,}632 \pm 0{,}064 + 0{,}617 \pm 0{,}077 \times \log_{10}(p),$$

l'écart-type résiduel de cette relation s'élevant à 0,058.

La réduction marginale initiale de l'erreur de prédiction pour chaque multiplication de l'effectif par un facteur 10 est donc d'autant plus faible que le concept† est de grande dimension. Cela s'explique par le lien entre le nombre de dimensions du concept binaire et la quantité relative d'information apportée par chaque individu : plus un concept binaire présente de dimensions, plus son espace contient d'individus distincts et donc plus faible est la quantité d'information sur ce concept détenue par un individu isolé.

IV.3.4. Cas particuliers

Comme cela a déjà été mentionné lors de la première phase de l'analyse, les forêts aléatoires† peuvent présenter dans certaines conditions particulières des performances en prédiction inférieures au vote à la majorité simple dans l'échantillon.

Ces conditions concernent tout d'abord les concepts†. Pour rappel, les familles principalement touchées ont déjà été énumérées au paragraphe IV.2.3, à savoir *équilibre*, *compteur* et *conjonction*. Au sein

de chaque famille, la fréquence de ce phénomène croît avec le nombre de dimensions, dont l'augmentation entraîne une baisse de la taille de la cible† positive mais également de la variation.

Nous avions également remarqué que les contre-performances des forêts aléatoires† dépendent également du niveau général de perturbation et de l'effectif d'apprentissage associé. Nous pouvons à présent mieux comprendre ces relations.

Les tableaux ci-dessous détaillent les occurrences des contre-performances des algorithmes RF par rapport au taux d'erreur de base† des 10 concepts† concernés, et cela par effectif d'apprentissage, par niveau de bruit et par taux de variables parasites†. Ces résultats confirment tout d'abord l'influence de l'effectif d'apprentissage ($\chi^2_{4dl} = 201{,}4$, $P < 0{,}001$), de niveau de bruit ($\chi^2_{4dl} = 75{,}81$, $P < 0{,}001$) et du taux de variables parasites ($\chi^2_{4dl} = 14{,}57$, $P = 0{,}006$). Les effets de ces deux derniers paramètres sont toutefois indépendants l'un de l'autre ($\chi^2_{16dl} = 2{,}043$, $P = 1{,}000$), ainsi que ceux de l'effectif d'apprentissage et du taux de variables parasites ($\chi^2_{16dl} = 11{,}98$, $P = 0{,}745$). Le seul effet de dépendance est celui observé entre l'effectif d'apprentissage et le niveau de bruit ($\chi^2_{16dl} = 42{,}88$, $P < 0{,}001$) : plus l'effectif d'apprentissage est élevé et plus la distribution des contre-performances se décale vers les taux de bruits élevés.

Ces observations sont compatibles avec les conclusions des analyses concernant la relation entre l'effectif d'apprentissage et l'erreur de prédiction. Chez les forêts aléatoires†, l'ordonnée à l'origine de cette relation, représentant le taux d'erreur moyen attendu pour un effectif de 50 individus, est sensible à une augmentation du niveau de perturbation général et plus particulièrement du niveaux de bruit. De plus, cette sensibilité est exacerbée lorsque la taille de la cible† positive (et donc l'erreur de base†) du concept† est réduite (§ IV.2.3). En combinant ces conditions, le taux d'erreur initial observé peut donc dépasser celui de base. Cependant, la pente continue de cette relation et l'absence de plateau limitant l'apprentissage permettent aux algorithmes *RF* de redescendre sous cette limite au fur et à mesure de l'augmentation de l'effectif d'apprentissage, tandis que l'algorithme *CART* y reste perché.

Tableau 22. Occurrences des cas de défection des algorithmes RF.500.Best en fonction de l'effectif d'apprentissage, du niveau de bruit et du taux de variables parasites (sur un total de 10 concepts pour chaque combinaison et 250 par effectif d'apprentissage). Effectifs de 50, 100 et 500 individus.

Effectif appren tissage.	Niveau de bruit (%)	Taux de variables parasites (%)					
		0	25	50	100	200	Total
50	0	4	5	5	5	6	25
	5	5	6	6	5	6	28
	10	5	7	6	6	6	30
	25	7	8	7	7	8	37
	50	8	8	9	10	10	45
	Total	29	34	33	33	36	**165**
100	0	2	4	4	5	6	21
	5	3	4	5	5	6	23
	10	4	4	6	5	6	25
	25	5	7	7	7	7	33
	50	7	8	8	9	9	41
	Total	21	27	30	31	34	**143**
500	0	0	0	1	2	1	4
	5	0	1	2	1	2	6
	10	0	2	2	2	4	10
	25	2	3	3	3	5	16
	50	4	6	6	7	7	30
	Total	6	12	14	15	19	**66**

Tableau 23. Occurrences des cas de défection des algorithmes RF.500.Best en fonction de l'effectif d'apprentissage, du niveau de bruit et du taux de variables parasites (sur un total de 10 concepts pour chaque combinaison et 250 par effectif d'apprentissage). Effectifs de 1000 et 5000 individus.

Effectif appren tissage.	Niveau de bruit (%)	Taux de variables parasites (%)					
1000	0	0	0	0	1	1	2
	5	0	0	0	0	0	0
	10	0	0	0	2	3	5
	25	0	2	2	3	3	10
	50	3	4	5	6	6	24
	Total	3	6	7	12	13	**41**
5000	0	0	0	0	0	0	0
	5	0	0	0	0	1	1
	10	0	0	0	0	1	1
	25	0	0	0	0	0	0
	50	0	2	4	2	3	11
	Total	0	2	4	2	5	**13**

IV.3.5. Conclusions

Comme on pouvait s'y attendre, la méthode des forêts aléatoires† livre globalement des résultats présentant un taux d'erreur inférieur à ceux fournis par les arbres CART. Cette constatation générale est cependant utilement nuancée et complétée par les différences de comportements affichées par les deux algorithmes face aux variations des caractéristiques des échantillons d'apprentissage† utilisés.

