Université De Strasbourg
UFR Mathématiques-Informatique

Stage de Master II Biostatistiques et Statistiques Industrielles
du 16.02.15 au 14.08.15

Institut de Recherches Internationales Servier
Pôle d'expertise Méthodologie et Valorisation des Données
Maître de stage: F. Couvelard

La recherche de sous-groupes par Virtual Twins

François Vieille

Paris, le 31 août 2015

vieille.francois@gmail.com

Table des matières

Remerciements

Je remercie en premier lieu mon maître de stage, Mr Couvelard, pour m'avoir accueilli, pour son écoute et son enseignement.

Je remercie également Mr Guillier pour son aide et son soutien qui ont été d'une grande utilité.

Je remercie Mme Gabarroca pour son acceuil au sein de son département.

Je remercie bien évidemment l'ensemble du PEX méthodologie et valorisation des données pour leur accueil très convivial.

Je remercie Capucine Py et Marie Bessot pour la relecture attentive du présent document.

Je remercie l'ensemble de mes collègues du Master de Strasbourg pour leur soutien.

Enfin, je remercie particulièrement le Bench pour sa bonne humeur et sa bonne ambiance.

Introduction

Dans le cadre de la deuxième année de Master Mathématiques et applications : Biostatistiques et statistiques industrielles de Strasbourg, il nous est proposé d'effectuer une immersion professionnelle au travers d'un stage d'une durée de six mois. C'est dans l'optique de pouvoir élargir mes connaissances, d'appliquer mes compétences, de découvrir un champ d'application de la Statistique qui est le data mining et de me confronter à une certaine réalité professionnelle, que j'ai réalisé mon stage à l'Institut de Recherches Internationales Servier.

Servier est le premier laboratoire pharmaceutique français indépendant et rayonne sur le marché mondial par son expertise et son savoir-faire. Les laboratoires Servier proposent des médicaments issus de leur propre recherche. Dans le but de déposer un nouveau produit sur le marché, il est obligatoire d'effectuer des essais cliniques très réglementés où les biostatistiques jouent un très grand rôle.

Le design d'un essai clinique a pour objectif de fournir une information définitive à propos du traitement testé. Il est très fréquent que l'essai clinique compare un nouveau traitement par rapport à un traitement comparateur. Dans ce cas, la conclusion du test est applicable à l'ensemble de la population considérée dans l'essai clinique.

L'émergence des thérapies ciblées et de la médecine personnalisée entraînent, de la part des laboratoires pharmaceutiques, une volonté de rechercher des sous-populations de patients chez qui le nouveau produit serait plus efficace, ou à l'inverse, chez qui ce produit aurait moins d'impact, voire même un impact nocif. Cette recherche est motivée par le fait que le mécanisme d'action d'un nouveau produit n'agit pas à l'identique chez tous les patients.

D'un point de vue statistique la recherche exploratoire de sous-groupes post essai clinique est un exercice dangereux. La multiplicité des tests et le manque de puissance peuvent provoquer des résultats erronés et non fiables. Depuis quelques années diverses méthodes de recherches de sous-groupes ont vu le jour. L'équipe de data mining des laboratoires Servier utilise déjà certaines d'entre elles. Afin d'élargir le champ des possibilités, ils s'intéressent également à une méthode récente : *Virtual Twins* . Elle a été proposé et formalisé par Jared Foster en 2011 [1].

L'objectif de mon stage est de comprendre et soumettre *Virtual Twins* aux contraintes imposées par les essais thérapeutiques afin d'évaluer l'intérêt de cette méthode.

Il sera présenté dans un premier temps le contexte du stage, à savoir le groupe Servier et le déroulement des essais thérapeutiques. Nous présenterons également ce qu'est le data mining. Seront introduit dans un second temps les outils nécessaires pour comprendre le mécanisme que *Virtual Twins* utilise : les arbres CART et les forêts aléatoires. Enfin, nous présenterons la méthode *Virtual Twins* suivie d'une application et de simulations qui nous permettrons de proposer des conclusions sur la méthode.

Chapitre 1

Contexte du stage

1.1 Institut de Recherches Internationales Servier

1.1.1 Le groupe Servier

L'entreprise Servier a été créée en 1954 à Orléans par Jacques Servier (1922 - 2014) et neuf collaborateurs. Aujourd'hui, l'institut SERVIER est le premier groupe pharmaceutique indépendant français et le deuxième groupe pharmaceutique français au niveau mondial. En 2014, le chiffre d'affaire du groupe s'élève à 4 milliard d'euros, et 75% de ce chiffre est réalisé à l'international : 92% des médicaments Servier sont prescrits hors de France.

Depuis presque 60 ans, les laboratoires Servier poursuivent leur développement, aussi bien en France qu'à l'international, où sont consommés 90% de leurs médicaments. Le groupe est présent dans plus de 140 pays, représentant plus de 21 000 collaborateurs, dont 5 000 en France.

La mission des laboratoires Servier est de mettre au point de nouveaux médicaments. Pour ce faire, leurs activités concernent toutes les étapes de la vie du médicament : recherche, production, information des médecins en France et à l'international ...

Par ailleurs, en 1996, face à l'émergence du marché des génériques, le Dr Servier a créé la filiale *Biogaran*. Avec près de 25% de parts de marché, la filiale occupe aujourd'hui une place majeure sur ce marché.

1.1.2 La recherche et le développement

Tous les médicaments mis sur le marché par les laboratoires Servier sont issus de sa propre recherche, et avec 30 médicaments innovants en 30 ans reconnus par tous les praticiens et vendus dans le monde entier, le laboratoire a démontré son savoir-faire et son expertise pharmacologique.

Près de 27% d'un chiffre d'affaires annuel du groupe est consacré à la recherche et au développement avec près de 3 000 collaborateurs dans ce secteur (R&D) et 20 Centres Internationaux de Recherche Thérapeutique (CIRT).

1.1.3 Les différents axes thérapeutiques

Les laboratoires Servier sont titulaires de plus de 27 000 brevets déposés pour plus de 50 000 molécules synthétisées. Le groupe s'assure un pôle de recherche dynamique et important dont les principales aires thérapeutiques sont :

- Cancérologie
- Pathologies cardiovasculaires (Cardiopathie ischémique, Artéro-thrombose, Hypertension artérielle, Insuffisance veineuse)
- Maladies métaboliques (Diabète)
- Pathologies du système nerveux central (Maladie neuro-dégénératives : Alzheimer, Parkinson. Maladie psychiatriques : dépression, anxiété, psychose)

- Rhumatologie

1.1.4 Pôle d'expertise Méthodologie et Valorisation des Données

Le pôle d'expertise *Méthodologie et Valorisation des Données* est au sein de la R&D et se compose des missions suivantes :

- Optimiser les méthodologies relatives aux études cliniques et pharmaco-épidémiologiques.
- Participer à la conception d'études cliniques et pharmaco-épidémiologiques, réaliser l'analyse des données issues de ces études en se basant sur des méthodes statistiques approuvées par les autorités de santé, procéder à l'interprétation des résultats et à la production de rapports d'études cliniques et d'analyses intégrées.
- Évaluer l'acceptabilité des produits du laboratoire dans une dimension populationnelle , détecter les signaux et évaluer les risques.
- Développer une expertise biostatistique et pharmaco-épidémiologique.

Le stage s'est déroulé au sein du département Data Mining (bleueté sur la Figure 2), dirigé par C. Gabarroca. Dans la suite, nous parlerons en détails de l'introduction du Data Mining au sein du laboratoire Servier. Cependant les objectifs du Data Mining sont :

- L'identification des meilleurs patients répondeurs.
- Caractériser les profils de patients à risque.
- Exploration des études non-positives.
- La recherche de sous-groupes.

1.2 Les essais thérapeutiques

1.2.1 Introduction

Dans la phase de développement d'un médicament, qui va de la découverte de la molécule à la mise sur le marché, la période entre la naissance de la substance chimique et le moment où un médicament peut être prescrit à un patient varie souvent entre sept et douze ans.

L'essai doit être *éthique* : il doit, en particulier, être précédé d'études chez l'animal dont l'un des buts est de sécuriser le passage aux essais cliniques, chez l'homme, ainsi que respecter certaines exigences réglementaires. Il doit également avoir une *valeur scientifique* (intérêt médical ou de santé publique) et faire l'objet d'une *méthodologie rigoureuse*. Enfin, l'essai doit respecter la loi sur la protection des personnes. En France, les essais cliniques ont fait l'objet d'une loi en 1988 qui a été révisée récemment : la loi Huriet dont les trois objectifs principaux sont de protéger les personnes qui se prêtent aux recherches biomédicales, de définir un ensemble de bonnes pratiques cliniques afin de renforcer la qualité scientifique et d'aider au développement européen et international des industries de santé.

La construction d'un essai thérapeutique est soumise à des contraintes nombreuses et de natures différentes. Le but de ces essais est d'obtenir l'autorisation de mise sur le marché (AMM) d'un produit, délivrée par les agences du médicament. Cela nécessite donc une analyse statistique rigoureuse conforme à la planification et adaptée à la nature des données.

1.2.2 Le développement pré-clinique

Dans un premier temps, des études sont réalisées chez l'animal, afin d'étudier les propriétés pharmacocinétiques, pharmacodynamiques et toxicologiques de la molécule. C'est au cours de cette étape que seront étudiés entre autres, les effets secondaires, la toxicité, les conséquences sur la reproduction et sur le développement éventuel de tumeurs. Les données recueillies sont indispensables au bon déroulement de l'étude car, au-delà de la phase pré-clinique, l'administration se fait sur l'homme.

1.2.3 Les études de phase I

Ces études ont pour objectif de définir la tolérance humaine du médicament, la dose maximale tolérée et, si possible, la posologie entraînant les premiers effets indésirables ainsi que celle entraînant les premiers effets pharmacodynamiques souhaités. Ces études sont le plus souvent réalisées chez quelques dizaines de sujets volontaires sains, sauf en oncologie où les patients sont tous des volontaires malades, en fin de vie, pour lesquels tous les autres traitements ont échoué.

1.2.4 Les études de phase II

Les études de phase II sont des études exploratoires qui visent à déterminer l'efficacité pharmacologique du médicament. Elles permettent de recueillir des données de tolérance de pharmacocinétique et de relation dose-effet, mais aussi de réaliser l'étude au sein de sous-groupes de populations fragiles (personnes âgées, insuffisants rénaux, hépatiques ...)

1.2.5 Les études de phase III

Les études de phase III sont des études confirmatoires qui permettent de prouver l'efficacité thérapeutique du médicament dans les différentes indications revendiquées. L'effet thérapeutique du produit est étudié sur un groupe de patients homogène souffrant de la maladie à traiter. Il est mesuré sur un paramètre précis et comparé par rapport à un placebo ou à un traitement de référence. On assigne aléatoirement le traitement de contrôle, on parle alors d'essais randomisés contrôlés. Chaque essai (quelques centaines à quelques milliers de personnes) est organisé selon un protocole précis : nombre de patients à inclure calculé à l'avance, essai conduit en double aveugle.

1.2.6 Autorisation de Mise sur le Marché (AMM)

L'AMM est l'une des dates les plus importantes dans la vie du médicament. Elle est nécessaire à la vente de toute nouvelle molécule, ainsi qu'à l'extension d'indication d'un produit déjà sur le marché. Le dossier doit être déposé à l'ANSM (Agence Nationale de Sécurité du Médicament et de produits de santé) pour une procédure nationale, à l'EMEA (European Agency for Evaluation of Medical products) pour une procédure Européenne et à la FDA (Food and Drug Administration) pour une AMM aux Etat-Unis. Ce sont ces agences qui délivrent l'autorisation.

1.2.7 Les études de phase IV

Les études de phase IV ou de pharmacovigilance, ont lieu après l'AMM du médicament. Elles permettent de mieux évaluer l'utilité réelle et la tolérance du nouveau médicament au sein d'une population plus large et plus représentative dans des conditions usuelles de prescription et de suivi.

Chapitre 2

Le Data Mining

Le data mining, que l'on peut traduire littéralement par *fouille de données*, est l'application des techniques de statistique, d'analyse de données et d'apprentissage automatique à l'exploration et l'analyse sans *a priori* de grandes bases de données informatiques, en vue d'en extraire des informations nouvelles et utiles pour le détenteur de ces données.

L'essentiel des informations présentées ci-dessous sont tirées de l'excellent ouvrage de Stéphane TUFFERY [2], responsable statistique dans un grand groupe bancaire français, également enseignant à l'université de Rennes 1. Son livre est un formidable recueil des techniques de data mining et de leurs utilisations.

2.1 L'histoire du Data Mining

Il est naturel de se demander ce qui différencie la statistique dite classique du data mining, quelle est sa spécificité et qu'est-ce qu'apporte le data mining à des problématiques auparavant traditionnellement traitées par la statistique. Pour mieux comprendre cette discipline, on distingue trois grandes ères.

Depuis la fin du XIX^{e} siècle jusqu'aux années 1950, c'est la statistique classique qui règne. L'informatique n'a pas encore été inventé, les moyens de calculs sont donc manuels et très limités. la statistique est alors caractérisée par de petits volumes étudiés, de l'ordre de quelques centaines d'individus. Les variables étudiées sont recueillies selon un protocole spécial (échantillonnage, plan d'expérience...). Les modèles sont issus de la théorie des probabilités et sont confrontés aux données : de fortes hypothèses existent sur les lois statistiques suivies (e.g. linéarité, normalité, homoscédasticité).

Les données sont collectées et analysées dans un cadre strict, souvent scientifique, en vue de vérifier une théorie, laquelle peut être réfutée par le résultat d'un test. Les fondements de la statistique mathématique, mais aussi d'importantes méthodes prédictives remontent à cette période, telles que la régression logistique (Joseph Berkson, 1944).

A partir des années 1960, l'apparition du calcul informatique révolutionne la discipline, en permettant des calculs bien plus complexes et surtout rapides. Il est alors possible de s'occuper de plus gros volumes de données, allant jusqu'à quelques milliers d'individus et quelques dizaines de variables. La représentation visuelle des données prend également une place importante.

C'est une période de grande créativité théorique, pendant laquelle sont inventées de nombreuses méthodes fondamentales encore très utilisées aujourd'hui. L'analyse de données (Jean-Paul Benzécri et John Wilder Tukey) est en plein essor, notamment l'analyse factorielle.

C'est à partir des années 1990 que le data mining prend son envol. Cette période n'est pas seulement caractérisée par une explosion des ressources informatiques et de la quantité de données à traiter, mais aussi par une profonde évolution du rôle de l'analyse quantitative. C'est dans cette évolution que se différencie le data mining de la statistique, même s'il reprend les outils de cette dernière, son rôle et ses applications diffèrent : les données traitées possèdent plusieurs millions ou dizaines de millions d'individus, avec plusieurs centaines ou milliers de variables. Ces variables

peuvent être aussi bien numériques, textuelles et même parfois contenant des images.

Les modèles sont issus des données et on tente parfois d'en tirer des éléments théoriques, il y a de faibles hypothèses sur les lois statistiques suivies.

Les données sont parfois imparfaites, avec des erreurs de saisie, de codification, des valeurs manquantes. De plus, les données sont recueillies avant l'étude, et souvent à d'autres fins.

L'utilisation du data mining se déroule souvent en entreprise, où il est nécessaire de faire des calculs rapides et parfois en temps réel.

Un modèle de data mining peut être vu comme une application informatique, et nécessite alors une connaissance technique des outils informatiques à utiliser.

Les méthodes utilisées sont empruntées à la statistique, mais aussi à la théorie de l'information et à l'apprentissage automatique (*machine learning*).

Depuis ces dernières années, dans le monde des entreprises les termes *data mining* ou *data science* sont particulièrement virulent. En effet, ces techniques qui en premier lieu ne servaient qu'au cercle fermé de la science, ont su s'exporter rapidement au sein des entreprises : c'est le passage d'une économie de demande à une économie de l'offre. La pression concurrentielle, les nouvelles attentes des consommateurs, ainsi que parfois les nécessités réglementaires ont largement contribués à l'expansion du data mining.

2.2 Les secteurs d'activité et les applications

Cette section est à titre informative, et ne prétend pas être exhaustive.

Le data mining est présent dans la gestion de la relation client dans le but d'identifier des prospects les plus susceptibles de devenir clients, identifier des clients les plus rentables. L'idée est de chercher le meilleur taux de réponse lors des campagnes marketing. L'intérêt peut être aussi pour trouver les meilleures implantations pour les agences d'une banque ou les établissements d'une chaîne de magasins, en déterminant des profils de magasins en fonction de leur localisation et des chiffres d'affaires générés par leurs différents rayons.

On retrouve également le data mining dans le domaine du marketing stratégique : aide à la création de promotions, à la conception de nouveaux produits, à la découverte d'associations inattendues de produits. De manière générale le data mining est employé pour une meilleure compréhension de la clientèle, en vue de l'adaptation de la communication et de la stratégie commerciale de l'entreprise.

Dans le domaine de la gestion du risque, et la détection de fraude, l'apport du data mining n'est pas négligeable. En astrophysique, le data mining est utile pour la reconnaissance de motifs ou de formes, notamment pour classer en étoile ou galaxie un nouveau corps céleste découvert au télescope.

Évidemment, le secteur médical est un grand utilisateur de statistique, ainsi le data mining s'y est aussi développé. Parmi les premières applications, on rencontre la détermination de groupes de patients susceptibles d'être soumis à des protocoles thérapeutiques déterminés. On y trouve aussi la recherche des facteurs de décès ou de survie dans certaines pathologies, à partir de données recueillies lors des essais cliniques, dans le but de choisir le traitement le plus approprié en fonction de la pathologie du patient.

D'autres secteurs comme les télécoms, l'industrie automobile, la vente par correspondance, l'industrie agro-alimentaire sont touchés par la fouille de données.

2.3 Le Data Mining en développement clinique

Les Laboratoires Servier ont ressenti le besoin d'utiliser les méthodes de Data Mining pour de nombreuses raisons.

Tout d'abord, les bases de données des essais cliniques sont parfaitement adaptées pour l'analyse : les données sont de bonne qualité (peu de données manquantes, homogénéité des réponses pour tous les individus...) et très riches (avec de nombreuses variables de types hétérogènes). Le fonctionnement des essais cliniques avec une méthodologie extrêmement rigoureuse, permet d'accumuler une grande quantité de données de qualité, et qui ultérieurement sont simplement (le cas le plus souvent) stockées dans les bases des données.

De plus bien que présentant des volumétries sans comparaison avec celles traitées en marketing, les résultats d'essai clinique contiennent de l'information non triviale, potentiellement utile. Actuellement, ces données sont peu ou sous-exploitées. Les plans d'analyse statistique mis en place n'ont pas pour but de fouiller les données, mais de s'en tenir au cadre strict de l'étude. Le Data Mining est une bonne alternative afin d'explorer les données et y trouver des pistes solides à investiguer d'avantages sur de nouvelles études.

David Hand, professeur au Imperial College de Londres, définit ainsi le Data Mining comme étant un *process of secondary analysis of large databases aimed at finding unsuspected relationships which are of interest or value to the database owners.*

Cette citation permet d'appuyer le fait que le Data Mining dans le développement clinique aide à transformer l'information, sous forme de données, en connaissance : il s'agit donc d'extraire de l'information ou générer des hypothèses utiles dans la compréhension clinique des résultats, d'identifier des profils de patients, de comprendre l'échec d'un essai ou préparer d'autres essais.

Il est important de comprendre le rôle complémentaire du Data Mining concernant les essais cliniques.

La Biostatistique intervient pour définir la procédure à employer dans le but de *confirmer* avec un niveau de certitude fixé l'efficacité du médicament codée sous forme d'hypothèse statistique.

A l'opposé, le Data Mining n'intervient que dans le but d'*explorer* les données, qui initialement ne sont pas collectées dans cet objectif. Le Data Mining permet d'extraire l'information de ces données sans hypothèses à priori. Cette exploration permettra de générer des hypothèses qui pourront nécessiter des études complémentaires. Comme nous le verrons plus tard, la recherche de sous-groupes rentre dans ce cadre.

Depuis la création de la cellule Data Mining, de nombreux projets ont été réalisés. La cellule a évolué et a pris de l'ampleur ; aujourd'hui, elle est rattachée à la Biostatistique et est constituée de deux-trois personnes.

