Forêts aléatoires pour données longitudinales de grande dimension

Louis Capitaine

université de BORDEAUX

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR
DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE SOCIÉTÉ, POLITIQUE,

SANTÉ PUBLIQUE

SPÉCIALITÉ SANTÉ PUBLIQUE, OPTION BIOSTATISTIQUE

Par **Louis CAPITAINE**

Forêts aléatoires pour données longitudinales de grande dimension

Sous la direction de : **Rodolphe THIEBAUT**
Co-directeur : **Robin GENUER**

Soutenue le 17 décembre 2020

Membres du jury :

M. Jean-Philippe VERT	Chercheur associé	Mines ParisTech	Rapporteur
M. Stéphane GAÏFFAS	Professeur	Université de Paris	Rapporteur
Mme. Cécile PROUST-LIMA	Directrice de recherche	INSERM	Présidente du jury
Mme. Julie JOSSE	Advanced reseacher	INRIA Montpellier	Examinatrice
M. Jérémie BIGOT	Professeur	Université de Bordeaux	Examinateur
M. Rodolphe THIÉBAUT	Professeur	Université de Bordeaux	Directeur de thèse
M. Robin GENUER	Maître de conférence	Université de Bordeaux	Co-directeur de thèse

Remerciements

Robin, tu as été le premier à me faire confiance, tu m'as fait découvrir le monde de la recherche et transmis le goût pour l'enseignement. Durant ces trois années, tu as toujours été très présent et à l'écoute. Tu m'as offert un accompagnement et non un encadrement. Tu as supporté mes humeurs, mes craintes, mes doutes et tu as toujours su m'offrir une oreille lorsque j'en avais le plus besoin. Tu n'as pas hésité à me suivre dans ma fascination pour les travaux de Maurice Fréchet. Je suis admiratif de ta persévérance. Au fur et à mesure de notre avancée dans les profondeurs de ces forêts, j'ai pu profiter de ton expérience et de tes conseils avisés. Dans ton antre des adorateurs de R, tu n'as jamais cessé de me proposer des cafés. Merci pour tout, camarade.

Rodolphe, tu m'as laissé pleinement m'exprimer au cours de cette thèse. Tu as toujours su prendre de la hauteur sur mes propositions et m'orienter lorsqu'il s'agissait d'applications. Je reste impressionné par ta maîtrise approfondie de plusieurs disciplines. Tu m'as souvent donné de précieux conseils qu'ils soient personnels ou professionnels. Enfin, tu as souvent valorisé mon travail et pour cela je t'en serai toujours reconnaissant. Merci chef.

Jean Philippe Vert et Stéphane Gaïffas, merci d'avoir accepté de rapporter ce travail. À l'ensemble des membres du jury, merci de me faire l'honneur de votre présence.

Jérémie, j'ai beaucoup appris à tes cotés, que ce soit en Master où durant ces derniers mois. Merci pour cette collaboration qui, je l'espère, ne sera pas la dernière.

Boris, je n'oublierai pas nos nombreux échanges dans ton bureau, ta gentillesse et ta bonne humeur communicative.

Sandrine, merci d'avoir supporté mon allergie aux papiers administratifs et de m'avoir évité une saisie de l'intégralité de mon patrimoine en payant pour moi les 19

centimes que je devais à l'Université.

À toute l'équipe SISTM, votre accueil m'aura permis de me sentir bien. Merci pour votre gentillesse. À tous ceux que j'ai pu croiser durant ces trois dernières années, que ce soit au cours d'une discussion ou d'un exposé. Merci.

William, cette thèse marque notre rencontre à Fréjus et le début d'une amitié. Merci pour ces nuits à discuter des civilisations, des traditions et de la France éternelle.

Mathieu et Emma, merci pour ces longues soirées durant lesquelles le temps semblait figé. Il m'a semblé avoir refait le monde des dizaines de fois ensemble. Nos échanges filmiques, philosophiques et scientifiques ont été une source d'inspiration permanente. Mathieu, j'ai encore tellement à apprendre à tes côtés.

Kéta, j'ai l'impression de ne jamais avoir vécu sans toi. De nos escapades en forêt ou de nos expéditions dans de lugubres complexes abandonnés, je n'oublierai jamais ton amour pour la République et la démocratie. À tes côtés la vie prend un sens comique, elle devient légère, presque facile. J'aime le ridicule des situations dans lesquelles nous sommes toujours embarqués. J'aime me replonger dans les souvenirs que nous nous sommes créés. Grandir à tes côtés *c'est porter en soi un chaos pour pouvoir mettre au monde une étoile dansante*. Merci.

À mes parents, vous m'avez offert une enfance extrêmement heureuse. Vous m'avez toujours soutenu dans mes choix et m'avez donné tout l'amour dont un fils peu rêver. Les retours chez vous durant ces trois années furent toujours de grandes bulles de respiration. Quentin, merci de m'offir une scène des frères Coen à chacune de tes apparitions. À ma famille, merci pour vos étrangetés, vos bizarreries qui m'ont tant apporté.

À Marine, *à la folie dont tu es la raison*. Je ne me souviens pas quand j'ai cessé de penser ma vie sans toi, cela fait si longtemps. Durant ces trois années tu as été mon plus grand soutien. Je me souviens de nos tendres soirées à essayer de résoudre nos problèmes respectifs sur la voix de Gainsbourg. Merci pour tes regards, tes fous rires, ton écoute. Merci pour ton amour.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Ce chapitre introductif vise à présenter, en cinq parties, les différentes notions qui seront abordées tout au long de ce manuscript et qui faciliteront la compréhension des travaux de thèse. Dans une première section, nous introduirons les éléments classiques de l'apprentissage statistique supervisé. La deuxième section sera consacrée aux méthodes d'ensembles dont est issue la méthode des forêts aléatoires. Puis, dans la troisième section, nous détaillerons la construction des forêts aléatoires introduites par Breiman (2001) et nous donnerons deux variantes de cette méthode qui s'avèrent être très utiles. En quatrième section, nous présenterons les problématiques liées aux données longitudinales de grande dimension et expliciterons les objectifs de cette thèse. Enfin, dans la cinquième et dernière section, nous résumerons les différents développements menés durant ce doctorat.

1.1 Apprentissage statistique

Cette section présente les notions de base de l'apprentissage statistique. Lorsqu'un phénomène physique (Radovic et al., 2018), biologique (Schrider and Kern, 2018), ou même financier (Yoo et al., 2005), est trop complexe pour proposer une modélisation déterministe, on fait appel à des méthodes d'apprentissage statistique. C'est le cas par exemple lorsque l'on cherche à faire de la détection d'émotions à partir de signaux vocaux (Noroozi et al., 2017) ou même lorsque l'on cherche à faire de la reconnais-

sance d'images (Bosch et al., 2007).

Les forêts aléatoires sont une méthode d'apprentissage statistique non-paramétrique introduite par Breiman (2001). Ses performances prédictives exceptionnelles aussi bien sur des petites comme des colossales bases de données en font une méthode très populaire et utilisée dans un vaste champ de domaines comme en génomique (Goldstein et al. (2010), Chen and Ishwaran (2012)), en biologie (Casanova et al., 2014), en écologie (Cutler et al. (2007), Prasad et al. (2006)), en chimie (Svetnik et al., 2004), en informatique (Alam and Vuong, 2013) et même en finance (Booth et al., 2015).

1.1.1 Problématique de prédiction

Dans cette partie, nous présentons les notions de base de l'apprentissage statistique supervisé. Nous renvoyons au livre de Hastie et al. (2009) pour une introduction claire et détaillée du domaine. Soit $\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ un échantillon d'apprentissage constitué de n vecteurs aléatoires indépendants et identiquement distribués ($i.i.d$), de même loi qu'un couple (X, Y). Le couple (X, Y) est indépendant de l'échantillon d'apprentissage $\mathcal{L}_n$ et sa loi, notée P, est inconnue. Notons $\mathcal{X}$ et $\mathcal{Y}$ les espaces dans lesquels vivent respectivement les variables aléatoires X et Y. $X \in \mathcal{X}$ est appelé vecteur des variables explicatives. Le nombre de coordonnées de ce vecteur, noté p, est le nombre de variables explicatives. On parle souvent de covariables ou simplement d'entrées pour dénommer les variables explicatives. $Y \in \mathcal{Y}$ est appelé variable à expliquer ou variable de sortie. Le but de l'apprentissage statistique est d'apprendre le lien qui existe entre les variables d'entrée X et la variable de sortie Y afin de pouvoir prédire pour n'importe quelle entrée $x \in \mathcal{X}$ fixée la sortie qui lui est associée. On introduit alors la notion de prédicteur ainsi que de prédiction.

Définition 1.1.1 (Prédicteur et prédiction)**.** On appelle prédicteur sur $\mathcal{X} \times \mathcal{Y}$ toute fonction mesurable $f : \mathcal{X} \longmapsto \mathcal{Y}$. On note $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ l'ensemble des prédicteurs sur $\mathcal{X} \times \mathcal{Y}$. Pour tout $x \in \mathcal{X}$, $f(x) \in \mathcal{Y}$ est la prédiction de l'entrée x par le prédicteur f.

Il est à noter que lorsque les espaces $\mathcal{X}$ et $\mathcal{Y}$ sont mesurables, l'espace $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ n'est pas vide. Par exemple, pour tout $C \in \mathcal{Y}$, la fonction constante $f : x \longmapsto C$ pour tout $x \in \mathcal{X}$ est bien un prédicteur. Dans un problème d'apprentissage statistique, on souhaite non seulement être capable de proposer un prédicteur mais aussi que ce dernier soit "efficace", c'est-à-dire que ses prédictions soient les plus proches possibles de la réalité. Afin de mesurer l'écart entre une prédiction et la réalité on fait appel à la notion de fonction de coût.

Définition 1.1.2 (Fonction de coût)**.** Soit $\mathcal{Y}$ un espace mesurable. On appelle fonction de coût sur $\mathcal{Y}$ toute application mesurable $L : \mathcal{Y} \times \mathcal{Y} \longmapsto \mathbb{R}^+$ vérifiant $L(y, y) = 0$ pour tout $y \in \mathcal{Y}$.

Étant donné un couple $(x, y) \in \mathcal{X} \times \mathcal{Y}$, un prédicteur $f \in \mathcal{F}(\mathcal{X}, \mathcal{Y})$ ainsi qu'une fonction de coût L, $L(f(x), y)$ quantifie l'écart entre la prédiction $f(x)$ et la réalité y. Plus cette quantité est faible plus le prédicteur prédit bien. De plus, lorsque la prédiction est exactement l'élément à prédire, c'est-à-dire $f(x) = y$, la fonction de coût est nulle. Donnons quelques exemples de fonctions de coût pour diverses espaces.

Exemple 1.1.1. Dans $\mathbb{R}$, la distance Euclidienne usuelle $L(y, y') = |y - y'|$ pour tout $y, y' \in \mathbb{R}$ est une fonction de coût.

Exemple 1.1.2 (Fonction de coût 0-1)**.** La fonction de coût 0-1 sur $\mathcal{Y}$ est définie par :

$$L(y, y') = \begin{cases} 0 & \text{si } y = y' \\ 1 & \text{sinon} \end{cases} \qquad \forall y, y' \in \mathcal{Y}$$

Exemple 1.1.3 (Espaces métriques et espaces normés)**.** Soit $(\mathcal{Y}, d)$ un espace métrique, c'est-à-dire un espace muni d'une distance d. La distance d est une fonction de coût. De plus si $(\mathcal{Y}, \|.\|)$ est un espace vectoriel normé de norme $\|.\|$, la distance $L_{\|.\|}$ associée à la norme $\|.\|$ définie par $L_{\|.\|}(y, y') = \|y - y'\|$ est une fonction de coût.

Une fonction de coût permet de quantifier l'erreur commise par un prédicteur pour tout couple $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Cependant, afin de sélectionner le meilleur prédicteur possible, c'est-à-dire celui qui fait le moins d'erreur de prédiction relativement à une

fonction de coût L, nous avons besoin de mesurer son coût de prédiction sur l'espace $\mathcal{X} \times \mathcal{Y}$ tout entier, et ce, relativement au couple aléatoire (X, Y). La variable aléatoire positive $L(f(X), Y)$ mesure le coût de l'erreur de prédiction de f sur le couple (X, Y). L'espérance de cette variable aléatoire définie l'erreur de généralisation du prédicteur f sur le problème d'apprentissage donné par le couple (X, Y).

Définition 1.1.3 (Erreur de généralisation)**.** Soit $f \in \mathcal{F}(\mathcal{X}, \mathcal{Y})$ un prédicteur sur $\mathcal{X} \times \mathcal{Y}$ et L une fonction de coût sur $\mathcal{Y}$. On définit l'erreur de généralisation du prédicteur f par :

$$\mathcal{R}_{P,L}(f) = \mathbb{E}\left[L\left(f(X), Y\right)\right]. \tag{1.1}$$

L'erreur de généralisation d'un prédicteur f dépend de la loi P du couple (X, Y) ainsi que de la fonction de coût L. Le meilleur prédicteur possible, c'est-à-dire celui qui minimise l'erreur de généralisation, est appelé prédicteur de Bayes.

Définition 1.1.4 (Prédicteur de Bayes)**.** Soit (X, Y) un vecteur aléatoire de $\mathcal{X} \times \mathcal{Y}$, soit L une fonction de coût sur $\mathcal{Y}$ et soit $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ l'ensemble des prédicteurs sur $\mathcal{X} \times \mathcal{Y}$, on définit l'erreur de généralisation de Bayes $\mathcal{R}^{\star}_{P,L}$ par :

$$\mathcal{R}^{\star}_{P,L} = \inf_{f \in \mathcal{F}(\mathcal{X}, \mathcal{Y})} \mathcal{R}_{P,L}(f) \tag{1.2}$$

On définit le prédicteur de Bayes comme le prédicteur $f^{\star} \in \mathcal{F}(\mathcal{X}, \mathcal{Y})$ dont l'erreur de généralisation est égale à l'erreur de généralisation de Bayes :

$$f^{\star} = \operatorname*{arg\,min}_{f \in \mathcal{F}(\mathcal{X}, \mathcal{Y})} \mathcal{R}_{P,L}(f) \tag{1.3}$$

En pratique, comme la loi P du couple (X, Y) est inconnue le prédicteur de Bayes n'est pas calculable. L'objectif est alors de construire un prédicteur seulement à partir de l'échantillon d'apprentissage $\mathcal{L}_n$ telle que son erreur de généralisation soit la plus proche possible de l'erreur de généralisation du prédicteur de Bayes. Il faut donc proposer une stratégie de construction de prédicteurs à partir d'échantillons d'apprentissage de taille quelconque. C'est ce que l'on appelle une règle d'apprentissage.

1.1.2 Règle d'apprentissage

Définition 1.1.5. On appelle règle d'apprentissage toute fonction mesurable

$$\widehat{f} : \bigcup_{n\geq 1} (\mathcal{X} \times \mathcal{Y})^n \longmapsto \mathcal{F}(\mathcal{X}, \mathcal{Y})$$

Pour toute règle d'apprentissage $\widehat{f}$ et pour tout échantillon d'apprentissage $\mathcal{L}_n$, on note $\widehat{f}(\mathcal{L}_n) \in \mathcal{F}(\mathcal{X}, \mathcal{Y})$ le prédicteur construit à partir de la règle d'apprentissage $\widehat{f}$ sur $\mathcal{L}_n$. Pour tout $x \in \mathcal{X}$, on note $\widehat{f}(\mathcal{L}_n; x)$ la prédiction de $\widehat{f}(\mathcal{L}_n)$ en x. Dans la pratique, une règle d'apprentissage se matérialise par un algorithme qui prend $\mathcal{L}_n$ en entrée et renvoie un prédicteur en sortie. Une fois une règle d'apprentissage (ou algorithme) définie pour construire des prédicteurs à partir d'échantillons de toute taille, il nous faut valider cette règle en montrant que l'erreur de généralisation des prédicteurs construits tend vers l'erreur de généralisation du prédicteur de Bayes, lorsque la taille de l'échantillon d'apprentissage tend vers l'infini. C'est ce qu'on appelle la consistance.

Définition 1.1.6 (Consistance)**.** Soit $\widehat{f}$ une règle d'apprentissage, soit $\mathcal{L}_n$ un n-échantillon *i.i.d* de même loi de probabilité P et soit L une fonction de coût. On dit que :

1. $\widehat{f}$ est faiblement consistante pour P lorsque :

$$\mathbb{E}\left[\mathcal{R}_{P,L}\left(\widehat{f}(\mathcal{L}_n)\right)\right] \underset{n\to+\infty}{\longrightarrow} \mathcal{R}^\star_{P,L}. \tag{1.4}$$

2. $\widehat{f}$ est fortement consistante pour P lorsque :

$$\mathcal{R}_{P,L}\left(\widehat{f}(\mathcal{L}_n)\right) \underset{n\to+\infty}{\longrightarrow} \mathcal{R}^\star_{P,L} \qquad \text{p.s} \tag{1.5}$$

Une règle d'apprentissage fortement consistante produit asymptotiquement des prédicteurs optimaux (au sens de l'erreur de généralisation). Lorsqu'une règle d'apprentissage est faiblement consistante, elle produit asymptotiquement des prédicteurs optimaux en moyenne. Dans la pratique, nous ne disposons pas d'échantillons de taille infinie. Si nous voulons obtenir le meilleur prédicteur possible, il nous faut donc com-

parer les erreurs de généralisation des différentes règles d'apprentissage sur l'échantillon dont on dispose. Cependant, l'erreur de généralisation dépend de la loi du couple (X, Y) qui est inconnue, nous ne pouvons donc pas la calculer directement, il faut donc l'estimer.

1.1.3 Estimation de l'erreur de généralisation

Une idée naïve pour estimer l'erreur de généralisation d'un prédicteur est de calculer l'erreur moyenne de prédiction (calculé à partir de la fonction de coût L) sur les observations de l'échantillon d'apprentissage $\mathcal{L}_n$. C'est ce qu'on appelle l'erreur d'apprentissage (parfois nommée erreur empirique).

Définition 1.1.7 (Erreur d'apprentissage)**.** Soit $\mathcal{L}_n$ un échantillon d'apprentissage, soit $\widehat{f}$ une règle d'apprentissage et L une fonction de coût sur $\mathcal{Y}$. On définit l'erreur d'apprentissage $\widehat{\mathcal{R}}\left(\widehat{f}, \mathcal{L}_n\right)$ de $\widehat{f}$ sur $\mathcal{L}_n$ par

$$\widehat{\mathcal{R}}\left(\widehat{f}, \mathcal{L}_n\right) := \frac{1}{n} \sum_{i=1}^{n} L\left(Y_i, \widehat{f}(\mathcal{L}_n; X_i)\right) \tag{1.6}$$

Ici, les données d'apprentissage sont utilisées deux fois, une première fois pour construire le prédicteur, une seconde fois pour calculer le coût de prédiction. L'erreur d'apprentissage $\widehat{\mathcal{R}}\left(\widehat{f}, \mathcal{L}_n\right)$ est donc un estimateur biaisé et a tendance à sous évaluer l'erreur de généralisation. Sélectionner un prédicteur sur la base de son erreur d'apprentissage mène souvent au phénomène de surapprentissage, c'est-à-dire à la sélection d'un prédicteur très peformant sur les données d'apprentissage mais qui obtiendra une erreur de prédiction bien supérieure sur un nouvel échantillon.
Afin d'éviter le surapprentissage, il s'agirait de construire notre prédicteur sur un échantillon d'apprentissage puis de le valider en calculant son erreur de prédiction empirique (relativement à une fonction de coût L) sur un échantillon indépendant. Dans la pratique, nous ne disposons que de l'échantillon $\mathcal{L}_n$. Une stratégie, appelée validation croisée (Kohavi et al., 1995), consiste à découper aléatoirement l'échantillon $\mathcal{L}_n$ en deux : un échantillon d'apprentissage, sur lequel sera construit notre prédicteur

et un échantillon de test, sur lequel seront calculés les coûts de prédictions. Notons Θ une variable aléatoire à valeurs dans $\mathcal{P}(\{1,\ldots,n\})$ l'ensemble des sous-ensembles de $\{1,\ldots,n\}$, on note $\mathcal{L}_n^\Theta := \{(X_j, Y_j);\ j \in \Theta\}$ le tirage de l'échantillon d'apprentissage selon la variable Θ. L'estimation de l'erreur de prédiction sur l'échantillon test est donnée par

$$\widehat{\mathcal{R}}\left(\widehat{f}, \mathcal{L}_n^\Theta\right) = \frac{1}{\#\left\{j : (X_j, Y_j) \notin \mathcal{L}_n^\Theta\right\}} \sum_{j:(X_j,Y_j)\notin\mathcal{L}_n^\Theta} L\left(Y_i, \widehat{f}(\mathcal{L}_n^\Theta; X_i)\right). \tag{1.7}$$

Remarque 1.1.1. Dans la pratique il y a plusieurs manières de découper $\mathcal{L}_n$ en échantillon d'apprentissage et échantillon test, c'est-à-dire de choisir la loi de la variable de tirage Θ. Une stratégie très répandue consiste à fixer une proportion $0 < \alpha < 1$ puis de tirer un sous-ensemble de taille $\lfloor n * \alpha \rfloor$ de $\{1,\ldots,n\}$.

Avec cette stratégie d'estimation de l'erreur de prédiction, l'erreur estimée dépend très fortement du découpage tiré. En effet, il est possible d'avoir un prédicteur qui soit plus avantagé que les autres en fonction du tirage. Afin d'éviter les problèmes liés à un tirage aléatoire unique, cette opération est répétée plusieurs fois.
Soit B un entier positif, soient $\Theta_1, \Theta_2, \ldots, \Theta_B$, B variables aléatoires i.i.d à valeurs dans $\mathcal{P}(\{1,\ldots,n\})$ et de même loi qu'une variable aléatoire Θ. Pour tout $i \in \{1,\ldots,B\}$ on note $\mathcal{L}_n^{\Theta_i} = \{(X_j, Y_j);\ j \in \Theta_i\}$ les échantillons d'apprentissage tirés selon $\Theta_1, \ldots, \Theta_B$. L'estimateur $\widehat{\mathcal{R}}^B(\widehat{f}, \mathcal{L}_n, \Theta)$ de l'erreur de généralisation par B découpages successifs en échantillon d'apprentissage et échantillon test est donné par :

$$\widehat{\mathcal{R}}^B(\widehat{f}, \mathcal{L}_n, \Theta) = \frac{1}{B}\sum_{i=1}^{B} \widehat{\mathcal{R}}\left(\widehat{f}, \mathcal{L}_n^{\Theta_i}\right). \tag{1.8}$$

Une autre stratégie très répandue d'estimation de l'erreur de généralisation est la méthode de validation croisée par $K - folds$ (Rodriguez et al., 2010). Soit $K \in \{2, 3, \ldots, n\}$, on note $\Theta_1, \ldots, \Theta_K$ une partition aléatoire de $\{1,\ldots,n\}$ en K sous-ensembles, appelés blocs et vérifiant $\bigcup_{i=1}^K \Theta_i = \{1,\ldots,n\}$ et $\Theta_i \cap \Theta_j = \emptyset$ pour tout $i \neq j$. Pour tout $i \in \{1,\ldots,K\}$, on note $\mathcal{L}_n^{-\Theta_i} := \{(X_j, Y_j);\ j \notin \Theta_i\}$ l'ensemble des observations privé des observations contenues dans $\mathcal{L}_n^{\Theta_i}$. On définit alors l'estimateur

de l'erreur de généralisation par la méthode des *K-folds* par :

$$\widehat{\mathcal{R}}^{K-folds}\left(\widehat{f}\right) = \frac{1}{K}\sum_{i=1}^{K}\sum_{j\in\Theta_i} L\left(Y_j, \widehat{f}(\mathcal{L}_n^{-\Theta_i}; X_i)\right) \tag{1.9}$$

Remarque 1.1.2. Dans la pratique, la partition $(\Theta_i)_{i=1}^K$ peut être tirée de la manière suivante : on permute aléatoirement les éléments de $\{1, \ldots, n\}$ puis on découpe de manière régulière l'ensemble permuté en K blocs de même taille.