L'algorithme RF affiche ainsi une sensibilité supérieure au niveau de bruit affectant l'échantillon d'apprentissage†, l'écart moyen de performance entre les deux algorithmes se réduisant lorsque le bruit augmente. Lorsque l'on modélise la relation existant entre l'erreur

prédiction et l'effectif d'apprentissage, on s'aperçoit que, pour l'algorithme CART, l'erreur initiale et le taux décroissance de celle-ci restent relativement stables sur toute la gamme de niveaux de bruit testés, tandis que ces mêmes paramètres se dégradent au fur et à mesure que le bruit augmente chez la méthode RF. Ce comportement va à l'encontre de la robustesse accordée aux techniques dérivées du *bagging*† face à ce type de perturbation (BAUER et KOHAVI, 1999; DIETTERICH, 2000).

L'introduction de variables parasites† affecte par contre de manière similaire la décroissance initiale du taux d'erreur chez les deux méthodes. L'amélioration des performances en prédiction avec l'augmentation de l'effectif d'apprentissage est ainsi moins rapide en présence de variables parasites, ces dernières diluant l'information utile à l'apprentissage présente dans chaque individu supplémentaire. La présence de ces variables non pertinentes entraîne également une augmentation du taux d'erreur associé aux effectifs faibles chez les forêts aléatoires†, tandis que la méthode CART reste sur ce point peu sensible.

Toutefois, la relative stabilité affichée par les arbres CART face aux perturbations de l'échantillon peut être expliquée par le taux d'erreur globalement élevé de cette méthode, supérieur à celui de la méthode RF et relativement proche du taux d'erreur de base† des concepts† lorsque les effectifs sont faibles, et ce même en présence de perturbations faibles ou nulles.

En outre, les deux méthodes diffèrent par la forme générale de la relation liant erreur de prédiction et l'effectif d'apprentissage. Alors que les forêts aléatoires† montrent une diminution continue et relativement régulière de cette erreur lorsque la taille de l'échantillon augmente (décroissance de type logarithmique), l'algorithme CART présente avec une occurrence accrue un plateau limitant cette baisse sur l'intervalle d'effectifs testé, et ce à un niveau bien supérieur aux taux d'erreur affichés par les forêts aléatoires et dont la valeur dépend de la variation du concept† étudié. Cela a donc pour effet de accentuer l'écart de performance existant entre les deux algorithmes en faveur des forêts aléatoires lorsque les tailles d'échantillon sont importantes (plus de 500 individus).

A l'inverse, dans certaines situations particulières (effectif d'apprentissage faible, perturbations élevées et cible† de taille réduite), la méthode des forêts aléatoires† peut présenter des prédictions de qualité inférieure à celle de la méthode CART, cette dernière se maintenant juste au niveau de l'erreur de base† des concepts† concernés. Cette différence peut s'expliquer par le fait que l'élagage† pratiqué sur l'arbre CART en présence d'un bruit dominant réduit l'arbre à sa racine†, et livre donc une estimation unique basée sur un vote à la majorité simple de l'échantillon, se rapprochant ainsi de l'erreur de base† du concept. Les forêts aléatoires continuent quant à elles de développer leurs arbres et d'extraire de l'information, qui dans ces cas particuliers est dominée par le hasard et contribue à augmenter le taux d'erreur final. Remarquons qu'aucun des estimateurs produits par les deux méthodes dans ces situations extrêmes ne livre de prédictions fiables, la nuance de comportement affichée par les deux méthodes ne justifie donc pas que l'on recommande l'usage privilégié de la méthode CART dans ces cas particuliers.

Globalement, on peut donc conclure que les forêts aléatoires† exploitent plus efficacement l'information contenue dans l'échantillon d'apprentissage† et ce tout au long de la gamme d'effectifs testés. L'erreur moyenne associée aux faibles effectifs est inférieure à celle des arbres CART, même si ce écart se réduit lorsque le niveau de perturbation de l'échantillon augmente. Cette différence en faveur des forêts aléatoires s'accentue encore avec les échantillons de grandes tailles, l'algorithme RF ne souffrant pas de limitation asymptotique de son apprentissage au contraire des arbres CART.

CONCLUSIONS ET PERSPECTIVES

L'objectif de la présente étude était de vérifier la robustesse de la méthode de classement par forêts aléatoires†, telle que développée par BREIMAN, 2001, dans une gamme de situations recouvrant les caractéristiques des données issues du domaine agronomique. Parmi celles-ci, on retrouve la présence dans les concepts† à estimer d'interactions souvent complexes, ainsi que des échantillons de taille modeste (plusieurs dizaines à quelques centaines d'individus), caractérisés par des mesures la plupart du temps entachées d'erreurs aléatoires et de pertinence très variable.

A cela s'ajoute une exploration des deux principaux paramètres d'entrée de l'algorithme *Random Forest* (nombre d'attributs présélectionnés aléatoirement et taille de la forêt), visant à déterminer la ou les combinaison(s) de ces paramètres dont l'usage pourrait être recommandé lors de l'emploi de cette technique de classement.

L'expérimentation conduite par simulation s'est donc attachée à reproduire les variations de ces différents facteurs selon un dispositif factoriel (Tableau 7, page 96), de manière à fournir une information aussi complète que possible concernant leurs effets individuels et mutuels sur la qualité de la prédiction fournie par l'estimateur, en prenant comme référence un algorithme classique de génération d'arbres de décision, CART. La performance de chaque prédicteur a été mesurée par le taux d'erreur affiché lors du classement d'un échantillon indépendant issu du même concept (taux d'erreur en généralisation estimé).

L'analyse des résultats de cette simulation a tout d'abord montré que pour garantir un taux d'erreur en généralisation minimal, il est préférable d'utiliser des forêts de décision comportant au moins cent, voire cinq cents arbres. La présélection complètement aléatoire des attributs lors de la construction des arbres doit quant à elle être abandonnée car elle livre des estimateurs peu fiables notamment en présence de variables non pertinentes, ce qui n'avait pas été mis en évidence dans l'étude originelle de BREIMAN, 2001. L'utilisateur lui préférera une sélection de type mixte, comportant la présélection aléatoire à chaque nouvelle partition d'un certain nombre d'attributs (fixé dans l'expérience à int($\log_2 M + 1$), M représentant le nombre total d'attributs† du jeu d'apprentissage†) suivi d'une sélection déterministe classique basée sur le critère de Gini au sein de ce sous-ensemble.