Chapitre 3

Les arbres de décision CART

3.1 Introduction

Dans ce chapitre, il sera présenté les arbres de décision dit arbres CART, pour Classification And Regression Tree. La technique de l'arbre de décision est l'une des plus intuitives et des plus connues du data mining. Elle répond à des problématiques de classification supervisées aussi bien dans le cadre descriptif que prédictif. Seul la regression logistique reste concurrente à l'arbre de décision.

La méthode CART a été pensée et fondée par Leo Breiman, Jerome H. Friedman, Richard A. Olshen et Charles J. Stone et formalisée dans le livre *Classification And Regression Trees* [3]. Ce chapitre est essentiellement inspiré par cet ouvrage qui détaille au mieux la méthode.

Dans ce chapitre, une première section sera consacrée à la formalisation des problèmes de classification. Une seconde section présentera les méthodes de validation majeures. Enfin la dernière section portera sur la présentation théorique de la méthode et des divers aspects.

3.2 Classifieurs

Utilisée pour la reconnaissance de chiffre, pour nous céder un prêt à la banque, pour éviter les spam dans nos boîtes mails ou encore pour déterminer un patient à risque pour une certaine maladie, la classification se glisse dans tous les domaines. Le concept est d'utiliser des mesures sur un objet ou un individu pour chercher à prédire à quelle classe appartient l'objet ou l'individu.

Les techniques de classification ont souvent un but similaire, à savoir décrire les observations passées et prédire de futures observations.

Par exemple, à Los Angeles, les jours sont classifiés par rapport à leur niveau d'Ozone :

- class 1 : *non alert* (peu d'ozone)
- class 2 : *first-stage alert* (niveau modéré d'ozone)
- class 3 : *second-stage alert* (niveau élevé d'ozone)

Les données récueillies sont des mesures météorologiques, comme la température, l'humidité, les conditions atmosphériques et le niveau d'un certain nombre de polluants.

Les classifieurs ne sont pas construits par *hasard*. Ils sont basés sur une expérience passée. Par exemple, à Los Angeles, un jour chaud avec un haut taux de pollution a de fortes chances d'être suivi par un autre.

Plus formellement, dans la construction d'un classifieur, l'expérience passée est résumée dans ce qu'on appelle un échantillon d'apprentissage (*learning sample*). Cet échantillon est construit de N observations passées d'un phénomène que l'on rapporte sous forme de différentes mesures. Cet échantillon possède également la classification correcte de l'observation.

Remarque. *Lorsqu'on cherche à créer un classifieur sur un échantillon d'apprentissage qui possède déjà la classification exacte de l'observation, on parle de classification supervisée.*

L'échantillon d'apprentissage de la classification d'ozone contenait six ans de relevés de 400 mesures quotidiennes dans 30 lieux différents.

Définition 3.1. *Un échantillon d'apprentissage est définit comme tel :*
Soit une suite de variables aléatoires $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ *qui sont les réalisations i.i.d. d'une variable aléatoire* (X, Y)
(X, Y) *est régis par une loi inconnue, notée* $\mathbb{P}_{(X,Y)}$*, définis sur l'espace* $\mathcal{X} \times \mathcal{Y}$ *et l'on observe plusieurs réalisations. On a alors :*

- $\forall i = 1, \ldots, n\ X_i = (X_i^{(1)}, \ldots, X_i^{(p)}) \in \mathcal{X} \subset \mathbb{R}^p$
- $\forall i = 1, \ldots, n\ Y_i$ *prend ses valeurs dans* $\{1, \ldots, C\} := \mathcal{Y}$*, où* C *est le nombre de classes*
- *On note l'échantillon d'apprentissage* $\mathcal{L}_n$

Remarque. *Si nous cherchons à déterminer la survenue ou non d'une maladie, on a alors* $Y \in \{0, 1\}$*,* $Y = 1$ *si l'individu est malade et* $Y = 0$ *sinon.*

Définition 3.2. *Un classifieur est donc une application, notée* h*, de* $\mathcal{X}$ *dans* $\{0, \ldots, C\}$*,* $\forall X \in \mathcal{X}$ *on lui attribut la classe* $h(X)$*.*

Dans le cadre d'une classification supervisée, h *dépend de l'échantillon test* $\mathcal{L}$*, on a alors* $\forall X \in \mathcal{X}$ *on attribut la classe* $h(X, \mathcal{L})$*. Dans la littérature, l'écriture est raccourcie en* $h(X)$*. Enfin pour spécifier le nombre d'observations* n *dans l'échantillon d'apprentissage, il est naturel d'écrire le classifieur* $h_n(X, \mathcal{L}_n)$*.*

Lors d'une classification via l'application $h(.)$, il est tout à fait possible que ce dernier fasse des erreurs de classification. C'est pourquoi on associe à un classifieur son *erreur de classification*, on parle aussi de *risque*.

Définition 3.3. *L'erreur de classification est donnée par*

$$R(h) = \mathbb{P}(h(X) \neq Y | \mathcal{L}_n)$$

Définition 3.4.

On appelle h^* *le classifieur de Bayes associé au risque* R^* *tel que*

$$R^* = \min_h R(h)$$

Définition 3.5.
On définit $\pi(j) = \mathbb{P}(Y = j)\ \forall j = 1, \ldots, C$
On définit également la loi de $X|Y$*,* $\forall A \subset \mathcal{X}$ *et* $\forall j = 1, \ldots, C$*, on a* $\mathbb{P}(X \in A | Y = j) = \frac{\mathbb{P}(X \in A, Y = j)}{\pi(j)}$*.*
On lui associe une densité $f_j(.)$ *telle que*

$$\mathbb{P}(X \in A | Y = j) = \int_A f_j(x) dx$$

Théorème 1. *On définit alors explicitement* h^* *comme*

$$\forall x \in \mathcal{X}, h^*(x) = arg\max_i f_i(x)\pi(i)$$

Et le risque de Bayes, ou erreur de Bayes

$$R^* = 1 - \int \max_j (f_j(x)\pi(j)dx)$$

Demonstration. *Soit h un classifieur quelconque, on a alors*

$$\begin{aligned}\mathbb{P}(h(X)=Y) &= \sum_{j=1}^{C}\mathbb{P}(h(X)=Y|Y=j)\pi(j)\\ &= \sum\int_{h(X)=Y} f_j(x)\pi(j)dx\\ &= \int\left[\sum_{j=1}^{C}\mathbf{1}_{\{h(x)=j\}}f_j(x)\pi(j)\right]dx\end{aligned}$$

Et pour une valeur fixe x, on a :

$$\sum_{j=1}^{C}\mathbf{1}_{\{h(x)=j\}}f_j(x)\pi(j) \leq \max_j(f_j(x)\pi(j))$$

Alors :

$$\begin{aligned}\mathbb{P}(h(X)=Y) &= \int\left[\sum_{j=1}^{C}\mathbf{1}_{\{h(x)=j\}}f_j(x)\pi(j)\right]dx\\ &\leq \int\max_j(f_j(x)\pi(j))dx\\ &\leq \mathbb{P}(h^*(X)=Y)\end{aligned}$$

Remarque. *Dans le cas où $(Y)=\{0,1\}$, on a :*

$$h^*(x)=\begin{cases}1 & si \quad \mathbb{P}(Y=1|X=x)>0.5\\ 0 & si \quad \mathbb{P}(Y=1|X=x)<0.5\end{cases}$$

Définition 3.6. *Soit $(h_n)_{n\in\mathbb{N}}$ une suite de classifieur d'un même phénomène. h_n est dit consistant si $R(h_n)\to^{\mathbb{P}} R^*$*

3.3 Méthodes de validation

On définit alors $R(h_n)$, noté aussi R, comme la *vraie* erreur de classification de h_n (cf section 3.2). Cependant, dans des problématiques concrètes cette quantité est inconnu car la loi de la variable aléatoire (X,Y) est inconnu. Seul $\mathcal{L}_n$ est à notre disposition afin de construire le classifieur h_n et estimer R.

Trois types majeurs d'estimations sont possibles.

3.3.1 Par resubstitution

Une fois que le classifieur h_n est construit, on réutilise $\mathcal{L}_n$ pour calculer la proportion de cas mal classifié. On note cette proportion R^{resub} tel que,

$$R^{resub}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{\{h_n(x_i)\neq y_i\}}$$

En utilisant le même échantillon pour estimer R et construire h_n cela provoque un problème. En effet, toutes les méthodes classification cherche à minimiser $R^{resub}(h_n)$, donc utiliser cette quantité pour estimer R peut être biaisé de façon optimiste. On parle de sur-apprentissage, plus courant en anglais sous le nom de *overfitting*. Ce risque apparaît lorsque le classifieur explique trop en détails les données, qui ne sont qu'un échantillon de la population.

Il sera alors préférable d'utiliser une autre méthode de validation.

3.3.2 L'échantillon test

La seconde méthode consiste à découper l'échantillon de base $\mathcal{L}_n$ en deux échantillons $\mathcal{L}_{n_1}$ et $\mathcal{L}_{n_2}$ où n_i est le nombre d'observations dans $\mathcal{L}_{n_i}$. h est construit à partir de l'échantillon $\mathcal{L}_{n_1}$, dénommé l'échantillon d'apprentissage, plus connu en anglais sous le terme *train set*. Puis, $\mathcal{L}_{n_2}$, appelé échantillon test (*test set*), est utilisé pour estimer R

On écrit alors

$$R^{ts}(h_n) = \frac{1}{n_2} \sum_{(x,y)\in\mathcal{L}_{n_2}} \mathbf{1}_{\{h(x)\neq y\}}$$

Remarque. *L'échantillon test doit être issu de la même distribution que les réalisations de l'échantillon d'apprentissage et toutes les observations doivent être indépendantes tout échantillon confondu.*

Il est habituel de scinder l'échantillon de base en un tiers, deux tiers. Les deux tiers définissant l'échantillon d'apprentissage, et le dernier tiers l'échantillon test. Donc de toute évidence, l'un des inconvénients de cette méthode est de réduire l'échantillon d'apprentissage. Si l'échantillon de base est grand, cela ne pose pas de problème majeur.

3.3.3 Validation croisée k-fold

Plus connu sous son nom anglais *k-fold cross validation*, cette méthode consiste à découper aléatoirement l'échantillon $\mathcal{L}_n$ en k échantillons, de même effectif de préférence, puis d'appliquer la méthode de l'échantillon test k fois. On note les k échantillons $\mathcal{L}_1, \ldots, \mathcal{L}_k$. Pour chaque $m = 1, \ldots, k$, nous utilisons l'échantillon d'apprentissage $\mathcal{L}_n - \mathcal{L}_m$, i.e. les observations de $\mathcal{L}_n$ qui ne sont pas dans $\mathcal{L}_m$, sur lequel nous construisons le classifieur h_n^m. Et puisque aucune observation de $\mathcal{L}_m$ n'est utilisée dans la construction de h_n^m, $\mathcal{L}_m$ est utilisé comme échantillon test pour estimer $R(h_n^m)$, que l'on note R^{ts} tel que :

$$R^{ts}(h_n^m) = \frac{1}{N_m} \sum_{(x,y)\in\mathcal{L}_m} \mathbf{1}_{\{h(x)\neq y\}}$$

où N_m est l'effectif de $\mathcal{L}_m$. La procédure est répétée k fois, de telle sorte que chaque $\mathcal{L}_m$ soit exactement utilisé une fois comme échantillon test. On définit alors une nouvelle estimation de R, par :

$$R^{cv}(h_n) = \frac{1}{k} \sum_{m=1}^{k} R^{ts}(h_n^m)$$

La validation croisée k-fold a cette particularité d'utiliser l'ensemble des observations pour estimer une quantité donnée.

Remarques. *La valeur de k est traditionnellement choisie à 10, mais ça n'est pas une règle.*
Si $k = n$, i.e. l'effectif de l'échantillon initial, cette méthode consiste à construire l'échantillon d'apprentissage sur l'ensemble de l'échantillon initiale, excepté une observation. Ceci est appelé leave-one-out.

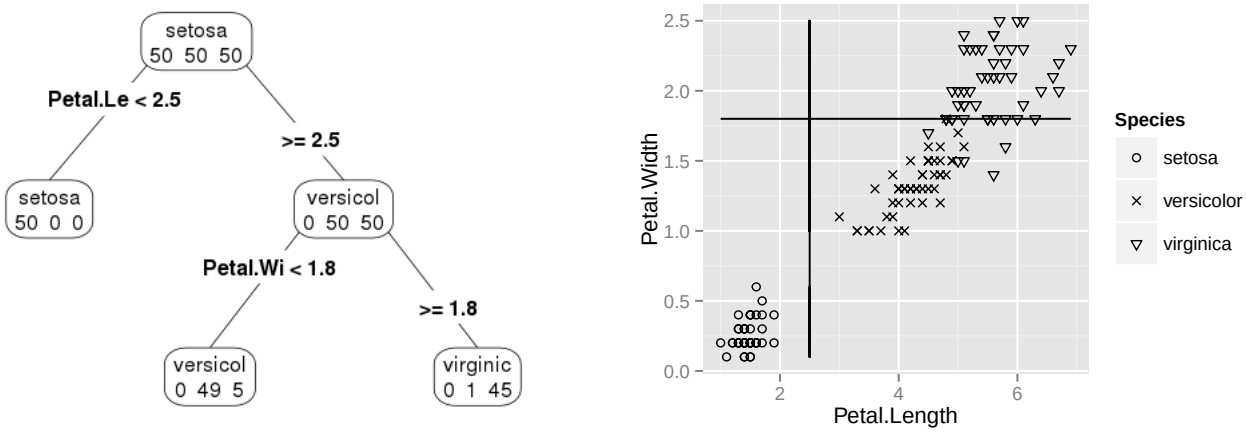
Pour conclure avec les méthodes de validation, les trois décrites ci-dessus ne sont pas les seules, mais elles ont le mérite d'être largement utilisées. D'autre part, dans ce rapport, elles ont été présentées dans le cadre d'estimation du risque d'un classifieur, naturellement, elles peuvent être utilisées pour une infinité d'autres modèles.

3.4 Les arbres de classification CART

L'arbre de classification CART est un classifieur par partitionnement récursif, l'arbre construit est dit binaire. L'objectif de la méthode est subdiviser l'espace de départ des mesures afin de créer des règles permettant d'assigner une classe aux observations respectant ces règles. On parle alors de partitionnement. De plus, ce partitionnement est dit récursif car la méthode pour subdiviser s'applique de manière récursive à chaque subdivision. Enfin, l'arbre construit se dit binaire car à chaque étape, l'espace est séparé en deux sous espaces.

Bien que la méthode n'ait pas été présentée en toute rigueur, il est nécessaire d'avoir une vision de ce qu'elle donne afin de mieux suivre la suite de ce chapitre. Pour cela, nous utiliserons le bien connu dataset *iris* du package `datasets` de `R` [4].

"This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica." [5]



(a) Arbre du dataset *iris* (b) Distribution des *Species*

FIGURE 3.1 – Visualisation de l'arbre de décision

La figure 3.1 (a) représente l'arbre de décision effectué sur le dataset *iris*. En haut de l'arbre, ce qu'on appelle la racine de l'arbre, nous retrouvons l'ensemble des observations avec les 3 espèces de fleurs confondues de même effectif, arbitrairement ce noeud (i.e. cette case de l'arbre) classifie l'ensemble des espèces comme `setosa`. Ensuite, l'algorithme initie le partitionnement à la recherche du meilleur seuil parmi toutes les variables explicatives afin de séparer au mieux les classes entre elles.

`Petal.Length` (i.e. la longueur de la pétale) satisfait ce critère de partitionnement. Une coupure est faite à 2.5 cm de cette variable. On parle également de *split*. Cela signifie que l'ensemble des fleurs satisfaisant `Petal.Length` < 2.5 cm tombent dans le noeud gauche du split. Les autres fleurs vont dans le noeud droit du split.

Le noeud gauche classifie les observations en `setosa`, en effet aucune autre espèce n'est présente dans ce noeud, le noeud est dit pur. Cela signifie également que l'algorithme s'arrête pour cette *branche*, i.e. cette partie de l'arbre.

Le noeud droit possède 50 espèces `versicolor` et 50 espèces `virginica`, ce noeud classifie arbitrairement les observations en `versicolor`. L'algorithme étant récursif, il continue à chercher un seuil séparant au mieux ces deux classes.

Le seuil est 1.8 cm pour `Petal.Width`. En effet, ce split crée deux nouveaux noeuds, qui dif-

férencient les deux espèces. Le noeud gauche possède 49 espèces `versicolor` et 5 `virginica`, il n'est donc pas pur, cependant l'ensemble des fleurs de ce noeud sont classifiées `versicolor`. C'est ce qu'on appelle le vote majoritaire.

Il en est de même pour le noeud de droite. Le partitionnement s'achève à cette étape, l'arbre est donc de profondeur deux.

la figure 3.1 (b) représente les espèces en fonction la longueur et de la largeur de leurs pétales. On remarque que les règles qu'établit l'arbre de décision sont particulièrement pertinentes. Ces règles permettent à la fois d'expliquer les données, mais aussi d'établir de futures prédictions. Par exemple si l'on possède une nouvelle observation qui a pour particularité d'avoir la longueur de ses pétales supérieur à 3cm et de largeur inférieur à 1.5 cm, alors elle sera classifiée comme `versicol`. La probabilité de mal la classifier sachant ces deux mesures est de $1 - 49/54$. La probabilité de mal classifié une observation sans connaître ses mesures est de $6/150 = 0.04$.

Dans la suite, nous formaliserons la construction de l'arbre, en particulier la recherche du meilleur split, les critères d'arrêt et l'élagage de l'arbre.

3.4.1 Croissance de l'arbre

Soit $\mathcal{L}_n$ un échantillon d'apprentissage de taille n. Pour rappel, $\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$, ce sont les réalisations i.i.d. de la variable d'intérêt (X, Y). $Y \in \{1, \ldots, C\}$ et $X \in \mathbb{R}^p$. Nous verrons plus tard comment faire lorsque des variables explicatives sont qualitatives.

Quelques notations :

- On note T l'arbre de classification
- t un noeud de l'arbre
- On définit $\tilde{T}$ comme étant l'ensemble des noeuds (ou des feuilles) de l'arbre
- t_R (resp. t_L) le noeud droit (resp. gauche) de t
- $p(t)$ est la probabilité qu'une observation tombe dans le noeud t
- p_R (resp. p_L) la probabilité qu'une observation tombe dans le noeud droit (resp. gauche) sachant qu'elle était dans t
- $p(j|t)$ est la probabilité d'obtenir la classe j sachant qu'on se trouve dans le noeud t

La meilleure coupure

La recherche du meilleur split se fait grâce à une fonction d'impureté qui doit vérifier les propriétés suivantes :

Définition 3.7. *Une fonction d'impureté est une fonction ϕ définie sur l'ensemble*

$$\left\{ (p_1, \ldots, p_C), \forall j \in \{1, \ldots, C\}, p_j \geq 0, \sum_{j=1}^{C} p_j = 1 \right\}$$

vérifiant

- *ϕ admet un unique maximum en $(1/C, \ldots, 1/C)$*
- *ϕ admet un minima en chaque e_i (où e_i a toutes ses coordonnées nulles sauf la i-ème)*
- *ϕ est une fonction symétrique de $p_1, \ldots, p_C$*

La fonction d'impureté est utilisé pour mesurer l'impureté d'un noeud t ou bien de l'arbre T.

Définition 3.8. *On appelle $i(t)$ l'impureté du noeud t définie par la relation*

$$i(t) = \phi((p(1|t), \ldots, p(C|t))$$

On peut également définir l'impureté d'un noeud t dans l'arbre total T :

$$I(t) = p(t)i(t)$$

Définition 3.9. *On appelle $I(T)$ l'impureté de l'arbre total T telle que*

$$I(T) = \sum_{t \in \tilde{T}} I(t)$$

Définition 3.10. *On définit alors la diminution d'impureté entre un noeud et ses deux fils obtenus par la coupure s, par la relation :*

$$\Delta i(s,t) = i(t) - p_R i(t_R) - p_L i(t_L)$$

La recherche du meilleur split s consiste à maximiser la différence entre l'impureté d'un noeud t et de ses deux fils, t_L et t_R.