Tout comme pour les découpages successifs, où l'on doit déterminer la proportions d'observations dans l'échantillon d'apprentissage, la difficulté ici est la sélection du nombre de blocs K (Arlot and Lerasle, 2016). Une variante de la méthode par validation croisée *K-folds* est la méthode du *Leave-One-Out*, elle consiste en un *K-folds* avec $K = n$ ce qui nous évite d'avoir à choisir un paramètre et qui rend l'estimateur non aléatoire. Nous verrons dans la Section 1.3.2 que la méthode des forêts aléatoires intègre une stratégie d'estimation ingénieuse de l'erreur de prédiction.
En apprentissage statistique, la plupart des problématiques de prédiction se scindent en deux cadres distincts : la régression et la classification. Ils se différencient par la nature de la variable de sortie Y.

1.1.4 Régression

On parle de régression lorsque la variable de sortie $Y \in \mathcal{Y}$ est continue, typiquement lorsque $\mathcal{Y} = \mathbb{R}$ ou un intervalle de $\mathbb{R}$. Lorsque $\mathcal{Y}$ est multivariée *i.e.* lorsque $\mathcal{Y} = \mathbb{R}^d$ ou un intervalle de $\mathbb{R}^d$, on parle de régression multivariée. Pour une présentation extêmement claire, précise et agréable de la régression, nous renvoyons sans réserve à l'excellent Györfi et al. (2006).

Exemple 1.1.4. Un exemple de régression univariée peut être la prédiction de la température dans la ville de Twin Peaks à un temps donné. De manière similaire, si l'on souhaite prédire la température dans les villes de Twin Peaks, Bordeaux et Saint-Rabier à un temps donné on est dans un cadre de régression multivariée.

On modélise le lien entre les variables d'entrées X et la variable de sortie Y par le modèle statistique de régression non-paramétrique suivant :

$$Y = g(X) + \epsilon \tag{1.10}$$

La fonction g, appelée fonction de régression, est inconnue, elle représente le lien entre X et Y. La variable aléatoire réelle ϵ représente l'erreur de mesure. On suppose qu'en moyenne l'erreur de mesure est nulle conditionnellement à X *i.e.* $\mathbb{E}(\epsilon|X) = 0$ presque surement. De par cette hypothèse, on peut réecrire la fonction de régression g sous la forme :

$$g(X) = \mathbb{E}(Y|X) \tag{1.11}$$

presque surement.

Le modèle de régression non paramétrique (1.10) n'impose aucune containte sur la fonction g. En effet, contrairement aux modèles paramétriques, comme en régression linéaire simple ou multiple la fonction g ne dépend a priori d'aucun paramètre. De par l'absence totale de contraintes, mis à part sur la distribution conditionnelle du bruit pour des raisons d'identifiabilité, le modèle non paramétrique (1.10) est le modèle de régression avec un bruit additif le plus général que l'on puisse écrire.

Remarque 1.1.3. Il est tout-à-fait possible de considérer des modèles non paramétriques avec un bruit qui ne serait pas additif (Firth, 1988). On peut par exemple considérer le modèle non paramétrique multiplicatif qui s'écrit $Y = g(X)\epsilon$. Cependant, considérer un modèle dans lequel le bruit intervient différement que de manière additive ne fait pas vraiment sens dans les applications de cette thèse et n'est pas très répandu dans la littérature. On ne considère donc que des modèles avec bruit additif.

Dans le cadre de la régression, la fonction de coût la plus communément utilisée dans $\mathbb{R}$ est la fonction de coût quadratique. Elle est donnée par :

$$L(y, y') = (y - y')^2 \qquad \forall y, y' \in \mathbb{R}$$

Pour tout prédicteur $f \in \mathcal{F}(\mathcal{X}, \mathbb{R})$, l'erreur de prédiction généralisée associée au coût quadratique est donnée par

$$\mathcal{R}_P(f) = \mathbb{E}\left[(f(X) - Y)^2\right] \tag{1.12}$$

En utilisant le modèle non paramétrique (1.10), on peut réecrire l'erreur quadratique :

$$\mathcal{R}_P(f) = \mathbb{E}\left[(f(X) - g(X))^2\right] + \mathbb{E}(\epsilon^2) \tag{1.13}$$

On en déduit que dans le cadre du modèle non paramétrique (1.10), le risque de Bayes est donné par $\mathcal{R}_P^\star = \mathbb{E}(\epsilon^2)$ et donc que la fonction de régression g est un prédicteur de Bayes.

Propriété 1.1.1. Soit $f \in \mathcal{F}(\mathcal{X}, \mathbb{R})$ un prédicteur de Bayes, alors sous les hypothèses du modèle non paramétrique (1.10),

$$f(X) = g(X) \tag{1.14}$$

presque sûrement.

Démonstration. Soit f un prédicteur de Bayes, alors $\mathcal{R}_P(f) = \mathcal{R}_P^\star = \mathbb{E}(\epsilon^2)$ ce qui implique par l'équation (1.13) que $\mathbb{E}[(f(X) - g(X))^2] = 0$ et donc que $f(X) = g(X)$ presque surement. □

Ainsi dans le cadre du modèle non paramétrique (1.10), l'erreur de prédiction la plus faible que puisse obtenir un prédicteur est donnée par $\mathcal{R}_P^\star = \mathbb{E}(\epsilon^2)$. Cette erreur ne dépend que de la variance et de la moyenne du bruit ϵ. Ce résultat vient confirmer l'intuition selon laquelle l'erreur de mesure (ou le bruit) est le seuil infranchissable de toute méthode, aussi bon soit un prédicteur, il ne pourra jamais être plus précis qu'une quantité de bruit intrinsèque aux erreurs de mesure. Ainsi, l'amélioration des techniques de mesures physiques et en particulier l'amélioration de leur précision sont une condition nécessaire à l'amélioration des capacités prédictives des méthodes.
Dans le cadre de la régression, la proposition (1.1.1) nous dit que la fonction de régres-

sion g est unique à un ensemble négligéable près. De plus, d'après l'équation (1.13), trouver le prédicteur le plus performant possible, au sens de l'erreur quadratique, revient à estimer la fonction de régression g le mieux possible. Ainsi, dans le cadre du modèle non paramétrique (1.10), le problème de prédiction est un problème d'estimation et inversement. Ceci n'est pas le cas dans le cadre de la classification.

1.1.5 Classification

On parle de classification lorsque l'espace $\mathcal{Y}$ est fini, c'est-à-dire lorsque la variable de sortie Y prend un nombre fini de valeurs. Dans ce cas, au lieu de parler de prédicteur, on parlera de classifieur. De manière similaire, plutôt que de parler d'erreur de généralisation ou d'erreur de prédiction, on parlera d'erreur de classification. Les différentes valeurs possibles pour la variable sont appelées classes (on parle aussi parfois de modalités). Nous renvoyons à l'excellent Devroye et al. (2013) pour plus de détails sur le cadre de la classification.

Exemple 1.1.5. Un exemple de problème de classification dans le domaine médical est la prédiction du stade d'évolution d'une maladie chez un patient. Par exemple si une maladie possède M stades allant stade 1 au stade M, le 1 codant pour "non malade" jusqu'au stade M codant pour "phase terminale".

Dans le cadre de la classification, une fonction de coût naturelle est la 0-1 définie par :

$$L(y, y') = \mathbb{1}\{y \neq y'\} \qquad y, y' \in \mathcal{Y} \tag{1.15}$$

où $\mathbb{1}$ est la fontion indicatrice. Pour tout classifieur $f \in \mathcal{F}(\mathcal{X}, \mathcal{Y})$, l'erreur de classification de f associée à la fonction de coût 0-1 est alors donnée par :

$$\mathcal{R}_P(f) = \mathbb{E}\left(\mathbb{1}\left\{f(X) \neq Y\right\}\right) = \mathbb{P}\left(f(X) \neq Y\right) \tag{1.16}$$

Le classifieur de Bayes $f^\star$ pour la fonction de coût 0-1 est alors donné par :

$$f^\star(x) = \underset{y \in \mathcal{Y}}{\arg\min} \mathbb{P}\left(Y = y | X = x\right) \qquad \forall x \in \mathcal{X} \tag{1.17}$$

Contrairement à la régression, la problématique de classification n'équivaut pas à un problème d'estimation. En effet, ici il n'est pas nécessaire de bien estimer les probabilités conditionnelles de chacune des classes pour prédire la bonne classe. Par exemple, dans un problème à deux classes $\mathcal{Y} = \{0, 1\}$, si $\mathbb{P}(Y = 0|X = x) = 0.95$ alors le classifieur de Bayes prédit la classe 1. Si on a estimé cette probabilité à 0.55 alors notre classifieur prédira quand même la classe 1 quand bien même l'estimation de la probabilité conditionnelle est très éloignée de la réalité.
On peut faire le lien avec la fonction de régression g définie par l'équation (1.11). Notons $\mathcal{Y} = \{0, \ldots, M\}$ l'espace de sortie constitué de $M + 1$ classes, on peut dans ce cas montrer que la fonction f définie par

$$f(x) = \arg\min_{c \in \mathcal{Y}} |c - g(x)| \qquad \forall x \in \mathcal{X}. \tag{1.18}$$

est bien un classifieur de Bayes pour la fonction de coût 0-1.

Remarque 1.1.4. Il est toujours possible, a une transformation bijective près, de ramener un problème de classification à $M + 1$ classes au problème de classification sur $\{0, \ldots, M\}$.

La règle d'apprentissage des forêts aléatoires (modulo de petites modifications) peut aussi bien s'utiliser dans un cadre de classification que dans un cadre de régression. La règle d'apprentissage des forêts aléatoires appartient à une famille de règles d'apprentissage appelées méthodes d'ensemble.

1.2 Méthodes d'ensemble

L'idée générale des méthodes d'ensemble est de construire q prédicteurs puis de les aggréger pour former un nouveau prédicteur censé combiner le meilleur des prédicteurs qui le composent. Nous renvoyons au très bon article de Dietterich (2000) pour une présentation des méthodes d'ensemble. On donne ici une définition générale de la notion de fonction d'aggrégation :

Définition 1.2.1 (Fonction d'aggrégation)**.** On appelle fonction d'aggrégation toute fonction mesurable

$$\Psi : \bigcup_{q \geq 1} \mathcal{Y}^q \longmapsto \mathcal{Y}. \tag{1.19}$$

Dans le cadre de la régression, la fonction d'aggrégation choisie en général est la moyenne. Plus précisément, soient $\widehat{f}_1, \ldots, \widehat{f}_q$, q prédicteurs qui peuvent avoir été construits à partir de règles d'apprentissage différentes. On peut alors définir le prédicteur aggrégé $\overline{f}_q$ par la moyenne des q prédicteurs individuels :

$$\overline{f}_q(x) = \frac{1}{q} \sum_{i=1}^{q} \widehat{f}_i(x) \qquad \forall x \in \mathcal{X} \tag{1.20}$$

Dans le cadre de la classification, la fonction d'aggrégation utilisée est la classe majoritairement prédite par les q prédicteurs $\widehat{f}_1, \ldots, \widehat{f}_q$. Au cours de cette thèse, nous serons amenés à considérer des espaces continus bien plus complexes dans lesquels il n'y a a priori pas d'opérations d'addition et donc pas de notion de moyenne au sens défini ci-dessus. Dans de tels espaces nous serons amenés à considérer des fonctions d'aggrégation différentes, ce sera d'ailleurs un des principaux apports de cette thèse. Illustrons de manière très simple les avantages de l'aggrégation de prédicteurs en se donnant un cadre simplifié. On considère $Z_1, \ldots, Z_q$, q variables aléatoires réelles identiquement distribuées, de même variance $\sigma^2 > 0$ et de corrélation $-1 < \rho < 1$. On définit $\overline{Z}_q$ l'aggrégation de ces q variables aléatoires par la moyenne $\overline{Z}_q = \sum_{i=1}^{q} Z_i$. Un calcul simple permet de montrer que la variance de $\overline{Z}_q$ est donnée par :

$$Var\left(\overline{Z}_q\right) = \sigma^2 \rho + \frac{1-\rho}{q} \sigma^2 < \sigma^2 \tag{1.21}$$

Ici, le fait d'aggréger les différentes variables Z_i permet d'obtenir une variable aléatoire ayant une variance plus petite. Cette dernière dépend de la corrélation des différentes variables aléatoires, plus elle est proche de zéro, plus la variance est faible. Considérons maintenant que ces Z_i sont des prédicteurs. Cet exemple très simple nous apprend que les prédicteurs individuels doivent être les moins corrélés les uns des autres et donc les plus différents possible. On doit s'approcher du cas idéal où

les prédicteurs sont indépendants entre eux. Enfin, les prédicteurs individuels doivent être tous relativement bons. Cependant, il ne s'agit pas ici de réduire la variance de prédicteurs, tous individuellement mauvais. Dans la pratique, il est impossible d'avoir des prédicteurs indépendants, surtout lorsqu'ils sont construits à partir du même échantillon d'apprentissage $\mathcal{L}_n$. Il convient alors de mettre en place des stratégies pour construire des prédicteurs les plus décorrélés les uns des autres à partir d'un échantillon d'apprentissage commun $\mathcal{L}_n$. C'est l'objectif des stratégies présentées ci-dessous.

1.2.1 *bagging*

La méthode du Boostrap Aggregating, appelée *bagging*, a été introduite par Breiman (1996). Le principe du *bagging* est de construire des prédicteurs individuels à partir de la même règle d'apprentissage $\widehat{f}$ sur des sous-échantillons de $\mathcal{L}_n$ tirés aléatoirement, appelés échantillons de *bootstrap*, puis de les aggréger en un prédicteur final. L'idée de cette méthode est qu'en donnant des entrées différentes à une même règle d'apprentissage, cette dernière produira des prédicteurs différents.

Définition 1.2.2 (Prédicteur du *bagging*)**.** Soit $\mathcal{L}_n$ un échantillon d'apprentissage, soit $\widehat{f}$ une règle d'apprentissage et soit $\Theta_1, \ldots, \Theta_q$, q variables aléatoires *i.i.d* de même loi qu'une variable Θ à valeurs dans $\mathcal{P}(\{1, \ldots, n\})$. On appelle échantillon de *bootstrap* les sous-échantillons $\mathcal{L}_n^{\Theta_1}, \ldots, \mathcal{L}_n^{\Theta_q}$ tirés à partir des variables aléatoires $\Theta_1, \ldots, \Theta_q$ selon

$$\mathcal{L}_n^{\Theta_i} = \{(X_j, Y_j)\,,\ j \in \Theta_i\} \qquad \forall i = 1, \ldots, q$$

On appelle prédicteur du *bagging* l'aggrégation des prédicteurs individuels $\widehat{f}\left(\mathcal{L}_n^{\Theta_1}\right), \ldots,$ $\widehat{f}\left(\mathcal{L}_n^{\Theta_q}\right)$ construits à partir des échantillons de *bootstrap*.

Le prédicteur du *bagging* dépend de la variable de tirage Θ des échantillons de *bootstrap* $\mathcal{L}_n^{\Theta}$. Dans la pratique, cette variable représente le tirage avec remise de n éléments de $\{1, \ldots, n\}$. Il est tout à fait possible de considérer d'autres manières de tirer les sous-échantillons de *bootstrap*. On peut par exemple fixer $k < n$ et tirer k

éléments de $\{1, \ldots, n\}$ sans remise.

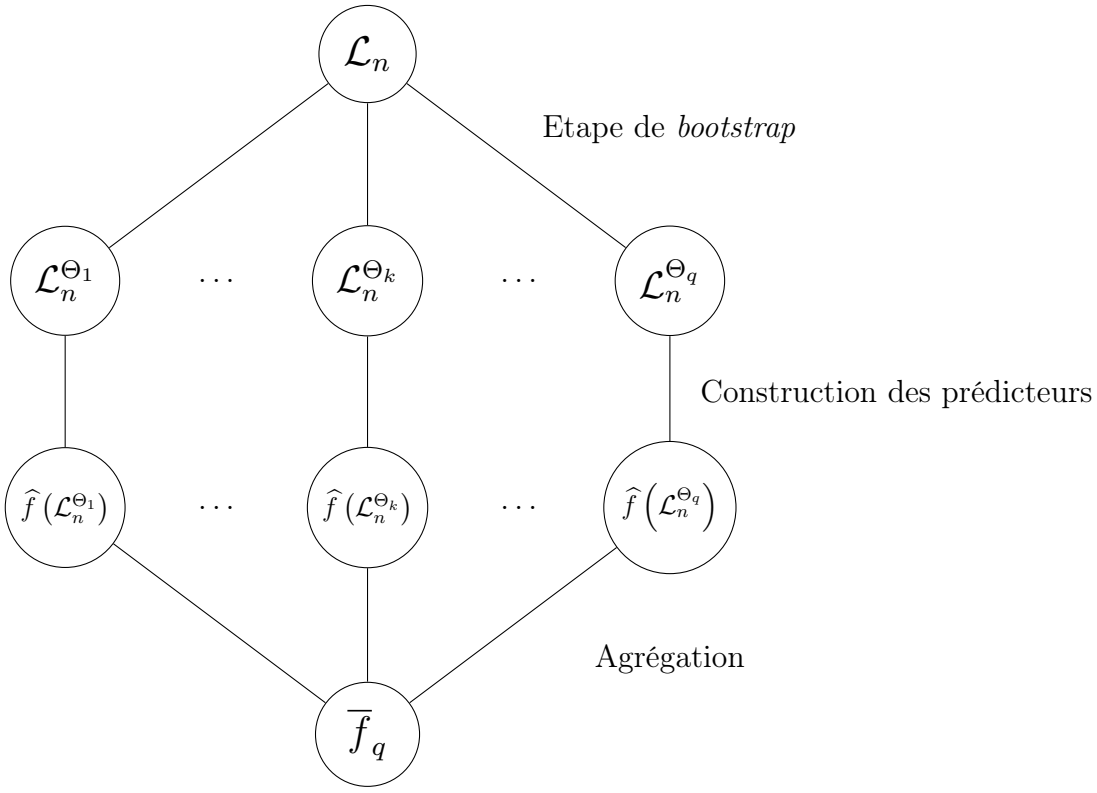

FIGURE 1-1 – Schéma du *bagging*

Une des grandes forces du *bagging* est sa capacité à produire, à partir de règles d'apprentissage non consistantes, des prédicteurs aggrégés consistants. C'est le cas de la règle d'apprentissage par plus proche voisin qui consiste à prédire par la valeur de la variable réponse (ou la classe) de l'observation la plus proche dans l'espace des variables explicatives. Biau et al. (2008) ont montré que faire du *bagging* sur cette règle d'apprentissage, qui n'est pas consistante, menait à un prédicteur consistant.

Dans la méthode du *bagging* les prédicteurs sont tous construits indépendamment les uns des autres. Cette méthode peut donc être implémentée en parallélisant la construction des prédicteurs.

1.2.2 *boosting*

La méthode du *boosting* a été introduite par Freund et al. (1996). Tout comme pour le *bagging*, l'idée générale est de construire des prédicteurs sur des échantillons de *bootstrap* à partir d'une même règle d'apprentissage $\widehat{f}$. Cependant, contrairement à la méthode du *bagging* où les échantillons de *bootstrap* sont tirés indépendamment les uns des autres, ici le tirage Θ_i de l'échantillon de *bootstrap* $\mathcal{L}_n^{\Theta_i}$ dépend des performances du prédicteur $\widehat{f}\left(\mathcal{L}_n^{\Theta_{i-1}}\right)$ construit sur l'échantillon de *bootstrap* précédent. Soit Θ_1 le tirage du premier échantillon de *bootstrap* $\mathcal{L}_n^{\Theta_1}$ et soit $\widehat{f}(\mathcal{L}_n^{\Theta_1})$ le prédicteur construit sur cet échantillon. Sur ce premier tirage, chaque observation a la même probabilité d'être tirée. On tire ensuite le deuxième échantillon de *bootstrap* $\mathcal{L}_n^{\Theta_2}$ mais cette fois le tirage Θ_2 n'est pas uniforme. Chaque observation (X_i, Y_i) a une probabilité d'être tirée qui dépend de l'erreur de prédiction $L\left(Y_i, \widehat{f}(\mathcal{L}_n^{\Theta_1}, X_i)\right)$ du premier prédicteur sur ce couple pour une fonction de coût L donnée. Le principe est d'augmenter la probabilité d'être tirée pour une observation mal prédite et inversement, de diminuer la probabilité d'être tirée pour une observation bien prédite. Une fois le nouvel échantillon $\mathcal{L}_n^{\Theta_2}$ tiré, on construit le prédicteur $\widehat{f}(\mathcal{L}_n^{\Theta_2})$. On tire ensuite un nouvel échantillon de *bootstrap* $\mathcal{L}_n^{\Theta_3}$ en fonction de l'erreur de prédiction de $\widehat{f}\left(\mathcal{L}_n^{\Theta_2}\right)$ sur $\mathcal{L}_n$ et ainsi de suite jusqu'à obtenir une collection de q prédicteurs. Finalement, le prédicteur du *boosting* est obtenu en aggrégeant les q prédicteurs construits $\widehat{f}\left(\mathcal{L}_n^{\Theta_1}\right), \ldots, \widehat{f}\left(\mathcal{L}_n^{\Theta_q}\right)$. L'idée générale du *boosting* est d'améliorer ses faiblesses plutôt que de renforcer ses points forts. Afin d'améliorer les capacités prédictives globales, on va se focaliser au fur et à mesure sur les observations mal prédites par la règle d'apprentissage, et donc construire des prédicteurs spécifiquement sur les données difficiles à prédire.

Remarque 1.2.1. Comme la méthode du *boosting* est séquentielle, contrairement à la méthode du *bagging* il n'est pas possible de paralléliser la construction des différents prédicteurs pour accélérer les temps de construction du prédicteur final.

Les méthodes du *bagging* et du *boosting* ont donc toutes deux recours à des stratégies de sous-échantillonnage aléatoire de l'échantillon d'apprentissage pour construire des prédicteurs différents à partir d'une règle d'apprentissage commune. Toujours

dans l'idée de décorréler les prédicteurs, Breiman (2000a) introduit la méthode de perturbation des sorties.

1.2.3 Perturbation des sorties

Il s'agit toujours de donner à une règle d'apprentissage commune des échantillons d'apprentissage les plus différents les uns des autres. Cependant, au lieu de tirer aléatoirement des sous-échantillons de $\mathcal{L}_n$, on va perturber aléatoirement les observations de la variable de sortie Y. Ainsi, pour une même observation X_i en entrée, la variable réponse associée Y_i sera toujours différente (dans le cas continu). Pour chaque perturbation nous avons un échantillon d'apprentissage différent (qui ne diffère que par les observations de la variable réponse), la règle d'apprentissage est alors appliquée sur ces échantillons pertubés, puis on aggrège les différents prédicteurs.

Remarque 1.2.2. Comme pour la méthode du *bagging* pour le tirage aléatoire des sous-échantillons, les perturbations de Y sont effectuées indépendamment les unes des autres. Il est ainsi possible de construire chaque prédicteur en parallèle.