L'utilisation de cette combinaison de paramètres conduit à l'obtention de classificateurs globalement plus précis que les arbres CART isolés (jusqu'à –10,6% de taux d'erreur moyen sur l'ensemble des concepts testés, soit une baisse relative de 36,4% de l'erreur moyenne du classificateur). Sur des échantillons d'effectifs faibles à modérés (inférieurs à 1000), cet écart se réduit toutefois avec l'augmentation du niveau de perturbation global de l'échantillon, les performances moyennes des deux classificateurs se confondant dans les cas les plus perturbés (50% de bruit de fond aléatoire et plus de 25% de variables parasites†).

Toutefois, dans ces situations particulières, aucune des deux méthodes ne descend sous le taux d'erreur de base† du concept. Dans certains cas (perturbations extrêmes associées à une cible† de petite taille), les forêts aléatoires affichent même un taux d'erreur nettement supérieur à ce taux de base. Ce comportement montre que les limites internes de chacune des méthodes ont été atteintes, ce qui est normal dans ces conditions volontairement extrêmes sur le plan de la difficulté d'apprentissage, et démontre que les bornes supérieures des facteurs de perturbation ont été fixées à une valeur suffisante pour couvrir l'entièreté de la gamme des comportements d'apprentissage.

Si on se limite aux conditions normales d'utilisation de telles méthodes de classement† (c'est-à-dire dans lesquelles la connaissance du concept reste accessible), la supériorité des prédictions des forêts

aléatoires† est sans équivoque. Les seuls cas de défection de cette dernière méthode ont concernés des situations d'apprentissage extrêmes, caractérisées par un effectif faible (50 à 100 individus), un bruit de fond et un taux de variables parasites très élevés (respectivement supérieurs à 25% et 100%) et des distributions de classe cible très déséquilibrées, cette dernière variable étant dès lors facilement "noyée" sous les perturbations. Signalons que dans ces situations, les arbres CART étaient la plupart du temps réduit à leur racine, réduisant le processus de classement à un vote à la majorité simple sur l'échantillon d'apprentissage.

En s'intéressant plus avant à l'utilisation de l'échantillon d'apprentissage† par chaque algorithme, on constate que la méthode des forêts aléatoires† exploite plus efficacement l'information présente dans les individus de l'échantillon d'apprentissage. En effet, le taux d'erreur moyen de cette méthode associé aux échantillons de faibles effectifs est inférieur à celui de la méthode CART. En outre la décroissance de cette erreur liée à l'augmentation de la taille de l'échantillon est plus rapide chez les forêts aléatoires. Enfin, contrairement aux arbres CART, ces dernières ne souffrent pas d'un effet asymptotique limitant leur courbe d'apprentissage pour les échantillons de grande taille, asymptote dont le niveau s'élève avec la variation du concept† et donc son niveau d'interaction. Cette dernière constatation montre que l'avantage des forêts aléatoires tend à croître avec l'augmentation de l'effectif d'apprentissage, les rendant encore plus compétitives par rapport à l'algorithme CART sur les échantillons de grande taille (supérieure à mille individus), ainsi que sur les problèmes marqués par de forts niveaux d'interactions.

La décroissance du taux d'erreur avec l'augmentation de la taille de l'échantillon d'apprentissage† est de nature logarithmique chez les deux algorithmes testés. Cela signifie que la réduction du taux d'erreur présente un coût élevé en terme d'échantillonnage, la taille de l'échantillon devant être multipliée par une constante pour chaque pourcent d'erreur gagné (en moyenne, pour les deux algorithmes et des effectifs modérés, $\times 1.21$).

Le taux de décroissance de l'erreur se réduit avec le niveau de perturbation de l'échantillon (bruit de fond et variables parasites). Il

apparaît donc clairement que ces techniques d'apprentissage réputées robustes tirent néanmoins un avantage non négligeable de l'usage d'un échantillon entaché du moins d'erreurs possibles et caractérisé par des variables judicieusement sélectionnées. Toutefois, dans le domaine biologique qui nous intéresse ici, ces perturbations sont généralement intrinsèques aux individus observés et non à l'opération de mesure. Elles sont donc difficilement réductibles et entraînent une perte de précision qui reste cependant modérée grâce aux différents mécanismes de sélection et d'agrégation des méthodes par arbres et par forêts.

Si la relation générale existant entre l'effectif de l'échantillon d'apprentissage et le taux d'erreur associé aux forêts aléatoires reste constante dans sa nature (logarithmique décroissante), ses paramètres exacts dépendent à la fois des caractéristiques du concept† sous-jacent (variation, taux d'erreur de base†, dimension) et de l'échantillon d'apprentissage (taille, bruit de fond et attributs parasites†), dont la plupart sont peu ou pas connus lors de la phase de construction de l'estimateur. Il n'existe donc pas de règle générale permettant de déterminer *a priori* le nombre d'observations nécessaires pour atteindre une précision donnée. La taille optimale de l'échantillon doit donc être déduite de manière itérative, en se basant sur les résultats de cycles de construction/validation issus d'échantillons de taille croissante pour en extraire les paramètres de la relation empirique liant celle-ci au taux d'erreur. Plus rapide mais beaucoup plus grossière, cette estimation peut également utiliser la valeur moyenne du taux de décroissance observé au cours de cette étude (réduction du taux d'erreur d'1% lorsque l'échantillon croît de 21%) pour situer approximativement la taille d'échantillon nécessaire au départ d'un essai préliminaire unique.