On peut chercher à maximiser $\Delta i(s,t)$, mais également de manière équivalente $\Delta I(s,t)$ tel que

$$\Delta I(s,t) = I(t) - I(t_R) - I(t_L)$$

Finalement, maximiser cette quantité revient à trouver la meilleur variable $X^{(l)}$ qui réduit au maximum l'impureté du noeud t, telle que l'ensemble des observations respectant $X^{(l)} < s$ vont dans le noeud de gauche t_L, et l'ensemble des observations respectant $X^{(l)} \geq s$ vont dans le noeud de droite t_R. On a, par construction $\{i, i \in t, i = 1, \ldots, n\} = \{i, i \in t \text{ et } X_i^{(l)} < s\} \cup \{i, i \in t \text{ et } X_i^{(l)} \geq s\}$. Et en conséquence, l'ensemble des noeuds terminaux de l'arbre T forme une partition disjointe des observations.

Réduire l'erreur de classification

Étant donnée que la méthode cherche à classifier, il est assez naturel de proposer une fonction d'impureté en relation avec l'erreur de classification.

Pour cela on note j^* la classe qui est affectée à un noeud t, telle que : $j^* = arg\max_j p(j|t)$. Sauf cas particulier que nous aborderons brièvement plus tard, j^* correspond à la classe majoritaire du noeud t.

Définition 3.11. *On définit alors la fonction erreur de classification $r(t)$ d'une observation sachant qu'elle est dans le noeud t telle que*

$$\forall t \in \tilde{T},\ r(t) = 1 - p(j^*|t)$$

Et appelle $R(t)$ l'erreur de classification d'une observation au noeud t telle que

$$R(t) = p(t)r(t)$$

En posant $r(t) = \phi(p_1, \ldots, p_C) = 1 - \max_j p_j$ comme fonction d'impureté, on montre qu'elle vérifie bien les trois propriétés de la définition 3.7. On chercher alors à maximiser la quantité $r(t) - p_L r(t_L) - p_R r(t_R)$.

Dans le chapitre 4, section 1 de l'ouvrage CART [3], il est mentionné que cette fonction d'impureté n'était pas satisfaisante. L'une des raisons est qu'elle ne récompense pas assez les noeuds pures. Finalement, essayer de réduire l'erreur de classification provoque une perte d'information.

Les critères utilisés en pratique

On définit alors une nouvelle classe de fonction d'impureté telle que $i(t) = \sum_{j=1}^{C} f(p(j|t))$ et f vérifie

- $f(0) = f(1) = 0$
- $f(p) = f(1-p)$
- $f''(p) < 0, 0 < p < 1$ (i.e. f concave)

Définition 3.12. *Un candidat, largement utilisé est alors le critère de GINI, définit par*

$$i(t) = \sum_{j \neq i} p(j|t)p(i|t)$$

C'est le cas où $f(p) = p(1-p)$

Définition 3.13. *Un second candidat est le critère d'information*

$$f(p) = -plog(p)$$

Ce critère est plus connu sous le nom d'entropie de Shannon, issu de la théorie de l'information.

Dans le cas où $C = 2$, le choix de split entre Gini et ce second candidat est le même, comme le montre la figure 1 en annexe.

3.4.2 Critère d'arrêt

Il existe plusieurs critères qui permettent de stopper l'algorithme pour un noeud donné. L'existence des paramètres qui vont être énoncés ont pour but d'empêcher au classifieur de sur-apprendre, c'est à dire décrire trop en détails les données, qui ne sont qu'un échantillon du phénomène qui cherche à être compris ou prédit.

Pour cela il est possible de :

- limiter la profondeur de l'arbre
- fixer le nombre minimal d'observations dans un noeud terminal
- fixer le nombre minimal d'observations nécessaire pour créer un split

Ces paramètres sont aussi importants afin de limiter le temps de calcul.

3.4.3 Elagage de l'arbre

En complément de ces paramètres cités précédemment, une technique d'élagage assez avancée est utilisée afin d'éviter le sur-apprentissage. L'élagage d'un arbre consiste, après l'avoir construit de taille maximale, à couper certaines branches pour le rendre moins complexe tout en gardant sa capacité de prédiction.

On définit alors une mesure R_α tenant compte de la complexité de l'arbre telle que $R_\alpha(T) = R(T) + \alpha|\tilde{T}|$, où $|\tilde{T}|$ est le nombre de noeuds terminaux dans T. α est appelé le paramètre de complexité, qui pénalise l'erreur lorsque l'arbre est trop complexe.

Pour chaque valeur de α, on définit un sous-arbre $T(\alpha) \leq T_{max}$ qui minimise R_α, c'est à dire $R_\alpha(T(\alpha)) = \min_{T \leq T_{max}} R(T)$

- Pour $\alpha = 0$ on cherche simplement à minimiser R et $T_0 = T_{max}$
- Plus α est grand, plus la pénalisation liée à la complexité de l'arbre est importante.
- Pour $\alpha = +\infty$, l'arbre sera réduit à sa racine.

On a les résultats suivants, démontrés dans [3] :

1. Si T_1 et T_2 sont des sous-arbres de T, $R_\alpha(T_1) = R_\alpha(T_2)$ alors soit T_1 est un sous-arbre de T_2, soit T_2 est un sous-arbre de T_1.
2. Si $\alpha > \beta$ soit $T_\alpha = T_\beta$, soit T_α est un sous-arbre de T_β.

Le premier résultat implique que l'ont peut définir de manière unique T_α comme le plus petit sous-arbre pour lequel R_α est minimisé.

Le second résultat implique que toutes les valeurs possibles de α peuvent être regroupées dans m intervalles.

On cherche donc ici à construire une suite d'arbre décroissante et emboîtée partant de l'arbre final T_{max} jusqu'à la racine $\{t_1\}$, $T_{max} < T_1 < T_2 < \ldots < \{t_1\}$, de sorte qu'il ne reste plus qu'à choisir quel arbre dans cette suite possède le meilleur compromis complexité/prédiction. Afin de construire cette suite, on procède de la manière suivante :

1. On supprime chaque noeud enfant d'un même noeud s'ils ont les même classes. L'arbre T_{max} est alors réduit en un arbre noté T_1.
2. On construit l'arbre T_2 comme suit :

Pour tout noeud t, on note T_t l'arbre ayant pour racine le noeud t, et $\{t\}$ l'arbre constitué du seul noeud t. On note donc

$$\begin{aligned} R_\alpha(\{t\}) &= R(t) + \alpha \\ R_\alpha(T_t) &= R(T_t) + \alpha|\tilde{T}_t| \end{aligned}$$

Tant que $R_\alpha(T_t) < R_\alpha(\{t\})$, T_t a un meilleur taux erreur-complexité que le simple noeud $\{t\}$. Mais à une valeur critique notée α_c, les deux erreurs deviennent égales, puis s'inverse. C'est à dire qu'il est préférable de supprimer de l'arbre T_1 le sous arbre T_t. Pour définir ce seuil α_c, on résout l'inégalité :

$$R_\alpha(T_t) < R_\alpha(\{t\}) \iff \alpha \leq \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$$

$$\text{Et } R(t) - R(T_t) > 0$$

Posons, pour tout $t \in T_1$,

$$g_1(t) = \begin{cases} \dfrac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}, & \text{si } t \notin \tilde{T}_1 \\ +\infty, & \text{sinon} \end{cases}$$

On définit alors t'_1 comme étant le *plus faible lien*, de T_1 tel que

$$g_1(t'_1) = \min_{t \in T_1} g_1(t)$$

et on pose

$$\alpha_2 = g_1(t'_1)$$

Le noeud t'_1 est le *plus faible lien* dans le sens où lorsque α augmente, c'est le premier noeud tel que $R_\alpha(\{t\})$ est inférieur ou égal à $R_\alpha(T_t)$. Alors on va préférer $\{t'_1\}$ à $T_{t'_1}$

On définit alors un nouvel arbre $T_2 < T_1$ en supprimant le sous-arbre $T_{t'_1}$. Puis on recommence la procédure d'élagage jusqu'à ce que $T_k = t_1$ (i.e. la racine).

Il reste donc à trouver le meilleur arbre de la suite ainsi créée. Pour cela on peut utiliser une méthode de validation vu précédemment dans la section 3.3. Ainsi, il suffit d'estimer les erreurs de classification de chaque sous-arbres de la suite et de sélectionner le meilleur.

Pour un cas concret, nous renvoyons vers une vignette du package `rpart` [6].

3.4.4 Consistance des arbres

Cette partie est tirée du mémoire de diplôme de l'ENS de Erwan Scornet [7] qui résume parfaitement en quelques lignes l'essentiel du chapitre 15 du livre CART [3].

On dispose d'un échantillon d'apprentissage $\mathcal{L}_n$. Pour tout noeud t, notons $p_N(t)$ la probabilité empirique qu'une donnée tombe dans $\{t\}$. Soit $\tilde{T_N}$ une certaine partition. On pose $\tilde{T_N(x)} = \left\{t \in \tilde{T_N}, x \in t\right\}$ et on note $\delta(t) = \sup_{x,x' \in t} |x - x'|$, i.e. le diamètre de $\{t\}$. De même on note $D_N = \delta(\tilde{T_N(x)})$ le diamètre de t contenant x.

Soit $h_N(.)$ un classifieur. On appelle classifieur de Bayes et on le note h^*, le classifieur qui minimise l'erreur de classification pour x

$$\sum_j C(h(x)|j)P(j|x)$$

où

$$C(i,j) = \begin{cases} \text{le coût de classifier une observation de classe } i \text{ en classe } j, \text{ si } i \neq j \\ 0, \text{sinon} \end{cases}$$

Remarque. *Habituellement la matrice de coût est faite de 0 sur la diagonale et de 1 sinon. On retrouve ainsi les formulations d'erreur $r(.)$ précédemment introduite.*

Un classifieur naturel h_N est donc construit en minimisant l'erreur de classification empirique pour x

$$\sum_j C(h(x)|j)p_N(j|x)$$

On peut montrer que cela revient à minimiser l'expression suivante :

$$\sum_{n, X_n \in \tilde{\tau_N(x)}} C(h(x)|Y_n)$$

ce qui est exactement la façon dont on assigne une classe à un noeud terminal après avoir construit l'arbre. Le classifieur associé à un arbre de décision est consistant sous certaines hypothèses

Théorème 2. *Soit $k_N > 0$ tel que*

$$p_N(t) \geq k_N \frac{logN}{N}$$

Supposons de plus que

1. $\lim_N k_N = +\infty$

2. $lim_N D_N(X) = 0$ *(en probabilité)*

Alors h_N est un classifieur consistant, c'est à dire que l'erreur moyenne due à ce classifieur tend vers la meilleure erreur moyenne envisageable (en probabilité).

3.4.5 Pour aller plus loin...

Il y aurait beaucoup de choses à rajouter sur les arbres de décision CART, et afin de donner des pistes de lecture, voici quelques autres points qui méritent d'être énoncés.

- Il est tout à fait possible d'utiliser des covariables qualitatives. Si on choisit une variable qualitative, la fonction d'impureté va rechercher quel groupe de classes parmi cette variable qualitative maximise $\Delta i(s,t)$. Il faut alors penser à réduire le nombre de classes d'une variable qualitative, car l'algorithme va essayer toutes les combinaisons, ce qui peut être coûteux.
- Il est également possible d'avoir des covariables avec des valeurs manquantes. L'algorithme utilise des *surrogates*, c'est à dire des autres variables explicatives qui sont liées à la façon de séparer le jeu de données de la variable initiale.
- L'algorithme produit également une liste de variables importantes. Ceci est basé sur le gain de Gini, ou d'un autre critère.
- De même, il est possible d'avoir accès à la liste des meilleures split pour chaque noeud.
- La modification des *priors* (des proportions de classes du jeu de données) peut être envisageable.
- La matrice de coût est également modifiable.
- Une fonction d'impureté appelé Twoing peut être utilisée lorsque $C > 2$. Là où Gini essaye d'isoler la plus large des classes du jeu de données, Twoing cherche plutôt à équilibrer au premier split les effectifs de classes puis à les isoler. Idéalement.
- CART permet aussi les arbres de régression. L'idée dans ce cas est de minimiser la variance au sein d'un noeud. Les arbres de classification ont été beaucoup plus mis en avant car répondant à plus de problématiques, c'est pourquoi la littérature est moins dense à propos des arbres de régression.

L'ensemble des informations présentées ci-dessus peuvent être approfondies en lisant la vignette du package `rpart` [6], en consultant le site web et les excellents tutoriels de Salford System ou bien en se procurant l'ouvrage entier sur CART [3].

Chapitre 4

Les forêts aléatoires

4.1 Introduction

Ici, on se place toujours dans le cadre de la classification. Le *bagging* est ce qu'on appelle une méthode d'ensemble, elle consiste à créer un classifieur unique composé d'un ensemble de classifieurs (du même type). Assigner une classe à une observation revient alors à assigner le vote majoritaire de l'ensemble des classifieurs.

Cette idée de bagging pour les arbres de décision est présentée par Leo Breiman en 1996, connue sous le nom de Tree Bagging. De puis ces années là, plusieurs méthodes d'ensemble telles que le *random subspace* introduit par Ho en 1998, l'*Extra-Tree* pensé par Geurts et al. en 2006, ou encore la méthode *Randomization* pensée par Dietterich en 1999. Toutes ces méthodes d'ensemble d'arbres appartiennent à la famille des forêts aléatoires. Certaines forêts ont été introduites dans le seul but de faciliter l'étude théorique des forêts aléatoires [8]. D'autres pour améliorer les performances ou la vitesse de calcul. Il est important de retenir néanmoins que d'une forêt à une autre, seul quelques points techniques varient, ce qui les rendent toutes similaires.

Dans la suite de ce chapitre nous parlerons de la forêt aléatoire RI (pour *Random Input*), celle introduite et formalisée par Leo Breiman en 2001 [9]. A l'heure actuelle, c'est la forêt de loin la plus utilisée, et autant performante, si ce n'est pas mieux, comparée aux autres forêt de la famille.

Il est à noter que bien que les forêts aléatoires soient largement utilisées, qu'elle aient des performances concurrentielles et que le mécanisme les régissant soit relativement simple, elles restent néanmoins encore peu comprises et font l'objet d'études théoriques [8].

Dans une première section nous verrons le lien qui existe entre les arbres CART et la fôrets aléatoire, ainsi que la construction d'une telle forêt, puis quelques aspects théoriques. Ensuite, il sera présenté les divers atouts d'une forêt aléatoire, comme la matrice de proximité, les variables d'importance, etc ... Enfin, nous finirons par présenter un exemple simple d'utilisation en rapport avec le domaine médical.

4.2 Construction de la forêt

Une forêt, comme son nom l'indique est constitué de plusieurs arbres. L'aléatoire est introduit à deux niveaux, comme nous allons le voir. On dispose d'un échantillon d'apprentissage $\mathcal{L}_n$ (cf section 3.2). On pose K le nombre d'arbres présents dans la forêt, ainsi que M, $1 \leq M \leq p$. Où, pour rappel, p est le nombre de variables explicatives. A l'itération $k = 1, \ldots, K$, voici les étapes de créations de la forêt :

1. Nous tirons un échantillon bootstrap T_k issu de $\mathcal{L}_n$ choisi par la loi θ_k
2. Un arbre de décision CART $h_k(X, T_k, \theta_k)$ est construit, de la manière suivante :
   (a) L'échantillon d'apprentissage est bien T_k
   (b) A chaque noeud, M variables sont sélectionnées **aléatoirement**.
   (c) Parmi ces M variables, le meilleur split est calculé, comme décrit en section 3.4.1.

(d) L'arbre est construit **au maximum**, idéalement avec qu'une observation dans les feuilles terminales.

3. Le classifieur est ajouté à l'ensemble $\{h_1(X, T_1, \theta_1), \ldots, h_{k-1}(X, T_{k-1}, \theta_{k-1})\}$

Remarques. *Les θ_k sont des réalisations i.i.d. d'une variable aléatoire θ. Il faut bien comprendre que cette variable θ est la variable portant l'aléatoire de l'arbre. Dans notre cas, elle définit la façon de tirer l'échantillon bootstrap, et la sélection des M variables à chaque noeud.*

Ainsi on définit un nouveau type de classifieur, appelés des classifieurs randomisés. On l'écrit $h_k(X, T_k, \theta_k)$, qu'on notera dans la suite $h_k(X, \theta_k)$.

Il est à remarquer que $h_k(X, \theta_k)$ est une réalisation de la variable aléatoire $h(X, \theta)$.

Lors de la prédiction d'une nouvelle observation notée X_{n+1}, la classe Y_{n+1} associée à X_{n+1} est alors la classe majoritaire de l'ensemble $\{h_k(X_{n+1}, \theta_k),\ k = 1, \ldots, K\}$. C'est ce qu'on appelle le vote majoritaire, ou encore un classifieur moyen.

Remarque. *Dans le cas d'un problème à deux classes tel que $C = \{1, 0\}$, on note*

$$\begin{aligned} P &= \textstyle\sum_{k=1}^{K} \mathbf{1}(h(X_{n+1}, \theta_k)) = 1) \\ Q &= \textstyle\sum_{k=1}^{K} \mathbf{1}(h(X_{n+1}, \theta_k)) = 0) \end{aligned}$$

Alors si $P > Q$, $Y_{n+1} = 1$, sinon $Y_{n+1} = 0$.

Il est important de comprendre la raison pour laquelle ce type de méthode, à savoir de *bagging*, est un coup de génie, la thèse de Robin Genuer en rend compte [10]. Initialement, comme nous l'avons vu pour CART, un classifieur seul est construit de sorte qu'il soit optimisé pour la prédiction ou la description. Les méthodes d'ensemble, au contraire, mettent en commun un ensemble de prédicteur dans le but de d'explorer l'ensemble des solutions et des règles de prédictions. On attend du prédicteur final qu'il soit meilleur que chacun des prédicteurs qui le constituent : l'union fait la force.

Prenons une réalisation X, et plaçons-nous dans un problème à deux classes. Comme nous l'avons fait remarquer, pour que l'ensemble $\{h_k(X, \theta_k),\ k = 1, \ldots, K\}$ commette une erreur de classification, il faut au moins que la moitié des classifieurs de l'ensemble se trompe. Intuitivement, cela n'arrive pas souvent, car même si un classifieur h_k commet des erreurs, il est moins probable que la majorité des classifieurs commette la même erreur pour la même observation X. Donc on en vient à penser que les classifieurs, entre eux, doivent être différents. Mais cela n'est pas suffisant, il faut aussi que le classifieur individuel soit relativement bon : là où un prédicteur se trompe, les autres doivent *prendre le relais* en ne se trompant pas.

En résumé, on retient qu'une méthode d'ensemble, comme les forêts aléatoires nécessite de vérifier deux points :

- Chaque classifieur individuel doit être relativement bon. En effet, si ce n'est pas le cas, même en assemblant ces derniers, le résultat final ne sera pas performant non plus.
- L'ensemble des classifieurs doivent être différents les uns des autres. Si ce n'est pas le cas, les assembler n'apporte pas ou très peu d'intérêt.

Leo Breiman dans son article [9] avance que les forêts aléatoires convergent. La convergence ici signifie que le risque de classification de la forêt tend vers une valeur limite quand le nombre d'arbres K augmente. Cela signifie que la forêt aléatoire ne pose pas de problème de sur-apprentissage. Il donne également une borne à cette valeur limite. La démonstration qui suit est issue de l'article de Leo Breiman et d'une version traduite de la thèse de Simon Bernard [11].

Définition 4.1. *On définit la fonction marginale d'un ensemble de classifieur* $\{h_k(.,\theta_k),\ k=1,\ldots,K\}$, *telle que*

$$mg(X,Y) = \frac{1}{K}\sum_{k=1}^{K}\mathbf{1}(h_k(X)=Y) - \max_{j\neq Y}\frac{1}{K}\sum_{k=1}^{K}\mathbf{1}(h_k(X)=j)$$

Cette fonction calcule la différence entre la proportion de vote pour la classe correcte et la proportion de vote pour la deuxième *meilleure* classe. Évidemment, le souhait est d'avoir une différence assez grande, cependant si la marge est négative, c'est que l'ensemble s'est trompé de classe.