1.2.4 *random subspace*

Une stratégie pour produire des estimateurs différents est d'appliquer la règle d'apprentissage sur des parties de l'espace des variables explicatives $\mathcal{X}$, et non sur $\mathcal{X}$ tout entier. Dans le cas où $\mathcal{X} = \mathbb{R}^p$, Tin Kam Ho (1998) introduit la méthode dite de *random subspace*. L'idée de la méthode est de tirer aléatoirement un sous-ensemble de variables (et donc de coordonnées dans $\mathbb{R}^p$) puis d'appliquer la règle d'apprentissage seulement sur les variables tirées. Le prédicteur final est obtenu par aggrégation des prédicteurs construits sur les différents sous-espaces de $\mathbb{R}^p$.

Remarque 1.2.3. Les tirages des coordonnées sur lesquelles est appliquée la règle d'apprentissage sont indépendants. Tout comme pour le *bagging* et la méthode de perturbation des sorties, il est donc possible de construire les différents prédicteurs en parallèle.

Les quatre méthodes d'ensemble présentées ci-dessus utilisent une couche d'aléatoire pour produire des échantillons d'apprentissage différents. Ces méthodes ne sont pas cloisonnées et peuvent tout à fait être combinées dans le but de produire des prédicteurs encore plus différents. Par exemple, il est possible de faire un *bagging* en perturbant les observations de sortie pour les échantillons de *bootstrap* tirés. Tout comme il est possible de faire du *boosting* en tirant pour chaque échantillon de *bootstrap* un sous-ensemble de coordonnées sur lequel notre prédicteur sera construit. Néanmoins, il est plus optimal, au sens des temps de calculs, de combiner les méthodes de *bagging*, de perturbation des sorties et *random subspace* car ces stratégies permettent toutes de paralléliser la construction des prédicteurs individuels. La méthode des forêts aléatoires est elle-même une combinaison de *bagging* et d'une variante de la méthode *random subspace*.

1.3 Forêts aléatoires

La méthode des forêts aléatoires a été introduite par Breiman (2001) et appartient à la famille des méthodes d'ensemble. Son principe général est d'aggréger des prédicteurs par arbres CART randomisés, qui sont une variante de la méthode des arbres CART. Cette méthode d'apprentissage est la brique de matière essentielle à la construction des forêts aléatoires. Nous détaillons cette méthode dans la prochaine section.

1.3.1 Arbres CART

La méthode d'apprentissage statistique par arbre CART (pour **C**lassification **A**nd **R**egression **T**rees) a été introduite par Breiman et al. (1984). Cette règle d'apprentissage fonctionne aussi bien en régression qu'en classification. L'idée est de partitionner récursivement et de manière dyadique l'espace des variables explicatives $\mathcal{X}$.
Considérons le cas où toutes les variables explicatives sont toutes continues *i.e.* $\mathcal{X} = \mathbb{R}^p$. On adopte la notation $X_i = \left(X_i^{(1)}, \ldots, X_i^{(j)}\right)$ où $X_i^{(j)}$ est la $j$ème variable explicative pour l'observation i. Partant de l'espace $\mathbb{R}^p$ tout entier, appelé noeud racine,

on découpe l'espace en deux régions. Pour cela, on introduit la notion de découpage (*split* en anglais) :

Définition 1.3.1. On appelle découpage tout couple $(j, s) \in \{1, \dots, p\} \times \mathbb{R}$. Soit $t \subseteq \mathbb{R}^p$, on définit le noeud fils gauche t_g et noeud fils droit t_d de t associés au découpage (j, s) par

$$t_g = \left\{x \in t,\ x^{(j)} \leq s\right\} \qquad t_d = \left\{x \in t,\ x^{(j)} > s\right\} \tag{1.22}$$

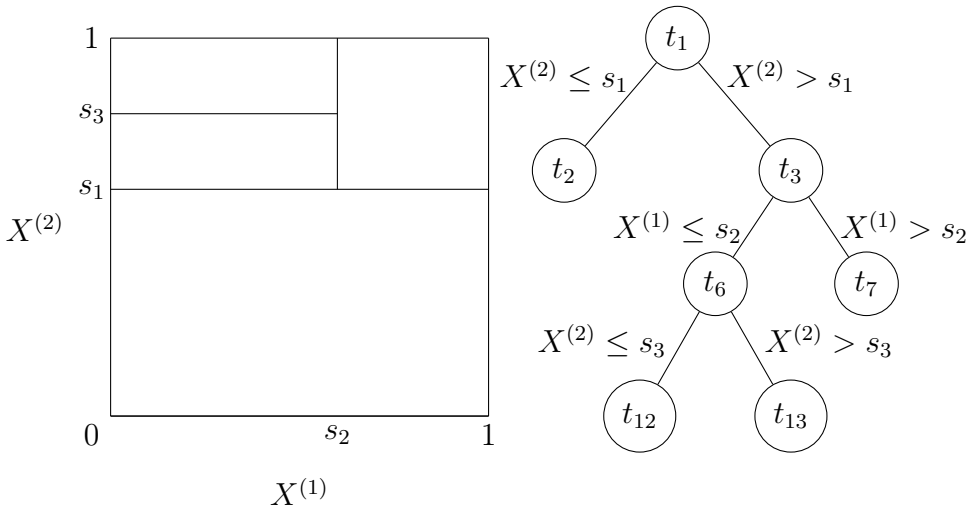

FIGURE 1-2 – Une partition du carré unité et son arbre CART associé.

Pour tout découpage (j, s), les observations (X_i, Y_i) telles que la $j$ème variable de X_i est plus petite que le seuil s vont dans le noeud fils gauche et toutes celles qui sont plus grandes que s vont dans le noeud fils droit. Dans la méthode introduite par Breiman, les découpages se font toujours perpendiculairement à un axe j de $\mathbb{R}^p$. Cependant, il existe des méthodes dont les découpages peuvent se faire de manière oblique (Menze et al., 2011).

Remarque 1.3.1 (Découpage sur une variable explicative discrète)**.** Dans cette section nous nous restreignons seulement aux variables explicatives continues *i.e.* $\mathcal{X} = \mathbb{R}^p$. Cependant la méthode des arbres CART s'adapte très bien sur des variables explicatives discrètres. En effet, un découpage d'un noeud t selon une variable discrète

est simplement une partition en deux groupes des classes présentes dans le noeud t pour la variable de découpe. Par exemple, si la $j$ème variable explicative contient les classes $\{2,4,5\}$ dans le noeud t alors on découpe le noeud selon une des partitions $\{2\} \cup \{4,5\}$; $\{4\} \cup \{2,5\}$ et $\{5\} \cup \{2,4\}$.

L'objectif du découpage est de séparer l'espace tout entier en deux régions les plus "homogènes" possible. Pour ce faire on va mesurer l'hétérogénéité des noeuds après découpage par ce que l'on appelle l'impureté de ces derniers. La notion d'impureté n'est pas la même en régression ou en classification. Dans le cadre de la régression, il n'y a pas de définition formelle de l'impureté d'un noeud. Dans les arbres CART introduits par Breiman et al. (1984), on mesure l'impureté $I(t)$ du noeud t par la variance empirique des observations de la variable réponse dans ce noeud, *i.e.* :

$$I(t) = \frac{1}{N_n(t)} \sum_{i:X_i \in t} \left(Y_i - \overline{Y}_t\right)^2 \tag{1.23}$$

où $\overline{Y}_t$ est la moyenne empirique des observations de la variable de réponse dans le noeud t et $N_n(t)$ est le nombre d'observations dans le noeud t. Dans le cadre de la classification, on dispose d'une définition formelle donnée par Breiman :

Définition 1.3.2 (Fonction d'impureté)**.** Soit $\mathcal{Y} = \{1, \ldots, L\}$. On appelle fonction d'impureté I toute fonction à valeurs positives définie sur l'ensemble

$$\left\{ (p_1, \ldots, p_L)\,;\ \forall i = 1, \ldots, L,\ p_i \geq 0;\ \sum_{i=1}^{L} p_i = 1 \right\}$$

vérifiant les propriétés suivantes :

1. I admet un unique maximum en $\left(\frac{1}{L}, \ldots, \frac{1}{L}\right)$.
2. I admet un minimum en chaque $e_i = (0, \ldots, 0, \underbrace{1}_{i-\text{ème}}, 0, \ldots, 0)$, les éléments de la base canonique de $\mathbb{R}^L$.
3. I est une fonction symétrique en les $p_1, p_2, \ldots, p_L$.

La première propriété d'une fonction d'impureté est qu'elle doit atteindre son maximum s'il y a une distribution uniforme des classes dans le noeud, c'est l'état de

l'hétérogénéité maximale. La deuxième propriété nous indique que l'impureté d'un noeud doit être minimale s'il n'y a, dans ce noeud, que des éléments de la même classe. Enfin, la troisième propriété nous indique que l'impureté ne dépend que des proportions des classes dans un noeud et non des classes qui y sont attribuées. Par exemple, dans un cadre de classification binaire $\mathcal{Y} = \{0, 1\}$, un noeud qui contient 40% de 0 et 60% de 1 aura la même impureté qu'un noeud qui contient 60% de 0 et 40% de 1. La fonction de split utilisée dans l'algorithme CART est l'indice de Gini G qui est défini par :

$$G\left(p_1, \ldots, p_L\right) = 1 - \sum_{i=1}^{L} p_i^2 \tag{1.24}$$

A partir de la fonction d'impureté donnée par l'indice de Gini G, on définit l'impureté du noeud t par $I(t) = G\left(\widehat{p}_1(t), \ldots, \widehat{p}_L(t)\right)$ où $\widehat{p}_i(t)$ est la proportion d'éléments de la classe i dans le noeud t.

Exemple 1.3.1. D'autres fonctions d'impureté peuvent être utilisées. On donne ici deux exemples très courants dans la littérature, l'entropie de Shannon et l'erreur de classification :

— **L'entropie de Shannon** (Lin, 1991) :

$$H\left(p_1, \ldots, p_L\right) = -\sum_{i=1}^{L} p_i \log p_i$$

— **L'erreur de classification :**

$$C\left(p_1, \ldots, p_L\right) = 1 - \max_{1 \leq i \leq L} p_i$$

Il est important de souligner qu'utiliser une fonction d'impureté différente, ou plus simplement un calcul différent de l'impureté des noeuds, mène à des découpages différents. Ainsi, il paraît naturel d'adapter la fonction d'impureté à la problématique considérée. Par exemple, pour adapter l'algorithme des arbres CART à l'analyse de survie, Ishwaran et al. (2008) ont considéré un score du log rank comme fonction d'impureté. Dans le Chapitre 3, nous introduirons une nouvelle fonction d'impureté

afin d'adapter le critère de découpage des arbres CART au cadre de la régression dans des espaces métriques généraux.

Plus l'impureté d'un noeud est faible, plus il est homogène en termes des Y_i qu'il contient. L'objectif du découpage est alors de minimiser l'impureté des noeuds fils obtenus, c'est-à-dire, de diminuer le plus possible l'impureté associée à ce découpage. On mesure la diminution de l'impureté associée au découpage (j, s) par :

$$\Delta(j, s) := I(t) - p(t_g)I(t_g) - p(t_d)I(t_d)$$

où t_g et t_d sont respectivement les noeuds fils gauche et droit du noeud t et $p(t_q)$ et $p(t_d)$ sont respectivement les probabilités d'être dans le noeud t_g et t_d. Dans la pratique, ce sont les proportions d'observations de $\mathcal{L}_n$ dans les noeuds fils gauche et droit.
On note $(j^\star, s^\star)$ le découpage optimal, c'est-à-dire celui qui maximise la diminution de l'impureté :

$$(j^\star, s^\star) = \underset{(j,s)\in\{1,\dots,p\}\times\mathbb{R}}{\arg\max}\ \Delta(j, s) \tag{1.25}$$

Une fois le noeud racine découpé, on se restreint aux deux noeuds fils obtenus que l'on découpe aussi en deux selon le même procédé. On répète le processus sur les nouveaux noeuds fils obtenus et ainsi de suite. Un noeud n'est pas découpé s'il vérifie une règle d'arrêt. Il y a plusieurs manières de déterminer une règle d'arrêt. On peut par exemple ne pas découper un noeud si le nombre d'observations dans ce noeud est inférieur à un nombre pré-déterminé. Il est aussi possible de se donner un critère d'arrêt basé sur la réduction de l'impureté $\Delta(j, s)$, par exemple en ne découpant pas un noeud si le meilleur découpage mène à une réduction de l'impureté inférieur à un seuil préalablement fixé.

Définition 1.3.3. Soit T un arbre, on appelle feuille tout noeud de T qui n'est pas découpé. On note $\widetilde{T}$ l'ensemble des feuilles de T.

Définition 1.3.4. On appelle arbre maximal, noté $T_{\max}$, tout arbre pleinement déve-

loppé *i.e.* tout arbre dont les feuilles vérifient toutes le critère d'arrêt préalablement défini.

Dans la pratique, la règle d'arrêt utilisée est de ne pas découper un noeud pur, c'est-à-dire un noeud composé seulement des mêmes observations pour la réponse. En régression, cela revient à avoir des feuilles qui ne contiennent qu'une seule observation. En classification, cela revient à ne pas découper un noeud qui ne contient que des observations d'une même classe. Dans cette thèse, nous ne considèrerons que cette règle d'arrêt.

A tout arbre CART, on associe une partition de $\mathbb{R}^p$ (voir la Figure 1-2), chaque feuille de l'arbre code pour une cellule de la partition. Le prédicteur (ou classifieur) par arbre CART est un prédicteur constant par morceaux sur les cellules de la partition récursivement construite.

Définition 1.3.5 (Prédicteur par arbre)**.** Soit T un arbre, on définit le prédicteur $\widehat{f}_T$ associé à l'arbre T par

$$\widehat{f}_T(x) = \sum_{t \in \widetilde{T}} \overline{Y}_t \mathbb{1}\{x \in t\} \tag{1.26}$$

— En régression : $\overline{Y}_t$ est la moyenne des Y_i dans la feuille t.

— En classification : $\overline{Y}_t$ est la classe majoritaire des Y_i dans la feuille t.

Dans un problème d'apprentissage statistique, on n'utilise jamais le prédicteur associé à l'arbre maximal seul pour prédire. En effet, ce prédicteur a tendance à être surajusté aux données d'apprentissage et donc à avoir une très faible capacité de généralisation, de par le fait que l'arbre maximal minimise l'erreur de prédiction sur les données d'apprentissage $\mathcal{L}_n$. Par conséquent, l'erreur de prédiction sera significativement plus élevée sur des nouveaux échantillons. Ceci s'explique par le fait que plus un arbre est développé, plus la variance de son prédicteur associé augmente. Ainsi, pour un échantillon d'apprentissage $\mathcal{L}_n$, l'arbre maximal est le prédicteur par arbre le plus variable possible et donc le moins général possible. La Figure 1-3 illustre très simplement ce phénomène de surapprentissage. Elle représente la partition associée à

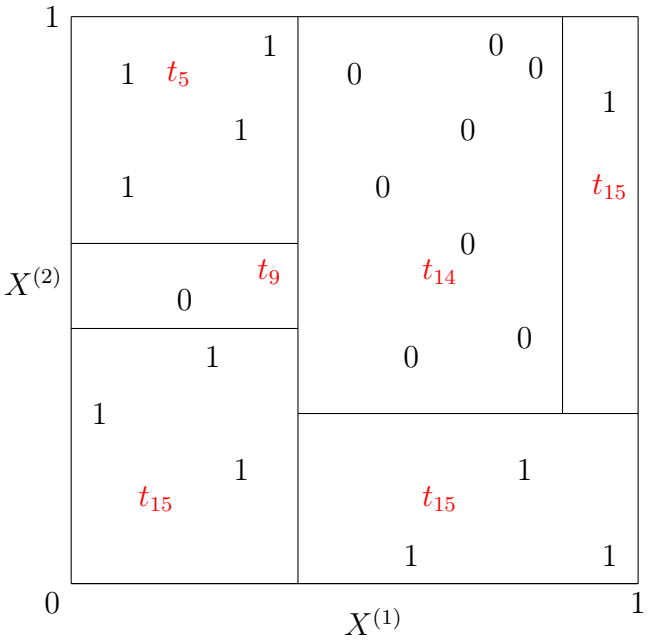

FIGURE 1-3 – Exemple de partition associée à un arbre maximal sur le cube unité

un arbre maximal construit sur le carré unité dans $\mathbb{R}^2$ pour un problème de classification binaire *i.e.* $\mathcal{Y} = \{0, 1\}$. Chaque cellule ne contient que des réponses de la même classe. Ainsi, l'erreur d'apprentissage du prédicteur associé à cet arbre est nulle. Plus particulièrement, considérons la cellule t_9. Pour toute nouvelle observation se situant dans cette cellule, le prédicteur associé à cette partition (et donc à cet arbre) prédira la classe 0. De par la répartition des données sur cet exemple, il nous paraît assez naturel de dire que cette dernière est une observation aberrante. On se retrouve alors à prédire la classe 0 sur toute une région de l'espace qui apparaît clairement comme étant une région dans laquelle on devrait prédire 1. Il en est de même pour la cellule t_{15}, qui par la nature des découpages (perpendiculaire à un axe) mène à prédire 1 dans une très grande région dans laquelle on devrait plutôt prédire la classe 0. Le prédicteur associé à l'arbre maximal est donc extrêmement sensible aux données, et en particulier aux données aberrantes.

Le nombre de cellules du partitionnement, et donc le nombre de feuilles de l'arbre, mesure le niveau d'ajustement du prédicteur aux données d'apprentissage. L'idée

pour éviter ce phénomène de sur-apprentissage est de produire un arbre avec moins de feuilles, et donc un partitionnement avec moins de cellules. L'erreur d'apprentissage pénalisée par le nombre de feuilles de l'arbre est alors le critère de sélection de l'arbre optimal.

Définition 1.3.6 (Erreur d'apprentissage pénalisée)**.** Soit T un arbre et $\widehat{\mathcal{R}}(T;\mathcal{L}_n)$ l'erreur d'apprentissage du prédicteur associé à T. Pour tout $\alpha > 0$, on définit l'erreur d'apprentissage pénalisée $\widehat{\mathcal{R}}_\alpha(T;\mathcal{L}_n)$ par

$$\widehat{\mathcal{R}}_\alpha(T;\mathcal{L}_n) = \widehat{\mathcal{R}}(T;\mathcal{L}_n) + \alpha \# T \tag{1.27}$$

où $\#T$ est le nombre de feuilles de T.

L'arbre finalement sélectionné $T_\alpha^\star$ est le sous-arbre de $T_{\max}$ qui minimise l'erreur d'apprentissage pénalisée :

$$T_\alpha^\star = \underset{T \preceq T_{\max}}{\arg\min}\ \widehat{\mathcal{R}}_\alpha(T;\mathcal{L}_n) \tag{1.28}$$

Breiman a demontré l'existence et l'unicité d'un tel arbre pour tout $\alpha > 0$ ainsi que pour toute fonction de coût. La procédure de sélection du sous-arbre optimal au sens de l'erreur de'apprentissage pénalisée est appelée l'élagage. On renvoie à Gey and Nedelec (2005) pour plus de détails sur cette étape.

Remarque 1.3.2. Notons que pour $\alpha = 0$ alors le meilleur arbre est bien l'arbre maximal $T_{\max}$.

Gestion des données manquantes

Dans les applications réelles, il n'est pas rare d'avoir des données manquantes. C'est le cas lorsqu'un patient oublie de répondre à une question, lorsqu'une électrode ne fonctionne pas, lorsqu'un gène a un niveau d'expression inférieur au seuil de détection technique. Que les données manquantes soient issues d'un problème technique ou qu'elles soient intrinsèques à la problématique étudiée, elles sont toujours un défi supplémentaire à l'analyse des données. La stratégie la plus répandue pour s'accommoder

des données manquantes est l'imputation : on cherche à les remplacer en s'approchant le plus possible des valeurs qu'elles auraient dû être (Husson et al. (2019), Audigier et al. (2016), Schafer and Olsen (1998), Li et al. (2004)). Au moment de la prédiction, la méthode des arbres CART intègre une gestion des données manquantes sans avoir à les imputer. Le principe est le suivant : pour sélectionner le meilleur découpage il a été nécessaire de calculer l'ensemble des découpages possibles pour le noeud en question. Pour chaque découpage alternatif, on calcule le nombre de désaccords avec le découpage optimal. Les découpages alternatifs, appelés *surrogate splits*, sont alors classés par ordre croissant du nombre de désaccords. Si une observation est manquante pour le découpage optimal, on choisit le *surrogate splits* ayant le moins de désaccords avec la règle optimal et dont la variable de découpage n'est pas manquante pour cette observation. Bien que cette stratégie permette de gérer les données manquantes sans avoir à faire de l'imputation de données, Feelders (1999) a illustré la supériorité de l'imputation par rapport à l'approche par *surrogate splits*. C'est le prix de la liberté et de l'indépendance vis-à-vis d'une procédure extérieure d'imputation des données manquantes. Pour conclure, la construction des arbres CART se caractérise par trois notions essentielles :

1. le découpage

2. la règle d'arrêt

3. la stratégie de prédiction dans les feuilles de l'arbre sélectionné

Pour adapter la méthode des arbres CART à des cadres plus complexes que la régression univariée ou la classification, il est alors intéressant de jouer sur ces trois notions. Le travail fondateur de cette thèse repose sur l'adaptation de ces principes afin de pouvoir traiter le cadre plus large de l'apprentissage sur des espaces métriques généraux.

1.3.2 Les forêts aléatoires RF-RI

Construction des forêts aléatoires RF-RI

On détaille dans cette section la construction des forêts aléatoires RF-RI (pour ***R**andom **Forests** with **R**andom **I**nputs*) classiques introduites par Breiman (2001). Cette version des forêts aléatoires est aujourd'hui massivement employée dans de nombreux challenges d'apprentissage statistique. Tout comme pour le *bagging*, la construction des forêts aléatoires RF-RI se décompose en trois phases :

1. Le tirage des échantillons de *bootstrap*
2. La construction des prédicteurs par arbres CART aléatoires
3. L'aggrégation des prédicteurs individuels

La première consiste à tirer les q échantillons de *bootstrap* $\mathcal{L}_n^{\Theta_1}, \ldots, \mathcal{L}_n^{\Theta_q}$. La deuxième étape consiste à construire une variante aléatoire des arbres CART de la manière suivante : pour chaque noeud de l'arbre, plutôt que de mettre en compétition l'ensemble des variables d'entrée, on tire aléatoirement m variables explicatives. On cherche ensuite le meilleur découpage sur ces m variables uniquement. De plus, il n'y a pas d'étape d'élagage dans la construction de ces arbres. La dernière étape consiste en la construction du prédicteur par RF-RI qui est l'aggrégation des différents prédicteurs par arbres CART aléatoires.

Toute méthode d'ensemble a besoin de prédicteurs individuels les plus différents les uns des autres. La méthode des arbres CART étant totalement déterministe, la stratégie des forêts aléatoires pour rendre les arbres différents est d'ajouter deux couches d'aléa. La première est de construire les arbres sur des échantillons d'apprentissage différents, c'est l'étape de *bootstrap* (commune au *bagging*). La deuxième couche d'aléa consiste à restreindre, à chaque découpage, l'ensemble des variables de découpe possibles à m variables tirées aléatoirement. Les m variables explicatives mises en compétition à chaque découpage sont tirées uniformément et sans remise parmi toutes les variables. Ce paramètre est fixé avant la construction des arbres et est le même pour tous les arbres. Lorsque $m = p$ la méthode RF-RI correspond à la

méthode du *bagging* avec des arbres CART maximaux. À l'opposé, lorsque $p = 1$, il n'y a aucune mise en compétition des variables entre elles et la variable de découpe est tirée aléatoirement. Dans ce dernier cas, seul le seuil de découpage est optimisé. Ce paramètre est le plus important puisqu'il règle la quantité d'aléa des arbres qui composent la forêt. Si m est trop faible alors les prédicteurs individuels seront trop mauvais et leur aggrégation mènera à un mauvais prédicteur. Si m est trop grand alors les arbres ne seront pas suffisamment différents. Il convient donc de toujours optimiser ce paramètre. Le deuxième paramètre d'une forêt aléatoire est q le nombre d'arbres. Ce paramètre s'il est suffisamment grand n'a que très peu d'influence sur les performances de la forêt dans la pratique.