La supériorité des forêts aléatoires sur les arbres CART en matière de qualité des prédictions est donc établie dès les échantillons de faible effectif et sur une large gamme de niveaux de perturbation et de complexité d'apprentissage (à l'exception des cas particuliers dont il a déjà été question). Cependant, ce gain de performance s'établit au détriment de la lisibilité du classificateur, formé de plusieurs centaines d'arbres et donc peu enclin à une représentation synthétique simple et aisément transmissible, au contraire de l'algorithme CART.

Parmi les solutions les plus évidentes à ce problème figurent les méthodes de synthèse structurelle de collections d'arbres telles que développées par SHANNON et BANKS, 1999 et déjà évoquées au paragraphe I.4.4, qui délivrent un arbre unique né du consensus établi sur la structure moyenne du groupe de départ. Cependant, ces méthodes se heurtent à la diversité structurelle des arbres formant les forêts aléatoires, très élevée par construction car forcée par la présélection aléatoire des attributs. L'arbre issu de ces méthodes ne devrait pas être utilisé en tant qu'estimateur, dont les performances seraient inférieures à la forêt originale, mais bien comme un support informatif destiné à la compréhension du phénomène de classement.

Toutefois, l'algorithme de génération des forêts aléatoires fournit lui-même une série d'informations très utiles sur ce plan, sous la forme du calcul des importances des attributs descripteurs. Ces valeurs mesurent par divers biais (§ I.5.4) le pouvoir prédictif de chaque attribut du jeu de données. Cette information offre donc un point de vue quantitatif sur le rôle de chaque variable sur la qualité finale de la prédiction. A titre d'exemple, si nous reprenons le jeu de données réelles *Soybean* traité au paragraphe I.5.5, nous obtenons les valeurs d'importance reprises à la figure 24. Ce type d'information quantitative complète parfaitement l'information structurelle qualitative délivrée par l'arbre consensus et permet une meilleure compréhension du classificateur *Random Forests*, comblant partiellement le manque de représentation synthétique inhérent à sa structure agrégative.

Les méthodes de synthèse structurelle d'ensembles d'arbres sont malheureusement encore peu diffusées hors des laboratoires de recherches et donc indisponibles pour un utilisateur lambda des forêts aléatoires. L'algorithme CART pourrait éventuellement constituer une alternative viable dans les cas où celui-ci fournit un estimateur non trivial, bien que suboptimal. Si nous comparons la figure 24 avec celle présentant un arbre CART construit sur les mêmes données (Figure 11, page 66), on constate une excellente correspondance entre la structure de l'arbre et les valeurs d'importance issues de l'algorithme *Random Forests*, les attributs sélectionnés par CART correspondant effectivement aux valeurs d'importance les plus élevées.

Lorsque que les exigences en matière de classement portent à la fois sur les performances en prédiction et sur la compréhension du processus de classement, nous recommandons donc l'utilisation d'une forêt aléatoire pour les tâches de prédiction, accompagnée d'une part d'un arbre de structure consensuelle ou à défaut d'un arbre CART et d'autre part d'un graphique illustrant l'importance des variables comme indicateurs du fonctionnement interne du classificateur généré.

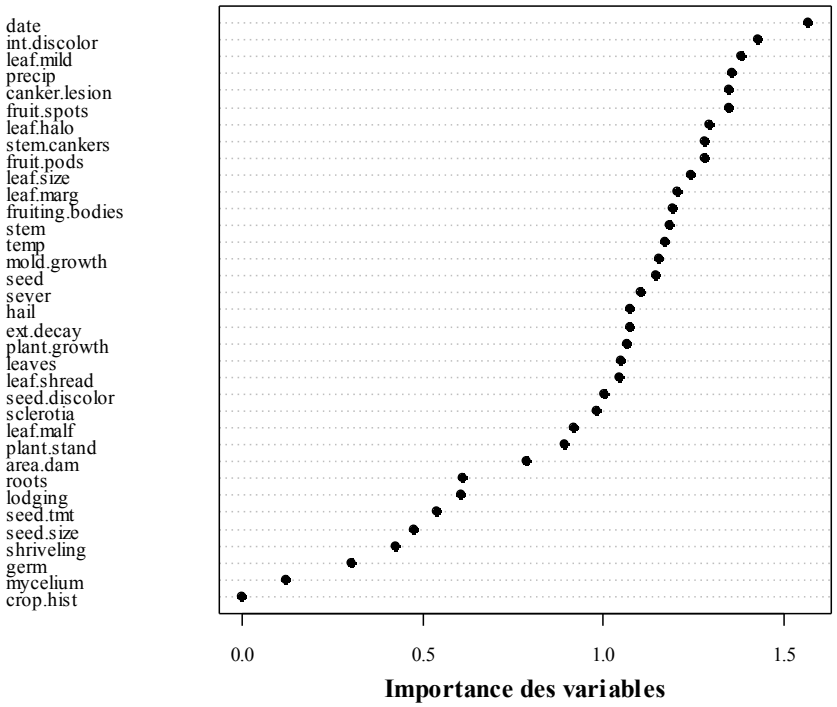


Figure 24. Valeurs estimées de l'importance des variables du jeu de données Soybean, classées par ordre décroissant.

Certaines propriétés particulières des forêts aléatoires soulevées au cours du présent travail méritent un complément d'étude. Ainsi, nous avons pu constaté que la défaillance de ces classificateurs était principalement liée aux concepts présentant des cibles† fortement dissymétriques. L'influence de ce paramètre devrait être explorée de manière plus systématique et étendue aux cas de cibles multivaluées, qui par rapport aux cibles binaires offrent un degré de liberté supplémentaire via la multiplicité des distributions envisageables (une classe de faible effectif *vs* des classes équilibrées de grande taille,

nombreuses classes de faibles effectifs, classes de tailles régulièrement croissantes, etc.).

Nous avons brièvement abordé l'utilisation des forêts aléatoires pour des tâches de classification non supervisée[82] au cours du paragraphe I.5.4. A nouveau leurs performances relatives par rapport aux méthodes classiques de type hiérarchique ou non sont encore peu connues. Une étude comparative systématique du même type que la présente recherche permettrait de lever ces incertitudes.