Définition 4.2. *On définit l'erreur généralisée, qui n'est rien d'autre que l'erreur de classification R introduite en section 3.2, telle que*

$$PE = \mathbb{P}(mg(X,Y)<0)$$

Théorème 3. *Breiman montre alors que pour toute séquence* $\theta_1,\ldots,\theta_K$ *qui sont des réalisations i.i.d. d'une variable aléatoire* θ

$$PE \longrightarrow^{K\to\infty} \mathbb{P}_{X,Y}\left(\mathbb{P}_\theta(h(X,\theta)=Y) - \max_{j\neq Y}\mathbb{P}_\theta(h(X,\theta)=j) < 0\right) := PE^*$$

Cette formule exprime la probabilité globale d'une forêt aléatoire d'avoir une marge négative, quelles que soient les valeurs $(x,y)\in\mathcal{X}\times\mathcal{Y}$, et donc la probabilité que celle-ci se trompe dans sa prédiction pour un individu quelconque de cette population.

Remarque. *Lorsque* $K\to\infty$ *La forêt est infinie, c'est à dire que le vote majoritaire se fait sur l'espace entier des classifieurs.*

Cette valeur étant théorique, il est possible de la borner afin d'avoir une estimation de sa valeur. Pour cela Leo Breiman introduit la fonction de marge (non empirique cette fois), définie par :

$$mr(X,Y) = \mathbb{P}_\theta(h(X,\theta)=Y) - \max_{j\neq Y}\mathbb{P}_\theta(h(X,\theta)=j)$$

On définit la *force* d'une forêt aléatoire par

$$s = \mathbb{E}_{X,Y}mr(X,Y)$$

En supposant $s >= 0$, l'inégalité de Chebychev donne :

$$PE^* \leq \frac{\mathrm{var}(mr)}{s^2}$$

La variance d'une marge d'une forêt n'était pas très explicite, on introduit alors la fonction de marge brut :

$$rmg(\theta,X,Y) = \mathbf{1}(h(X,\theta)=Y) - \mathbf{1}(h(X,\theta)=\hat{j}(X,Y))$$

où $\hat{j}(X,Y) = arg\max_{j\neq Y}\mathbb{P}_\theta(h(X,\theta)=j)$

On considère θ et θ' **indépendants, de même distribution**, et on note alors

$$mr(X,Y)^2 = \mathbb{E}_{\theta,\theta'}[rmg(\theta,X,Y)rmg(\theta',X,Y)]$$

On en déduit facilement

$$\begin{aligned}\mathrm{var}(mr) &= \mathbb{E}_{\theta,\theta'}[cov_{X,Y}rmg(\theta,X,Y)rmg(\theta',X,Y)]\\ &= \mathbb{E}_{\theta,\theta'}[\rho(\theta,\theta')\sigma(\theta)\sigma(\theta')]\end{aligned}$$

où $\rho(\theta, \theta')$ est la corrélation entre $rmg(\theta, X, Y)$ et $rmg(\theta', X, Y)$, et $\rho(\theta)$ est l'écart type de $rmg(\theta, X, Y)$, pour θ et θ' fixés.

On note $\bar{\rho}$, la valeur moyenne de la corrélation telle que $\bar{\rho} = \mathbb{E}_{\theta,\theta'}[\rho(\theta, \theta')\sigma(\theta)\sigma(\theta')]/\mathbb{E}_{\theta,\theta'}[\sigma(\theta)\sigma(\theta')]$, on a alors

$$\begin{aligned} \text{var}(mr) &= \bar{\rho}\mathbb{E}_{\theta,\theta'}[\sigma(\theta)]\mathbb{E}_{\theta,\theta'}[\sigma(\theta)'] && \text{par indépendance des } \theta \\ &= \bar{\rho}\left(\mathbb{E}_{\theta}[\sigma(\theta)]\right)^2 && \text{les distributions étant les mêmes} \\ &\leq \bar{\rho}\mathbb{E}_{\theta,\theta'}[\text{var}(\theta)] && \text{Inégalité de Jensen} \end{aligned}$$

Puis, on écrit

$$\mathbb{E}_{\theta}[\text{var}(\theta)] \leq \mathbb{E}_{\theta}(\mathbb{E}_{X,Y}[rmg(\theta, X, Y)])^2 - s^2 \leq 1 - s^2$$

On peut alors définir une borne supérieur de PE^* qui a le mérite d'être interprétable.

$$PE^* \leq \frac{\bar{\rho}(1 - s^2)}{s^2} = \frac{\bar{\rho}}{s^2} - 1 \tag{4.1}$$

Cette écriture permet de signaler deux éléments importants qui jouent un rôle dans la quantité PE^* qui est, pour rappel, l'erreur asymptotique de classification de la forêt. Tout d'abord, si s^2 devient grand, PE^* diminue, avec $\bar{\rho}$ fixé. Qualitativement, cela revient à dire que l'ensemble des classifieurs doivent être de bons prédicteurs. Deuxièmement, si $\bar{\rho}$ diminue, PE^* diminue également, tout autre quantité fixée. De même, cela signifie que la corrélation entre deux arbres doit être minimale, autrement dit, les arbres doivent être différents entre eux. La formule 4.1 démontre alors parfaitement les deux critère cités en préambule de la démonstration.

Cette section achève le point de vue très théorique des forêts aléatoires. Pour aller plus loin le lecteur peut se rediriger vers des ouvrages plus complets comme les thèses [11], [10] et une publication de G. Biau [8]. A noter qu'un mémoire [12] semble mieux formaliser l'écriture des forêts aléatoires.

4.3 Les intérêts des forêts aléatoires

Contrairement à beaucoup d'autres méthodes de classification la forêt aléatoire impose très peu de contraintes sur l'échantillon d'apprentissage $\mathcal{L}_n$. Voici les possibilités d'applications d'une forêt :

- Idéal pour un jeu de données volumineux, aussi bien en covariables qu'en observations. La construction d'un arbre coûte peu en temps de calcul. De plus, il est possible de découper la forêt pour faciliter le calcul, on parle alors de parallélisation.
- Le nombre de variables explicatives n'est pas limité, typiquement, p peut être largement supérieur à n.
- Les covariables peuvent être qualitatives ou quantitatives. Dans le cas qualitatif, la restriction est la même que pour les arbres CART (cf. 3.4.5).
- Initialement, les forêts aléatoires ne supportent pas les valeurs manquantes. Cependant, nous verrons dans la suite qu'elles proposent une méthode d'imputation.

4.3.1 Le *Out-Of-Bag* estimateur

Les méthodes de validation présentées en section 3.3 peuvent être utilisées pour estimer l'erreur de classification d'une forêt. Cependant, la forêt aléatoire offre, et plus largement le bagging, la possibilité d'estimer cette erreur de façon interne. Voyons comment.

La probabilité pour une observation d'être choisie dans l'échantillon bootstrap T_k au premier tirage est $\frac{1}{n}$. Donc, la probabilité pour une observation de ne pas être choisie dans l'échantillon bootstrap T_k au premier tirage est $1 - \frac{1}{n}$. Comme le tirage est avec remise, la probabilité pour une observation de ne pas être choisie dans l'échantillon bootstrap au tirage n est de $\left(1 - \frac{1}{n}\right)^n$.

Or, on a :

$$\lim_{n\to\infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e} \approx 0.368$$

Cela signifie qu'à chaque construction d'arbre, environ un tiers des observations ne sont pas utilisées pour la construction de l'arbre. Ce tiers, théorique, d'observation mis de côté est appelé l'*OOB* (pour *Out-Of-Bag*). A la fin de la construction de l'arbre nous pouvons écrire la proportion de vote *OOB* pour une classe j d'une observation (x, y) :

$$Q(x, j) = \frac{\sum_{k=1}^{K} \mathbf{1}\left(h(x, \theta_k) = j : \ (x, y) \notin T_k\right)}{\sum_{k=1}^{K} \mathbf{1}\left((y, x) \notin T_k\right)}$$

On peut alors estimer l'erreur de classification de la forêt aléatoire par l'*OOB* estimateur, définit par

$$R^{oob} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\left(arg\max_j \{Q(x_i, j) \neq y_i\}\right)$$

Le concept est d'utiliser l'ensemble des arbres non construits à partir d'une observation (x, y) donnée pour estimer sa classe.

Leo Breiman dans son article [9] avance que d'après des résultats empiriques, l'*OOB* estimate est autant précis que d'utiliser un échantillon test de la même taille que l'échantillon d'apprentissage. C'est pourquoi, il est possible de se passer d'un échantillon test, ce qui peut être un avantage suivant le contexte.

Cependant, comme l'*OOB* estimateur n'est calculé que sur un sous ensemble de la forêt, contrairement à l'échantillon test et comme l'erreur généralisée tend à décroître, il est naturel que l'*OOB* estimateur surestime l'erreur de classification. L'article de Matthew W. Mitchell rend compte de ce biais positif [13]. Enfin, Breiman termine par énoncer que pour n grand, l'*OOB* estimate est non biaisé.

Dans la pratique, il est habituel d'utiliser en complément un échantillon test, si le contexte le permet.

4.3.2 Matrice de proximité et imputation de valeurs manquantes

Une forêt aléatoire propose une matrice de proximité $n \times n$ où une cellule de coordonnée (i, j) correspond au nombre de fois que l'observation i de $\mathcal{L}_n$ se trouve dans le même noeud terminal d'un arbre que l'observation j de $\mathcal{L}_n$, puis ce nombre est divisé par K. La matrice est symétrique.

Cette matrice peut servir à flagger des outliers, notamment grâce à une analyse en composante principale sur cette matrice.

Mais dans bien des cas, la matrice de proximité est calculée dans le but d'imputer des valeurs manquantes dans le jeu de données. Pour cela, on procède à l'algorithme suivant :

- La première étape consiste à remplacer les valeurs manquantes avec la valeur moyenne de chaque variable, ou la catégorie la plus fréquente de la variable.
- La seconde étape consiste à :
  1. Construire une forêt avec les données complétées
  2. Construire la matrice de proximité

3. Pour une variable qualitative, remplacer les valeurs initialement manquantes par une moyenne pondérée des valeurs non manquantes de la variable, où les poids sont les proximités. Pour une variable catégorielle, les valeurs initialement manquantes sont remplacées par la classe dont la moyenne des proximités est la plus élevée.

La seconde étape est réitérée tant qu'elle n'atteint pas un critère d'arrêt, comme le nombre d'itération maximale, ou lorsque la différence d'OOB estimateurs entre deux imputations est inférieur à une borne donnée.

Adam Pantanowitz et Tshilidzi Marwala montre, par l'exemple, la nature compétitive de l'imputation par forêt aléatoire dans une étude sur le HIV.

Contrairement à beaucoup de méthode d'imputation, la seul contrainte est de posséder aucune valeurs manquantes au sein du vecteur $(Y_1, \ldots, Y_n)$.

C'est pour cette raison qu'une nouvelle méthode, basée sur les forêts aléatoires, est proposée par Daniel J. Stekhoven et Peter Bühlmann. Brièvement, elle consiste à permuter chaque variable comme variable à expliquer, et construire une forêt que pour les valeurs non manquantes, puis d'itérer. Ainsi le temps de calcul en souffre, mais le résultat a su montrer son avantage face à des méthodes comme KNN et MICE [14].

4.3.3 L'importance des variables

L'une des utilisations les plus fréquentes des forêts aléatoires se trouve dans le domaine de la sélection de variables. En effet, certains modèles, contrairement aux forêts aléatoires, ne supporte pas un nombre *grand* de variables explicatives. La forêt aléatoire peut calculer de manière interne l'importance d'une variable sur la qualité prédictive de la forêt en fonction des autres variables. Deux méthodes sont possibles.

L'importance par permutation

L'importance des variables par permutation est basée sur la fluctuation du taux d'erreur : pour un arbre h_k construit dans la forêt, on utilise l'échantillon OOB pour mesurer le nombre d'observations bien classées. Puis, pour une variable X_p on permute ses valeurs de façon à ce qu'il n'y ait plus de lien avec les valeurs de Y, puis on ré-utilise l'échantillon OOB dans l'arbre h_k, une nouvelle mesure d'observations bien classés est calculée. En soustrayant les deux mesures, avant permutation moins celle après permutation, on obtient une valeur qui permet de quantifier l'importance de la variable X_p au sein de l'arbre. Le procédé est itérer à chaque arbre de la forêt, en moyennant par le nombre d'arbre, cela nous donne une mesure d'importance de chaque variable.

L'interprétation est assez simple : plus l'importance d'une variable est élevée, plus la variable joue un rôle dans la classification totale. En effet, cela signifie qu'en perturbant l'ordre des valeurs de la variable, cela perturbe fortement de façon négative la classification.

A l'inverse, plus une valeur sera faible, moins la variable est liée à la mesure de qualité de la forêt, à savoir sa précision de classification.

Enfin, en pratique, l'ordre des valeurs d'importances importe plus que la valeur en tant que telle.

L'importance par mesure de Gini

Une seconde méthode pour calculer l'importance des variables est de s'appuyer sur la mesure d'impureté Gini présentée en 3.4.1. L'importance est calculée en moyennant la diminution de l'impureté d'un noeud d'une variable sur l'ensemble des arbres.

L'interprétation de la valeur est difficile, alors que l'ordre des variables donne sens, comme l'importance par permutation.

Néanmoins ces deux mesures sont à prendre avec des précautions, plusieurs articles sur le sujet ont pu dénoter un biais lorsque le jeu de données est constitué de plusieurs types de variables de natures différentes. Des variables qualitatives ayant beaucoup de catégories seront artificiellement avantagées par celles en ayant moins. De même, s'il existe des variables continues n'évoluant pas dans le même intervalle, leurs importances peuvent être biaisée. D'autre part, le biais vient aussi du ré-échantillonnage à la construction de chaque arbre. Un excellent article met en avant ces défauts et propose une nouvelle méthode afin de palier ce biais [15].

En conclusion, lorsque la forêt aléatoire a aussi le but de décrire les données, et de comprendre concrètement l'influence des variables, il faut avoir eu connaissance de ces biais.

4.4 Une application

Nous présentons dans cette partie une simple application des forêts aléatoires sur un jeu de données *Sepsis*. Ce jeu de données est le résultat d'une simulation de données d'un essai thérapeutique à deux bras de traitement. Cela signifie qu'un groupe de patients reçoit un traitement dont nous cherchons à prouver l'efficacité, et un second groupe de patients reçoit un traitement de contrôle.

La septicémie ou sepsis est une maladie infectieuse sévère, qui, si non traitée, entraîne la mort. Ici, on mesure la survie du candidat au 28ième jour.

Le jeu de données est composé de 470 patient, dont 153 dans le bras de contrôle, et 317 dans le bras de traitement. Pour information, le traitement ne montre pas son efficacité, avec une proportion de 0.34 survivant dans le bras de contrôle et 0.407 dans le bras de traitement, la différence de proportion n'est pas significative.

Voici les variables qui constituent le jeu de données

survival	1 pour la survie au bout de 28 jours, 0 sinon
THERAPY	1 pour le traitement actif, 0 pour le traitement de contrôle
TIMFIRST	Temps entre la prise de traitement et la défaillance du premier organe
AGE	Age du patient en années
BLLPLAT	Les plaquettes à baseline
blSOFA	Somme du sofa à baseline (score)
BLLCREAT	Créatinine à baseline
ORGANNUM	Nombre d'organes en défaillance à baseline
PRAPACHE	Score Pre-infusion Apache II
BLGCS	Echelle de Glasgow (état de conscience) à baseline
BLIL6	Concentration du serum IL-6 à baseline
BLADL	score *Activity of daily living* à baseline
BLLBILI	Bilirubine à baseline

TABLE 4.1 – Variables du jeu de données *sepsis*

Remarque. *sepsis est un jeu de donnée utilisé dans la présentation du package que nous avons développé* `aVirtualTwins`. *[16]. Ce package est stable, mais toujours en version de développement.*

On effectue une forêt aléatoire sur ces données avec pour variable à expliquer `survival`. On échantillonne aléatoirement un échantillon d'apprentissage composé des deux tiers de *sepsis*. L'échantillon test est alors composé d'un tiers des données. On lance une forêt aléatoire via le package `randomForest` [17] de 1000 arbres, laissant par défaut tous les autres paramètres, excepté le paramètre permettant la calcul de l'importance des variables et de la matrice de proximité.

```
randomForest(survival~., data = sepsis.train, ntree = 1000, importance = T,
xtest = sepsis.test[, -1], ytest = sepsis.test[,1])
```

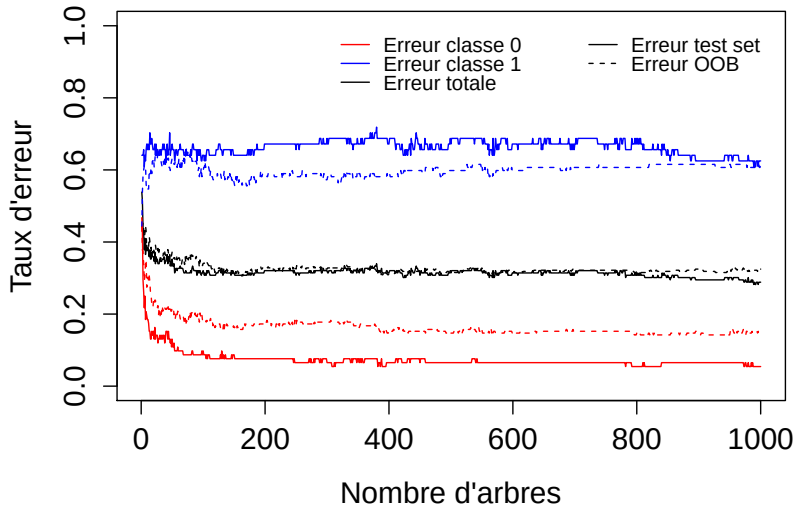


FIGURE 4.1 – Taux d'erreur OOB et de l'échantillon test de la forêt aléatoire

L'erreur globale est plutôt forte, environ 32.5% pour l'oob estimateur, et comme vu à la section 4.3.1, l'erreur sur l'échantillon test est inférieure : 28.9%.

Nous pouvons également visualiser les mesures d'importance des variables.

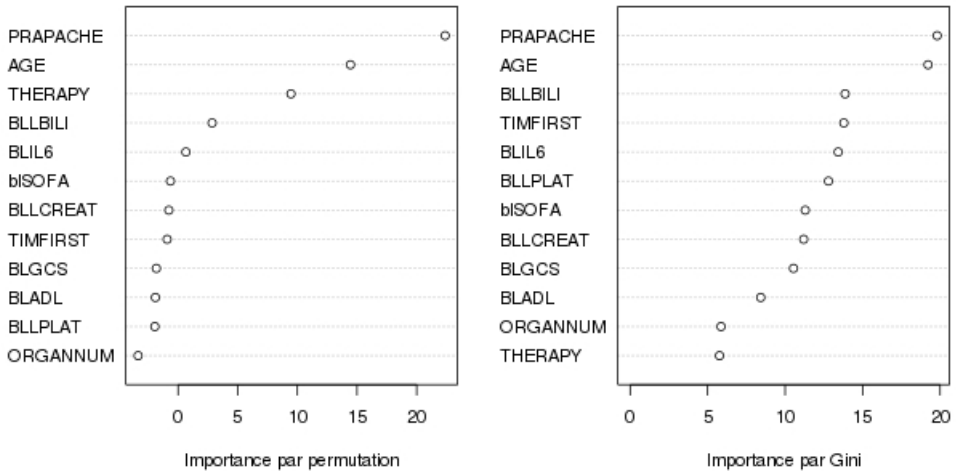


FIGURE 4.2 – Mesures d'importance des variables par la forêt aléatoire

Ici, les problèmes de biais peuvent se poser, cependant excepté la variable `THERAPY`, toutes les covariables sont continues, avec globalement un nombre de points uniques équivalents. Les deux graphiques de la figure 4.2 montrent une certaine homogénéité. Deux variables sortent du lot : `PRAPACHE` et `AGE`. Nous sommes également intéressés par la variable `THERAPY`, dans la réduction d'impureté, cette variable est mauvaise. La première raison, c'est son inefficacité a priori supposée, mais également le fait qu'elle soit binaire, et donc cette mesure est sans doute biaisée par rapport aux autres variables. Dans l'importance des variables par permutation, `THERAPY` se trouve à la troisième place, et est plutôt élevée : cela nous questionne car étant binaire et n'améliorant pas la réduction d'impureté, elle a un impact plutôt fort sur la réponse.