Remarque 1.3.3 (Gestion des données manquantes)**.** Contrairement aux arbres CART classiques, la méthode RF-RI ne gère pas les données manquantes. Il n'y a pas de procédure de gestion des données manquantes implémentée dans le paquet *R* `randomForest`. Ceci s'explique par le tirage aléatoire des variables de découpe potentielles. En effet, il est possible que les observations des m variables tirées soient toutes des données manquantes et donc qu'on soit incapable de classer cette observation dans un nouveau noeud. De plus, cette situation qui peut paraître très anecdotique arrive souvent dans des applications où il est courant d'avoir 50, 60 ou 70% de données manquantes. Ceci n'arrive pas dans le cadre des arbres CART classiques car toutes les variables sont mises en compétition.

Erreur OOB

Comme pour le *bagging*, les arbres qui composent la forêt aléatoire sont construits sur des échantillons de *bootstrap* $\mathcal{L}_n^{\Theta_1}, \ldots, \mathcal{L}_n^{\Theta_q}$. A l'instar d'un découpage en données d'apprentissage/test, chaque arbre n'a été construit que sur une partie des données. Il est donc possible de se servir des données non utilisées pour la construction de l'arbre pour estimer l'erreur de généralisation. On appelle échantillon OOB (pour ***O**ut **O**f **B**ag*) l'ensemble des observations non tirées dans l'échantillon de *bootstrap* $\mathcal{L}_n^{\Theta}$. Ce sont littéralement les échantillons "en dehors du sac". Pour chaque observation (X_i, Y_i) de l'échantillon d'apprentissage $\mathcal{L}_n$, on note O_i les indices des échantillons de

bootstrap qui ne contiennent pas cette observation *i.e.* les arbres qui n'ont pas été construits sur cette observation :

$$O_i = \left\{ j \in \{1, \dots, q\},\ (X_i, Y_i) \notin \mathcal{L}_n^{\Theta_j} \right\} \tag{1.29}$$

On peut alors donner la définition de la prédiction OOB de l'observation (X_i, Y_i) ainsi que de l'erreur OOB de la forêt aléatoire.

Définition 1.3.7 (Prédiction OOB)**.** Soit $\widehat{f}$ la règle d'apprentissage par arbre aléatoire. On définit la prédiction OOB de Y_i par

$$\widehat{Y}_i^{OOB} = \frac{1}{\#O_i} \sum_{j \in O_i} \widehat{f}\left(\mathcal{L}_n^{\Theta_i}, X_i\right) \tag{1.30}$$

Définition 1.3.8 (Erreur OOB)**.** On définit l'erreur OOB, notée $\widehat{\mathcal{R}}^{OOB}$ par :

$$\widehat{\mathcal{R}}^{OOB} = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \widehat{Y}_i^{OOB} \right)^2 \quad \text{en régression} \tag{1.31}$$

$$\widehat{\mathcal{R}}^{OOB} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\left\{ Y_i \neq \widehat{Y}_i^{OOB} \right\}} \quad \text{en classification} \tag{1.32}$$

Notons que la définition de prédiction OOB, et donc d'erreur OOB, est la même pour une règle d'apprentissage quelconque. Ces notions sont, modulo la règle d'apprentissage, identiques pour les forêts aléatoires RF-RI et le *bagging*.
L'erreur OOB est un estimateur naturel de l'erreur de généralisation de la forêt aléatoire en le sens qu'il tire pleinement parti des échantillons OOB, c'est-à-dire de la fraction de l'information qui n'est pas utilisée pour la construction de chaque arbre. Dans la définition 1.3.8, on se restreint au cas classique de la régression avec fonction de coût quadratique ainsi que celui de la classification avec fonction de coût 0-1. Or, l'erreur $\widehat{\mathcal{R}}^{OOB}$ peut être définie avec une fonction de coût quelconque L de la manière suivante :

$$\widehat{\mathcal{R}}^{OOB} = \frac{1}{n} \sum_{i=1}^{n} L\left(Y_i, \widehat{Y}_i^{OOB} \right) \tag{1.33}$$

C'est cette forme que nous utilisons dans le Chapitre 3. Enfin, l'erreur OOB dépend

de la règle d'apprentissage $\widehat{f}$, de la fonction de coût L, des échantillons de *bootstrap* $\mathcal{L}_n^{\Theta_1}, \ldots, \mathcal{L}_n^{\Theta_q}$ et donc de la loi de tirage Θ ainsi que de l'échantillon d'apprentissage $\mathcal{L}_n$. C'est par souci de lisibilité que nous avons opté pour la notation $\widehat{\mathcal{R}}^{OOB}$.

L'importance des variables

Il n'est pas rare dans certains domaines, comme en vaccinologie, en génétique ou même en imagerie cérébrale, de disposer de plusieurs milliers de variables explicatives. Il est alors intéressant, dans un but d'interprétation, de pouvoir déterminer quelles sont les variables réellement importantes pour expliquer la variable de sortie. En effet, toutes les variables explicatives ne sont pas nécessairement utiles pour expliquer la variable de sortie. Certaines variables ont même un effet de bruit et peuvent augmenter l'erreur de prédiction lorsqu'elles sont prises en compte. Ainsi, les méthodes de sélection de variables fournissent une réponse à ce double problème : réduire le nombre de variables explicatives pour améliorer les performances du prédicteur et obtenir un modèle plus interprétable.

L'importance des variables est un score qui est attribué à chaque variable explicative. Le score d'importance d'une variable d'entrée mesure l'intensité du lien qui lie cette dernière à la variable de sortie. L'idée générale pour mesurer l'importance d'une variable est de casser le lien entre cette variable et la sortie, puis, de calculer l'augmentation de l'erreur de prédiction obtenue. Plus cette erreur augmente, plus la variable est importante pour expliquer la sortie. Dans la méthode des forêts aléatoires, comme pour l'estimation de l'erreur de prédiction, on utilise les échantillons OOB pour calculer l'importance des variables. On détaille le calcul du score d'importance de la variable $X^{(j)}$. Pour tout $k \in \{1, \ldots, q\}$, on note OOB_k le $k$ième échantillon OOB. On note $errOOB_k$ l'erreur empirique sur l'échantillon OOB_k de l'arbre construit sur l'échantillon de *bootstrap* $\mathcal{L}_n^{\Theta_k}$ *i.e.*

$$errOOB_k = \frac{1}{\#OOB_k} \sum_{i:(X_i,Y_i)\in OOB_k} \widehat{f}(\mathcal{L}_n^{\Theta_k}, X_i) \tag{1.34}$$

On note $\widetilde{OOB}_k^j$ le $k$ième échantillon OOB sur lequel on a permuté aléatoirement les observations de la $j$ième variable. C'est l'étape à laquelle on brise le lien entre la variable $X^{(j)}$ et la sortie Y. On note $err\widetilde{OOB}_k^j$ l'erreur de prédiction sur l'échantillon permuté $\widetilde{OOB}_k^j$ de l'arbre construit sur $\mathcal{L}_n^{\Theta_k}$. Cette opération est répétée sur l'ensemble des échantillons OOB. Le score de l'importance de la variable $X^{(j)}$, notée $\mathcal{VI}(j)$ est alors l'augmentation moyenne de l'erreur de prédiction par arbre aléatoire :

$$VI(j) = \frac{1}{q}\sum_{k=1}^{q}\left(err\widetilde{OOB}_k^j - errOOB_k\right) \tag{1.35}$$

Remarque 1.3.4. L'importance d'une variable peut-être négative. Cela signifie que l'aléatoire apporte plus d'information que la variable en question.

Le score d'importance de la variable $X^{(j)}$ nous apporte de l'information sur l'intensité du lien qui la relie avec la réponse Y relativement aux autres variables. Notamment, le score $VI(j)$ est relatif à l'ensemble des variables considérées et est susceptible de changer si on mesure $VI(j)$ parmi un autre ensemble de variables. Enfin, notons que le score d'importance n'apporte aucune information sur le lien qu'il peut exister entre différentes variables explicatives.

Consistance des forêts aléatoires

On assiste depuis une dizaine d'années à une très forte augmentation du nombre de travaux visant à analyser les propriétés des forêts aléatoires. L'analyse théorique des forêts aléatoires est un défi extrêmement complexe de par la structure récurssive de la méthode, avec notamment la dépendance des tirages des variables de découpe potentielles à chaque noeud. Beaucoup de travaux ont consisté en l'analyse de versions simplifiées des forêts aléatoires. C'est le cas des forêts purement aléatoires introduites par Breiman (2000b). L'idée est de construire les arbres indépendamment des données d'apprentissage en tirant aléatoirement un noeud, puis une variable de découpe puis uniformément le seuil. Biau et al. (2008) démontrent la consistance de ces dernières dans le cadre de la classification. Biau (2012a) montrent ensuite la consistance d'une

version intermédiaire de forêts aléatoires où à chaque étape tous les noeuds sont découpés au milieu selon une variable tirée aléatoirement. Genuer (2012) introduit les arbres et forêts purement uniformément aléatoires (voir la Section 1.3.4). Robin Genuer montre dans ce travail non seulement que le passage d'un arbre à une forêt permet de réduire la variance d'un facteur de 3/4 mais aussi que cette classe simplifiée d'arbres et de forêts aléatoires atteint la vitesse minimax de convergence sur la classe des fonctions Lipschitziennes. Plus récemment, citons le travail remarquable de Scornet et al. (2015) qui montre la consistance d'une version des forêts aléatoires extrêmement proche de celle introduite par Léo Breiman pour un modèle de régression additif. Mentch and Hooker (2016) et Wager (2014) ont quant à eux étudié la normalité asymptotique des prédictions des forêts aléatoires et ont proposé des intervalles de confiance pour ces prédictions. Citons enfin le travail très récent de Mourtada et al. (2020) sur une version de forêts aléatoires pour l'apprentissage *online* (Anderson (2008)) et les forêts aléatoires de Mondrian introduites par Lakshminarayanan et al. (2014).

Paquet `randomForest`

La méthode RF-RI à été implémentée en Fortran par Breiman and Cutler (2005) puis à été importée sur le logiciel de programmation R par Liaw and Wiener (2002) dans le paquet `randomForest`. Il existe principalement, dans la fonction principale, deux paramètres qu'il revient à l'utilisateur de fixer voire optimiser : `ntree` qui est le nombre d'arbres composant la forêt et `mtry` qui est le nombre de variables tirées aléatoirement et mises en compétition à chaque découpage de noeud. Ces paramètres sont par défaut fixés de la manière suivante :

— **En régression** : `ntree`=500 et `mtry`=$p/3$

— **En classification** : `ntree`=500 et `mtry`=$\sqrt{p}$

Dans la pratique, `mtry` est le paramètre qui a le plus d'impact sur les performances de la forêt, Genuer et al. (2008) ont montré que les performances prédictives de la forêt pouvaient parfois être identiques à celles du *bagging* avec les paramètres par défauts. Par ailleurs, ils ont démontré l'importance d'optimiser correctement le paramètre du

`mtry`, en particulier dans le cadre de la grande dimension. Il ont aussi montré que bien souvent, un nombre d'arbres moins grand `ntree` mène à des capacités prédictives aussi bonnes qu'avec le paramètre par défaut `ntree`=500. Nous dirigeons le lecteur vers le récent livre de Genuer and Poggi (2019) qui constitue une excellente introduction à la pratique de forêts aléatoires sur R.

Pour une présentation extrêmement claire des forêts aléatoires et des problématiques associées nous renvoyons aux excellents Biau and Scornet (2016) (et les commentaires Arlot and Genuer (2016)) et Genuer and Poggi (2017).
Nous présentons maintenant deux variantes d'arbre et de forêts aléatoires qui seront rencontrées au Chapitre 3 de cette thèse : les Extra-trees puis les arbres et forêts purement aléatoires.

1.3.3 Extra-trees

Geurts et al. (2006) ont introduit la méthode Extra-trees, pour *extremly randomized trees*, une variante des arbres aléatoires utilisés dans la construction des forêts aléatoires RF-RI. A chaque découpage d'un noeud, m variables de découpe sont tirées. On tire ensuite aléatoirement le seuil de découpe pour chacune des m variables tirées au préalable. Dans la pratique, pour une variable explicative continue, cela revient à tirer le seuil de coupure uniformément entre la plus petite et la plus grande des valeurs contenues dans le noeud pour la variable de découpe considérée. Pour une variable discrète, cela revient à tirer une partition de deux sous-ensembles des classes présentes dans le noeud pour cette variable. On choisit ensuite le découpage qui minimise la réduction d'impureté sur la variable de sortie. Le principal avantage à ce type de construction est évidemment la faible complexité calculatoire de la méthode. En particulier, il n'y a pas d'optimisation du seuil de découpe pour les m variables tirées. En plus d'être facilement implémentable et d'avoir une complexité plus faible que la méthode de référence RF-RI, cette méthode obtient d'excellents résultats en prédiction, et parfois même de meilleurs que ceux obtenus pas la méthode RF-RI.
Enfin, notons qu'il est tout à fait possible de tirer, pour chacune des m variables, plu-

sieurs seuils de découpages possibles et de sélectionner le meilleur. C'est cette version que nous utiliserons dans le Chapitre 3.

1.3.4 Arbres et forêt purement uniformément aléatoires

Les méthodes des arbres et forêts purement uniformément aléatoires, notées respectivement PURT (pour ***P**urely **U**niformly **R**andom **T**rees*) et PURF (pour ***P**urely **U**niformly **R**andom **F**orest*) ont été introduites par Genuer (2012). L'idée générale est de construire les prédicteurs en n'utilisant pas l'échantillon d'apprentissage $\mathcal{L}_n$. Pour chaque noeud, la variable ainsi que le seuil de découpage sont tirés aléatoirement. En dimension un, lorsque $p = 1$, construire un arbre PURT à k feuilles revient donc à tirer indépendamment $k - 1$ seuils de découpages.
Cette version d'arbres et de forêt aléatoires a été introduite pour faciliter l'étude des propriétés théoriques des prédicteurs associés. Arlot and Genuer (2014) proposent une analyse théorique complète des propriétés du biais de ces prédicteurs. Ils montrent en particulier que le passage des arbres aux forêts améliore les capacités prédictives.

1.4 Données longitudinales de grande dimension

A partir de problématiques d'analyse de données réelles surgissent fréquemment de nouvelles questions méthodologiques. En santé publique, les données dites longitudinales ainsi que les données de grande dimension sont des sujets d'étude omniprésents, nécessitant perpétuellement la création de nouveaux outils mathématiques adaptés (Thiébaut et al., 2014). Nous détaillerons dans cette section les problématiques spécifiques à ce type de données. Nous décrirons par la suite les deux essais vaccinaux qui ont motivé cette thèse et auxquelles nous consacrerons l'application des méthodes développées ici.

1.4.1 Données de grande dimension

Il y a quelques décennies, la stratégie d'acquisition de données consistait à ne mesurer que quelques variables. On s'appuyait sur les connaissances scientifiques pour restreindre l'ensemble des variables à mesurer. De plus, certaines informations n'étaient pas mesurables. C'est le cas de l'expression génique où il a fallu attendre l'apparition de la technologie des biopuces puis du séquençage pour pouvoir récolter ce genre de données. Ainsi, on se retrouvait dans la grande majorité des cas avec plus d'observations n que de variables p. L'amélioration constante des techniques d'acquisition de données, la diminution globale des coûts ainsi que la digitalisation du monde ont mené à un changement de paradigme. Aujourd'hui, on ne cible plus de petits ensembles de variables, on mesure tout ce qu'il est possible de mesurer. Ce changement de paradigme engendre des bases de données de plusieurs milliers voire centaines de milliers de variables alors que le nombre d'observations reste de quelques dizaines voire quelques centaines. C'est ce que l'on appelle la grande dimension.
Il n'y a pas de définition formelle à la grande dimension. On dit communément que l'on est dans un contexte de grande dimension lorsque $p > n$, c'est-à-dire que le nombre de variables explicatives p dépasse le nombre d'observations n.
Le contexte de la grande dimension est devenu monnaie courante dans certains domaines. C'est le cas en analyse de données génétiques où l'on dispose de plusieurs milliers de gènes pour relativement peu de patients. C'est aussi le cas en imagerie où chaque pixel d'une image code pour trois variables (les niveaux de bleu, de rouge et de vert). Par exemple, une image dans la définition classique *Full HD* de 1920×1080 pixels est caractérisée par plus de 6 millions de variables. Nous renvoyons à l'excellent Donoho et al. (2000) pour plus d'exemples de domaines souvent confrontés à la grande dimension.

Bien qu'il semble que l'augmentation fulgurante du nombre de variables mesurées soit une opportunité pour inclure le plus d'information possible dans nos modèles de prédiction, c'est en fait une malédiction. Ce terme de malédiction de la grande

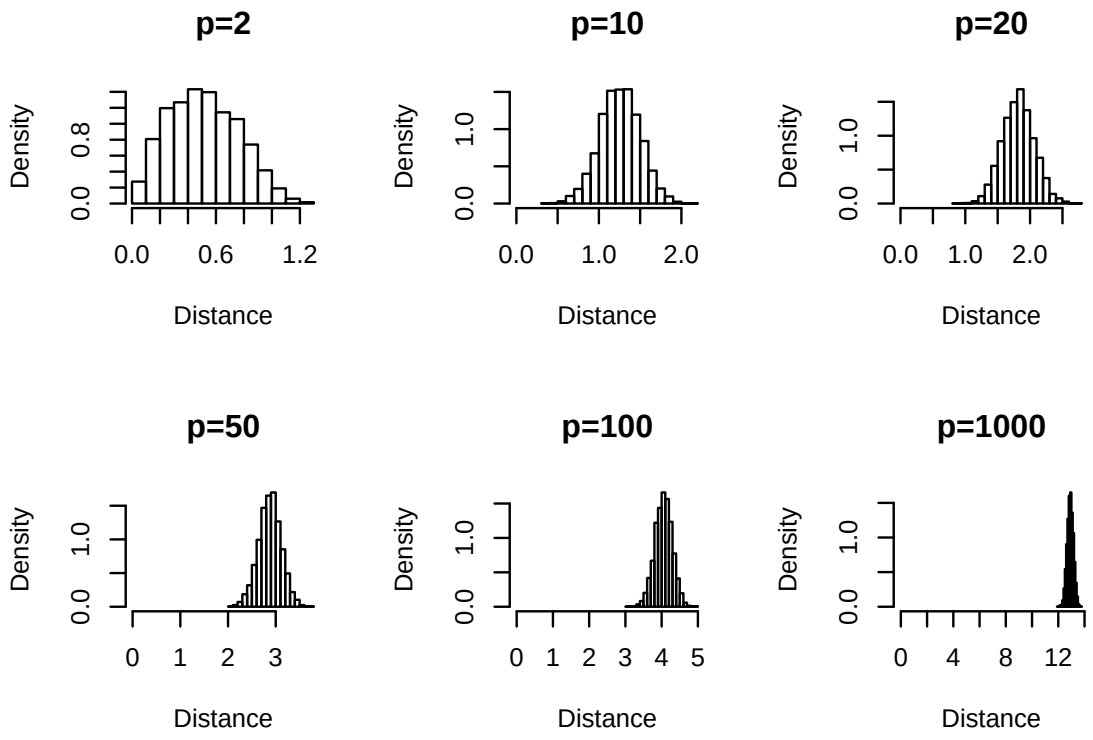

FIGURE 1-4 – Histogrammes de la distance ℓ_2 entre $n = 100$ points tirés uniformément dans l'hypercube $[0, 1]^p$ pour $p = 2,\ 10,\ 20,\ 50,\ 100$ et 1000

dimension a été introduit par Bellman (2015). Cela renvoie à tous les problèmes qui apparaissent lorsque l'on passe en grande dimension. Un grand nombre de méthodes statistiques développées dans le cas classique $p < n$ échouent quand il s'agit de la grande dimension. Par exemple, il n'est pas possible de faire une régression linéaire multiple en grande dimension car l'estimation des paramètres de régression nécessitent d'inverser la matrice des variables explicatives qui n'est pas de plein rang (car $p > n$) et donc qui n'est pas inversible. De plus, toutes les méthodes à moyenne locale échouent en grande dimension dans la mesure où tous les points sont à des distances similaires les unes des autres et qu'il n'y a pas de voisinnage en grande dimension (France et al., 2012). La Figure 1-4 illustre la répartition de la distance ℓ_2 entre $n = 100$ points tirés uniformément dans l'hypercube $[0, 1]^2$. On remarque qu'à mesure que la dimension p grandit, les points s'éloignent les uns des autres tout en étant de plus en plus à même distance les uns des autres, au point qu'ils semblent alors quelque peu perdus dans l'immensité des dimensions s'accumulant. D'où, les méthodes classiques à moyenne locale telles que les k-means ou les estimateurs à noyaux du type Nadaraya-Watson ne sont pas adaptées à la grande dimension (Francois et al., 2005).

On pourrait arguer qu'il suffit de faire augmenter le nombre de points n dans l'hypercube à mesure que le nombre de dimension p augmente, cette solution est illusoire. Afin d'illustrer l'impossibilité de faire croitre le nombre d'observations n suffisamment rapidement, on donne ici un exemple célèbre. On note $V_p(r)$ le volume de la boule de dimension p et de rayon r qui est donné par :

$$V_p(r) = \frac{\pi^{p/2}}{\Gamma\left(p/2+1\right)} r^p \underset{p\to\infty}{\sim} \left(\frac{2\pi e r^2}{p}\right)^{p/2} (p\pi)^{-1/2} \tag{1.36}$$

où Γ est la fonction $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$ pour tout $x > 0$. Soient $X_1, \ldots, X_n$, n variables aléatoires *i.i.d* uniformément distribuées dans l'hypercube $[-0.5, 0.5]^p$. On peut alors montrer que la probabilité qu'au moins un X_i appartienne à la boule $B(0, r)$

centrée en 0 et de rayon r est majorée par :

$$\mathbb{P}\left(\exists i \in \{1, \ldots, n\},\ X_i \in B(0,r)\right) \leq n\mathbb{P}\left(X_1 \in B(0,r)\right) = nV_p(r) \tag{1.37}$$

Cette probabilité tend vers 0 tant que $n = o\left(V_p(r)^{-1}\right)$. Or, le volume $V_p(r)$ tend vers zéro lorsque p augmente, à vitesse supérieure de l'exponentielle. En utilisant alors $V_p(r)^{-1}$ en tant que borne inférieure du nombre d'observations nécessaire pour que la probabilité d'avoir au moins une observation dans le voisinage de zéro (à distance r) soit non nulle, on obtient les résultats suivant : en dimension $p = 30$ nous avons besoin de $n = 45630$ observations, en dimension $p = 50$, nous avons besoin de $n = 5.710^{12}$ observations, en dimension $p = 200$ nous avons besoin de plus d'observations que le nombre estimé de particules dans l'univers.

Au-delà de ces problèmes, d'autres phénomènes étranges apparaissent. Nous en citons quelques uns donnés dans le très bon livre de Giraud (2014), dont nous recommandons vivement la lecture pour une présentation claire et détaillée des problèmes liés à la grande dimension :

- Les densités deviennent plates, par exemple la densité de la Gaussienne multivariée standard atteint son supremum en $(2\pi)^{-\frac{p}{2}}$.
- Les phénomènes rares apparaissent souvent.
- De petites fluctuations sur plusieurs axes mènent à de gigantesques fluctuations globales.
- Dans le cadre des tests, le taux de fausses découvertes explose en grande dimension.