Enfin, le développement fulgurant des biotechnologies a ouvert ces dernières années un domaine d'analyse entièrement nouveau, au sein duquel les forêts aléatoires occupent une place de choix (BUREAU, DUPUIS, HAYWARD, FALLS et VAN EERDEWEGH, 2003; ZHANG, YU et SINGER, 2003; BUREAU, DUPUIS, FALLS, LUNETTA, HAYWARD, KEITH et VAN EERDEWEGH, 2005; SHI, SELIGSON, BELLDEGRUN, PALOTIE et HORVATH, 2005). Les données issues des analyses du génome et du protéome possédant des caractéristiques internes souvent très éloignées de celles provenant de l'expérimentation agronomique classique (notamment par leur dimensionnalité très élevée), la recherche concernant les performances des algorithmes de type *Random Forests* dans ces situations formera sans aucun doute un axe de développement futur de ces méthodes.

[82] en anglais : *clustering*.

RÉFÉRENCES BIBLIOGRAPHIQUES

AIZENSTEIN H.J. [1993]. *On the learnability of disjunctive normal form formulas and decision trees*. PhD Thesis, University of Illinois, Urbana-Champaign, 123 p.

AMIT Y. et GEMAN D. [1997]. Shape Quantization And Recognition With Randomized Trees. *Neural Comp.*, **9**(7), 1545-1588.

BAUER E. et KOHAVI R. [1999]. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, **36**(1-2), 105-139.

BELSON W.A. [1959]. Matching and prediction on the principle of biological classification. *Applied Statistics*, **8**(2), 65-75.

BLAKE C.L. et MERZ C.J. [1998]. *UCI Repository of machine learning databases* [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, consulté le 25/06/2002.

BLUMER A., EHRENFEUCHT A., HAUSSLER D. et WARMUTH M.K. [1987]. Occam's Razor. *Information Processing Letters*, **24**(6), 377-380.

BREIMAN L. [1996a]. Bagging predictors. *Machine Learning*, **24**(2), 123-140.

BREIMAN L. [1996b]. Heuristics of instability and stabilization in model selection. *Ann. Statist.*, **24**(6), 2350–2383.

BREIMAN L. [1996c]. Some properties of splitting criteria. *Machine Learning*, **24**(1), 41-47.

BREIMAN L. [1998]. Arcing classifiers. *Ann. Statist.*, **26**(3), 801-849.

BREIMAN L. [2000]. Randomizing Outputs to Increase Prediction Accuracy. *Machine Learning*, **40**(3), 229-242.

BREIMAN L. [2001]. Random forests. *Machine Learning*, **45**(1), 5-32.

BREIMAN L. [2002]. *Manual - Setting Up, Using, And Understanding Random Forests V3.1* [ftp://ftp.stat.berkeley.edu/pub/users/breiman/Using_random_forests_v4.0.pdf]., consulté le 01/10/2004.

BREIMAN L. [2003]. *Manual - Setting Up, Using, And Understanding Random Forests V4.0* [ftp://ftp.stat.berkeley.edu/pub/users/breiman/Using_random_forests_v4.0.pdf]., consulté le 01/10/2004.

BREIMAN L. et ADELE C. [2003]. randomForest: Breiman's random forest for classification and regression. R package, version 3.9-6. R port by Andy Liaw and Matthew Wiener.

BREIMAN L., FRIEDMAN J.H., OLSHEN R. et STONE C. [1984]. *Classification and Regression trees*. Belmont, CA, Wadsworth International Group.

BREMNER A.P. et TAPLIN R.H. [2002]. Modified classification and regression tree splitting criteria for data with interactions. *Austral. & New Zealand J. Stat.*, **44**(2), 169-176.

BRESLOW L.A. et AHA D.W. [1997]. Simplifying Decision Trees: A Survey. *Knowledge Engineering Review*, **12**, 1-40.

BUNTINE W. [1992]. Learning Classification Trees. *Statist. Comput.*, **2**, 63-73.

BUNTINE W. et NIBLETT T. [1992]. A further comparison of splitting rules for decision tree induction. *Machine Learning*, **8**(1), 75-85.

BUTTREY S.E. et KARO C. [2002]. Using k-nearest-neighbor classification in the leaves of a tree. *Comput. Stat. Data Anal.*, **40**(1), 27-37.

CESTNIK B. et BRATKO I. [1991]. *On estimating probabilities in tree pruning*. Fifth European Working Sessions on Learning. Springer-Verlag. 138-150.

CHOU P.A. [1991]. Optimal partioning for classification and regression trees. *IEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(4), 340-354.

CLEVELAND W.S. [1979]. Robust Locally Weighted Regression and Smoothing Scatterplot. *J. Amer. Stat. Assoc.*, **74**(368), 829-836.

DIETTERICH T.G. [2000]. An Experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization. *Machine Learning*, **40**(2), 139-157.

ESPOSITO F., MALERBA D. et SEMERARO G. [1997]. A comparative analysis of methods for pruning decision trees. *IEEE Trans. Pattern. Anal. Mach. Intell.*, **19**(5), 476-491.

FAYYAD U.M. et IRANI K.B. [1992]. *The attribute selection problem in decision tree generation*. Tenth National Conference on Artificial Intelligence, San Jose, CA. The AAAI Press / The MIT Press. 104-110.

FISHER R.A. [1925]. *Statistical methods for research workers*. Edinburg (UK), Oliver and Boyd, 239 p.

FIX E. et HODGES J.L. [1989]. Discriminatory Analysis - Nonparametric Discrimination: Consistency Properties. *Int. Stat. Rev.*, **57**(3), 238-247.

FRANK E. et WITTEN I.H. [1998]. *Using a Permutation Test for Attribute Selection in Decision Trees*. 15th International Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA. 152-160.

FREITAS A.A. [2001]. Understanding the crucial role of attribute interaction in data mining. *Artif. Intell. Rev.*, **16**(3), 177-199.