Nous décidons alors de construire une seconde forêt sur ces trois variables, à savoir `THERAPY`, `PRAPACHE` et `AGE`, cette fois sans échantillon test pour bénéficier de toute l'information et avec 1500 arbres.

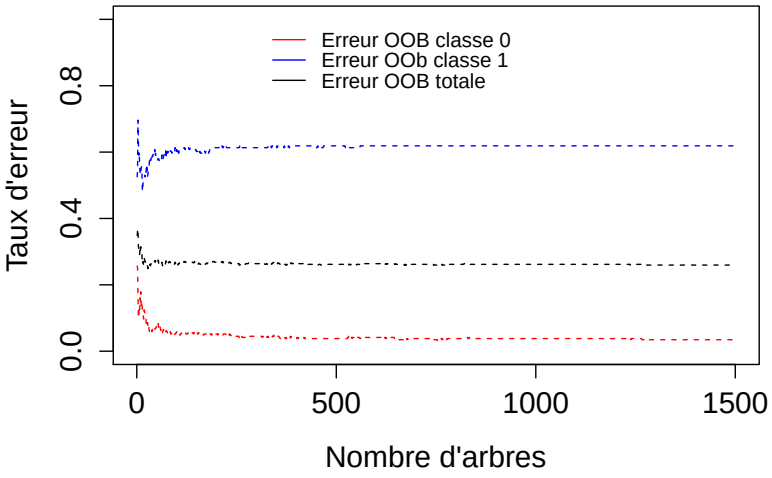


FIGURE 4.3 – Taux d'erreur OOB de la seconde forêt aléatoire

Le taux d'erreur OOB global tombe à 26%. Seulement, nous nous attendions à ce que l'erreur de la classe 1 tombe aussi, mais cela n'est pas le cas. Quoi qu'il en soit, il semble y avoir un signal avec ces trois variables.

Remarque. *Il est possible de mesurer la qualité prédictive d'un classifieur en proposant une courbe ROC. Dans notre cas, il s'agit de faire varier le paramètre appelé* cutoff *afin d'altérer la façon dont la classe est assignée, en l'occurence le vote majoritaire. La courbe ici ne sera pas tracée car proche de la diagonale et donc n'apportant pas d'informations supplémentaires.*

Le package `aVirtualTwins` approfondit ce jeu de données en appliquant la méthode *Virtual Twins* à titre d'exemple. Nous présentons dans la suite du rapport cette méthode.

Chapitre 5

La recherche de sous-groupes et Virtual Twins

5.1 La recherche de sous-groupes avec effet traitement

Lors d'essais cliniques randomisés, la recherche de sous-groupes a pour but d'identifier des sous populations de patients qui bénéficient au mieux d'un traitement selon des critères dit de *baseline* (i.e. des valeurs de covariables à l'initiation du traitement du patient). C'est ce que l'on appelle l'*efficacité*, plus connue sous le nom anglais *efficacy*.

Il existe également une deuxième configuration où la recherche de sous-groupes a son importance. Un essai clinique peut montrer l'efficacité d'un traitement contre un traitement de référence. Mais si le nouveau traitement entraîne plus d'événements indésirables que le comparateur cela pose problème. Dans ce cas, il est possible de rechercher des sous-groupes qui sur-concentrent l'apparition des effets indésirables. C'est ce que l'on appelle des problématiques de *sécurité*, plus connu sous le nom anglais *safety*. On dit alors qu'on explique le signal de *safety*.

Ce type de recherches est au coeur d'une tendance actuelle : la médecine personnalisée. Un traitement peut être adapté pour une typologie de patient, mais pas forcément pour la population entière.

La recherche de sous-groupes nourrit les controverses [18]. D'un point de vue statistique, c'est un dangereux exercice (i.e. multiplicité des tests, manque de puissance), pour lequel les résultats ne peuvent être fiables. Cependant, une partie de la communauté avance qu'on ne peut nier les fortes raisons biologiques des sous-groupes, et de ce fait on peut se permettre d'envisager transgresser certaines règles statistiques [19].

Quoi qu'il en soit, dans le cadre du stage ici présenté, et de ce qui est mené au sein de l'équipe data mining, nous nous plaçons dans le cadre exploratoire de la recherche de sous-groupes. En exploratoire, l'identification d'un bon nombre de sous-groupes est privilégiée, nous nous soucions moins de la puissance des tests, ou alors du contrôle du risque de première espèce. Cette première phase a pour but de générer des futures hypothèses dans l'objectif de mieux connaître le produit, la pathologie ou même d'envisager d'éventuels essais cliniques.

En effet, l'agence européenne du médicament (EMA) cite l'ICH 9 (International Conference of Harmonisation) à ce propos "Any conclusion of treatment efficacy (or lack thereof) or safety based solely on exploratory subgroup analyses are unlikely to be accepted." [18]

C'est pourquoi quelques sous-groupes sont sélectionnés **pour la phase confirmatoire** à l'aide d'un clinicien et de la consistance des sous-groupes, évaluée à travers l'utilisation de diverses méthodes de recherche de sous-groupes, ou d'un échantillon test.

Exemple

Dans le cadre du jeu de données *sepsis* présenté en section 4.4, nous nous trouvons dans un cas d'*efficacy*, où le traitement ne montre pas son efficacité dans la base totale puisque l'incidence (taux de survivant) dans le bras de traitement est de 41% alors que celle dans le bras de contrôle est de 34%. Cette différence n'est pas significative au test du khi-deux. Par construction de ce jeu de données simulé, nous savons qu'il existe un sous-groupe de patients bénéficiant de manière significative

du traitement. Ce sous-groupe de 128 patients est donné par les observations `PRAPACHE > 26` et `AGE > 49.8`, l'incidence est de 31% dans le bras contrôle, et de 83.7% dans le bras de traitement. Cette différence est significative.

5.2 L'approche classique

L'approche classique pour identifier des sous-groupes en statistiques est de créer un modèle qui inclut les interactions entre le traitement et les covariables.

Si l'on considère un essai clinique randomisé à deux bras, avec une variable réponse binaire notée $Y, Y = \{0, 1\}$, un indicateur du traitement noté T, $T = \{0, 1\}$ et des covariables notées $X = \left(X^{(1)}, \ldots, X^{(p)}\right) \in \mathcal{X} \subset \mathbb{R}^p$. L'objectif est alors de trouver un sous-espace de $\mathcal{X}$, noté A, pour lequel l'effet du traitement est meilleur que dans l'espace $\mathcal{X}$.

Par exemple, on pourrait considérer une régression logistique de la forme :

$$logit(P(Y = 1|T, X)) = \alpha + \beta T + \gamma h(X) + \theta T w(X) \tag{5.1}$$

Où h est une fonction de $\mathcal{X}$ dans $\mathbb{R}$, généralement $h(X) = X^{(1)} + \ldots + X^{(p)}$. w est une fonction de $\mathcal{X}$ dans $\mathbb{R}$, qui définit quelles interactions avec le traitement le modèle inclut. Par exemple, $w(X) = \sum_{j=1}^{p} X^{(j)} + \sum_{j \neq l} X^{(j)} X^{(l)}$. α et β sont des paramètres inconnus de dimension un. γ est un vecteur ligne de paramètres de dimension p. θ est un vecteur ligne de dimension égale au nombre de termes d'interactions inclus.

Pour la recherche de sous-groupes, $w(X)$ est le terme important. Seulement, seuls les termes présents dans l'interaction seront porteurs de sous-groupes. Un modèle avec toutes les interactions n'est pas vraiment faisable, au delà des problèmes de multiplicité des tests associés rendant l'interprétation des résultats biaisée, si le nombre d'observations est inférieur au nombre de paramètres à estimer, aucun modèle linéaire n'est envisageable. Pour *sepsis*, si nous voulions investiguer des sous-groupes définis par trois variables, il aurait fallu au minimum 1124 observations.

Ce que nous pouvons retenir d'un modèle comme (5.1), c'est qu'il est nécessaire de connaître à l'avance les sous-groupes susceptibles de définir A.

5.3 Virtual Twins

Ces dernières années, plusieurs méthodes ont vu le jour afin de trouver des sous-groupes, dont une en 2011, a été pensé et présenté par Jared Foster : *Virtual Twins* [1].

Virtual Twins est une méthode récente qui cherche à identifier sur une cible donnée des sous-groupes de patients avec un bénéfice significatif du nouveau traitement sur le comparateur. *Virtual Twins* présente l'intérêt de modéliser une cible binaire sur chacun des traitement et pour chaque patient. La différence entre les traitements est ensuite modélisée pour être la plus grande possible au sein d'un sous-groupe de sujets.

Concrètement, ***Virtual Twins* repose sur deux grandes étapes simples mais très ouvertes** :

On considère des données comme présentées dans la section 5.2.

1. **1er étape**, $\forall i = 1, \ldots, n$ *Virtual Twins* **estime** $\mathbb{P}(Y_i = 1|T_i = 1, X_i)$, notée $\hat{P_{1i}}$ et $\mathbb{P}(Y_i = 1|T_i = 0, X_i)$, notée $\hat{P_{0i}}$, qui sont respectivement, la probabilité de faire l'événement sachant le nouveau traitement, et la probabilité de faire l'événement sachant le traitement de contrôle.

2. **2nd étape**, *Virtual Twins* **explique** la différence $Z_i := \hat{P_{1i}} - \hat{P_{0i}}$. Ce qui correspond à la variation de la réponse d'un patient en fonction du traitement.

Il est laissé libre à l'utilisateur de choisir quelles méthodes peuvent fournir ces deux étapes. Dans l'article original de *Virtual Twins* [1], l'auteur utilise les forêts aléatoires pour la première étape, puis les arbres CART pour la seconde.

Dans ce rapport nous utiliserons ces deux méthodes : la forêt aléatoire a su prouver, par sa théorie, et ses diverses applications son avantage pour ce type de classifications, l'arbre CART est une excellente méthode de description d'un jeu de données, comme vu précédemment.

5.3.1 Etape 1 : La forêt aléatoire

Dans cette partie, nous faisons le lien avec les écritures introduites dans les chapitre 3 et 4. Dans un premier temps, nous construisons une forêt aléatoire qui est un ensemble de classifieurs $\{h_k(.,\theta_k),\ k=1,\dots,K\}$. La variable explicative étant Y, chaque classifieur qui constitue la forêt est construit sur les variables (X,T), on a alors un ensemble de classifieurs qui est $\{h_k((X,T),\theta_k),\ k=1,\dots,K\}$. On peut alors estimer pour chaque patient la probabilité $\mathbb{P}(Y_i=1|T_i,X_i)$ qui correspond à $\hat{P_{1i}}$ si $T_i=1$, et qui correspond à $\hat{P_{0i}}$ si $T_i=0$. Voici comment calculer cette estimation :

$$\mathbb{P}(Y_i=1|T_i,X_i)=\frac{1}{K}\sum_{k=1}^{K}\mathbf{1}\left(h_k((X_i,T_i),\theta_k)=1\right)$$

Dans un second temps, on altère le jeu de données en remplaçant, pour chaque patient son traitement par celui qui ne lui a pas été donné. Mathématiquement, on a alors le nouveau traitement noté T', $T'=1-T$. Autrement dit, pour chaque patient i, $T'_i=1-T_i$, i.e.

$$\forall i=1,\dots,n,\ T'_i=\begin{cases}1 & \text{si } T_i=0\\ 0 & \text{si } T_i=1\end{cases}$$

Enfin, en utilisant le caractère prédictif des classifieurs, on peut alors estimer pour chaque patient la probabilité $\mathbb{P}(Y_i=1|T'_i,X_i)$ qui correspond à $\hat{P_{1i}}$ si $T'_i=1$, et qui correspond à $\hat{P_{0i}}$ si $T'_i=0$

Le résultat de cette étape permet d'avoir pour chaque patient sa probabilité de faire l'événement sachant les deux traitements. Mathématiquement, on possède la suite $\left(\hat{P_{1i}},\hat{P_{0i}}\right)_{i=1,\dots,n}$

Les variantes

D'après les simulations numériques, il est possible d'améliorer les performances de la méthode en incluant en variables explicatives les interactions entre le traitement et les X, que l'on note $X\mathbf{1}(T=1)$ et $X\mathbf{1}(T=0)$. La forêt donne alors un ensemble de classifieurs qui est $\{h_k((X,T,X\mathbf{1}(T=1),X\mathbf{1}(T=0)),\theta_k),\ k=1,\dots,K\}$.

Remarque. *Avec les interactions, il est alors obligatoire que les variables quantitatives soit transformées en variables binaires.*

Il existe également une autre variante qui divise l'échantillon en deux échantillons suivant le traitement. On note $\mathcal{L}^1$ l'ensemble des patients tels que $T=1$, et on note $\mathcal{L}^0$ l'ensemble des patients tels que $T=0$.

On construit une première forêt sur l'échantillon $\mathcal{L}^1$, ce qui nous donne pour tous les patients appartenant à ce groupe, $\hat{P_{1i}}$. De même on construit une seconde forêt sur l'échantillon $\mathcal{L}^0$, ce qui nous donne pour pour tous les patients appartenant à ce groupe, $\hat{P_{0i}}$. Puis, on altère les traitements, en posant $T'=1-T$. Enfin, la forêt non utilisée pour calculer $\hat{P_{ji}}$ est utilisée pour calculer $\hat{P_{(1-j)i}}$.

5.3.2 Etape 2 : l'arbre de décision CART

La deuxième étape consiste à utiliser la suite $\left(\hat{P_{1i}}, \hat{P_{0i}}\right)_{i=1,\dots,n}$ afin de construire deux nouvelles variables Z et Z^* définies pour tout i par

$$Z_i = \hat{P_{1i}} - \hat{P_{0i}}$$

$$Z_i^* = \begin{cases} 1 & Z_i \geq c \\ 0 & Z_i < c \end{cases}$$

où c est un seuil choisi en fonction de l'effet attendu du traitement.

Notre domaine d'intérêt est l'ensemble des $Z_i^* = 1$ car ce sont les observations qui bénéficient positivement du traitement. Pour connaître les variables explicatives qui définissent ce domaine d'intérêt nous construisons un arbre de classification CART, comme vu précédemment au chapitre 3, sur la variable binaire Z^*.

Nous définissons alors $\hat{A}$, l'estimation de l'ensemble A (i.e du sous-groupe recherché), l'ensemble des noeuds terminaux de l'arbre tels que la classe qui leur est affectée est 1. Cela revient à définir une règle sur les observations qui permet d'appartenir à $\hat{A}$. Par exemple, $\hat{A}$ peut être défini par une règle : $\left\{X^{(1)} > 0 \cap X^{(5)} < 3\right\} \cup \left\{X^{(1)} < 0 \cap X^{(20)} = 1\right\}$, c'est ce qu'on appelle un sous-groupe. Dans ce cas, nous dirons qu'il y a deux sous groupes : $\left\{X^{(1)} > 0 \cap X^{(5)} < 3\right\}$ et $\left\{X^{(1)} < 0 \cap X^{(20)} = 1\right\}$.

On utilise donc l'arbre ici à des fins de description, mais il est tout à fait possible d'évaluer ce sous groupe en l'appliquant à un échantillon test.

Variante

Une variante de l'arbre de classification est possible en appliquant à la variable Z un arbre de régression, puis en déterminant avec le même seuil c quels noeuds terminaux définissent $\hat{A}$. Nous avons décidé de ne pas tenir compte de cette méthode car en faisant varier c dans la première méthode, l'approche semble proche de cette seconde méthode. De plus, des résultats préliminaires ont montré des résultats équivalents pour ces deux approches. Et enfin, il s'avère que cette méthode est moins interprétable.

5.4 Sélection de sous-groupes

Une fois en possession de sous-groupes, il faut trouver une méthode pour les évaluer, en sélectionner et y accorder une crédibilité.

Avant cela, nous pouvons faire une remarque sur l'ensemble $\hat{A}$ créé. Étant donné qu'il est défini grâce à une structure sous forme d'arbre, la première variable rencontrée fera toujours partie des sous groupes définis. *Virtual Twins* semble plutôt définir un sous-groupe (que l'on peut voir comme plusieurs sous sous-groupes), plutôt qu'une multitude. Dans la pratique, ce n'est que parmi ce sous-groupe $\hat{A}$ que nous choisirons alors **un seul** sous ensemble noté S qui sera alors notre **sous-groupe d'intérêt**. Pour faire suite à notre exemple, nous prendrions par exemple $S = \left\{X^{(1)} > 0 \cap X^{(5)} < 3\right\}$ car il correspondrait aux critères ci dessous :

Tout d'abord, à chaque sous groupe noté S, nous associons une mesure de qualité définie par le risque relatif, que nous notons RR, défini par

$$RR(S) = \frac{\mathbb{P}(Y = 1 | T = 1, X \in S)}{\mathbb{P}(Y = 1 | T = 0, X \in S)}$$

On cherche alors à trouver des sous-groupes qui possèdent un RR élevé (par rapport au RR de l'échantillon total) si l'événement d'intérêt est 1. Dans le cas contraire, on cherche un RR plus faible.

En *efficacy*, si l'essai est neutre, l'échantillon total possède un $RR \approx 1$. Le sous-groupe recherché devra posséder un RR bien supérieur à 1, si l'événement désirable est associé à 1. Le complémentaire n'est pas étudié.

En *safety*, l'échantillon total possède un $RR > 1$. Le sous-groupe recherché est un sous groupe qui concentre un maximum d'événements indésirables, ce que l'on associe à un RR élevé, très supérieur à 1. Le but indirect est de retrouver un $RR \approx 1$ dans le complémentaire de S, noté ${}^cS$.

La taille du sous-groupe est également importante. En *safety*, nous privilégions les petits sous-groupes. En *efficacy*, nous privilégions les plus grands sous-groupes.

Un sous-groupe est aussi caractérisé par une profondeur, qui est le nombre de variable qui le compose. Idéalement, nous nous concentrons sur des sous-groupes de profondeur maximale deux voire trois.

Un sous-groupe sera présenté à un clinicien s'il possède une forme de consistance, c'est à dire qu'il n'est pas seulement dû au hasard. Cette partie est délicate, et l'équipe de data mining de Servier est plutôt partisane de la recherche de sous-groupes via différentes approches assez avancées, comme SIDES [20] ou Interaction Trees. Une méthode également robuste est d'utiliser un échantillon test, ou faire de la validation croisée.

Le problème est d'estimer sans biais $RR(S)$, Jared Foster propose cinq méthodes dont trois sont présentées ci-dessous.

La première étant de calculer $RR(S)$ empiriquement (par resubstitution) grâce aux données, ce qui peut provoquer dans le cas de petits échantillons un problème de sur-apprentissage.

La second est appelé SND (pour *Simulate New Data*), et on estime $RR(S)$, grâce aux P_{ij}, par

$$\hat{RR}(\hat{S}) = \frac{\frac{1}{|\hat{S}|}\sum_{X_i \in \hat{S}, T_i=1} \hat{P_{i1}}}{\frac{1}{|\hat{S}|}\sum_{X_i \in \hat{S}, T_i=0} \hat{P_{i0}}}$$

Comparé à la première, elle dépend moins des données et possède théoriquement moins de biais.

La troisième est d'utiliser la validation croisée k-fold pour calculer les $\hat{P_{ij}}$, puis d'appliquer l'une des deux méthodes précédentes.

Les deux autres méthodes qui ne seront pas présentées ici, ont vocation à estimer la qualité de construction de $\hat{A}$, qui est variable, tandis qu'ici, nous avons sélectionner empiriquement un sous groupe S fixe dont nous voulons connaître sa consistance. Comme en pratique la sélection passe par une connaissance du domaine et une validation croisée par d'autres techniques, ces deux méthodes ont été mises de côtés. Cependant, arrivé en fin stage, et avec plus de recul, il semblerait qu'une adaptation de l'une d'entre elle est possible.

Finalement, **l'interprétation et le rationnel clinique** du sous-groupe est de loin le critère le plus **important** pour la sélection de sous-groupes.