Enfin, Donoho et al. (2000) rappellent que l'optimisation d'une fonction Lipschitzienne de p variables nécessite de l'ordre de $(1\backslash\epsilon)^p$ opérations sur une grille pour obtenir une approximation à ϵ près du minimiseur ou maximiseur de la fonction. Il en est de même pour l'intégration. Ainsi, certaines méthodes très utilisées et efficaces en dimension 1, 2 ou 3 se voient inutilisables en grande dimension du fait de leur trop grande complexité calculatoire. Fort heureusement pour nous, en grande dimension les données ne se répartissent pas de manière uniforme dans $\mathbb{R}^p$. Par exemple, l'inten-

sité des pixels d'une image n'est pas purement aléatoire mais possède une structure géométrique. De plus, dans l'immense majorité des cas, les données se concentrent dans des sous-espaces de $\mathbb{R}^p$ de bien plus petite dimension. Comme expliqué dans la partie précédente, beaucoup de variables sont inutiles et ne participent pas à expliquer la sortie, il convient alors d'extraire les seules variables qui ont un intêret. On verra dans les applications que cela revient en pratique à ne sélectionner que quelques dizaines de variables parmi plusieurs dizaines de milliers, comme chercher une aiguille dans une botte de foin.

La méthode des forêts aléatoires est très performante que ce soit en petite dimension lorsque $p < n$ ou bien en grande dimension lorsque $p > n$. En partant de l'hypothèse que la fonction de régression g définie par (1.11) dépend seulement d'un sous-ensemble de variables utiles, noté $\mathcal{S}$ pour *Strong*, Biau (2012b) montre pour un modèle simplifié des forêts aléatoires, appelé forêt centrée, que sa vitesse de convergence ne dépend que de $|\mathcal{S}|$ et non de p. Ceci en fait une méthode totalement adaptée à la grande dimension. De plus, le score d'importance des variables calculé avec les forêts aléatoires permet de faire cette sélection de variables utiles. Citons deux méthodes de sélection de variables basées sur les arbres CART et forêts aléatoires. Poggi and Tuleau (2006) introduisent une stratégie de sélection de variables basées sur un score d'importance calculé à partir des arbres CART. La méthode VSURF (Genuer et al., 2010), disponible sous forme de paquet R (Genuer et al., 2015) procède à une sélection de variable automatique. Cette méthode procède en deux étapes : la première consiste à ordonner les variables explicatives selon le score d'importance obtenu par forêt aléatoire. La deuxième étape est d'introduire petit-à-petit les variables explicatives (dans l'ordre pré-établi).

1.4.2 Données longitudinales

On parle de données longitudinales lorsque l'on dispose de mesures répétées dans le temps pour chaque sujet. L'interêt principal de mesurer les variables d'un même patient à des temps différents est de pouvoir analyser et mettre en évidence des dy-

namiques dans le phénomène étudié. Dans bien des domaines, la variable que l'on souhaite expliquer varie au cours du temps conjointement aux variables explicatives. Par exemple, le prix d'une maison peut varier dans le temps en fonction de facteurs comme la pression du marché immobilier, l'inflation ou la valorisation du terrain. Ces facteurs évoluent au cours du temps et peuvent être mis en relation avec le prix pour expliquer sa dynamique temporelle. Dans le domaine médical, l'intégration de données longitudinales permet de dégager des marqueurs dont les dynamiques sont susceptibles d'expliquer l'évolution de la maladie. La mise en évidence de trajectoires caractéristiques du développement futur d'une maladie permet en outre une prise en charge des patients parfois même des années avant le développement de cette dernière. Par exemple, Wagner et al. (2018) montrent qu'une glycémie élevée, une pression sanguine faible et une perte de poids sont des facteurs de risques pour le développement futur de maladies cardiovasculaires ainsi que de la démence.

La plupart des méthodes d'apprentissage supposent que toutes les observations sont indépendantes. Or, dans le cadre des données longitudinales, les observations provenant d'un même patient ne sont pas indépendantes, elles possèdent une certaine corrélation qu'il convient de prendre en compte. Le modèle linéaire à effets mixtes introduit dans Laird and Ware (1982) adapte le célèbre modèle de régression linéaire au cadre des données longitudinales. On en donne ici une description non exhaustive. Supposons qu'on dispose de n sujets (ou patients), les variables du patient i sont observées n_i fois au cours du temps. On note Y_{ij} la j-ième observation du patient i au temps t_{ij}. Y_{ij} est modélisée par :

$$Y_{ij} = X_{ij}\beta + Z_{ij}b_i + \varepsilon_{ij} \tag{1.38}$$

où X_{ij} est le $p \times 1$ vecteur des variables explicatives pour la mesure j du sujet i, β est le vecteur des effets fixes, b_i est le $q \times 1$ vecteur des effets aléatoires associés au $1 \times q$ vecteur Z_{ij} contenant des covariables, et ε_{ij} l'erreur de mesure. Pour tout $i = 1, \ldots, n$ les b_i sont indépendantes. Les ε_{ij} sont également indépendantes pour tout $i = 1, \ldots, n;\ j = 1, \ldots, n_i$. De plus les b_i et ε_{ij} sont mutuellement indépendantes. Les

ε_{ij} sont supposés distribuées selon une normale $\mathcal{N}(0, \sigma^2)$, les b_i sont distribuées selon une normale $\mathcal{N}(0, B)$ où B est une matrice définie positive de taille $q \times q$.
Dans le modèle (1.38), il est considéré que l'évolution au cours du temps de la variable réponse Y_i pour le sujet i varie autour d'un comportement moyen $X_i\beta$. Ce comportement moyen est commun à tous les sujets, il est supposé être linéaire en les variables qui composent la matrice X_i. Les variations individuelles autour de ce comportement moyen sont caractérisées par le vecteur des effets aléatoires b_i qui est spécifique à chaque individu. Enfin, le terme ε_{ij} représente l'erreur de mesure.
Le modèle (1.38) ne prend pas explicitement en compte les corrélations serielles qui peuvent exister entre les différentes mesures issues d'un même individu. Afin de prendre explicitement en compte ces corrélations, Diggle and Hutchinson (1989) rajoutent un processus stochastic autorégressif ω_i. Ce modèle est communément appelé le modèle linéaire stochastique à effets mixtes :

$$Y_{ij} = X_{ij}\beta + Z_{ij}b_i + \omega_i\left(t_{ij}\right) + \epsilon_{ij} \tag{1.39}$$

Notons que la fonction de variance-covariance du processus ω_i caractérise la structure des corrélations sérielles des différentes mesures d'un individu. Bien qu'initialement ce processus soit autorégressif, il est tout à fait possible de considérer des processus plus généraux comme le mouvement Brownien fractionnaire. C'est ce que nous ferons dans le Chapitre 2.
Il n'existe pas de solution analytique pour estimer les différents paramètres en même temps dans les modèles (1.38) et (1.39), notamment parce que les paramètres de la variance sont inconnus et sont donc aussi à estimer. Pour ce faire, on utilise l'algorithme itératif ***Expectation-Maximization*** (noté **EM**) introduit dans Dempster et al. (1977). L'idée générale de l'algorithme est d'alterner entre une étape **E** et une étape **M** jusqu'à convergence d'un critère donné. L'étape **E** consiste en l'estimation du vecteur des effets fixes β, des effets aléatoires b_i ainsi que du processus stochastique ω_i pour les paramètres de la variance fixés et donnés par ceux estimés à l'itération précédente. L'étape **M** consiste à considérer les effets fixes β, les effets aléatoires b_i ainsi

que le processus stochastique ω_i comme connus et donnés par ceux estimés à l'étape **E** puis d'estimer les paramètres de la variance, soit la matrice de variance covariance B des effets aléatoires, les paramètres caractéristiques de la fonction de covariance du processus stochastique ainsi que la variance résiduelle σ^2. Pour plus de détails sur l'estimation des différents paramètres, nous renvoyons à l'excellent livre "Modèles biostatistiques pour l'épidémiologie" (Commenges and Jacqmin-Gadda, 2015).
Les modèles (1.38) et (1.39) bien que très utilisés en biostatistiques (Koerner and Zhang (2017), Verbeke and Molenberghs (2009), Zhang and Davidian (2001)) échouent complètement lorsque l'on passe dans le cadre de la grande dimension, c'est-à-dire lorsque le nombre d'effets fixes p dépasse largement le nombre total d'observations $\sum_{i=1}^{n} n_i$. C'est le cas des deux essais vaccinaux considérés dans cette thèse, l'essai DALIA-I et l'essai LIGHT.

1.4.3 Essai vaccinal DALIA-I

DALIA-I est un essai vaccinal de phase 1 composé de 19 patients atteints du VIH. L'un des objectifs de l'essai était d'évaluer la réponse immunitaire au vaccin. Lors de la première période, les 19 patients ont tous reçu le vaccin thérapeutique sous forme de quatre injections à la semaine 0, 4, 8 et 12, tout en étant sous traitement antirétroviral. Après la période de vaccination s'en suit une interruption du traitement antirétroviral (ATI pour Analytical Treatment Interruption) à la semaine 24. Les patients ont été suivis jusqu'à la semaine 48. Nous disposons de 14 temps de mesure, dont 5 en pré-ATI et 9 en post-ATI. En pré-ATI, 32 979 transcrits sont mesurés par des biopuces (microarray Illumina) et en post-ATI, la concentration plasmatique d'ARN du VIH (qui est log-transformée) est également mesurée (voir figure 3-12 Chapitre 3). On renvoie à Lévy et al. (2014) et Hejblum et al. (2015) pour une description complète du jeu de données. Le but est de mettre en relation l'évolution des transcrits pendant la phase de traitement avec l'évolution de la charge virale après l'interruption du traitement. L'intérêt ici est double, on veut à la fois sélectionner les gènes qui expliquent au mieux les variations de la charge plasmatique mais aussi être capable, pendant la phase de traitement, de prédire l'évolution future de cette charge à partir du compor-

tement des gènes sélectionnés. C'est un problème d'apprentissage crucial. En effet, y apporter une réponse satisfaisante permettrait de mettre en place des dispositifs de médecine personnalisée. On pourrait ajuster le schéma de vaccination d'un patient en fonction de la dynamique de certains gènes en réponse au vaccin. Cependant, ce problème se révèle être extrêmement complexe. En effet, les temps de mesures ainsi que le nombre de mesures diffèrent entre la phase de traitement et la phase après l'interruption du traitement. Ainsi l'immense majorité des méthodes échouent car il est impossible d'apparier chaque temps d'observation pendant la phase de traitement à une mesure après l'arrêt du traitement. Pour éviter cette problématique, certaines stratégies visent à compacter l'information en un seul point, en prenant par exemple la différence entre le premier temps de mesure et le dernier. Par exemple, Thiébaut et al. (2019) ont proposé une analyse en ne prenant que les expressions des gènes à un temps de mesure pendant la phase de traitement et l'ont mise en relation avec le maximum de la charge virale après l'interruption du traitement. En faisant cela, il est possible d'utiliser les méthodes classiques d'apprentissage, mais cela se fait au prix d'une perte colossale d'information : la dynamique des gènes et celle de la charge virale. De plus, le plan d'expérience est déséquilibré : les différents patients ont des trajectoires de tailles différentes ce qui fait que certains patients ont leurs variables mesurées à des temps où d'autres ne le sont pas. Ces différents problèmes techniques font de l'analyse de cet essai un véritable défi qu'il convient de relever en développant une stratégie qui utilisera toute l'information que le jeu de données contient.

1.4.4 Essai vaccinal LIGHT

LIGHT est un essai vaccinal composé de 97 patients atteints du VIH. Pour chaque patient, on dispose de 1 à 4 temps d'observations, auxquels sont mesurés plus de 17000 expressions de gènes par séquençage ainsi que la proportion de cellules T CD4+ obtenue par cytométrie en flux. Chez les patients atteints du VIH, les cellules T CD4+ sont les principales cibles du virus, ce qui entraîne lors de sa progression, une diminution de la proportion de ces cellules. L'objectif est de prédire la proportion de cellules T CD4+ à partir de l'évolution de l'expression des gènes afin d'anticiper l'évolution

du virus et donc de mettre en place des stratégies personalisées de traitement. Le deuxième objectif à l'analyse de cet essai est de sélectionner les gènes dont l'évolution dans le temps explique le mieux possible la proportition de cellules T CD4+. La principale difficulté pour l'analyse des données de cet essai vient de la très grande variabilité du nombre de mesures par patient (entre 1 et 4). Notons que jusqu'à présent, aucune méthode d'apprentissage n'a permis de retirer de l'information pertinente de ce jeu de données.

1.4.5 Forêts aléatoires pour données longitudinales

Les données issues des essais vaccinaux DALIA-I et LIGHT correspondent au cadre de la grande dimension. Dans ce contexte, la plupart des méthodes d'apprentissage en biostatistiques échouent, c'est par exemple le cas pour le modèle linéaire à effets mixtes (1.38). La méthode des forêts aléatoires présente le bon goût, en plus d'avoir d'excellentes performances, d'être parfaitement adaptée à la grande dimension (Caruana et al., 2008). Cependant, cette dernière traite indépendamment toutes les observations. Or, les observations provenant d'un même individu ne sont évidemment pas indépendantes les unes des autres. L'objectif de cette thèse est d'adapter la méthode des forêts aléatoires RF-RI au cadre des données longitudinales. On souhaite que les adaptations proposées préservent les avantages des forêts aléatoires tout en prenant en compte les éventuelles corrélations des mesures provenant des mêmes individus. De plus, les méthodes développées devront s'accomoder le plus possible aux difficultées souvent rencontrées dans le cadre des données longitudinales : des temps d'observations différents pour les variables explicatives et la variable de sortie (comme pour DALIA-I par exemple), des temps d'observations différents pour les patients, pouvoir traiter des données manquantes ainsi que de traiter des plans d'expérience déséquilibrés (comme pour LIGHT).

1.5 Apports de la thèse

Dans cette section, nous introduisons les différents apports de la thèse. Pour adapter la méthode des forêts aléatoires aux données longitudinales, nous avons développé deux approches radicalement différentes. La première utilise les modèles mixtes classiquement employés en biostatistiques. La deuxième consiste à construire des prédicteurs par arbres sur des espaces métriques en adaptant les éléments caractéristiques de ces derniers : critère de découpage, la stratégie de prédiction dans les feuilles et le critère d'arrêt.

1.5.1 L'approche par modèles mixtes

Étant donnée la problématique d'analyse de données longitudinales décrite précédemment, il paraît naturel d'utiliser une approche par modèles mixtes. Les modèles linéaires à effets mixtes (1.38) et (1.39) sont extrêmement populaires en biostatistiques. Cependant, ces modèles possèdent deux inconvénient majeurs. Le premier est que le comportement moyen est supposé être linéaire en les variables explicatives, ce qui est plutôt restrictif. Le deuxième problème est d'ordre technique. En grande dimension, lorsque le nombre d'effets fixes p dépasse largement le nombre d'observations n, ces modèles ne sont pas utilisables. Afin de régler ces problèmes, nous avons considéré une généralisation de ce modèle appelé modèle semi-paramétrique à effets-mixtes stochastique.

Considérons des données longitudinales composées de n individus, le $i$ème patient ayant n_i temps d'observation. On note Y_{ij} (pour tous les $i = 1, \ldots, n$ et $j = 1, \ldots, n_i$), la réponse du $i$ème individu au temps d'observation t_{ij}. On modélise Y_{ij} par :

$$Y_{ij} = f(X_{ij}) + Z_{ij}b_i + \omega_i(t_{ij}) + \varepsilon_{ij} \tag{1.40}$$

où X_{ij} est le $p \times 1$ vecteur des variables explicatives, $f : \mathbb{R}^p \longrightarrow \mathbb{R}$ est une fonction inconnue, b_i est le $q \times 1$ vecteur d'effets aléatoires associés au $1 \times q$ vecteur des covariables Z_{ij}, $\omega_i(t)$ est un processus stochastique et ε_{ij} désigne un erreur de mesure.

Dans le modèle (1.40) la fonction inconnue f modélise le comportement moyen de la variable réponse. Cette fonction est commune à tous les individus. Les variations individuelles de Y autour de cette fonction de comportement moyen sont modélisées par les effets aléatoires qui sont spécifiques à chaque individu. Le processus stochastique ω_i modélise les corrélations serielles des mesures entre elles. Enfin, ϵ_{ij} représente l'erreur de mesure commise pour la $j$ième observation de l'individu i.

Dans cette première approche présentée Chapitre 2, on se propose d'estimer le comportement moyen f par la méthode des forêts aléatoires. De cette manière, on utilise la grande flexibilité des forêts aléatoires pour approcher des comportements complexes, qui sont potentiellement non linéaires en les effets fixes, ainsi que sa capacité à être particulièrement adaptée au contexte de la grande dimension. A l'instar des modèles (1.38) et (1.39), il n'existe pas de solution analytique pour estimer des différents paramètres du modèle semi-paramétrique à effets-mixtes (1.40). On utilise ici une variante de l'algorithme EM introduite dans Hajjem et al. (2014). Afin de prendre en compte la structure de corrélation des mesures d'un même individu, nous utilisons une procédure introduite par Sela and Simonoff (2012) de réallocation des valeurs des feuilles des arbres qui composent la forêt. Enfin, l'utilisation des forêts aléatoires pour estimer f permet aussi de récupérer l'importance des variables pour chacun des effets fixes, et donc, de sélectionner les variables qui expliquent au mieux les variations du comportement moyen. Nous introduisons un score sur l'importance des variables qui mesure la stabilité du classement des variables selon leur importance. Ce critère peut être utilisé pour optimiser les paramètres de la forêt.

La méthode développée a fait l'object d'un article publié dans *Statistical Methods in Medical Research* (doi : `https://doi.org/10.1177/0962280220946080`) et a été intégralement implémentée dans un paquet R appelé `LongituRF` disponible sur le CRAN ainsi que sur github : `https://github.com/Lcapitaine/LongituRF`

1.5.2 L'approche métrique

La deuxième approche proposée dans cette thèse diffère fondamentalement de la première. Il n'est ici pas question d'estimer une partie d'un modèle avec une forêt

aléatoire mais bien d'adapter toute la construction de cette dernière au cadre de la régression sur des espaces métriques, et donc par extension, sur des espaces de courbes discrètes. Le point de départ de cette deuxième méthode débute à la lecture de ces quelques lignes de Maurice Fréchet dans Fréchet (1948) *"Le Calcul des probabilités a été implicitement ou explicitement, jusqu'à une époque récente, l'étude des nombres aléatoires et des points aléatoires dans un espace à une, deux ou trois dimensions (probabilités géométriques). Depuis peu, on a souvent cherché à étendre les résultats obtenus aux séries aléatoires, aux vecteurs aléatoires et aux fonctions numériques aléatoires de variables numériques certaines. Mais la nature, la science et la technique offrent de nombreux exemples d'éléments aléatoires qui ne sont, ni des nombres, ni des séries, ni des vecteurs, ni des fonctions. Telles sont par exemple, la forme d'un fil jeté au hasard sur une table, la forme d'un oeuf pris au hasard dans un panier d'oeufs. On a ainsi une courbe aléatoire, une surface aléatoire. On peut aussi considérer d'autres éléments mathématiques aléatoires : par exemple des transformations aléatoires de courbe en courbe. On peut aussi rencontrer des éléments qui n'ont pas jusqu'ici été décrits mathématiquement. On étudiait les nombres aléatoires obtenus en choisissant au hasard une ville dans un pays donné et en notant un nombre relatif à cette ville, comme sa population, le nombre de ses maisons, etc. Mais on peut aussi considérer des éléments attachés à une ville choisie au hasard et qui ne peuvent se décrire par l'intermédiaire d'une des notions mathématiques usuelles : nombre, fonction, courbe, etc. Par exemple, la moralité de sa population, son état d'esprit politique, l'impression de beauté qu'elle donne, etc. C'est une catégorie nouvelle d'éléments aléatoires. Sans aller jusque là, il paraît certain que l'urbanisme conduira à étudier des éléments aléatoires tels que la forme d'une ville prise au hasard, vérifier ainsi par exemple, d'une manière scientifique l'hypothèse de la tendance au développement des villes vers l'Ouest, etc.*

Nous voyons ainsi que si l'on s'est occupé surtout jusqu'ici en Calcul des Probabilités et en Statistique Mathématique, des nombres aléatoires, ce n'est pas parce que l'étude d'éléments aléatoires de natures très diverses, ne s'impose pas aussi bien dans les applications. C'est surtout parce que, comme dans toutes les sciences, on s'est attaché

d'abord à ce qui était plus simple ou plus facile."

Pour reprendre les termes de Maurice Fréchet, on ne s'imaginerait pas sommer la forme de deux oeufs, multiplier la moralité de deux populations ou la forme de deux villes. Ces opérations n'auraient pas de véritable sens. On s'imagine bien, par contre, quantifier la proximité entre deux formes d'oeufs, déterminer qu'une moralité est plus proche qu'une autre, dire que Paris ressemble plus à Londres qu'à Tokyo. La notion commune à tous ces exemples est la notion de distance, de quantification de la proximité entre deux éléments de même nature. Dans cette thèse, il n'est pas question d'oeufs ou de villes mais de trajectoires et plus précisement de leurs formes. Si les trajectoires des gènes d'un patient caractérisent par leurs formes, la tendance du phénomène étudié, il convient alors de les replacer dans le cadre général des espaces métriques, où seule la notion de distance est définie. Pour cette deuxième approche, tous les espaces considérés sont des espaces métriques *i.e.* des espaces seulement équipés de la notion de distance.

Le cadre d'apprentissage de cette deuxième méthode est le suivant. Nous considérons un échantillon d'apprentissage $\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, X_n)\}$ composé d'observations *i.i.d.* de même loi qu'un couple d'objets aléatoires $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, où $\mathcal{X}$ est un produit d'espaces métriques p (qui peuvent être non ordonnés) $(\mathcal{X}_1, d_1) \times \cdots \times (\mathcal{X}_p, d_p)$, et où $\mathcal{Y}$ est aussi un espace métrique muni de la distance $d_{\mathcal{Y}}$. Dans ce cadre, l'espace $(\mathcal{X}_i, d_i)$ code pour la $i$ème variable explicative. L'objectif principal est d'adapter la méthode des forêts aléatoires pour apprendre la relation entre X et Y.

L'élément central dans la construction d'une forêt aléatoire est l'arbre CART. Comme mentionné dans la Section 1.3.1, un arbre CART se caractérise par trois notions essentielles :

— le découpage

— la règle d'arrêt

— la stratégie de prédiction dans les feuilles

La notion de découpage des arbres CART classiques définie par l'équation (1.22) n'a plus de sens dans des espaces métriques quelconques. En effet, ces espaces ne sont

pas nécessairement ordonnés. Il se peut donc qu'on ne puisse pas déterminer si un élément est plus grand qu'un certain seuil s qui est, lui aussi, un élément de l'espace métrique considéré. En effet, pour reprendre un des exemples de Maurice Fréchet, il n'est a priori pas naturel de dire qu'une forme de fil est supérieure à une autre forme de fil. Nous adaptons alors le principe de découpage aux espaces métriques généraux en utilisant seulement la notion de distance. L'idée générale de nos découpages est que si la variable $X^{(j)}$ est fortement reliée à la variable de sortie Y alors des observations proches dans $(\mathcal{X}_j, d_j)$ le seront aussi dans $(\mathcal{Y}, d_{\mathcal{Y}})$ relativement aux autres mesures. Il convient alors de trouver comment séparer les éléments de la variable $X^{(j)}$ en deux groupes les plus homogènes possibles dans $(\mathcal{X}_j, d_j)$, puis de sélectionner le meilleur découpage parmi toutes les variables. A l'instar des arbres CART, le critère de découpage choisi est la maximisation de la réduction de la variance, à la différence que nous adoptons la variance de Fréchet sur la variable réponse. La moyenne et la variance de Fréchet introduites dans Fréchet (1906) sont les généralisations respectives des notions de moyenne et de variance dans des espaces métriques généraux. Les noms des deux méthodes, les arbres de Fréchet et les forêts aléatoires de Fréchet, proviennent de ces deux notions. La règle d'arrêt utilisée est la construction des arbres jusqu'à obtenir des arbres maximaux *i.e.* des arbres dont les feuilles ont une variance de Fréchet nulle. De plus, on étend la procédure d'élagage au nouveau type d'arbre introduit. On utilise ensuite naturellement la notion de moyenne de Fréchet pour prédire les valeurs des différentes feuilles. De plus, nous donnons un théorème de consistance pour les prédicteurs par régressogrammes construits à partir de méthodes de partitionnement sur $\mathbb{R}^p$ et à valeurs dans un espace métrique de diamètre fini. Finalement, nous appliquons ce résultat au cadre des arbres purement aléatoires.