FREUND Y. et SCHAPIRE R.E. [1999]. A Short Introduction to Boosting. *J. Japanese Soc. Artificial Intelligence Res.*, **14**(5), 771-780.

FRIEDMAN J.H. [1977]. A Recursive Partitioning Decision Rule for Nonparametric Classification. *IEEE Trans. Comput.*, **26**(4), 404-408.

FRIEDMAN J.H., HASTIE T. et TIBSHIRANI R. [2000]. Additive Logistic Regression: a Statistical View of Boosting. *Ann. Statist.*, **28**(2), 337-407.

GLESER M.A. et COLLEN M.F. [1972]. Towards automated medical decisions. *Comput. Biomed. Res.*, **5**(2), 180-189.

GOODMAN L.A. et KRUSKAL W.H. [1954]. Measures of association for cross classifications. *J. Amer. Stat. Assoc.*, **49**(4), 732-762.

GUEGUEN A. et NAKACHE J.-P. [1988]. Méthode de discrimination basée sur la construction d'un arbre de décision binaire. *Rev. Stat. Appl.*, **XXXVI**(1), 19-38.

HASTIE T. et PREGIBON D. [1990]. *Shrinking trees.* Technical Report. AT&T Bell Laboratory, NJ. 21 p.

HO T.K. [1998]. The Random Subspace Method for Constructing Decision Forests. *IEEE Trans. Pattern. Anal. Mach. Intell.*, **20**(8), 832-844.

HYAFIL L. et RIVEST R.L. [1976]. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, **5**(1), 15-17.

IHAKA R. et GENTLEMAN R. [1996]. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, **5**(3), 299-314.

KASS G.V. [1980]. Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics*, **29**(2), 119-127.

KORS J.A. et HOFFMANN A.L. [1997]. Induction of decision rules that fullfill user-specified performance requirements. *Pattern Recognit. Lett.*, **18**, 1187-1195.

KOTHARI R. et DONG M. [2001]. Decision Trees For Classification: A Review and Some New Results. In: PAL S.R. et PAL A. (ed.). *Pattern Recognition: From Classical to Modern Approaches*, World Scientific, 1169-1184.

LEE T.C.M. [2001]. An introduction to coding theory and the two-part minimum description length principle. *Int. Stat. Rev.*, **69**(2), 169-183.

LI X. et DUBES R.C. [1986]. Tree classifier design with a permutation statistic. *Pattern Recognit.*, **19**(3), 229-235.

LOH W.-Y. et SHIH Y.-S. [1997]. Split Selection Methods for Classification Trees. *Statist. Sinica*, **7**, 815-840.

LÖPEZ DE MÄNTARAS R. [1991]. A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning*, **6**(1), 81-92.

MARTIN J.K. [1997]. An Exact Probability Metric for Decision Tree Splitting and Stopping. *Machine Learning*, **28**(2-3), 257-291.

MARTIN J.K. et HIRSCHBERG D.S. [1995]. *The Time Complexity of Decision Tree Induction.* Technical Report 95-27. Dept. of Information & Computer Science, University of California, Irvine. 26 p.

MATHEUS C.J. et RENDELL L.A. [1989]. Constructive induction on decision trees. In: *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, Morgan Kaufmann, 645-650.

MCCULLOCH W.S. et PITTS W.H. [1943]. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, **5**, 115-133.

MCSHERRY D. [1999]. Strategic induction of decision trees. *Knowledge-Based Systems*, **12**, 269-275.

MINGERS J. [1987]. Experts systems - rule induction with statistical data. *Journal of the Operational Research Society*, **38**(1), 39-47.

MINGERS J. [1989a]. An Empirical Comparison of Pruning Methods for Decision Tree Induction. *Machine Learning*, **4**(2), 227-243.

MINGERS J. [1989b]. An Empirical Comparison of Selection Measures for Decision-Tree Induction. *Machine Learning*, **3**(4), 319-342.

MOLA F. et SICILIANO R. [1997]. A fast splitting procedure for classification trees. *Statist. Comput.*, **7**(3), 209-216.

MORGAN J.N. et SONQUIST J.N. [1963]. Problems in the analysis of survey data, and a proposal. *J. Amer. Stat. Assoc.*, **58**(302), 415-434.

MUGGEO V.M.R. [2003]. Estimating regression models with unknown break-points. *Statist. Med.*, **22**(19), 3055 - 3071.

MUGGEO V.M.R. [2004]. segmented: functions to estimate break-points of segmented relationships in regression models. R package, version 0.1-4.

MURTHY S.K. [1998]. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Mining and Knowledge Discovery*, **2**(4), 345-389.

NAZAR K. et BRAMER M.A. [1998]. Concept dispersion, feature interaction and their effect on particular sources of bias in machine learning. *Knowledge-Based Systems*, **11**, 275-283.

NIBLETT T. et BRATKO I. [1987]. *Learning Decision Rules in Noisy Domains*. Expert Systems 86, Cambridge. Cambridge University Press.

OLIVER J.J. [1993]. *Decision Graphs - An Extension of Decision Trees*. Fourth International Workshop on Artificial Intelligence and Statistics. 343-350.

PAGALLO G. et HAUSSLER D. [1990]. Boolean feature discovery in empirical learning. *Machine Learning*, **5**(1), 71-99.

PALM R. [1994]. L'analyse discriminante décisionnelle : principes et application. *Notes de Statistique et d'Informatique* (Gembloux) 1994/4, 41p.

PARZEN E. [1962]. On estimation of a probability density function and mode. *Ann. Math. Statist.*, **33**(3), 1065-1076.

PAYNE H.J. et MEISEL W.S. [1977]. An algorithm for constructing optimal binary decision trees. *IEEE Trans. Comput.*, **26**(9), 905-916.

PAZZANI M.J., MERZ C., MURPHY P., ALI K., HUME T. et BRUNK C. [1994]. *Reducing Misclassification Costs*. 11th International Conference of Machine Learning, New Brunswick. Morgan Kaufmann. 217-225.

PÉREZ E. et RENDELL L.A. [1995]. *Using multidimensional projection to find relations*. 12th International Conference on Machine Learning, Palo Alto, CA. Morgan Kaufmann Inc. 447-455.