Une fois un sous-groupe sélectionnée et confirmé, il apporte alors une connaissance supplémentaire au produit et à la pathologie. Ces informations sont précieuses et peuvent débouchées à des articles scientifiques et de possibles nouvelles études.

5.5 État de l'art

L'article original de *Virtual Twins* écrit par Jared Foster est accompagné d'une simulation : le jeu de données est généré par le modèle type donné par l'équation 5.1. $n = 400$, $p = 15$, le sous

groupe est de profondeur deux et les $X \sim \mathcal{N}(0,1)$. Il compare le modèle logistique et *Virtual Twins* . Ce qui est intéressant de noter c'est que *Virtual Twins* trouve 2 à 3 fois plus souvent les termes du sous-groupe que le modèle logistique, mais trouve également plus de variables inutiles que la régression logistique.

Dans le cas où les données sont simulées sans sous-groupe, la modèle logistique ne trouvera aucune interaction à 85%, tandis que *Virtual Twins* trouvera quasiment à chaque fois un sous-groupe.

Virtual Twins est une approche non paramétrique de la recherche de sous-groupe. Ces dernières années plusieurs méthodes sont apparues de façon indépendantes [21] basées sur des approches de régressions paramétriques, comme MOB, Interaction trees, STIMA. SIDES est également une approche sans régressions, mais contrairement à *Virtual Twins* , SIDES propose plusieurs sous-groupes qui peuvent s'intercepter. MOB et STIMA peuvent être utilisé dans d'autres contexte qu'avec *effet traitement*. Néanmoins toutes ces méthodes utilisent le partitionnement récursif. La conclusion donnée dans l'article comparant ces méthodes montre bien qu'il faut s'aider de plusieurs méthodes pour proposer une bonne explication de la base.

GUIDE propose également une méthode de recherches de sous-groupes : Gi, Gc, et Gr [22]. Les auteurs de ces méthodes les comparent à *Virtual Twins* , SIDES et Interaction Trees sur trois scénarios pas simples. Mais nous retrouvons que seul *Virtual Twins* et Gs trouvent un sous-groupe alors qu'il n'y en a pas.

L'un des scénario implique un sous-groupe définit par deux variables qui font également parties des variables prédictives sans interactions, dans ce cas *Virtual Twins* est légèrement moins performant que les autres méthodes, excepté Gs.

Le scénario final est construit avec un sous-groupe, dont les variables apparaissent seulement dans l'interaction avec le traitement. Dans ce dernier cas, *Virtual Twins* et Gs sont les plus performantes car elles trouvent à environ 90% la première et la deuxième variables du sous groupe. Les scénarios ont été construits avec 100 variables avec trois catégories (pour imiter des biomarqueurs génétiques) pour $n = 100$. Les incidences dans chaque bras varient entre 20% et 80%.

Comme nous pouvons le constater, les méthodes ne manquent pas pour la recherche de sous-groupes avec effet traitement, chacune d'entre elle a sa particularité et son type d'application.

Jusqu'à maintenant les applications proposées avec *Virtual Twins* dans les différents articles ne correspondent pas à l'utilisation que Servier pourrait en faire. C'est pourquoi dans la suite, nous proposons une étude de cas et une série de simulations. Dans le but d'appliquer *Virtual Twins* de façon simple et automatique, j'ai proposé de développer une routine sous `R`, qui a donné lieu à un package du nom de `aVirtualTwins` [16].

Chapitre 6

Etude de cas et simulations

6.1 Étude de cas de safety

Par soucis de confidentialité, l'étude qui va être présentée ci-après ne sera divulgué : le nom du médicament, le nom de l'étude, la date, les variables, les effectifs exacts, etc... seront inconnus. Seules les données statistiques importent dans notre cas.

L'étude est un cas de *safety* d'un nouveau traitement contre un comparateur, elle présente environ 9000 patients et 99 variables dont la variable cible représentant la présence ou non de l'effet indésirable d'intérêt. Nous notons $Y = 1$ l'événement d'intérêt, et $Y = 0$ sinon.

Il y a une trentaine de variables continues qui sont principalement des mesures de concentrations (exemples : lymphocyte, sodium). Le reste étant des variables binaires représentant des antécédents médicaux, des prises de médicaments (exemples : multi-vitamines et fer) ou encore des symptômes à noter (exemples : mal de dos). Ces variables sont précises, elles ont donc peu d'événements : de l'ordre de quelques pour-cents.

Il y a environ 4500 patients dans chaque bras. Les incidences (i.e. pourcentage de l'événement d'intérêt, ici l'effet indésirable) sont de 1.6% dans le bras du nouveau traitement et 0.9% dans le bras de contrôle. Cette différence de proportions est significative au khi-deux. Il y a donc un problème : le nouveau traitement entraîne plus d'effets indésirables que le comparateur. Le RR global est de $\frac{0.016}{0.009} \approx 1.78$, ce qui est beaucoup.

Première étape de *Virtual Twins*

Nous appliquons alors *Virtual Twins* sur cette étude. Il va se poser la problématique des classes déséquilibrées, car nous avons 117 cas d'intérêt contre 8883. la forêt aléatoire cherche à minimiser le risque de classification, ici, le plus simple est de classifier tous les cas dans la classe majeur, le risque est alors de $\frac{117}{8883} \approx 1.3\%$. Chao Chen propose deux solutions [23], l'une est d'utiliser les poids donnés aux classes qui se répercute sur la fonction d'impureté mais cette technique n'est pas (ou mal) implémentée dans le package `R randomForest`. La second technique consiste à bootstrapper pour augmenter la taille de la classe minoritaire. Nous décidons de plutôt réduire la classe majeure et d'augmenter le nombre d'arbre.

Nous proposons huit façons d'exécuter *Virtual Twins* présentée dans le tableau 6.1. Nous appelons forêt avec contrôle lorsque nous construisons l'échantillon bootstrap de chaque arbre de la forêt avec le même nombre d'observations de chaque classe, ici, il serait de 117 observations de classe $Y = 1$, et 117 événements de classe $Y = 0$. Nous appelons forêt avec interactions lorsque nous ajoutons les termes $X\mathbf{1}(T)$ comme décrit au paragraphe 5.3.1. Nous appelons sensibilité la probabilité que la forêt classifie une observation comme faisant un événement indésirable sachant que $Y = 1$. Nous appelons la spécificité la probabilité que la forêt classifie une observation comme ne faisant pas un événement indésirable sachant que $Y = 0$. Et le risque est la probabilité de mal classer une observation.

Numéro	Forêt	Interaction	Contrôle	Spécificité	Sensibilité	Risque
1	Simple	non	non	1	0	0.013
2	Simple	oui	non	1	0	0.013
3	Simple	non	oui	0.885	0.28	0.123
4	Simple	oui	oui	0.866	0.342	0.141
5	Double	.	non	1 / 1	0 / 0	0.009 / 0.016
6	Double	.	oui	0.88 / 0.891	0.262 / 0.361	0.126 / 0.117
7	5-fold	oui	non	1	0	0.013
8	5-fold	oui	oui	0.934	0.193	0.075

TABLE 6.1 – Différentes approches de *Virtual Twins*

Numéro	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Sd
1	-0.63	0	0	0	0.01	0.65	0.07
2	-0.15	0	0	0.01	0.02	0.18	0.02
3	-0.5	0	0.01	0.01	0.02	0.5	0.04
4	-0.15	0.06	0.11	0.12	0.17	0.42	0.08
5	-0.16	0	0	0.01	0.02	0.17	0.02
6	-0.37	-0.06	0.01	0.01	0.07	0.43	0.1
7	-0.14	0	0	0	0.01	0.17	0.02
8	-0.16	0.05	0.1	0.1	0.15	0.38	0.08

TABLE 6.2 – Distribution de la variable Z

Les histogrammes des différentes distributions de Z en annexe 3 illustrent le tableau 6.2.

Nous nous intéressons à la distribution de Z, puisqu'elle correspond à la variation de la probabilité de faire l'événement de chaque patient par rapport au traitement. Plus Z est grand, plus cette variation est importante, et par conséquent plus le nouveau traitement influe sur la probabilité de faire l'événement indésirable.

Dans le cas des approches non contrôlées (qui correspondent 1, 2, 3, 5 et 7) la distribution n'est pas étalée, la variance est faible (la variance de l'approche 1 est artificiellement forte à cause de deux outliers). Au mieux nous pourrons expliquer qu'une très faible variation, ce qui n'est pas satisfaisant.

Dans le cas des approches 4, 6 et 8, les distributions sont étalées (respectivement une variance de 0.8, 0.1 et 0.8). Nous pouvons remarquer que les distributions de l'approche 4 et 8 sont, au centième près, équivalente : elles sont centrées en 0.1, de variance 0.8 et les quartiles sont sensiblement les mêmes. Il est intéressant de noter qu'au moins un quart des patients ont une variation de probabilité supérieur à environ 0.15. Les deux approches se distinguent par leur risque de classification : l'approche 8 possède un risque de 7.5% alors que l'approche 4 a un risque de 14.1%.

La distribution 6, bien qu'étant la plus variable, est centrée en 0.01, et un quart des patients ont une variation de seulement 0.7. Cette distribution est douteuse car elle ne montre pas une tendance à faire l'événement indésirable sous le nouveau traitement, ce qui est justement le but de cette recherche de sous-groupes. De plus, construisant deux forêts, l'une pour $T = 1$ et l'autre pour $T = 2$, on divise encore le nombre de cas dans chaque échantillon bootstrap.

Nous notons que le contrôle de l'échantillon bootstrap de chaque arbre en plus des termes d'interactions semble donner à la distribution Z une variance plus élevée.

Deuxième étape de *Virtual Twins*

Nous choisissons les approches 4, 6 et 8 pour appliquer la deuxième partie de la méthode *Virtual Twins* . La seconde étape consiste à choisir un seuil c et construire un arbre de décision sur la variable Z^*, comme vu en section 5.3.2.

Pour chaque forêt nous faisons varier le seuil c afin de voir ce que donne l'algorithme lorsque nous devenons plus restrictif sur l'ensemble de patients que nous cherchons à expliquer. Pour cela, nous définissons quatre seuils qui sont quatre déciles de la distribution Z, à savoir le décile 5 (la médiane), le décile 6, le décile 7 et le décile 8. Ce qui nous permet d'expliquer respectivement la moitié, 40%, 30% et 20% des plus grandes variations de Z. Pour rappel, les grandes variations de Z impliquent que le traitement a une forte influence sur la probabilité de faire l'événement d'intérêt.

Nous avons fait le choix également de stopper la construction de l'arbre à deux niveaux car nous voulons trouver des sous groupes simple et de profondeurs deux maximum.

Les résultats sont inscrits dans le tableau 6 en annexe. La colonne *déc.* signifie *décile*. Nous avons rendu les variables anonyme en leur attribuant un numéro. Nous voyons que seules les trois variables $X^{(1)}$, $X^{(2)}$ et $X^{(3)}$ interviennent dans les sous-groupes identifiés. Les tailles de sous-groupes sont comprises entre 4535 (50% de la base) et 1014 (11% de la base). Les estimations du RR ne sont pas du même ordre de grandeur par *snd* que par *resub*, cependant la relation $RR^{snd}(S) > RR^{snd}({}^cS) \iff RR^{resub}(S) > RR^{resub}({}^cS)$ excepté pour le premier sous-groupe de l'approche 8.

On remarque également que pour l'approche 4 et 6, plus c augmente, plus l'effectif du sous-groupe diminue. Concernant l'approche 8 ce n'est pas le cas, mais aussi bien $RR^{resub}(S)$ que $RR^{snd}(S)$ augmente quand c augmente, ce qui s'avère faux pour l'approche 4.

Discussion et sélection

Une fois ces quelques sous-groupes trouvés, il faut en sélectionner quelques-uns qui seront à présenter aux cliniciens. Les façons de faire peuvent être différentes, cependant nous allons premièrement nous concentrer sur l'effectif du sous-groupe. En effet, nous sommes dans un cas de *safety*, et l'objectif est de trouver un petit sous-groupe qui sur-concentre l'événement d'intérêt afin de proposer une contre-indication, et que le complémentaire ait un RR non significatif.

Les trois sous-groupes qui présente un intérêt sont les trois combinaisons possibles des variables que l'on trouve au décile 8 de chaque approche. Dans l'approche 6, le sous groupe est d'effectif 1982, ce qui représente environ 22% de la base. Nous devrions pas le sélectionner, mais il possède un RR fort. Les deux autres sous-groupes possèdent également un RR^{resub} élevé (≈ 10).

Bien entendu, *Virtual Twins* seul ne suffit pas, et il faut s'aider d'autres méthodes (comme SIDES), pour étayer ce type de résultat.

Une fois que cela est fait, nous pouvons nous adresser aux cliniciens et présenter les hypothèses. Dans ce cas, la variable $X^{(2)}$ était non interprétable. La variable $X^{(1)}$ a des impacts qui étaient déjà connus pour ce médicament. Quant à l'influence de la variable $X^{(3)}$, cela a permis de découvrir une restriction sur le nouveau produit. A l'heure actuelle, la recherche de sous-groupe a pu permettre une contre indication approuvée par les autorités réglementaires sur le médicament afin d'éviter un risque pour les patients le prenant.

6.2 Simulations

Dans le but de pouvoir positionner la méthode, comprendre ces points faibles, ces champs d'applications dans les contraintes qu'imposent les essais cliniques, nous procédons à quelques simulations.

Les jeux de données répondant à la demande de recherche de sous-groupes au sein du laboratoire comprennent souvent entre 100 et 1000 variables. Nous choisirons 120 variables explicatives.

La variable réponse peut être binaire ou continue, mais que l'on binarise souvent. Dans notre cas, nous nous concentrons sur une réponse binaire. Comme vu précédemment dans l'exemple le taux de l'événement d'intérêt est habituellement faible, de l'ordre de 15% à 30% pour les études d'*efficacy*, et de l'ordre de quelques pourcents en cas de *safety*.

Concernant le nombre de patients, les études sont variables, nous choisirons de préférence un nombre de patients élevés, de l'ordre de quelques milliers de patients. Les variables explicatives sont majoritairement des variables binaires, principalement avec une incidence faible (0.5% à 20%).

Enfin, nous privilégions les sous-groupes de profondeur deux, composés d'une variable binaire et une variable continue.

6.2.1 Méthode de simulation

Nous expliquons ici la façon dont nous simulons nos jeux de données. Introduisons les notations suivantes : n est l'effectif total. n_1 est le nombre de patients dans le bras du nouveau traitement, noté $T = 1$. n_0 est le nombre de patients dans le bras du traitement comparateur, noté $T = 0$. S est le sous-groupe définit au préalable. n^S (resp. n^{cS}) est le nombre de patient dans le sous-groupe S (resp. dans le complémentaire ${}^cS$). n_i^S est le nombre de patients dans le sous groupe dans le bras de traitement $T = i$ ($i = 1$ ou 2). n_i^{cS} est le nombre de patients dans le complémentaire du sous groupe dans le bras de traitement $T = i$ ($i = 1$ ou 2). n^S (resp. n^{cS}) est le nombre de patient dans le sous groupe (resp. le complémentaire du sous groupe).

Enfin on note I_i^S (resp. I_i^{cS}) l'incidence de l'événement d'intérêt dans le sous groupe (resp. le complémentaire du sous groupe) et le bras de traitement $T = i$. Nous notons également I_i l'incidence dans le bras de traitement $T = i$. Plus clairement nous présenterons les plans de simulations sous la forme du tableau 6.3.

Ensemble	Effectif total	Effectif $T = 1$	Effectif $T = 0$	Inc. $T = 1$	Inc. $T = 0$
S	n_1^S	n_1^S	n_0^S	I_1^S	I_0^S
${}^cS$	n_0^{cS}	n_1^{cS}	n_0^{cS}	I_1^{cS}	I_0^{cS}
$\mathcal{X}$	n	n_1	n_0	I_1	I_0

TABLE 6.3 – Plan de simulation type

Nous simulons 120 $X^{(i)}$ variables indépendantes et non corrélées en suivant le tableau 6.4.

Remarque. *La variable* $X^{(i)}, i = 101, \ldots, 120 \sim \mathcal{B}(0.03)$ *pour la simulation 7.*

i	1 à 20	21 à 30	31 à 40	41 à 50	51 à 60	61 à 80	81 à 100	101 à 120
$\mathcal{L}(X^{(i)})$	$\mathcal{N}(0,1)$	$Log\mathcal{N}(0,1)$	$\mathcal{B}(0.5)$	$\mathcal{B}(0.4)$	$\mathcal{B}(0.3)$	$\mathcal{B}(0.2)$	$\mathcal{B}(0.1)$	$\mathcal{B}(0.05)$

TABLE 6.4 – Lois des variables explicatives simulées

Puis, nous affectons sans aléatoire le traitement. Si l'essai clinique est équilibrée, nous affectons au $n/2$ premiers patient la valeur 1 à la variable T_i, pour désigner le nouveau traitement. La moitié restante se voit attribuée la valeur 0.

Enfin pour tout $i = 1, \ldots, n$, nous simulons la variable aléatoire Y_i comme étant une loi de Bernoulli comme décrit dans le tableau 6.5.

	$T_i = 0$	$T_i = 1$
$i \in S$	$\mathcal{B}(I_0^S)$	$\mathcal{B}(I_1^S)$
$i \in {}^cS$	$\mathcal{B}(I_0^{cS})$	$\mathcal{B}(I_1^{cS})$

TABLE 6.5 – Loi de la variable réponse simulée

6.2.2 Description des simulations

En annexe 7, le tableau résume les simulations effectuées que nous présenterons. Nous avons choisis de tester trois scénarios d'*efficacy* et trois de *safety*, que nous avons augmenté par la suite par quatre autres scénarios.

Remarque. *Il existe une multitude de scénarios possibles que l'on pourrait tester, cependant il est nécessaire de faire un choix. Les propositions suivantes sont majoritairement des scénarios que le laboratoire a déjà rencontrés dans le passé.*

Efficacy

En *efficacy*, le principe est de fixer l'incidence I_1 et I_0, de définir un sous groupe de patients assez large, et de définir $I_1^S >> I_0^S$ de façon à avoir une différence significative. Le but n'est pas de définir une différence précise, comme pourrait le faire un vrai essai clinique avec le calcul de la taille d'échantillon, mais de créer artificiellement une sous population qui répond mieux au traitement de manière significative. L'incidence dans le groupe complémentaire est maintenu équivalente (i.e. $I_1^{cS} \approx I_0^{cS}$ et le test n'est pas significatif.). A noter que le test utilisé est celui du khi-deux.

La simulation 1 propose un taux de réponse à 15%, et un sous-groupe de taille 3000, ce qui représente 15% de la base. La simulation 2 possède un taux de réponse à 20%, et un sous-groupe de taille 180, pour une taille total de l'échantillon à 1000 patients. La simulation 3 a un taux de réponse de 30%, et un sous-groupe représentant 15% de la base dont la taille totale est de 3000. Contrairement aux deux autres simulations, la simulation 3 a deux bras déséquilibrés, cela signifie que les effectifs dans les deux bras de traitement sont différents. Ici, $n_1 = 2000$ et $n_0 = 1000$.

Safety

En *Safety*, le principe est de fixer l'incidence I_1 et I_0 de façon à ce qu'elles soient significatives, mais sans être trop différentes. Ici nous choisissons $I_1/I_2 \approx 2$ pour nos simulations. Nous construisons alors un sous-groupe de petit effectif qui sur-concentre l'événement, de façon à ce que $I_1^S >> I_0^S$. Dans le complémentaire, on a $I_1^{cS} \approx I_0^{cS}$ et cette différence n'est pas significative. Les tests ici sont utilisées à titre informatif, car ce qui importe le plus c'est de bien définir une construction déséquilibrée au sein d'une même population.

La simulation 4 possède un sous-groupe avec un risque relatif de 6 sur 6% de la base ($n = 10000$). La simulation 5 ne définit le sous-groupe que par une seule variable continue qui couvre 20% de la base ($n = 6000$) avec un risque relatif de 4, mais les bras sont déséquilibrés, $n_1 = 5000$ et $n_0 = 1000$.

Les simulations 6 et 7 sont des variantes de la simulation 4, nous divisons par deux la taille du sous-groupe et multiplions par deux le risque relatif, dans le cadre d'un sous-groupe de profondeur deux, puis un.