Par suite, nous introduisons la méthode de régression par forêt aléatoire de Fréchet, qui est une agrégation d'arbres de Fréchet obtenue en utilisant la moyenne de Fréchet comme fonction d'agrégation. En nous inspirant des Extra-trees introduits par Geurts et al. (2006) (voir Section 1.3.3) nous proposons une version "extrêmement randomisée" de nos arbres et forêts aléatoires de Fréchet qui s'avère être plus

robuste, plus rapide et utilisable tant que la notion de distance existe.

Ces développements mathématiques dépassent largement le cadre des données longitudinales et permettent notamment de mettre en relation des objets de nature très différente comme des images, des formes, des scalaires, des facteurs et des courbes. Nous présentons quelques analyses de simulations sur de tels objets au Chapitre 3.

Cees nouvelles méthodes ont fait l'objet d'un article soumis et ont été intégralement implémentées dans un paquet R disponible sur github : https://github.com/Lcapitaine/FrechForest

Chapitre 2

Random forests for high-dimensional longitudinal data

Capitaine Louis[1], **Genuer Robin**[1], **Thiébaut Rodolphe**[1]

[1] INSERM U1219 Bordeaux Population Health Research Center, INRIA Bordeaux Sud-Ouest, SISTM Team, Bordeaux University, Bordeaux, France

Accepted in *Statistical Methods in Medical Research*

DOI : https://doi:10.1177/0962280220946080

Abstract

Random forests are one of the state-of-the-art supervised machine learning method and achieve good performance in high-dimensional settings where p, the number of predictors, is much larger than n, the number of observations. Repeated measurements provide, in general, additional information, hence they are worth accounted especially when analyzing high-dimensional data. Tree-based methods have already been adapted to clustered and longitudinal data by using a semi-parametric mixed effects model, in which the non-parametric part is estimated using regression trees or random forests. We propose a general approach of random forests for high-dimensional longitudinal data. It includes a flexible stochastic model which allows the covariance structure to vary over time. Furthermore, we introduce a new method which takes intra-individual covariance into consideration to build random forests. Through simulation experiments, we then study the behavior of different estimation methods, especially in the context of high-dimensional data. Finally, the proposed method has been applied to an HIV vaccine trial including 17 HIV infected patients with 10 repeated measurements of 20000 gene transcripts and blood concentration of human

immunodeficiency virus RNA. The approach selected 21 gene transcripts for which the association with HIV viral load was fully relevant and consistent with results observed during primary infection.

2.1 Introduction

Random forests (RF henceforth), introduced by Breiman (2001), are one of the state-of-the-art machine learning method (Fernández-Delgado et al., 2014). In several domains, RF achieve good prediction performance for high-dimensional data, where the number of predictors p is much larger than the number of observations n (e.g., Cutler et al. (2007) ; Chen and Ishwaran (2012)). On the other hand, theoretical results have also been recently obtained for RF. Scornet et al. (2015) proved a consistency result for RF in the context of additive regression models. Mentch and Hooker (2016) and Wager (2014) studied asymptotic normality of RF predictions and proposed confidence intervals for those predictions. We refer to Biau and Scornet (2016) for further reading on that matter.

When the number of predictors p is much larger than the number of observations n, *i.e.* $p >> n$, application of RF must be done with care since RF parameters, specifically the number of variables randomly picked at each node of a tree, must be carefully tuned to optimize their prediction performance (Genuer et al. (2008)). Some recent improvements have been suggested to deal especially with high-dimensional data. Zhu et al. (2015) introduced ideas from reinforcement learning within the tree-based model framework, in order to focus more efficiently on relevant variables during the tree building. Linero (2018) developed Bayesian regression trees (Chipman et al., 1998) with sparsity, by using appropriate priors.

The case, where repeated measurements are available, we focus on, is quite specific. Indeed, with longitudinal data, within unit variations bring information in addition to the between units variations. Even though an outcome does not change over time, getting repeated observations increases information about the link between predictors and the outcome. Compared to survival (i.e. censored) data, for which there is a large bulk of work (see Hothorn et al. (2005) ; Ishwaran et al. (2008), Ishwaran et al. (2010) ;

Steingrimsson et al. (2019), among others), less has been done to adapt random forest approaches to repeated measurements or clustered data. The analysis of longitudinal data requires to take into account the specific correlation structure, as in mixed effects models (Laird and Ware (1982) ; Verbeke and Molenberghs (2009)). Concerning tree-based methods, some approaches proposed to adapt the splitting nodes criterion to longitudinal data : Segal (1992) adapted a multivariate approach to split nodes, Eo and Cho (2014) used polynomial mixed effects models inside each node, while more recently, in the framework of clinical trials, Wei et al. (2020) combine mixed effects models with regression splines and use a likelihood ratio test at each node. On the other hand, Hajjem et al. (2011) and Sela and Simonoff (2012) independently introduced tree-based methods using a semi-parametric mixed effects model, in which the non-parametric part is estimated using regression trees. Fu and Simonoff (2015) studied an alternative method which uses conditional inference trees (Hothorn et al., 2006) instead of CART tress (Breiman et al., 1984), while Hajjem et al. (2014) have extended their methodology with the use of RF instead of regression trees. Their common estimation procedure is based on an Expectation Maximization (EM) algorithm (McLachlan and Krishnan, 1997), which iterates between estimation of the fixed part (with a regression tree or a RF) and estimation of the random part parameters. The work of (Hajjem et al., 2011), Hajjem et al. (2014)) focused on clustered data only. The correlation structures considered were much simpler than what is requested for longitudinal data. Recently, Kundu and Harezlak (2019) also proposed to use mixed effects model on different clusters corresponding to the leaves of a regression tree applied on baseline data only, and Calhoun et al. (2020) developed random forests that handle repeated measurements for classification problems.

Finally, all those previous works considered standard data, where n was always larger (and often much larger) than p. Hence, the potential gain due to repeated measurements and the behavior of the approaches in an high-dimensional context were not studied, although applications in such context are skyrocketing.

In this work, we propose a general approach of random forest method for high-dimensional longitudinal data. First, we develop a flexible stochastic semi-parametric

mixed effects model and introduce a new RF method for longitudinal data. We compare all available tree-based methods for longitudinal data in an extensive simulation study especially in the context of high-dimensional data. Finally, the proposed method has been applied to real data from a therapeutic vaccine trial in HIV-infected patients.

All existing and proposed methods have been implemented together in an R (R Core Team, 2019) package called `longituRF` [1].

2.2 The semi-parametric stochastic mixed effects model

Let us consider longitudinal data with n individuals, the ith individual having n_i observations over time. Suppose Y_{ij} (for all $i = 1, \ldots, n$ and $j = 1, \ldots, n_i$), the response of the ith individual at time t_{ij}, satisfies

$$Y_{ij} = f(X_{ij}) + Z_{ij} b_i + \omega_i(t_{ij}) + \varepsilon_{ij} \tag{2.1}$$

where X_{ij} is the $p \times 1$ vector of covariates, $f : \mathbb{R}^p \longrightarrow \mathbb{R}$ is the unknown mean behavior function, b_i is a $q \times 1$ vector of random effects associated with a $1 \times q$ vector of covariates Z_{ij}, $\omega_i(t)$ is a stochastic process used to model serial correlation and ε_{ij} denotes a measurement error.

For all $i = 1, \ldots, n$ the b_i are independent, as well as the $\omega_i(t)$. And the ε_{ij} are also independent for all $i = 1, \ldots, n;\ j = 1, \ldots, n_i$. We assume that $b_i, \omega_i(t)$ and ε_{ij} are mutually independent. We also suppose that the ε_{ij} are normally distributed as $\mathcal{N}(0, \sigma^2)$, the b_i are normally distributed as $\mathcal{N}(0, B)$ where B is a $q \times q$ positive definite matrix and $\omega_i(t)$ is a centered Gaussian process with covariance function $Cov\left(\omega_i(t), \omega_i(s)\right) = \gamma^2 \Gamma(s, t)$ depending on a parameter γ^2. More precisely, we denote the covariance matrix of the stochastic process ω_i for the ith individual by $\left(\gamma^2 \Gamma\left(t_{ij}, t_{ik}\right)\right)_{1 \leq j,k \leq n_i} = \gamma^2 K_i$ where K_i is a positive definite matrix. It should be noted

1. Available at https://github.com/Lcapitaine/longituRF

that function Γ, which depends only on the measurement times, fully determines the covariance structure of the stochastic process. For example, in the case of a standard Brownian motion, Γ is defined by $\Gamma(s,t) = \min(s,t)$. The parameter γ^2 tunes the variability of the stochastic process. We will also consider the case where Γ depends on a additional parameter α in the next section.

We consider in model (2.1) that the evolution of the response variable for the ith individual Y_i over time varies around a mean behavior function given by f. These variations specify the individual trajectories around f and are driven by the random effects b_i and the stochastic process $\omega_i(t)$ for the ith individual. Note that if the function f is assumed linear then model (1) reduces to the linear stochastic model of Diggle and Hutchinson (1989)Diggle and Hutchinson (1989).

(Zhang et al., 1998) already considered a semi-parametric stochastic mixed effects model but with f a function of the time only, hence model (2.1) can be seen as a generalization of their model. Hajjem et al. (2011), Hajjem et al. (2014) considered non-stochastic model, *i.e.* model (2.1) without the stochastic process $\omega_i(t)$, because they only worked with clustered data. The closest approach to ours is the one of Sela and Simonoff (2012), which took into account the serial correlation by the use of autoregressive processes in semi-parametric mixed effects model but that was not extended to random forests.

In the following, we will consider high-dimensional cases where p, the number of variables of f, is much larger than $N = \sum_{i=1}^{n} n_i$ the total number of observations.

2.3 Estimation

We consider the vectorized form of model (2.1) as follows, for all $i = 1, \ldots, n$:

$$Y_i = f_i + Z_i b_i + \omega_i + \varepsilon_i \tag{2.2}$$

where $f_i = (f(X_{i1}), \ldots, f(X_{in_i}))^T$, $Y_i = (Y_{i1}, \ldots, Y_{in_i})^T$, $Z_i = [Z_{i1}, \ldots, Z_{in_i}]^T$, $\omega_i = (\omega_i(t_{i1}), \ldots, \omega(t_{in_i}))^T$ and $\varepsilon_i = (\varepsilon_{i1}, \ldots, \varepsilon_{in_i})^T$.

A common ground among previous works is to use an adaptation of the Maximum Li-

kelihood (ML)-based EM algorithm (as described in Wu and Zhang (2006)), to estimate all quantities (all unknown parameters and the unknown mean behavior function) of model (2.1). The main principle of the estimation procedure is given in Algorithm 1, while further details can be found in the two following sections. Remark that all existing methods apply this estimation procedure, the difference being *i)* in the methodology used to estimate the mean behavior function f at step 1 of Algorithm 1 and *ii)* in the fact that they include or not a stochastic process in the model.

Algorithm 1: General estimation procedure for model (2.1)

initialization : Let $r = 0$, $\widehat{b}_{i,(0)} = 0_q$, $\widehat{\omega}_{i,(0)} = 0_{n_i}$, $\widehat{B}_{(0)} = I_q$, $\widehat{\gamma}^2_{(0)} = 1$ and $\widehat{\sigma}^2_{(0)} = 1$.

repeat

1. Set $r = r + 1$, compute $\widetilde{Y}_{ij,(r-1)} = Y_{ij} - Z_{ij}\widehat{b}_{i,(r-1)} - \widehat{\omega}_{ij,(r-1)}$ estimate f in the standard regression framework (with all N observations) :

$$\widetilde{Y}_{ij,(r-1)} = f\left(X_{ij}\right) + \varepsilon_{ij}$$

   to get $\widehat{f}_{i,(r)}$.
   Then predict $\widehat{b}_{i,(r)}$ and $\widehat{\omega}_{i,(r)}$ using $\widehat{B}_{(r-1)}, \widehat{\gamma}^2_{(r-1)}, \widehat{\sigma}^2_{(r-1)}$ and $\widehat{f}_{i,(r)}$.
2. Update $\widehat{B}_{(r)}$, $\widehat{\gamma}^2_{(r)}$ and $\widehat{\sigma}^2_{(r)}$ using $\widehat{f}_{i,(r)}$, $\widehat{b}_{i,(r)}$ and $\widehat{\omega}_{i,(r)}$,

until *convergence*;

2.3.1 Mean behavior function estimation

At step 1 of Algorithm 1, the mean behavior function f could actually be estimated with any regression method, but in this work we focus on tree-based methods. Hajjem et al. (2011) introduced **MERT** (**M**ixed **E**ffects **R**andom **T**rees), which consists in using a regression tree to estimate f in a model that does not include any stochastic process (because they focus on clustered data). More precisely, they used CART (Classification and Regression Trees, Breiman et al. (1984)), which consists in a binary data-driven recursive partitioning of the explanatory variables space. At each step of the partitioning, a node is split into two child nodes. Hence, the resulting partition can naturally be associated to a binary tree which is called a CART tree. Furthermore, we stress that each node splitting is optimized among all explanatory variables and that the CART algorithm works with two steps : the maximal

tree building followed by a pruning step, which ensure to get the tree-structured predictor with the best prediction performance.

Later, Hajjem et al. (2014) introduced **MERF** (**M**ixed **E**ffects **R**andom **F**orest) in which f is estimated using RF, again without including a stochastic process in the model. RF (Breiman, 2001) are obtained by aggregating a collection of randomized CART trees, where the aggregation consists in averaging individual trees predictions. Each tree is a maximal tree, built using random perturbations : first, it is built on a bootstrap sample of the learning set, and secondly, at each step of the partitioning, the best split is optimized among a randomly drawn subset of explanatory variables. The size of the subset of variables, often called `mtry`, has usually a strong impact on RF performance : if `mtry` is too small, individual trees would give too poor predictions, and if `mtry` is too high, the collection of trees could be not diverse enough (Díaz-Uriarte and Alvarez De Andres (2006) ; Genuer et al. (2008)). RF naturally estimate the prediction error with the Out-Of-Bag (OOB) error as follows : to predict the response of one particular observation of the learning set, only trees built on bootstrap samples not containing this observation are aggregated. Furthermore, OOB samples (made of observations not selected in bootstrap samples) are also used to compute a variable importance (VI) score. For a fixed variable, the VI score of this variable is defined as the mean increase of the error of a tree on its associated OOB sample after a random permutation of this variable values.

Independently, Sela and Simonoff (2012) introduced **REEMtree** (**R**andom **E**ffects **E**xpectation **M**aximization Tree) in a model that includes serial dependencies between observations with the use of an autoregressive process. **REEMtree** uses a CART tree T as a first step in the estimation of f. Once T is built, the associated partition (of the explanatory variables space) is used to fit a linear mixed effects model. More precisely, let Φ^i be the indicator matrix defined by $\Phi^i_{j\ell} = \mathbb{1}_{\{X_{ij} \in g_\ell\}}$ where g_ℓ is the ℓth leaf of tree T and consider the following linear mixed effects model (which we write directly in the framework of model (2.2)) :

$$Y_i = \Phi^i \mu_T + Z_i b_i + \omega_i + \varepsilon_i \ .$$

The vector of the leaves values μ_T is estimated by :

$$\widehat{\mu}_T = \left(\sum_{1 \leq i \leq N} (\Phi^i)^T V_i^{-1} \Phi^i \right)^{-1} \left(\sum_{1 \leq i \leq N} (\Phi^i)^T V_i^{-1} Y_i \right)$$

with $V_i = Var(Y_i) = Z_i B Z_i^T + \gamma^2 K_i + \sigma^2 I_{n_i}$ for all $i = 1, \ldots, N$. The advantage of this method is that the leaves values are updated by taking into account intra-individual covariance matrix V_i (instead of taking the simple mean of values as in **MERT**). Finally, f_i is estimated by $\widehat{f}_i = \Phi^i \widehat{\mu}_T$.

In this article, we propose a novel method, called **REEMforest**, which consists in aggregating a collection of randomized **REEMtrees**. More precisely, consider L randomized trees (as in standard RF) $T_1, \ldots, T_L$. Let $\Phi^{i,\ell}$ be the indicator matrix associated with the ℓth random tree T_ℓ and $\widehat{\mu}_{T_\ell}$ the vector of leaves values of T_ℓ estimated within the stochastic linear mixed effects model :

$$Y_i = \Phi^{i,\ell} \mu_{T_\ell} + Z_i b_i + \omega_i + \varepsilon_i \ .$$

f_i is thus estimated by :

$$\widehat{f}_i = \frac{1}{L} \sum_{\ell=1}^{L} \Phi^{i,\ell} \widehat{\mu}_{T_\ell} \ .$$

All details about **REEMforest** can be found in Algorithm 2.

To sum up, **MERT**, **MERF** and **REEMtree** are the already existing methods and we introduce **REEMforest** method that generalizes all previous methods. Our method is extended to random forest which is an important component, especially in the context of high dimensional data and the stochastic part of the model includes a general Gaussian process (Ornstein-Uhlenbeck process and fractional Brownian motion) in addition to the random effects. In the following, when a stochastic process is indeed included in the model, we add an **S** (for **S**tochastic) at the beginning of the method names, so in this case we denote the methods by **SMERT**, **SMERF**, **SREEMtree** and **SREEMforest** respectively.

2.3.2 Prediction of random effects and stochastic process

Once $\widehat{f}_i$ has been computed (by either of the previously described methods), the predictions for the random effects b_i and the stochastic processes ω_i for fixed parameters $\left(B, \gamma^2, \sigma^2\right)$

Algorithm 2: REEMforest algorithm

initialization : Let $r = 0$, $\widehat{b}_{i,(0)} = 0_q$, $\widehat{\omega}_{i,(0)} = 0_{n_i}$, $\widehat{B}_{(0)} = I_q$, $\widehat{\gamma}^2_{(0)} = 1$ and $\widehat{\sigma}^2_{(0)} = 1$.

repeat

1. Set $r = r + 1$, compute $\widetilde{Y}_{ij,(r-1)} = Y_{ij} - Z_{ij}\widehat{b}_{i,(r-1)} - \widehat{\omega}_{ij,(r-1)}$.
   Estimate f in the standard regression framework (with all N observations) with a RF, build the $\Phi^{i,\ell}$ matrices for every tree T_ℓ in the forest and estimate the leaves values $\widehat{\mu}_{T_\ell}$. Aggregate the updated trees to get

$$\widehat{f}_{i,(r)} = \frac{1}{L}\sum_{\ell=1}^{L} \Phi^{i,\ell}\widehat{\mu}_{T_\ell}.$$

   Then predict $\widehat{b}_{i,(r)}$ and $\widehat{\omega}_{i,(r)}$ for all $i = 1, \ldots, n$

$$\widehat{b}_{i,(r)} = \widehat{B}_{(r-1)} Z_i^T \widehat{V}^{-1}_{i,(r-1)} \left(Y_i - \widehat{f}_{i,(r)}\right)$$
$$\widehat{\omega}_{i,(r)} = \widehat{\gamma}^2_{(r-1)} K_i \widehat{V}_{i,(r-1)} \left(Y_i - \widehat{f}_{i,(r)}\right)$$

2. Update $\widehat{B}_{(r)}$, $\widehat{\gamma}^2_{(r)}$ and $\widehat{\sigma}^2_{(r)}$:

$$\widehat{B}_{(r)} = \frac{1}{n}\sum_{i=1}^{n} \left\{\widehat{b}_{i,(r)}\widehat{b}^T_{i,(r)} + \widehat{B}_{(r-1)} - \widehat{B}_{(r)} Z_i^T \widehat{V}^{-1}_{i,(r-1)} Z_i \widehat{B}_{(r)}\right\}$$
$$\widehat{\gamma}^2_{(r)} = \frac{1}{N}\sum_{i=1}^{n} \{\widehat{\omega}^T_{i,(r)} K_i^{-1} \widehat{\omega}_{i,(r)} + \widehat{\gamma}^2_{(r-1)} \left(n_i - \widehat{\gamma}^2_{(r-1)} tr\left(\widehat{V}^{-1}_{i,(r-1)} K\right)\right)\}$$
$$\widehat{\sigma}^2_{(r)} = \frac{1}{N}\sum_{i=1}^{n} \widehat{\varepsilon}^T_{i,(r)}\widehat{\varepsilon}_{i,(r)}\widehat{\sigma}^2_{(r-1)} tr\left(\widehat{V}^{-1}_{i,(r-1)}\right)$$

   with $\widehat{\varepsilon}_{i,(r)} = Y_i - \widehat{f}_{i,(r)} - Z_i\widehat{b}_{i,(r)} - \widehat{\omega}_{i,(r)}$
   $\forall i = 1, \ldots, n$.

until *convergence*;

are obtained by taking their conditional expectations given the data Y_i. The best linear unbiased predictors **BLUP** are thus :

$$\widehat{b}_i = B Z_i^T V_i^{-1} \left(Y_i - \widehat{f}_i\right)$$
$$\widehat{\omega}_i = \gamma^2 K_i V_i^{-1} \left(Y_i - \widehat{f}_i\right) \quad .$$

This ends step 1 of Algorithm 1.