PÉREZ E. et RENDELL L.A. [1996a]. *Learning despite concept variation by finding structure in attribute-based data*. 13th International Conference on Machine Learning, San Francisco, CA. Morgan Kaufmann Inc. 391-399

PÉREZ E. et RENDELL L.A. [1996b]. *Statistical variable interaction: focusing multiobjective optimization in machine learning*. First International Workshop on Machine Learning, Forecasting and Optimization (MALFO'96), July 9-12, Universidad Carlos III de Madrid. Leganés, Madrid, Spain.

PÉREZ E., VILALTA R. et RENDELL L.A. [1996]. *On the Importance of Change of Representation in Induction*. Invited talk. Workshop of Inductive Learning for the 1996 Canadian Conference on Artificial Intelligence.

PRÉVOT H. [2004]. *Comparaison de méthodes statistiques et neuronales pour l'établissement d'équations de calibrage en spectrométrie de réflexion diffuse dans le proche infrarouge*. Thèse de doctorat, Faculté universitaire des Sciences agronomiques, Gembloux, 382 p.

QUINLAN J.R. [1986]. Induction of decision trees. *Machine Learning*, **1**(1), 81-106.

QUINLAN J.R. [1987]. Simplifying decision trees. *Int. J. Man-Machine Studies*, **27**, 221-234.

QUINLAN J.R. [1989]. *Unknown attribute values in induction*. Proceedings of the sixth international workshop on Machine learning, Ithaca, NY. Morgan Kaufmann. 164-168.

QUINLAN J.R. [1993]. *C4.5: Programs for Machine Learning*. Los Altos, CA, Morgan Kaufmann, 302 p.

QUINLAN J.R. et RIVEST R.L. [1989]. Inferring decision trees using the minimum description length principle. *Inform. and Comput.*, **80**(3), 227-248.

RAGAVAN H. et RENDELL L.A. [1993]. *Lookahead feature construction for learning hard concepts*. 10th International Conference on Machine Learning. 252-259.

RENDELL L.A. et CHO H. [1990]. Empirical learning as a function of concept character. *Machine Learning*, **5**(3), 267-298.

RENDELL L.A. et SESHU R. [1990]. Learning hard concepts through constructive induction: framework and rationale. *Computational Intelligence*, **6**, 247-270.

RISSANEN R. [1983]. A universal prior for integers and estimation by minimum description length. *Ann. Statist.*, **11**(2), 416-431.

ROSENBLATT M. [1956]. Remarks on some nonparametric estimates of the density function. *Ann. Math. Statist.*, **27**(3), 832-837.

ROUSSEEUW P.J. et LEROY A.M. [1987]. *Robust regression and outlier detections*. New York, Wiley, 329 p.

RUEY-HSIA L. [2001]. *Instability of decision tree classification algorithms*. PhD Thesis, University of Illinois, Urbana-Champaign, 86 p.

SAFAVIAN S.R. et LANDGREBE D. [1991]. A Survey of Decision Tree Classifier Methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, **21**(3), 660-674.

SAPORTA G. [1990]. *Probabilités, analyse de données et statistique*. Paris, Technip, 493 p.

SARKAR D. [2004]. lattice: Lattice Graphics. R package, version 0.10-14.

SEGAL M.R. [2004]. *Machine Learning Benchmarks and Random Forest Regression* [http://repositories.cdlib.org/cbmb/bench_rf_regn]. Center for Bioinformatics & Molecular Biostatistics, consulté le 12/10/2004.

SETIONO R. et LIU H. [1998]. Fragmentation problem and automated feature construction. In: *Proceedings of the 10th International Conference on Tools with Artificial Intelligence*, Taipei, Taiwan, 208-215.

SHANNON W.D. et BANKS D. [1999]. Combining classification trees using MLE. *Statist. Med.*, **18**(6), 727-740.

SHIH Y.-S. [1999]. Families of splitting criteria for classification trees. *Statist. Comput.*, **9**(4), 309-315.

SHIH Y.-S. [2001]. Selecting the best splits for classification trees with categorical variables. *Statist. Probab. Lett.*, **54**(4), 341-345.

SHLIEN S. [1990]. Multiple binary decision tree classifiers. *Pattern Recognit.*, **23**(7), 83-88.

SHLIEN S. [1992]. Nonparametric classification using matched binary decision trees. *Pattern Recognit. Lett.*, **13**, 83-87.

SONQUIST J.N. et MORGAN J.N. [1964]. *The detection of interaction effects.* Monograph 35. Survey Research Center, Institute for Social Research, University of Michigan.

TALMON J.L. [1986]. A multiclass nonparametric partitioning algorithm. *Pattern Recognit. Lett.*, **4**(1), 31-38.

TAYLOR P.C. et SILVERMAN B.W. [1993]. Blocks diagrams and splitting criteria for classification trees. *Statist. Comput.*, **3**(4), 147-161.

THERNEAU T.M. et ATKINSON E.J. [1997]. *An introduction to recursive partitioning using the RPART routines.* Technical Report 61. Mayo Clinic, Section of Statistics. 33 p.

THERNEAU T.M. et ATKINSON E.J. [2003]. rpart: Recursive Partitioning. R package, version 3.1-12. R port by Brian Ripley, S-PLUS 6.x original at http://www.mayo.edu/hsr/Sfunc.html.

TODOROVSKI L. et DZEROSKI S. [2003]. Combining Multiple Models with Meta Decision Trees. *Machine Learning*, **50**, 223-249.

TUKEY J.W. [1952]. *Allowances for Various Types of Error Rates.* Unpublished IMS adress, Chicago, IL.

UTGOFF P.E. et CLOUSE J.A. [1996]. *A Kolmogorov-Smirnoff Metric for Decision Tree Induction.* Technical Report 96-3. University of Massachusetts, Department of Computer Science, Amherst, MA.

VENABLES W.N. et RIPLEY B.D. [1997]. *Modern Applied Statistics with S-Plus. 2nd Ed.* New york, Springer, 548 p.