Les simulations 8 et 9 sont des variantes de la simulation 5, nous divisons par deux la taille du sous-groupe sans changer le risque relatif. L'un des cas est pour un sous-groupe de profondeur deux, l'autre reste le même que la simulation 5.

La simulation 10 est une simulation moins réel mais qui apporte des informations sur *Virtual Twins* .

Quelle(s) approche(s) de *Virtual Twins* utilisée(s) ?

Nous l'avons vu, il existe beaucoup de variantes à *Virtual Twins* . Motivé par de précédentes simulations non rapportées ici, nous avançons qu'il existe peu de différence entre les forêt doubles, ou simple. De même, les forêts que nous appelons contrôlées n'ont pas montré d'écarts de résultats précédemment lorsque les forêt simples échouaient. Enfin, le choix d'ajouter les termes d'interactions ne se pose pas réellement, le papier original préconise leurs utilisations [1], et nous avons également établi qu'ils amélioraient la distribution de la variable Z.

Nous appliquons alors à chaque scénario une forêt simple (notée S). Nous réalisons parfois en supplément une forêt contrôlée (notée SC).

Concernant les paramètres de la fonction `randomForest`, nous les laissons par défaut, excepté le paramètre `ntree` auquel nous affectons la valeur 1000.

Ces choix sont de toute évidence discutables, mais essayer toutes les méthodes nous aurait coûter un temps de calcul démesuré. De plus, une simulation n'est pas un cas réel, l'objectif alors est de positionner la méthode vis à vis de plusieurs scénarios.

Chaque scénario de simulation est effectué 1000 fois. Et nous enregistrons les sous-groupes trouvés pour chaque décile de la distribution de Z.

De plus, on définit deux mesures que l'on enregistre pour une partie des simulations. Le recouvrement :

$$\text{recouvrement} = \frac{|\hat{A} \cap S|}{|S|}$$

et la taille relative

$$\text{taille relative} = \frac{|\hat{A}|}{|\hat{A} \cap S|}$$

où $\hat{A}$ est l'ensemble trouvé par *Virtual Twins* comme définit en 5.3.2. S (confondu à A) est le sous-groupe construit. Enfin nous notons $|.|$ le cardinal de l'ensemble considéré. Plus le recouvrement et la taille relative sont proches de 1, mieux c'est. Les résultats de ces mesures se trouvent en annexe 8.

6.2.3 Résultats

Les graphiques en annexe 4 et 5 présentent les résultats sous forme de courbe des simulations. Pour chaque simulation ayant un sous-groupe de dimension deux, nous visualisons la courbe noir, qui représente combien de fois le premier split de l'arbre correspond à l'une des deux variables formant le sous-groupe construit, ce chiffre est divisé par le nombre d'arbre, pour chaque décile. Nous dénommons cette courbe *correct 1st split*. La seconde courbe correspond au nombre de fois où le bon sous groupe est trouvé par l'arbre, ce chiffre est divisé par le nombre d'arbre, ce qui donc la fréquence pour chaque décile. Nous dénommons cette courbe *correct subgroup*.

Remarque. *Nous décidons que le* bon sous groupe *est trouvé s'il existe dans $\hat{A}$ le sous-groupe S ou une partie de S. L'idée est de vérifier si au moins les deux variables forment $\hat{A}$, et si les inégalités sont dans le bon sens comparé à S. Ici nous ne présenterons pas la valeur des splits, mais nous nous sommes assurées que ceux ci était en moyenne ceux de S. Quoi qu'il en soit, la construction des sous-groupes implique que, si la bonne variable est choisie, la fonction d'impureté coupe par mécanique cette variable autour de la valeur que nous avons prédéfinie dans S. Enfin, l'objectif des simulations est plutôt centré sur la l'apparition des bonnes variables.*

Dans le cas où S est défini à l'aide d'une seule variable, les deux courbes sont confondues.

Efficacy

Les simulations en *efficacy* correspondent aux cas 1, 2 et 3. *Virtual Twins* dans le cas des simulations 1 et 2 permet de ne trouver qu'un quart du temps environ une des deux variables qui composent le sous groupe en first split, ce qui est très faible. Concernant la simulation 3, il y a plus d'une chance sur deux que la première variable fasse partie du sous-groupe dans le cadre de la forêt non contrôlée, et au décile 7. La forêt contrôlée ne donne pas de meilleurs résultats. *Virtual Twins* ne trouve presque jamais le bon sous-groupe dans ces trois simulations.

Les résultats à propos du recouvrement et de la taille relative ne sont pas étonnant pour la simulation 3. En restreignant les patients à expliquer, le sous-groupe estimé devient de taille plus petite et manque de précision puisque qu'il ne recouvre que 14% au décile 8 pour environ 2 fois cette taille recouverte. A noter également que contrôler la forêt n'améliore pas les résultats, au contraire.

Safety

Les simulations 4 à 10 correspondent à des cas de *Safety*. Concernant le premier split de la simulation 4 sans contrôle, plus le décile est élevé plus la fréquence semble diminuer de 66% à 56%. A l'inverse, avec forêt contrôlée cette fréquence augmente de 15% pour atteindre 58% au décile 8. La variable binaire est la variable majoritairement choisie en premier split. La fréquence du bon sous-groupe est très faible, avec une légère hausse aux déciles 7 et 8.

La simulation 6 est une variante de la simulation 4. S n'est formé que d'une seule variable binaire. Ici, la fréquence d'apparition du bon sous-groupe est élevée : de 94% au décile 5 à 99,4% au décile 8.

La simulation 7 est une variante de la simulation 6 : S est formé de deux variables. La fréquence d'apparition du bon premier split pour la forêt sans contrôle semble décroître de 79% pour le décile 5 à 71% pour le dernier décile. La variable binaire est également majoritairement choisie. Quant à la forêt contrôlée, l'augmentation de la fréquence d'apparition du bon premier split augmente fortement de 29% à 78%. Cette fois la variable continue est plus choisie aux déciles 5,6 et 7 pour finalement laisser place à la variable binaire qui est deux fois plus choisie au décile 8.

Concernant la fréquence d'apparition du bon sous-groupe, les deux forêts montrent une tendance à la hausse avec l'augmentation du seuil c, mais la forêt non contrôlée atteint une valeur deux fois plus élevée d'environ 50%.

la simulation 5 est construite à l'aide d'un sous-groupe de profondeur un formé d'une variable continue. La fréquence de bon sous-groupe est de 100% pour tous les déciles.

La simulation 8 est une variante de la simulation 5, le sous-groupe S est également formé d'une seule variable continue, et de même, la fréquence du bon sous-groupe est au maximum pour les quatre déciles.

La simulation 9 est une variante de la simulation 8 : le sous-groupe S est formé d'une variable continue et binaire. La fréquence d'apparition du bon premier split est de l'ordre de 99% pour la forêt S, alors que dans le cas de la forêt SC, elle n'atteint 98% que pour le dernier décile. La variable binaire est pratiquement toujours choisie. La fréquence du correcte sous-groupe est variable selon les déciles, mais atteint sa plus forte valeur au décile 8 avec 95% pour la forêt S, et 81% pour la forêt SC.

Pour la simulation 10, *Virtual Twins* trouve toujours en premier split une des deux variables formant le sous-groupe S, que ca soit en forêt S, ou SC. La fréquence d'apparition du bon sous-groupe est maximale à la valeur de 99% aux décile 7 et 8 pour les deux forêts.

Concernant la deuxième partie des résultats sur le recouvrement et la taille relative du sous-groupe estimé, nous remarquons, pour les simulations 4 et 7, que pour recouvrir S il faut que $\hat{A}$ soit de taille élevée. Réduire cette taille implique de moins recouvrir S. De plus on note que les tailles relatives, à recouvrement égal, sont plus importantes pour les forêts SC que S.

La simulation 9 est à part, puisque les deux valeurs tendent ensemble vers 1, ce qui signifie une précision très élevée de la région $\hat{A}$.

6.2.4 Discussion et synthèse des résultats

Virtual Twins appliqué à des scénarios d'*efficacy* laisse penser que la méthode n'est pas très efficace pour trouver le bon sous-groupe ou l'une des deux variables en premier split, surtout dans le cas où les bras sont équilibrés. Les résultats de recouvrement de la simulation 3 sont également très parlant : le sous groupe n'est absolument pas identifié. L'utilisation d'une forêt SC n'améliore pas les performances de la méthode.

Virtual Twins appliqué à des scénarios de *safety* donne des résultats assez hétérogènes. Nous pouvons distinguer le cas où le vrai sous-groupe n'est constitué que d'une seule variable, dans ce cas, *Virtual Twins* obtient de très bons résultats car la fréquence d'apparition du bon sous-groupe est proche de 100%.

Les scénarios 4 et 7 sont très proches dans leur construction : le scénario 7 divise par deux la taille de S de la simulation 4 mais pour un $RR(S)$ doublé. Nous pouvons d'ailleurs remarquer graphiquement à l'aide des courbes, la similitude des scénarios. Seulement, le scénario 7 donne de meilleurs résultats, ce qui laisserait penser que *Virtual Twins* détecte plus facilement les signaux très fort, même dans une sous-population de taille faible. La simulation 6 appuie également cette hypothèse.

Le scénario 8 est équivalent au scénario 5, à la différence près d'un sous-groupe S de taille deux fois moins grande. Cela n'a pas d'influence sur la méthode qui donnent de très bons résultats dans les deux cas.

Le scénario 9 reprend le scénario 8 en ajoutant une deuxième variable au sous-groupe, les résultats sont toujours très acceptable. Ces trois scénarios n'impliquent pas un $RR(S)$ très élevé, mais le déséquilibre des effectifs en faveurs du bras de traitement semble rendre la méthode *Virtual Twins* performante. De plus le compromis recouvrement et taille relative est excellent avec 90% de recouvrement et une taille relative à 1.1 au décile 8 de la forêt S peut suggérer des pistes d'applications de *Virtual Twins* .

Le scénario 10 est similaire au scénario 4, cependant le $RR(S)$ est de 8 sur 16% de la base. On retrouve une similitude avec l'exemple vu à la section 6.1. Les résultats sont très prometteurs mais peut être un peu moins réels car le signal créé ici est très fort dans une sous-population de grande taille. Néanmoins, cette simulation indiquent une performance de *Virtual Twins* assez remarquable.

De plus, nous n'avons pas rapporté les erreurs de classification de la classe minoritaire, mais

comme rapporté dans l'exemple du paragraphe 6.1, les classes sont totalement déséquilibrées. Malgré cela, *Virtual Twins* semble fonctionner car cette méthode se base plutôt sur une différence de probabilité que la classification en elle-même. D'ailleurs en voulant réguler cette erreur en contrôlant la forêt, cela ne semble pas améliorer les résultats, voire même les détériore.

Ces quelques simulations de scénarios nous permettent d'avancer deux hypothèses concernant l'application de *Virtual Twins* pour les essais cliniques rencontrés en laboratoire :

- *Virtual Twins* donne de bons résultats lorsque le signal est très fort dans un sous-groupe, même de petite taille.
- *Virtual Twins* donne de bons résultats lorsque le signal est moyennement élevé dans un sous-groupe déséquilibré en faveur du nouveau traitement.

Il semblerait donc que cet outil serait plutôt à positionner dans le cadre *safety*. Néanmoins, il est important de rappeler que les simulations et celles-ci en particulier ne rendent pas compte de la réalité d'un essai clinique. Seules les applications au cours du temps sur des données réelles pourront rendre compte de la véritable utilité d'une telle méthode.

6.3 Pour aller plus loin ...

Nous avons tentés plusieurs variantes de *Virtual Twins* en proposant des modèles seulement avec les variables importances données par les mesures d'importance, en dichotomisant les variables continues, avec forêts doubles contrôlées et non contrôlées. Ces variantes n'améliorent pas les performances de *Virtual Twins* , et parfois les détériorent.

Comme dit précédemment *Virtual Twins* repose sur deux étapes et les méthodes pour appliquer ces étapes sont multiples. Une méthode de boosting peut être envisageable pour la première étape. Les méthodes de boosting, comme AdaBoost, ont déjà prouvé leur performance en terme de classification. Nous avons essayé sur un exemple d'utiliser l'algorithme TreeNet (R) [24], mais cela n'a pas amélioré les performances de *Virtual Twins* . La seconde étape peut très bien être remplacée par des arbres obliques, ou des arbres conditionnels.

Virtual Twins peut être aussi utilisé dans le cas où la variable réponse est continue, ce qui serait une prochaine étape d'application. Enfin, *Virtual Twins* est aussi une méthode qui pourrait s'appliquer à des cas où il existe plus de deux traitements.

Conclusion

Durant six mois, j'ai participé à l'activité de la cellule de data mining du laboratoire Servier. Il m'a été confié d'étudier la méthode de recherche de sous-groupes *Virtual Twins* .

A travers cet objectif, il a fallu que je me familiarise avec le monde pharmaceutique, les essais thérapeutiques et en particulier avec la recherche de sous-groupes. Au travers de ce dernier point j'ai pu avoir une première vision de ce qu'est le data mining.

Dans un second temps, pour répondre à la problématique, l'apprentissage et la mise en pratique de nouvelles techniques statistiques et de *machine learning* a été un point important, sans quoi la compréhension de *Virtual Twins* n'aurait pas pu aboutir.

Enfin, l'application de *Virtual Twins* au travers d'exemples et de simulations a permis de positionner la méthode face aux différents scénarios qu'il est possible de rencontrer lors d'essais cliniques.

Durant ces six mois, j'ai pu travailler en collaboration avec mes responsables, mais également de façon autonome pour répondre à différents objectifs de parcours. Ce stage m'a également permis de me confronter aux contraintes d'un laboratoire pharmaceutique privé. De plus, Servier offre la possibilité à ses stagiaires de réellement participer à la vie de l'entreprise et du département auquel nous appartenons. Cela m'a permis d'avoir une vision assez globale du travail à fournir dans ce type de secteur en tant que data scientist.

D'un point de vue plus technique, cette expérience m'a donné la possibilité d'enrichir mes connaissances statistiques et de les appliquer. J'ai aussi eu la chance de pouvoir effectuer un stage dans le data mining, qui est un domaine pour lequel je porte beaucoup d'intérêt, et cela m'a permis d'acquérir quelques compétences propres à ce domaine. L'approfondissement de mes connaissances vis-à-vis de R s'est aussi avéré important.

En conclusion, ce stage de fin d'étude a été une réelle bonne expérience et m'encourage à continuer sur la voie du data mining, et pourquoi pas découvrir d'autres domaines.

A nouveau, je remercie l'ensemble du personnel du département PEX méthodologie et valorisation des données des laboratoires Servier.

Bibliographie

[1] Jeremy M.G. Taylor Jared C. Foster and Stephen J. Ruberg. Subgroup identification from randomized clinical trial data. *Stat med.*, 2011.

[2] Stéphane TUFFERY. *Data Mining et statistique décisionnelle.* TECHNIP, 2012.

[3] Richard A. Olshen et Charles J. Stone Leo Breiman, Jerome H. Friedman. *Classification And Regression Trees.* Chapman & Hall, 1984.

[4] R Development Core Team. *R : A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[5] *Edgar Anderson's Iris Data.* R documentation, package datasets.

[6] Mayo Foundation Terry M. Therneau, Elizabeth J. Atkinson. *An Introduction To Recursive Partionning Using the RPART package*, February 24, 2015.

[7] Erwan Scornet. Apprentissage et forêts aléatoires. Master's thesis, ENS, 2012.

[8] Gabor Lugosi Gérard Biau, Luc Devroye. Consistency of random forests and other averaging classifers. *Journal of Machine Learning Research 9*, 2008.

[9] Leo Breiman. Random forests. *Machine Learning, 45, 5-32*, 2001.

[10] Robin Genuer. *Forêts aléatoires : aspects théoriques, sélection de variables et applications.* PhD thesis, Université Paris-sud XI, 2010.

[11] Simon Bernard. *De l'Analyse des Mécanismes de Fonctionnement à la Construction Dynamique.* PhD thesis, Université de Rouen, 2009.

[12] Brice Zirakiza. Forêts aléatoires pac-bayésiennes. Master's thesis, Université LAVAL, 2013.

[13] Matthew W. Mitchell. Bias of the random forest out-of-bag (oob) error for certain input parameters. *Open Journal of Statistics*, 2011.

[14] Daniel J. Stekhoven et Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 2011.

[15] Achim Zeileis et Torsten Hothorn Carolin Strobl, Anne-Laure Boulesteix. Bias in random forest variable importance measures : Illustrations, sources and a solution. *BMC Bioinformatics*, 2007.

[16] Francois Vieille. *aVirtualTwins : Adaptation of Virtual Twins method from Jared Foster.*, 2015. R package version 0.0.0.2, www.github.com/prise6/aVirtualTwins.

[17] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3) :18–22, 2002.

[18] Guideline on the investigation of subgroups in confirmatory clinical trials. Technical report, European Medecines Agency.

[19] Alvan R. Feinstein. The problem of cogent subgroups : A clinicostatistical tragedy. *J Clin Epidemiol*, 1998.

[20] Dmitrienko A. Lipkovich I. Strategies for identifying predictive biomarkers and subgroups with enhanced treatment effect in clinical trials using sides. *Biopharm Stat.*, 2011.

[21] K. Van Deun I. Van Mechelen L. L. Doove, E. Dusseldorp. A comparison of five recursive partitioning methods to find person subgroups involved in meaningful treatment-subgroup interactions. *Springer*, 2013.

[22] Michael Man Wei-Yin Loh, Xu He. A regression tree approach to identifying subgroups with differential treatment effect. 2014.

[23] Leo Breiman Chao Chen, Andy Liaw. Using random forest to learn imbalanced data.

[24] Salford Systems. Treenet(r). www.salford-systems.com/products/treenet.