2.3.3 Variance components estimation

At step 2 of Algorithm 1, the estimation of the variance parameters are obtained by taking the conditional expectation of their maximum likelihood estimators given the data Y_i. Thanks to the conditional independence between the individuals we can write, for fixed f_i, $i = 1, \ldots, n$, the likelihood function associated to model (2.2) as follows :

$$\mathcal{L}(B, \gamma^2, \sigma^2; Y) = \prod_{i=1}^{n} \mathcal{L}_i \left(Y_i; B, \gamma^2, \sigma^2\right)$$

with $\mathcal{L}_i \left(Y_i; B, \gamma^2, \sigma^2\right)$ the density function on the vector Y_i. Moreover, since $Y_i | b_i, \omega_i \sim \mathcal{N}\left(f_i + Z_i b_i + \omega_i, \sigma^2 I_{n_i}\right)$, by using the independence of b_i, ω_i and ε_i we can easily write the likelihood function $\mathcal{L}$ as :

$$\begin{aligned}\mathcal{L}(B, \gamma^2, \sigma^2; Y) = \prod_{i=1}^{n} \frac{1}{(2\pi\sigma^2)^{\frac{n_i}{2}}} \exp\Bigg\{ -\frac{1}{2\sigma^2} \big(Y_i - f_i \\ - Z_i b_i - \omega_i\big) \big(Y_i - f_i - Z_i b_i - \omega_i\big)^T \Bigg\} \times \frac{1}{(2\pi)^{\frac{q}{2}} \sqrt{\det(B)}} \\ \times \exp\{\frac{1}{2} b_i^T B^{-1} b_i\} \times \frac{1}{(2\pi)^{\frac{n_i}{2}} \sqrt{\det\left(\gamma^2 K_i\right)}} \\ \times \exp\{\frac{1}{2} \omega_i^T \left(\gamma^2 K_i\right)^{-1} \omega_i\} \ .\end{aligned}$$

Using that $Y_i - f_i - Z_i b_i - \omega_i = \varepsilon_i$, the maximum likelihood estimators of B, γ^2 and σ^2 are :

$$\widetilde{B} = \frac{1}{n} \sum_{i=1}^{n} b_i b_i^T, \ \widetilde{\gamma}^2 = \frac{1}{N} \sum_{i=1}^{n} \omega_i^T K_i^{-1} \omega_i, \ \widetilde{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{n} \varepsilon_i^T \varepsilon_i$$

Because b_i, ω_i and ε_i are unknown these estimators are not computable, this is why we take the expectation given the data Y_i. The conditional expectations of the estimators $\widetilde{B}$ and $\widetilde{\sigma}^2$ are given in Wu and Zhang (2006) :

$$\widehat{B} = \mathbb{E}(\widetilde{B}|Y) = \frac{1}{n} \sum_{i=1}^{n} \left\{ \widehat{b}_i \widehat{b}_i^T + B - B Z_i^T V_i^{-1} Z_i B \right\}$$

$$\widehat{\sigma}^2 = \mathbb{E}(\widetilde{\sigma}^2|Y) = \frac{1}{N} \sum_{i=1}^{n} \left\{ \widehat{\varepsilon}_i^T \widehat{\varepsilon}_i + \sigma^2 tr(V_i^{-1}) \right\}$$

The conditional expectation of the maximum likelihood estimator $\widetilde{\gamma}^2$ given the data Y_i is

$$\begin{aligned}\widehat{\gamma}^2 = \mathbb{E}(\widetilde{\gamma}^2|Y_i) &= \frac{1}{N}\sum_{i=1}^{n} tr\left(\mathbb{E}(K_i^{-1}\omega_i\omega_i^T|Y_i)\right) \\ &= \frac{1}{N}\sum_{i=1}^{n} tr\left(K_i^{-1}\widehat{\omega}_i\widehat{\omega}_i^T + Cov(K_i^{-1}\omega_i, \omega_i|Y_i)\right) \\ &= \frac{1}{N}\sum_{i=1}^{n}\left(\widehat{\omega}_i^T K_i^{-1}\widehat{\omega}_i + \gamma^2\left(n_i - \gamma^2 tr\left(V_i^{-1}K_i\right)\right)\right)\end{aligned}$$

Estimators of variance parameters B, γ^2 and σ^2 at step 2 are thus given by $\widehat{B}$, $\widehat{\gamma}^2$ and $\widehat{\sigma}^2$ respectively.

Gaussian processes such as Ornstein-Uhlenbeck process and fractional Brownian motion have a variance-covariance function $Cov\left(\omega_i(s), \omega_i(t)\right) = \gamma^2\Gamma\left(s, t; \alpha\right)$ which depends on two parameters γ^2 and α. In this case, we can write the covariance matrix of the stochastic process ω_i for the ith individual as $\left(\gamma^2\Gamma\left(t_{ij}, t_{ik}; \alpha\right)\right)_{1\leq j,k\leq n_i} = \gamma^2 K_i(\alpha)$ with $K_i(\alpha)$ depending on α. There is no analytic maximum likelihood estimator of α. However, for a fixed value of α, the estimation procedure described in this section holds. Thus for $\mathcal{H} = \{\alpha_1, \ldots, \alpha_d\}$ an ensemble of possible values of α parameter, the estimator of α is

$$\widehat{\alpha} = \underset{\alpha\in\mathcal{H}}{\arg\max}\, l\left(B, \gamma^2, \alpha, \sigma^2; y\right)$$

where l is the log-likelihood function.

2.4 Simulation study

2.4.1 Simulation model

Explanatory variables

In this section we detail how the data matrix of the explanatory variables X is simulated. Our choices are motivated by the characteristics of the data coming from our application, which are transcriptomics data in a phase 1/2 vaccine trial (see the application section for more details), called the DALIA trial.

As usual in high dimensional contexts, we assume that a large majority of variables are not associated with the response variable Y (also known as a *sparsity* assumption). In our

study, those variables are simulated as i.i.d. random draws from a multivariate Gaussian distribution $\mathcal{N}(0, 3I_N)$, where I_N denotes the identity matrix of size N (recall that $N = \sum_{i=1}^{n} n_i$ denotes the total number of observations).

Moreover, since we deal with longitudinal data in the context of gene expression, we assume that some explanatory variables vary over time and that some explanatory variables are clustered into groups (which correspond to genes involved in the same biological pathway). Hejblum et al. (2015) highlighted ten examples of groups of genes with different temporal behaviors in the DALIA trial, and we mimic some of these trends by setting the following six behaviors over time in our simulations :

$$\begin{cases} C_{g_1}(t) = 2.44 + 0.04\left(t - \frac{3(t-6)^2}{t}\right) \\ C_{g_2}(t) = 0.5t - 0.1\,(t-5)^2 \\ C_{g_3}(t) = 0.25t - 0.05\,(t-6)^2 \\ C_{g_4}(t) = \cos(\frac{t-1}{3}) \\ C_{g_5}(t) = 0.1t + \sin(0.6t + 1.3) \\ C_{g_6}(t) = -0.03t^2 \end{cases} \tag{2.3}$$

The explanatory variables with a temporal behavior are then simulated as follows :

$$X^{(k)}(t) = C_{g_{(k)}}(t) + \zeta_k + \varepsilon_t$$

where $g_{(k)}$ is the group of the kth covariate $X^{(k)}$; $\zeta_k \sim \mathcal{N}(0, 0.1)$ corresponds to a random translation at the group level and $\varepsilon_t \sim \mathcal{N}(0, 0.2)$ is an additional time-dependent variability. Plots of Figure 2-1 give the explanatory variables trajectories associated to one simulated dataset under model (2.3). In the following, we investigate two situations with different values of the total number of variables, p, as well as different sizes of each group of variables with temporal behavior.

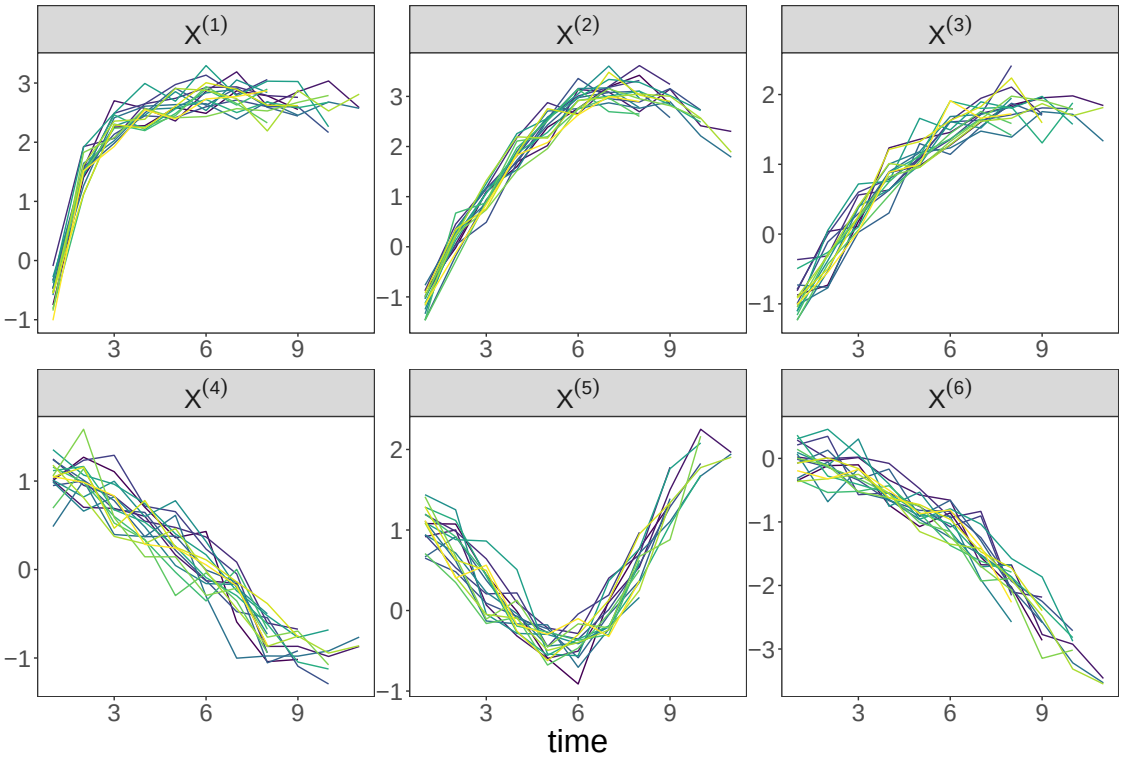

FIGURE 2-1 – Dynamics of explanatory variables (one curve per individual, $n = 17$) simulated under model (2.3) in the low-dimensional case.

Outcome variable

The two following models, which are special cases of model 2.2, are used to simulate the outcome variable Y. For all $i = 1, \ldots, n$:

$$Y_i = f_i + b_{0i} + z_i b_{1i} + \varepsilon_i \qquad \textit{(non-stochastic model)} \tag{2.4}$$

$$Y_i = f_i + b_{0i} + z_i b_{1i} + \omega_i + \varepsilon_i \qquad \textit{(stochastic model)} \tag{2.5}$$

where $(b_{0i}, b_{1i})^T \underset{i.i.d}{\sim} \mathcal{N}(0, B)$ with $B = \begin{pmatrix} 0.5 & 0.6 \\ 0.6 & 3 \end{pmatrix}$, ω_i is a Brownian motion with volatility $\gamma^2 = 0.8$ and $\varepsilon_i \underset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2 I_{n_i})$ with $\sigma^2 = 0.5$. In these models, the random effect b_1 is associated with an exogenous variable $Z = (z_1, \ldots, z_n)^T$, where $z_i \underset{i.i.d}{\sim} \mathcal{U}([0, 3])$ for $i = 1, \ldots, n$.

The mean behavior function depends on the dimension of the simulated data :

— In the *low-dimensional case* (with $p = 6$) :

$$f(x) = 1.3 \times \left(x^{(1)}\right)^2 + 2 \times \left|x^{(2)}\right|^{1/2} \tag{2.6}$$

— In the *high-dimensional case* (with $p = 8000$ and with at least 20 variables in the first two groups of explanatory variables) :

$$f(x) = 1.3 \times \left(\frac{1}{20} \sum_{g \in g_1^{20}} X^{(g)}\right)^2 + 2 \times \left|\frac{1}{20} \sum_{g \in g_2^{20}} X^{(g)}\right|^{\frac{1}{2}} \tag{2.7}$$

where g_1^{20} and g_2^{20} represent two sets of 20 genes randomly picked from the group g_1 and g_2 respectively.

The mean behavior function is actually quite the same in the two situations. The difference lies in the fact that in the high-dimension case, 40 variables are related to the response variable, against 2 in the low-dimension case. It is indeed reasonable, in high-dimensional genomic data, to assume that several genes coming from the same group are linked to the mean behavior function f.

2.4.2 Squared bias and prediction error

The different methods are compared in terms of squared bias (associated to each estimated quantity) and prediction performance, computed among M repetitions of the simulation.

Squared biases are defined as follows :

$$bias^2\left(\widehat{f}^M\right) = \frac{1}{n\#\mathbb{T}} \sum_{t\in\mathbb{T}} \sum_{i=1}^{n} \left\{\widehat{f}^M(X_i(t)) - f(X_i(t))\right\}^2$$

$$bias^2(\widehat{B}^M) = \frac{1}{q^2} \sum_{1\leq k,l\leq q} \left(\widehat{B}^M_{kl} - B_{kl}\right)^2$$

$$bias^2\left(\widehat{\gamma}^2_M\right) = \left(\widehat{\gamma}^2_M - \gamma^2\right)^2 \qquad bias^2\left(\widehat{\sigma}^2_M\right) = \left(\widehat{\sigma}^2_M - \sigma^2\right)^2$$

with

— $\mathbb{T}$ a fixed grid of times (different from the times of measurements),

— $\widehat{f}^M\left(X_i(t)\right) = \frac{1}{M}\sum_{m=1}^{M} \widehat{f}^m\left(X_i(t)\right) \quad \forall t \in \mathbb{T}, \forall i = 1, \ldots, n$

— $\widehat{f}^m$ the random forest returned by the algorithm after convergence, associated with the m-th repetition.

— $\widehat{B}^M = \frac{1}{M}\sum_{m=1}^{M} \widehat{B}^m, \quad \widehat{\gamma}^2_M = \frac{1}{M}\sum_{m=1}^{M} \widehat{\gamma}^2_m, \quad \widehat{\sigma}^2_M = \frac{1}{M}\sum_{m=1}^{M} \widehat{\sigma}^2_m$

— $\widehat{B}^m$ the estimation of B on the m-th repetition and similarly for $\widehat{\gamma}^2_m$ and σ^2_m.

To evaluate prediction performance, each simulated dataset is split into a learning set and a test set, the test set being made of the two last measurements of each individual. With $\mathcal{T}_i^\ell$ denoting the index of the i-th individual measurements in the ℓ-th test set, we define the prediction error as :

$$\frac{1}{2nM} \sum_{\ell=1}^{M} \sum_{i=1}^{n} \sum_{j\in\mathcal{T}_i^\ell} \left(Y_{ij} - \widehat{Y}^\ell_{ij}\right)^2$$

where $\widehat{Y}^\ell_{ij}$ is the predicted response variable (see (2.8) below), for the j-th measure of the i-th individual, given by the random forest returned by the algorithm after convergence.

The prediction of the response variable for the ith individual at time t is given by :

$$\widehat{Y}_i(t) = \widehat{f}\left(X_i(t)\right) + Z_i\left(t\right)\widehat{b}_i + \widetilde{\omega}_i(t) \tag{2.8}$$

with $X_i(t)$ and $Z_i\left(t\right)$ the fixed and random effects explanatory variables for the ith individual

at time t and

$$\widetilde{\omega}_i(t) = \begin{cases} \frac{1}{t_+ - t_-}\left[(t-t_-)\widehat{\omega}_i(t_-) + (t-t_+)\widehat{\omega}_i(t_+)\right] & \text{if } t_{i,1} \leq t \leq t_{i,n_i} \\ \mathbb{E}\left(\omega_i(t)|\widehat{\omega}_i(t_{i,1})\right) = \frac{\Gamma(t,t_{i,1})}{\Gamma(t_{i,1},t_{i,1})}\widehat{\omega}_i(t_{i,1}) & \text{if } t < t_{i,1} \\ \mathbb{E}\left(\omega_i(t)|\widehat{\omega}_i(t_{i,n_i})\right) = \frac{\Gamma\left(t,t_{i,n_i}\right)}{\Gamma\left(t_{i,n_i},t_{i,n_i}\right)}\widehat{\omega}_i(t_{i,n_i}) & \text{if } t > t_{i,n_i} \end{cases}$$

with $t_- = \max\left(\{s \in \{t_{i,1}, \ldots, t_{i,n_i}\}, s \leq t\}\right)$ and $t_+ = \min\left(\{s \in \{t_{i,1}, \ldots, t_{i,n_i}\}, s \geq t\}\right)$.

2.4.3 Results

The number of individuals n, is fixed to 17 (the same as in the DALIA trial) all along the simulation study, and the number of measurements n_i for the ith individual is randomly chosen (with uniform distribution) between 8 and 11 for every $i = 1, \ldots, n$, leading to an unbalanced design. We recall that the total number of observations is denoted by $N = \sum_{i=1}^{n} n_i$.

A low-dimensional case

We start by considering a low-dimensional example where $p = 6$. We have 6 explanatory variables in the dataset and all of them have a temporal behavior (given by Equation (2.3)). This framework allows to compare different tree-based methods as well as a linear mixed model for longitudinal data in a standard framework. First, we simulate one dataset under the non-stochastic model (2.4) using the mean behavior function f defined in Equation (2.6) and study the behavior of the **MERF** method on that dataset.

Figure 2-2 shows that the convergence of the ML-EM algorithm for the **MERF** method is quite affected by the `mtry` parameter value (the number of variables randomly drawn before optimizing the node splitting in the trees composing the RF). Standard RF are already sensitive to this parameter (Díaz-Uriarte and Alvarez De Andres (2006) ; Genuer et al. (2008)), but MERF is even more sensitive to it, because a sub-optimal value leads to a bad estimation of f which could also lead to bad predictions of random effects. In this example, `mtry` must be set at least equal to 3, in all our experiments we chose `mtry`=4.

We now simulate 100 datasets (again under model (2.4) with mean behavior function (2.6)) and study squared biases on estimations of quantities of interest (f, B, σ^2 and γ^2 when appropriate), given by the four tree-based methods (**MERT** and **REEMtree** for

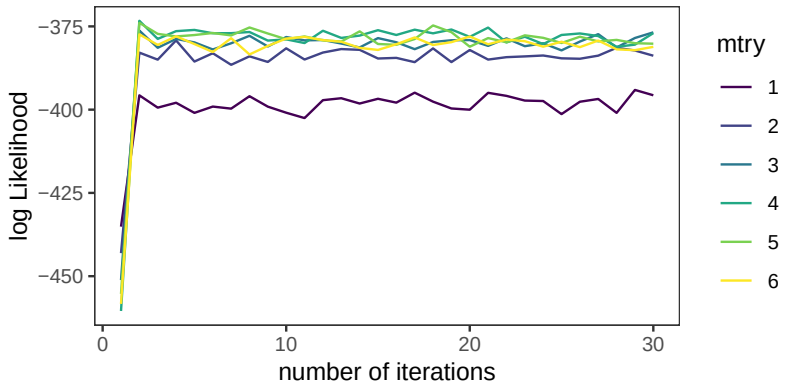

FIGURE 2-2 – Evolution of the log-likelihood against the number of iterations in **MERF** method for different `mtry` values, data simulated under model (2.4) in the low-dimensional case.

TABLE 2.1 – Squared bias of the estimated parameters, averaged on 100 datasets simulated under model (2.4) in the low-dimensional case, obtained with either well-specified models or misspecified models (which include a Brownian motion).

	f	B	γ^2	σ^2
Well-specified models				
LMEM	17.786	5.452	*	2.632
MERT	1.015	0.445	*	0.068
REEMtree	0.952	0.412	*	0.067
MERF	0.377	0.497	*	0.013
REEMforest	0.293	0.472	*	0.013
Misspecified models				
SLMEM	19.271	5.483	*	1.963
SMERT	1.287	0.476	*	0.112
SREEMtree	1.547	0.445	*	0.051
SMERF	0.495	0.504	*	0.033
SREEMforest	0.433	0.469	*	0.033

trees, **MERF** and **REEMforest** for forests) and also compare to the Linear Mixed Effects Model (**LMEM**) method. To study the robustness of the methods including stochastic process estimation, we also apply them by specifying a Brownian motion as stochastic process.

As shown in Table 2.1, when the models are well-specified, **LMEM** leads to much higher biases on all parameters compared to all other methods. Tree-based methods **MERT** and **REEMtree** are close to each other in terms of bias on f while **MERF** and **REEMforest**

which use random forests, provide a much better mean behavior estimation. Moreover, the squared bias on f for **REEMforest** is about 20% lower than **MERF** whereas the squared bias on f of **REEMtree** is only 6% lower than the one obtained with **MERT**. Hence, in this framework, taking into account the intra-individual covariance structure to evaluate the tree leaves values generates a much more important decrease of the squared bias on f when RF are used instead of CART. Furthermore, the squared bias obtained on the random effects covariance matrix B and the residual variance parameter σ^2 are lower for all four tree-based methods compared to **LMEM**, with forests estimating σ^2 much better than trees. Finally **REEMforest** gives slightly lower bias than **MERF**. In the case of misspecified models (that include a Brownian motion when there is no stochastic process), biases on f and σ^2 increase for almost all methods. It seems quite unavoidable since the covariance structure is changed, but in what follows we show that this increase does not harm much the results in terms of prediction performance. It is worth noting that, even in the misspecified case, the forest methods **SMERF** and **SREEMForest** still obtain a much lower bias than the ones obtained in the well-specified case by the other methods.

Next, we simulate 100 additional datasets under model (2.5) (still with f defined by Equation (2.6)) and compare the five methods described above first when the stochastic process is well-specified as a Brownian motion, secondly when it is misspecified as an Ornstein-Uhlenbeck process and then when no stochastic process is specified.

As shown is Table 2.2, when the models are well-specified, **SLMEM** again leads to much higher biases on all parameters compared to the other methods. Concerning tree-based methods, **SMERT** performs better than **SREEMtree** on f while **SREEMtree** leads to much lower biases on γ^2 and σ^2. Forests methods (**SMERF** and **SREEMforest**) still perform better than trees with squared biases almost all much lower. In the first misspecified case, in which the models do not include a stochastic process, we notice a very high increase in biases on B and σ^2 for all methods. Nevertheless the bias on f is quite stable for all methods except for **REEMtree** for which it decreases. Thus, not specifying a process when there is one does not prevent us from estimating the mean behavior function f very well. In the second misspecified case, *i.e.* when an Ornstein-Uhlenbeck process is used instead of the Brownian motion used to simulate the datasets, biases on B, σ^2 and especially on γ^2 increase. This was expected since those parameters help to model the residual variance and strongly depends on the specification of the stochastic process. However the bias on f

TABLE 2.2 – Squared bias of the estimated parameters, averaged on 100 datasets simulated under model (2.5) in the low-dimensional case, obtained with either well-specified models (that include a Brownian motion) or misspecified models (that include either no stochastic process or an Ornstein-Uhlenbeck process).

	f	B	γ^2	σ^2
Well-specified models				
SLMEM	16.647	3.791	0.129	2.555
SMERT	1.787	0.521	0.065	0.490
SREEMtree	2.200	0.493	0.019	0.144
SMERF	0.814	0.521	0.034	0.012
SREEMforest	0.779	0.510	0.036	0.013
Misspecified models				
LMEM	12.275	4.303	*	9.064
MERT	1.517	0.986	*	1.714
REEMtree	1.358	0.898	*	1.695
MERF	0.781	1.052	*	2.382
REEMforest	0.783	1.041	*	2.404
SLMEM	13.576	4.351	1.000	2.722
SMERT	1.764	0.783	0.138	0.601
SREEMtree	1.599	0.727	0.274	0.258
SMERF	0.832	0.902	0.513	0.029
SREEMforest	0.784	0.891	0.458	0.027

remains stable (except for **SREEMtree** for which it decreases), this illustrates the ability of the **(S)MERT**, **(S)REEMtree**, **(S)MERF** and **(S)REEMforest** methods to correctly estimate the mean behavior function f even when the stochastic process is misspecified.

Finally, we compare the different methods on their prediction capacity by computing prediction errors on 100 simulated datasets, either under model (2.4) or (2.5). For each dataset, a test set if formed by picking, for each individual i, the two last observations. This gives a test set containing $2n$ observations and a learning set with $N-2n$ observations. Breiman's RF is also included in this study in addition to the five methods already mentioned to illustrate the gain of taking into account the intra-individual correlation.