WALLACE C.S. et PATRICK J.D. [1993]. Coding decision trees. *Machine Learning*, **11**, 7-22.

WANG J.T.L. et ZHANG K. [2000]. Finding similar consensus between trees: an algorithm and a distance hierarchy. *Pattern Recognit.*, **34**(1), 127-137.

WEHBERG S. et SCHUMACHER M. [2004]. A Comparison of Non Parametric Error Rate Estimation Methods in Classification Problems. *Biom. J.*, **46**(1), 35-47.

WHITE A.P. et LIU W.Z. [1994]. Bias in information-based measures in decision tree induction. *Machine Learning*, **15**(3), 321-329.

WOLPERT D.H. et MACREADY W.G. [1999]. An Efficient Method To Estimate Bagging's Generalization Error. *Machine Learning*, **35**(1), 41-55.

ZHENG Z. [2000]. Constructing X-of-N Attributes for Decision Tree Learning. *Machine Learning*, **40**(1), 35-75.

GLOSSAIRE

Algorithme glouton
Algorithme qui recherche des solutions optimales locales pour approcher une solution globale

Arbre de décision
Structure hiérarchique arborescente, matérialisée par un graphe acyclique dont les vertex, appelés nœuds, portent des tests logiques et dont l'objectif est la prédiction d'une variable cible numérique ou catégorielle

Attribut descripteur ou **attribut**
Variable caractérisant un individu et servant de base au concept auquel appartient celui-ci

Attribut parasite
Attribut attaché à un individu n'apportant aucune information concernant le concept auquel appartient celui-ci

Bagging
Abréviation de *bootstrap aggregating*, méthode consistant à agréger les résultats plusieurs arbres de décision générés au départ d'un échantillon d'apprentissage randomisé par *bootstrap*.

Branche
Arc issu d'un nœud dit père d'un arbre de décision, correspondant à un des résultats possibles du test associé à ce nœud, et conduisant à un nœud fils

Cible

Ensemble des individus appartenant à une classe fixée, ou variable codant l'appartenance à une telle classe

Classement

Attribution d'une classe préexistante à un individu donné ; aussi appelé classification supervisée

Classification

Création de nouvelles classes par groupement d'individus présentant des caractéristiques semblables ; aussi appelé classification non supervisée

Concept

Ensemble de règles caractérisant l'appartenance d'un individu à une classe d'objets donnée

Convexité

Une fonction f est dite convexe si $\forall\ a, b, \exists\ \lambda \in [0, 1]$ tel que $f((1-\lambda)\ a + \lambda\ b) \leq (1-\lambda)\ f(a) + \lambda\ f(b)$

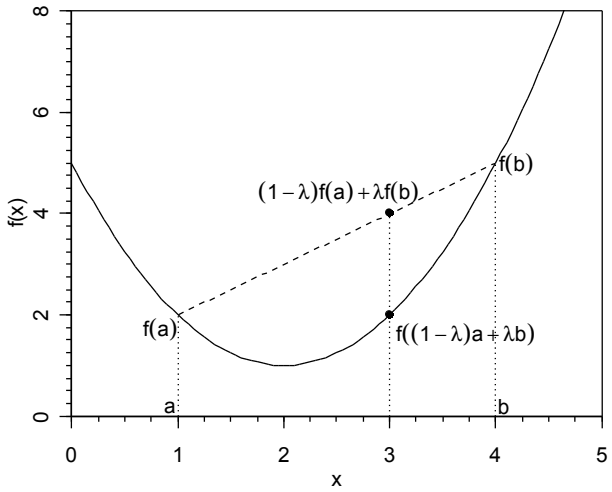


Figure 25. Illustration de la notion de convexité d'une fonction f sur $\Re$.

Critère de partition

Mesure de la pertinence d'une partition donnée dans le processus d'estimation de la variable cible

Echantillon d'apprentissage
Echantillon d'individus illustrant le concept, destiné à l'élaboration du modèle de prédiction

Echantillon test
Echantillon d'individus illustrant le concept, indépendant de l'échantillon d'apprentissage, et destiné à livrer une estimation de l'erreur en généralisation du modèle

Elagage
Extraction d'un sous-arbre par suppression d'une ou plusieurs branches d'un arbre entièrement développé

Feuille
Nœud terminal d'un arbre de décision, qui porte la prédiction associée aux individus qui lui appartiennent

Forêt aléatoire
Classificateur formé par l'agrégation par vote à la majorité simple des résultats de plusieurs arbres doublement randomisés par rééchantillonnage *bootstrap* et présélection aléatoire des attributs lors des partitions

Glouton
voir ***Algorithme glouton***

Impureté
Fonction convexe des probabilités de classe d'un nœud, qui est maximale lorsque toutes les classes sont équiprobables et minimale lorsqu'une classe unique a une probabilité égale à un.

OOB
voir ***Out-of-bag***

Out-of-bag (erreur…)
Taux d'erreur estimé dans les procédures d'agrégation d'arbres par *bagging* en utilisant comme échantillons tests les individus non repris dans les différents échantillons *bootstrap*

Racine
Point d'entrée d'un arbre de décision, contenant l'ensemble de l'échantillon d'apprentissage, et qui correspond au premier test

Rupture asymptotique
Zone correspondant à une valeur constante de la variable

dépendante dans une relation entre plusieurs paramètres ; se traduit par un modèle linéaire horizontal

Taux d'erreur apparent
voir ***Taux d'erreur en resubstitution***

Taux d'erreur de base
Taux d'erreur correspondant à l'attribution systématique de la classe majoritaire du concept

Taux d'erreur en généralisation
Espérance mathématique du taux d'erreur observé d'un estimateur lors de son application à un échantillon indépendant de celui d'apprentissage

Taux d'erreur en resubstitution
Taux d'erreur affiché par un prédicteur appliqué sur l'échantillon d'apprentissage qui a servi à le construire

TEG
voir ***Taux d'erreur en généralisation***

Variable parasite
voir ***Attribut parasite***