Annexes

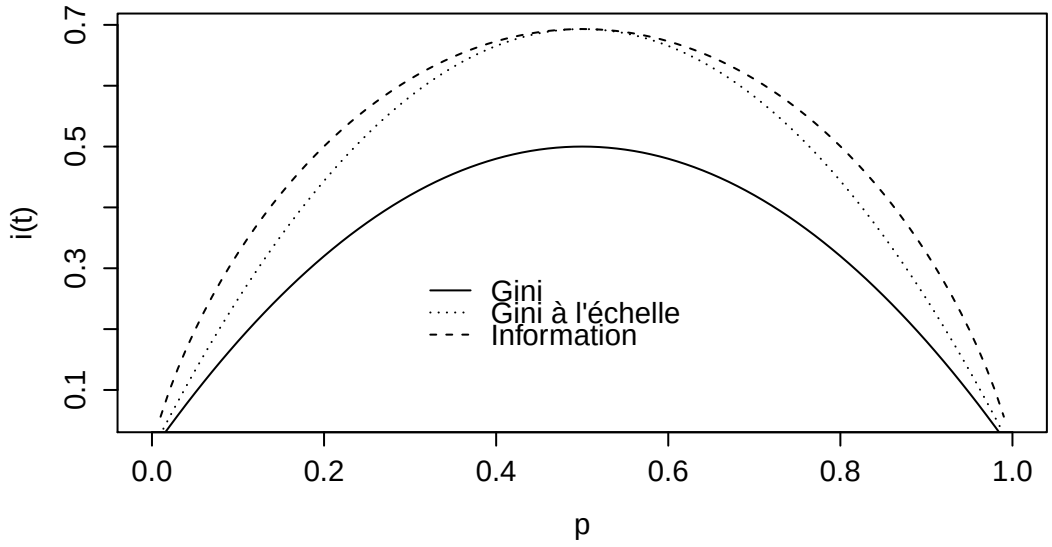


FIGURE 1 – Fonctions d'impureté lorsque $C = 2$

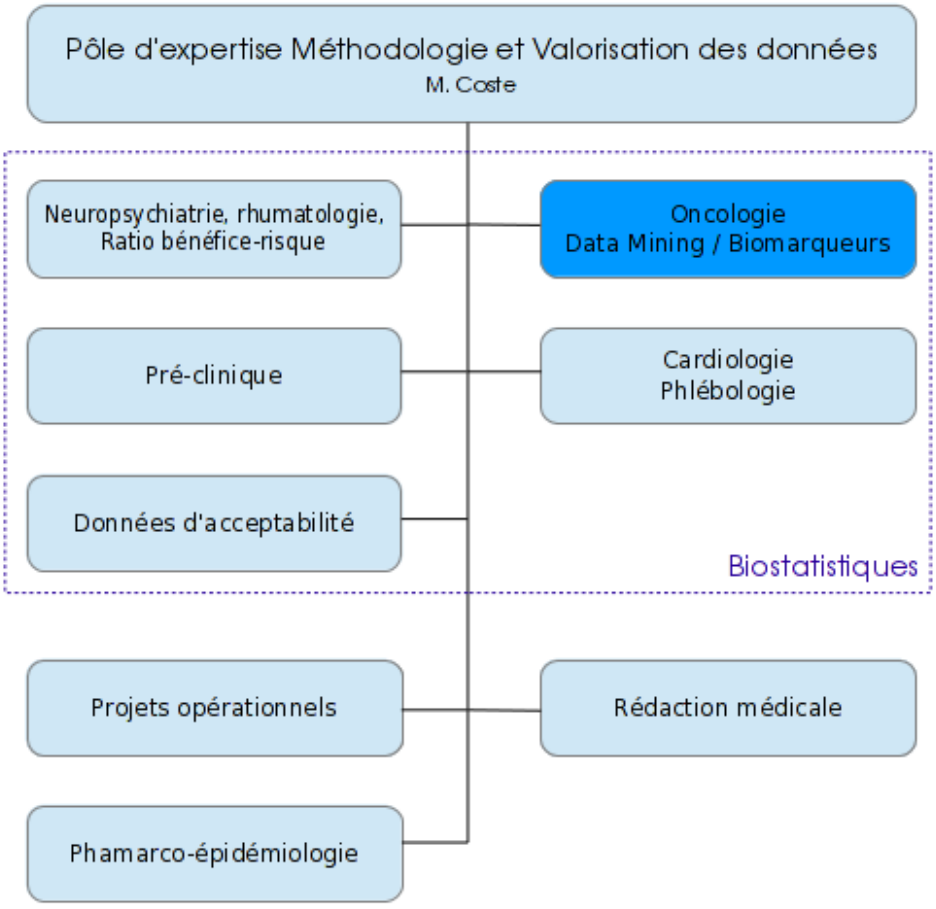


FIGURE 2 – Structure du PEX *Méthodologie et valorisation des données*

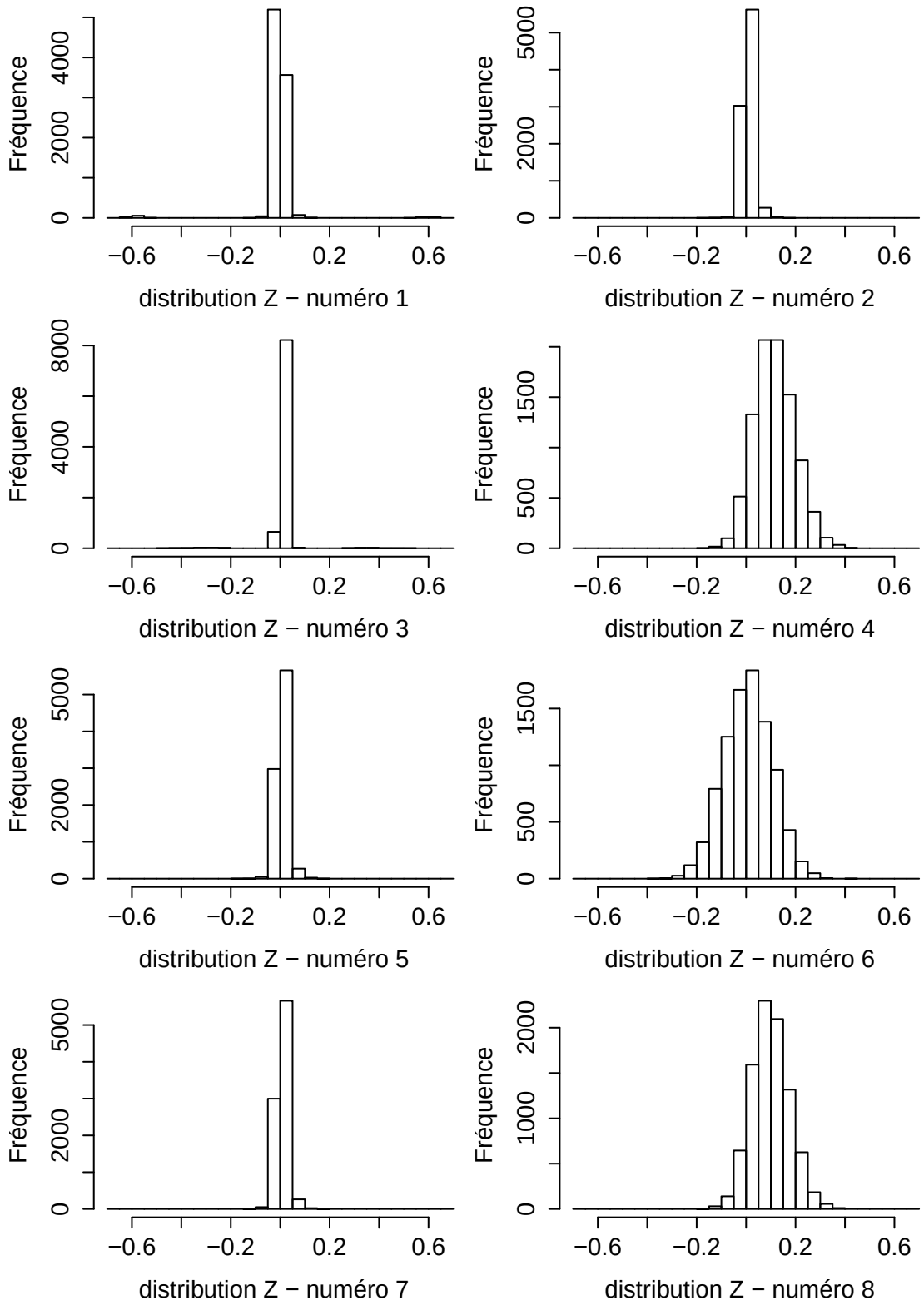


FIGURE 3 – Histogrammes des huit approches de l'étude de cas

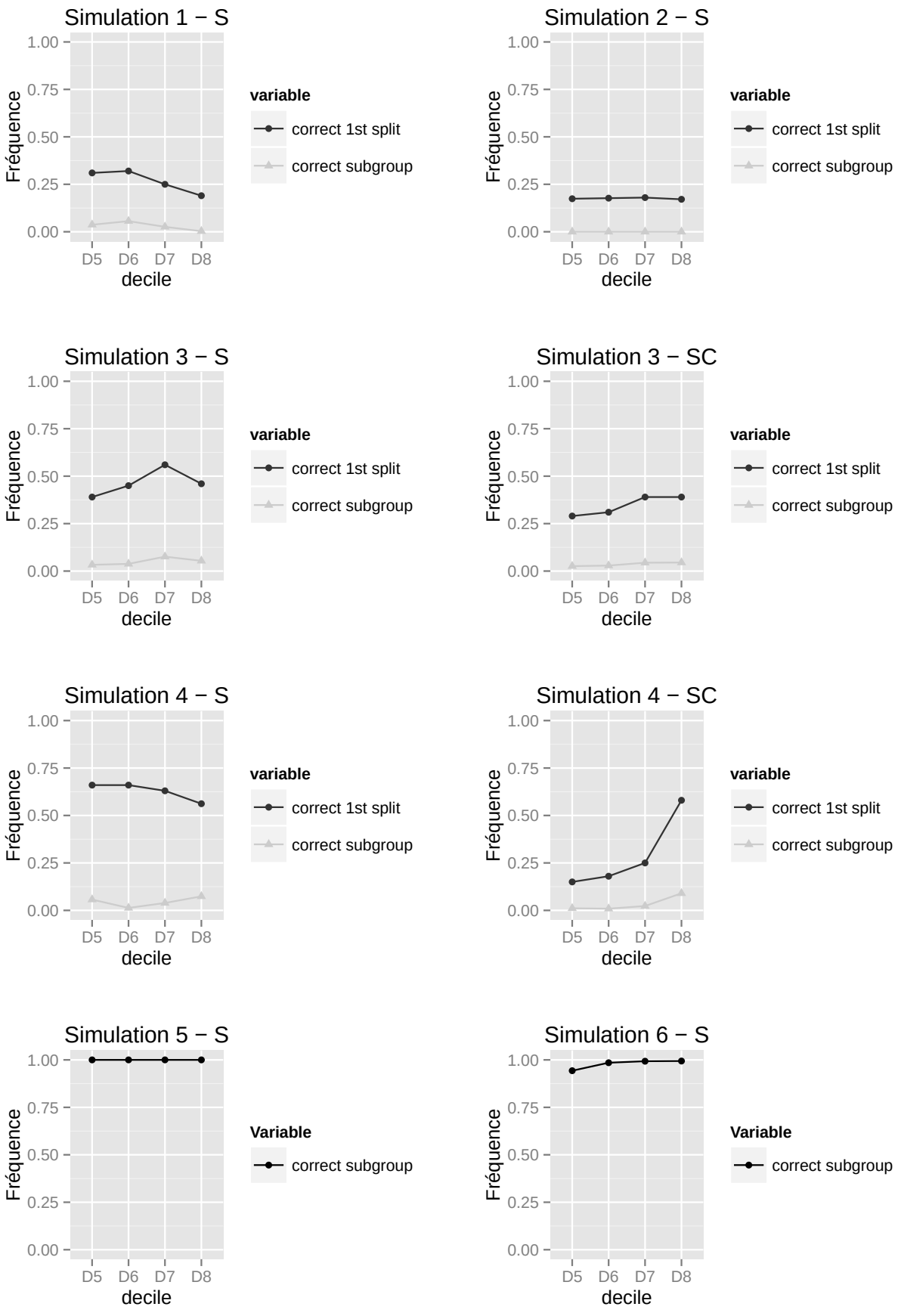


FIGURE 4 – Résultats des simulations en terme de splits (1)

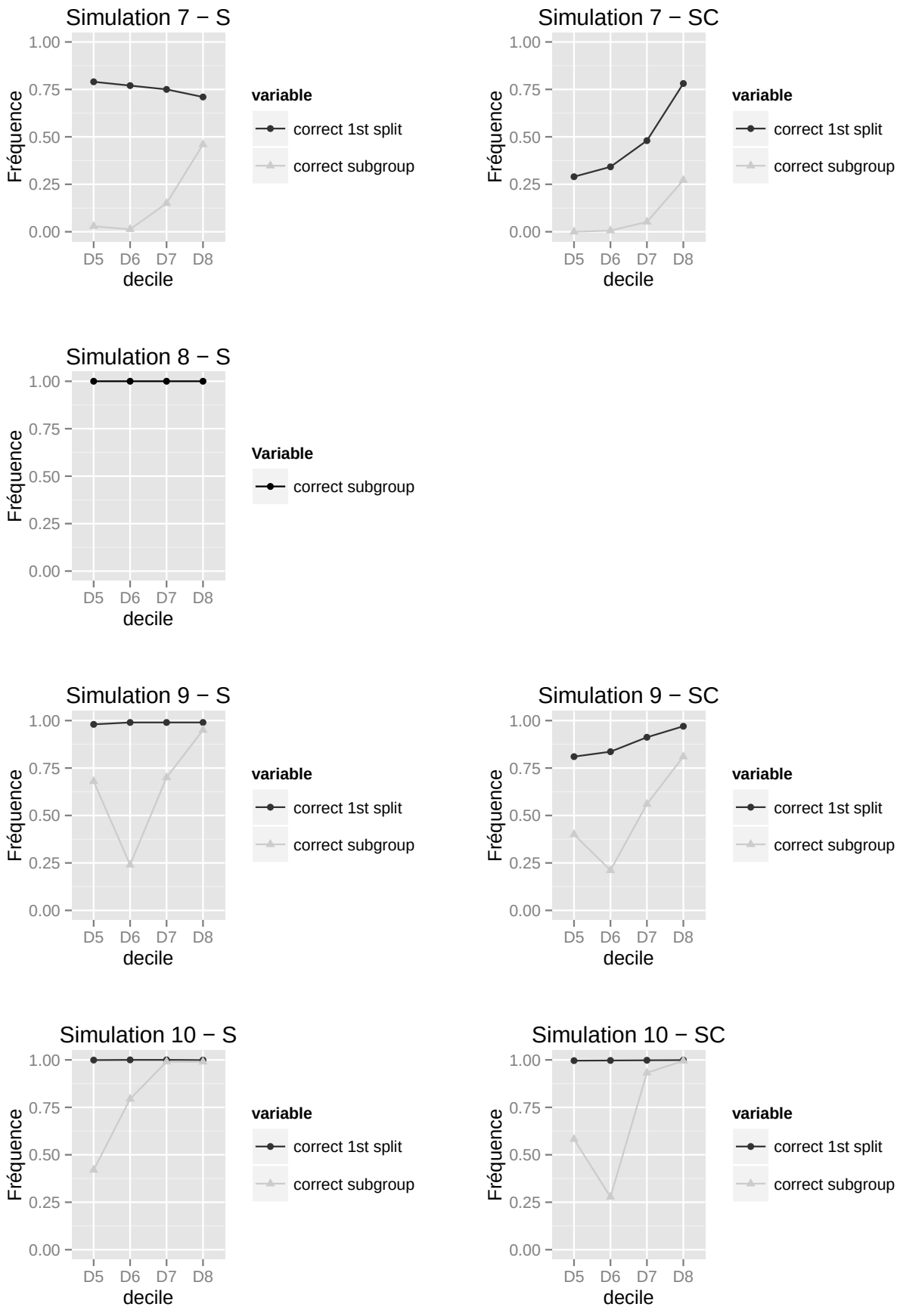


FIGURE 5 – Résultats des simulations en terme de splits (2)

Num.	Déc.	Valeur	Subgroup	Subgroup size	Treatment event rate	Control event rate	Treatment sample size	Control sample size	RR(S) (resub)	RR(S) (snd)	RR(^cS) (resub)	RR(^cS) (snd)
4	5	0.11	$X^{(1)} > 26.95$	3366	0.022	0.008	1730	1636	2.75	1.623	1.2	1.282
4	6	0.13	$X^{(1)} > 27.15 \cap X^{(2)} > 46.15$	2625	0.017	0.002	1340	1285	8.5	1.742	1.25	1.29
4	7	0.16	$X^{(1)} > 27.15 \cap X^{(2)} > 50.45$	2307	0.017	0.002	1184	1123	8.5	1.826	1.25	1.3
4	8	0.18	$X^{(1)} > 27.75 \cap X^{(3)} > 84.5$	1014	0.041	0.004	511	503	10.25	1.77	1.3	1.36
6	5	0.01	$X^{(2)} > 50.22$	4535	0.014	0.003	2323	2212	4.67	1.24	1.13	0.9
6	6	0.03	$X^{(2)} > 50.22 \cap X^{(1)} > 27.15$	2322	0.018	0.002	1195	1127	9	1.35	1.25	0.94
6	7	0.06	$X^{(2)} > 50.31 \cap X^{(1)} > 27.75$	2076	0.018	0.002	1058	1018	9	1.37	1.25	0.95
6	8	0.09	$X^{(2)} > 50.55 \cap X^{(1)} > 27.95$	1982	0.018	0.002	1013	969	9	1.37	1.25	0.95
8	5	0.10	$X^{(1)} < 25.85 \cap X^{(3)} > 82.5$	1339	0.022	0.012	673	666	1.833	1.351	1.67	1.4
8	5	0.10	$X^{(1)} > 25.85 \cap X^{(2)} > 45.24$	3454	0.018	0.005	1769	1685	3.6	1.64	1.25	1.27
8	6	0.12	$X^{(1)} > 25.85 \cap X^{(3)} > 87.5$	1120	0.034	0.007	552	568	4.86	1.65	1.3	1.35
8	7	0.14	$X^{(1)} > 26.55 \cap X^{(3)} > 84.5$	1322	0.037	0.009	654	668	4.11	1.68	1.2	1.34
8	8	0.16	$X^{(3)} > 86.5 \cap X^{(2)} > 47.96$	1178	0.029	0.003	585	593	9.667	1.76	1.4	1.34

TABLE 6 – Les différents sous-groupes

EFFICACY

Ensemble	Effectif total	Effectif $T=1$	Effectif $T=0$	Inc. $T=1$	Inc. $T=0$
SIMULATION 1					
$X^{(1)} > 0 \cap X^{(51)} = 1$	3000	1500	1500	0.18	0.12
${}^cS$	17000	8500	8500	0.145	0.155
$\mathcal{X}$	20000	10000	10000	0.15025	0.14975
SIMULATION 2					
$X^{(1)} > Q_{64} \cap X^{(31)} = 0$	180	90	90	0.3	0.16
${}^cS$	820	410	410	0.18	0.21
$\mathcal{X}$	1000	500	500	0.2016	0.201
SIMULATION 3					
$X^{(1)} > Q_{25} \cap X^{(61)} = 1$	450	300	150	0.42	0.3
${}^cS$	2550	1700	850	0.279	0.3
$\mathcal{X}$	3000	2000	1000	0.3	0.3

SAFETY

Ensemble	Effectif total	Effectif $T=1$	Effectif $T=0$	Inc. $T=1$	Inc. $T=0$
SIMULATION 4					
$X^{(1)} > Q_{40} \cap X^{(81)} = 1$	600	300	300	0.06	0.01
${}^cS$	9400	4700	4700	0.0145	0.01
$\mathcal{X}$	10000	5000	5000	0.0172	0.01
SIMULATION 5					
$X^{(1)} < Q_{20}$	1200	1000	200	0.2	0.05
${}^cS$	4800	4000	800	0.06	0.05
$\mathcal{X}$	6000	5000	1000	0.088	0.05
SIMULATION 6					
$X^{(101)} = 1$	300	150	150	0.12	0.01
${}^cS$	9700	4850	4850	0.0145	0.01
$\mathcal{X}$	10000	5000	5000	0.0186	0.01
SIMULATION 7					
$X^{(1)} > Q_{70} \cap X^{(81)} = 1$	300	150	150	0.12	0.01
${}^cS$	9700	4850	4850	0.0145	0.01
$\mathcal{X}$	10000	5000	5000	0.0186	0.01
SIMULATION 8					
$X^{(1)} < Q_{10}$	600	500	100	0.2	0.05
${}^cS$	5400	4500	900	0.068	0.05
$\mathcal{X}$	6000	5000	1000	0.0812	0.05
SIMULATION 9					
$X^{(1)} < Q_{50} \cap X^{(61)} = 1$	600	500	100	0.2	0.05
${}^cS$	5400	4500	900	0.068	0.05
$\mathcal{X}$	6000	5000	1000	0.0812	0.05
SIMULATION 10					
$X^{(1)} > Q_{60} \cap X^{(41)} = 1$	1600	800	800	0.08	0.01
${}^cS$	8400	4200	4200	0.01	0.01
$\mathcal{X}$	10000	5000	5000	0.0212	0.01

TABLE 7 – Simulations effectuées

SIMULATION 3

	S		SC	
Décile	recouvrement	taille relative	recouvrement	taille relative
5	0.72	4.6	0.66	4.84
6	0.5	2.6	0.46	3.4
7	0.39	1.73	0.31	2.3
8	0.14	1.67	0.14	2

SIMULATION 4

	S		SC	
Décile	recouvrement	taille relative	recouvrement	taille relative
5	0.92	5.28	0.64	13.4
6	0.75	3.15	0.45	11
7	0.59	2.6	0.38	7.5
8	NA	NA	0.47	2.6

SIMULATION 7

	S		SC	
Décile	recouvrement	taille relative	recouvrement	taille relative
5	0.85	8.2	0.69	24
6	0.79	6	0.55	17.8
7	0.66	4.5	0.49	10.8
8	0.5	2.7	0.61	3.5

SIMULATION 9

	S		SC	
Décile	recouvrement	taille relative	recouvrement	taille relative
5	0.99	3.5	0.95	4.9
6	0.98	2	0.9	2.5
7	0.95	1.4	0.88	1.6
8	0.9	1.1	0.89	1.2

TABLE 8 – Recouvrement et taille relative