When the models are well-specified, Figure 2-3 and 2-4 show that **(S)LMEM** reached a poor prediction ability, because f is not linear. **(S)MERT** and **(S)REEMtree** gave intermediate performance, whereas **(S)MERF** and **(S)REEMforest** reached the lowest test errors, with similar performance. Breiman's RF were not included in the graphs because

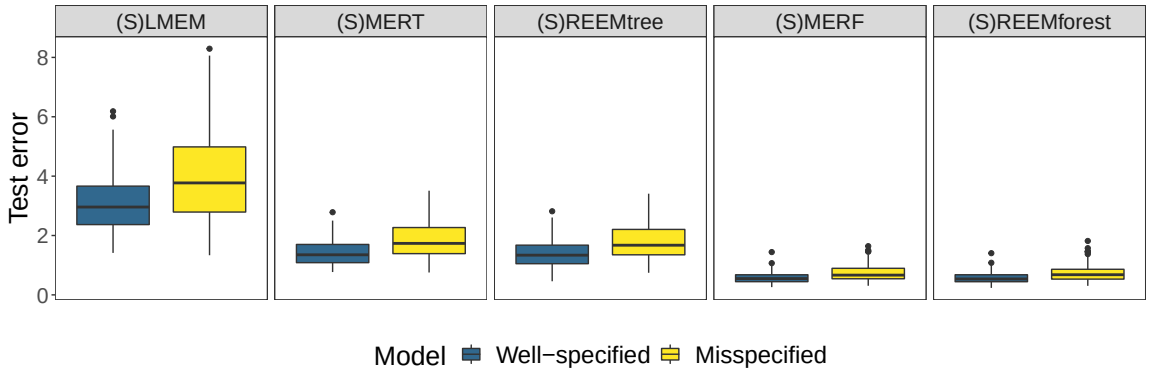

FIGURE 2-3 – Boxplots of the test errors computed on 100 simulated datasets in the low-dimensional case under model (2.4). For each method (in column) the prediction errors were obtained either with well-specified models (with the same parameters as those used to simulate the data) or with misspecified models (which use a Brownian motion while none was used to generate the data).

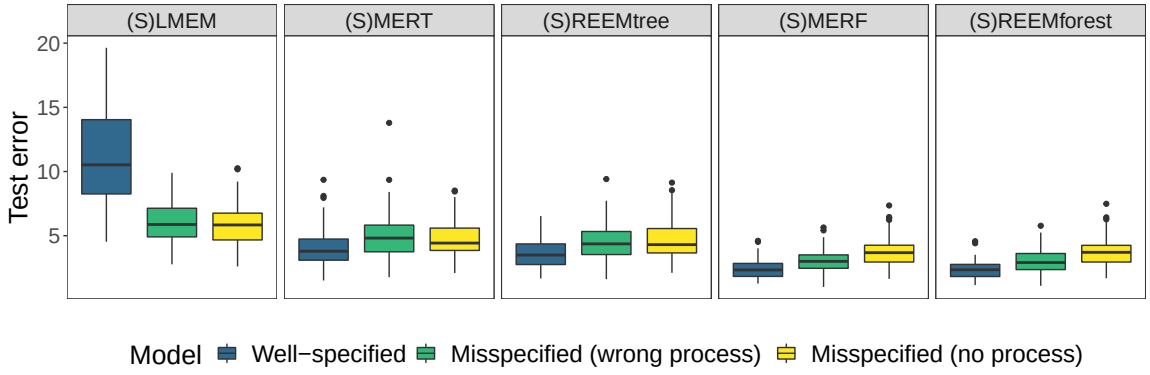

FIGURE 2-4 – Boxplots of the test errors computed on 100 simulated datasets in the low-dimensional case under model (2.5). For each method (in column) the prediction errors were obtained either with well-specified models (with the same parameters as those used to simulate the data) or with misspecified models, *i.e.* with an Ornstein-Uhlenbeck process instead of the Brownian motion used to generate the data (wrong process) or models without stochastic process (no process).

they are insensitive to changes in the model, but they reach quite high test errors (on average 6.74 and 13.15 for the datasets simulated under model (2.4) and (2.5) respectively).

In the misspecified cases, Figure 2-3 shows that using a stochastic process (Brownian motion), when there is none, increases the prediction error for all methods. This increase is more contained for the forest-based methods **SMERF** and **SREEMforest**. Figure 2-4 shows that specifying the wrong stochastic process increases prediction errors for **SMERT**, **SREEMTree**, **SMERF**, **SREEMForest** methods. In addition, when no stochastic process

is included in the model, prediction errors increase for all methods (except for **LMEM**). More precisely, for the tree-based methods, prediction errors are comparable the two misspecified cases, whereas for forest-based methods, the performance is worse when no stochastic process is included. It therefore seems preferable to always use a stochastic process.

Finally, to highlight the stability of **(S)REEMForest** under model misspecification, variable importance (VI) scores, computed with the RF returned after convergence of the method, are plotted in Figure 2-5 for both non-stochastic and stochastic models either with well-specified or misspecified models (for the stochastic model, we only consider the misspecified case where an Ornstein-Uhlenbeck process is used). We can see that variable importance is very stable in every settings, even with misspecified models. Note that, since the variables $X^{(2)}$ and $X^{(3)}$ have very similar trajectories (see Figure 2-1), their importance are close. This illustrates that misspecification of the stochastic process has limited impact on the search of variables that are strongly related to the outcome.

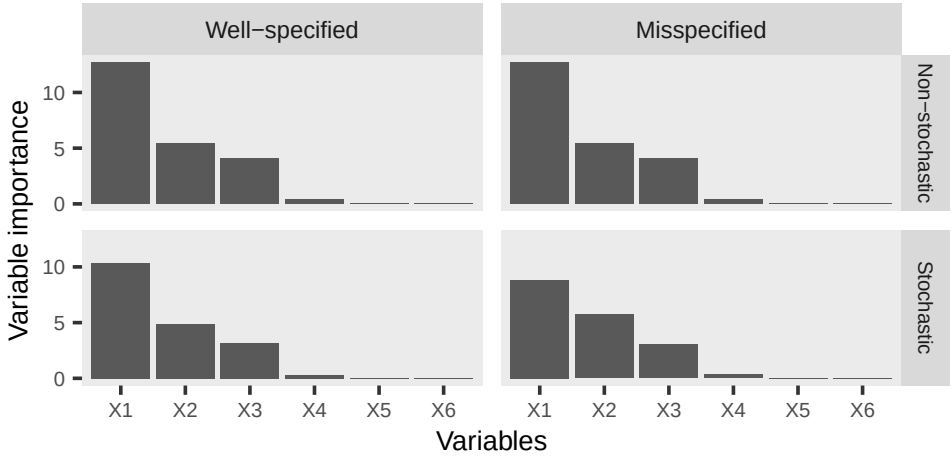

FIGURE 2-5 – Barplots of the RF variable importance scores, computed after convergence of the **REEMforest** method, obtained on one dataset in the low-dimensional case, simulated either under model (2.4) at the top or under model (2.5) at the bottom. Results obtained with well-specified models are on the left, while those with misspecified models are on the right.

As a conclusion, we demonstrate the benefits of RF approaches for longitudinal data analysis in a low-dimensional case, especially in terms of prediction error. Moreover, those methods appear rather robust to misspecification of the stochastic process. **REEMforest** exhibited a slight advantage compared to **MERF** in terms of validity of the estimation of the mean behavior function f and of the other parameters B, σ^2 and γ^2.

2.4.4 A high-dimensional case

For the high-dimensional context, we kept $n = 17$ but set $p = 8000$. We also set the size of each of the six groups containing explanatory variables with temporal behaviors (given by Equation (2.3)) to 266, leading to a total of 1596 variables that changed over time among the 8000 variables in the dataset.

First of all, according to Figure 2-2 for the low-dimensional case and some preliminary experiments, we fix the `mtry` parameter of RF to 5000 in all RF runs. This ensures convergence of ML-EM algorithms and avoids a too heavy computational load compared to an optimization of `mtry` for each RF.

TABLE 2.3 – Squared bias of the estimated parameters, averaged on 100 datasets respectively simulated under model (2.4) and (2.5) in the high-dimensional case.

	f	B	γ^2	σ^2
Non-stochastic model				
MERT	1.902	0.603	*	0.112
REEMtree	1.543	0.499	*	0.070
MERF	0.750	0.504	*	0.005
REEMforest	0.729	0.493	*	0.005
Stochastic model				
SMERT	5.229	0.926	0.113	0.590
SREEMtree	3.519	0.738	0.071	0.065
SMERF	1.378	0.511	0.024	0.010
SREEMforest	1.367	0.496	0.023	0.011

We simulated 100 datasets under model (2.4) (and 100 other datasets under model (2.5)) with the mean behavior function given by Equation (2.7), and computed squared biases on estimations given by the four tree-based methods : **(S)MERT**, **(S)REEMtree**, **(S)MERF** and **(S)REEMforest**. We did not compare anymore with **LMEM** which is not adapted to the high-dimensional setting.

For the non-stochastic scheme, the squared bias on f and all parameters obtained with **REEMforest** were slightly lower than the one obtained with the existing **MERF** method. For the stochastic model (2.5), the two forest-based methods gave similar bias on all parameters. As in the low dimensional context, forests led systematically to lower biases on all estimations compared to trees, especially for the estimation of f. Concerning trees, we note

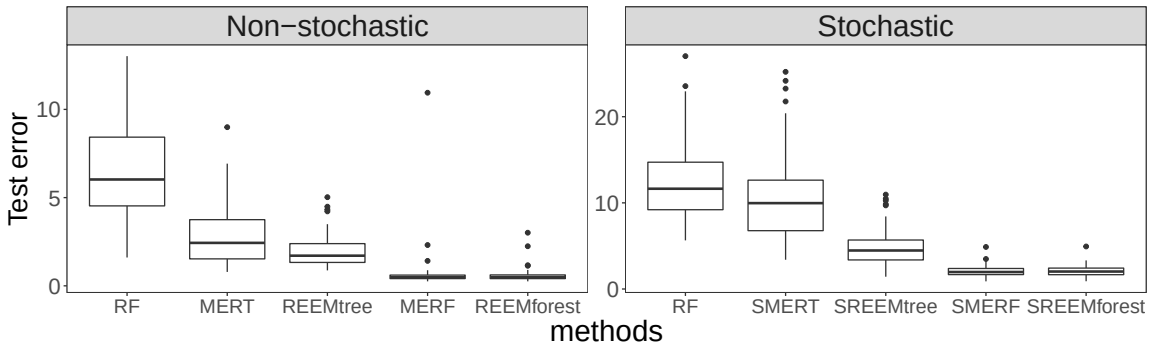

FIGURE 2-6 – Boxplots of test errors computed on 100 simulated datasets, either under model (2.4) on the left or model (2.5) on the right, in the high-dimensional case.

that **REEMtree** gave much more precise estimation of f compared to **MERT**, especially in the stochastic case. However, **(S)MERF** and **(S)REEMforest** performed quite similarly.

We estimated the prediction errors of the different methods on test samples consisting of the last two measurements of each individual trajectory in each of the simulated datasets (as in the low-dimensional case). As illustrated in Figure 2-6, forests reached very low prediction error estimations compared to trees. This last result was expected because RF perform better than trees most of the time (Breiman, 2001) and especially for high-dimensional data (Verikas et al., 2011). In addition, **(S)REEMtree** performed better than **(S)MERT**, whereas **(S)MERF** and **(S)REEMforest** gave similar results. This suggests that the gain brought by the update of leaves values is less visible after aggregating an ensemble of trees. It can also be seen that Breiman's RF (which assume independence between all observations in the data) are competitive compared to trees, especially compared to **(S)MERT**. Hence, in that case, the gain of using RF instead of trees roughly compensates the fact that Breiman's RF do not take into account the longitudinal feature of the data. We also studied biases and prediction errors obtained for the different methods under misspecification (not shown here) and we obtained results similar to those previously commented in the low-dimensional case. **(S)MERT**, **(S)REEMtree**, **(S)MERF** and **(S)REEMforest** methods remain quite robust under misspecification in the high dimensional case.

Finally, variable importance (VI) scores computed with the RF returned after convergence of the **REEMforest** method are plotted in decreasing order of VI in Figure 2-7 (only the 65 most important variables are plotted for the sake of clarity). This graph shows that the most important variables belong to one of the first three groups of explanatory variables.

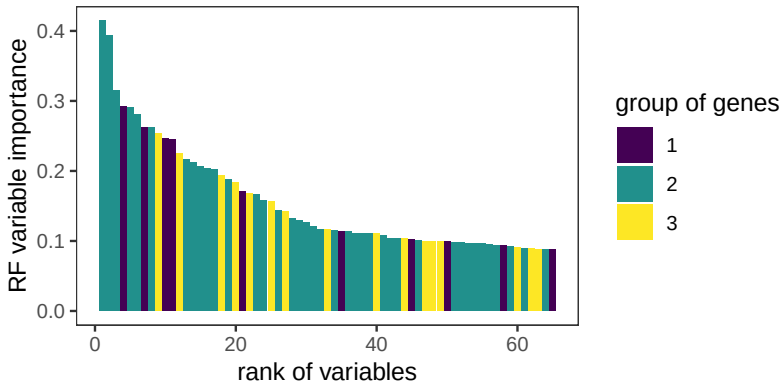

FIGURE 2-7 – Barplot of the first 65 sorted (in decreasing order) variable importance scores, computed after convergence of the **REEMforest** method applied on one dataset simulated under model (2.4) in the high-dimensional case.

This result is satisfactory because the mean behavior function (defined by Equation 2.7) depends on variables that belongs to the first two groups only and the third group is very close to the second one in terms of dynamics (see Equations 2.3).

2.5 Application to the DALIA vaccine trial

DALIA is a therapeutic phase 1/2 vaccine trial including 19 HIV-infected patients who received an HIV vaccine before stopping their antiretroviral treatment (HAART). For a full description of the DALIA vaccine trial we refer to Lévy et al. (2014).

At each harvest time, 32979 gene transcripts were measured as well as the plasma HIV RNA concentration (which was log-transformed) for every patient. In this application, we were interested in finding the gene transcripts associated to the HIV viral load dynamics after antiretroviral treatment interruption. The analysis was performed on the 17 patients with available data at the time of treatment interruption.

Figure 2-8 illustrates the dynamics of the viral replication after antiretroviral treatment interruption with a large between-individuals variability.

A random intercept and a Gaussian process were included in the model :

$$Y_{ij} = f\left(X_{ij}\right) + b_{0i} + \omega_i\left(t_{ij}\right) + \varepsilon_{ij} \; i = 1, \ldots, n \,;\; j = 1, \ldots, n_i \tag{2.9}$$

and we will refer to methods only using a random intercept as non-stochastic methods

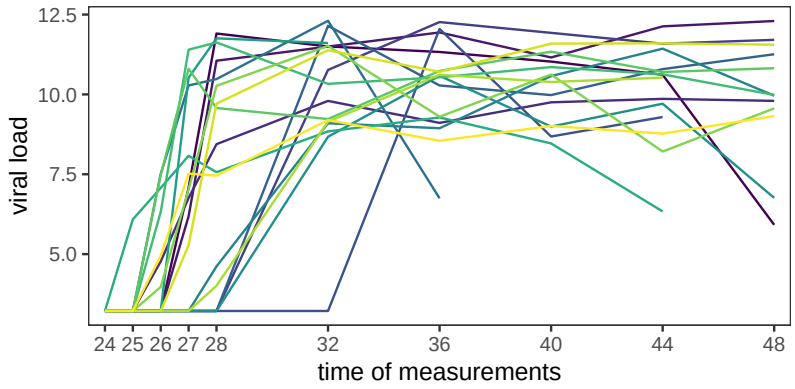

FIGURE 2-8 – Dynamics of plasma HIV viral load (one curve per patient) after anti-retroviral treatment interruption, DALIA vaccine trial.

(**MERF** and **REEMforest** in the following). Moreover, as suggested by the simulation experiments, we did not include **MERT** and **REEMtree** methods because of the really high dimension of the problem.

Prediction errors were evaluated with 25 training/test sets random splits. As in the simulation study, a test set was obtained by randomly drawing two observations for each individual. We chose the stochastic process (between an Ornstein-Uhlenbeck's process and a fractional Brownian motion) that minimized the estimated prediction error. Hence, the fractional Brownian motion with Hurst exponent $h = \frac{1}{2}$ which is the standard Brownian motion was selected. Finally, the `mtry` parameter was fixed to $9p/10 = 29681$ in all experiments of this section, according to the likelihood profile (Figure 2-9).

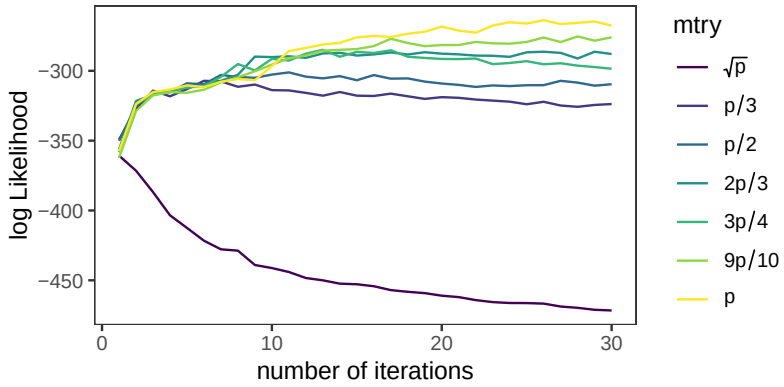

FIGURE 2-9 – Log-likelihood according to the number of iterations in **SREEMforest** from the model (2.9) with standard Brownian motion, DALIA trial.

As illustrated in Figure 2-10, Breiman's RF were comparable in terms of prediction

error with **MERF** and **REEMforest** which only included a random intercept. However, **SMERF** and **SREEMforest** outperformed RF, with a slight advantage to **SREEMforest**. This confirms, in this real dataset, that taking precisely into account the longitudinal aspect of the data in RF leads to a significant drop of the prediction error. Furthermore, this illustrates that the methods introduced in this article (**SMERF** which generalizes **MERF** in the stochastic model and **SREEMforest** which generalizes **SREEMtree**) are the most suited to analyze high-dimensional longitudinal data.

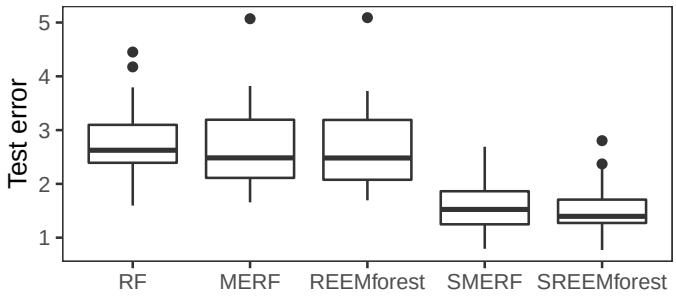

FIGURE 2-10 – Boxplots of test errors computed using 25 training/test sets random splits, for Breiman's RF, **MERF**, **REEMforest**, **SMERF** and **SREEMforest**, DALIA trial.

2.5.1 Variable selection using random forests

Once the algorithm (e.g., **SREEMforest**) has converged, a variable selection process is applied to select the genes the most associated with the viral load dynamics. More precisely, the last estimations of b_{0i} and ω_i (which are outputs of the algorithm) are subtracted from the output variable Y_i, for all i (as in step 1 of Algorithm 1) to come back to a classical regression framework (*i.e.*, with independent observations). Hence, the Variable Selection Using Random Forests method from Genuer et al. (2010) can be apply by using the `VSURF` package (Genuer et al., 2015).

This method is a fully automatic variable selection procedure based on RF and designed to deal with high dimensional data in a regression framework as well as in supervised classification. It works in three steps : i) first, the variables are sorted in decreasing order of RF variable importance (VI), then a data-driven threshold is computed to eliminate variables with low VI ; ii) variables left are then introduced (one by one according to the previous

order) in nested RF models and the one minimizing the OOB error is selected ; iii) a refined ascending sequence of RF models (obtained in a stepwise way) is then built and finally the last model of this sequence is returned.

2.5.2 Stability of the selected variables set

We illustrate the stability of the selected variables set by introducing a stability score and studying the behavior of this score against the RF parameter `mtry`.

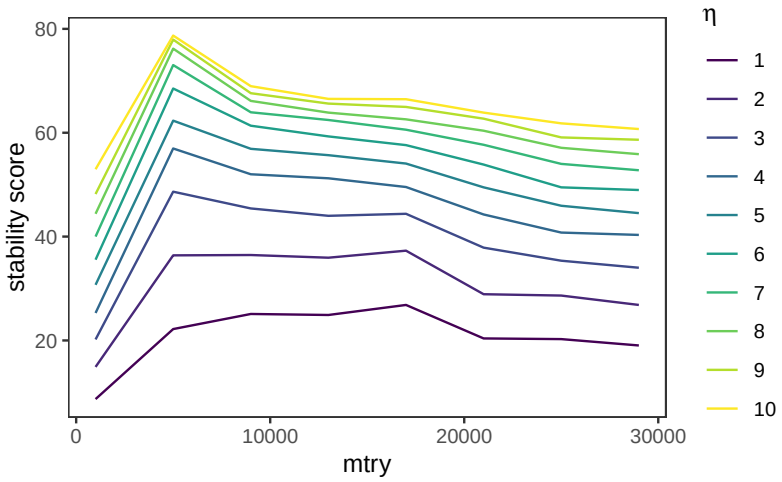

FIGURE 2-11 – Evolution of the mean stability score against the `mtry` parameter and the neighborhood size (η), restricted to the 50 most important variables, for the **SREEMforest** method, DALIA trial.

Let $\mathcal{V} = \{V_{(1)}, \ldots, V_{(p)}\}$ and $\mathcal{V}' = \{V'_{(1)}, \ldots, V'_{(p)}\}$ be the decreasing ordered variables respectively to the variable importance obtained with two runs of the **SREEMforest** method. Due to the randomness aspect of the RF, **SREEMforest** is random and the sequences $\mathcal{V}$ and $\mathcal{V}'$ may be different. Hence, we introduce a stability score $\mathcal{SS}$ which measures the difference between two ordered sequence $\mathcal{V}$ and $\mathcal{V}'$:

$$\mathcal{SS}^{\eta}\left(\mathcal{V}, \mathcal{V}'\right) = \frac{1}{p} \sum_{i=1}^{p} \mathbb{1}_{\left\{V_{(i)} \in \mathcal{B}\left(V'_{(i)}; \eta\right)\right\}}$$

with $\mathcal{B}\left(V'_{(i)};\eta\right) = \left\{V'_{(i-\eta)_+},\ldots,V'_{(i+\eta)_-}\right\}$ where

$$V'_{(i-\eta)_+} = \begin{cases} V'_{(1)} & \text{if } i-\eta \leq 0 \\ V'_{(i-\eta)} & \text{else} \end{cases} \quad \text{and} \quad V'_{(i+\eta)_-} = \begin{cases} V_{(p)} & \text{if } i+\eta \geq p \\ V'_{(i+\eta)} & \text{else} \end{cases} .$$

This score measures the proportion of variables ranked in a same neighborhood (η handles the size of the neighborhood). To stabilize the results, the score was computed with 30 pairs of sequences $\mathcal{V}$ and $\mathcal{V}'$ and the mean of the obtained stability scores was provided.

The computation of these stability scores was restricted to the 50 most important variables given by different runs of **SREEMforest** applied to the DALIA vaccine trial dataset. In Figure 2-11, we note that, except for `mtry` set to 1000, we obtained a stability score around 0.5 for a neighborhood size of 4. This means that for two lists of the 50 most important variables obtained with **SREEMforest**, approximately 50% of them were at the same rank ($\pm$4 ranks). For a neighborhood size larger than 8, the score can exceed 75%. In conclusion, for a wide range of `mtry` values, variable ranking results were quite consistent.

2.5.3 Biological results

The 21 variables selected by `VSURF` (applied after convergence of **SREEMforest**) were mainly transcripts (OAS, LY6E, HERC5, IFI/IFIT, EPSTI1, MX1, RSAD2, EIF2AK2, XAF1) associated to the interferon-α pathway. For instance, they all belongs to the Chaussabel's modules M1.2 and M3.4 annotated "Interferon" (Chaussabel and Baldwin, 2014). Interferon pathway is highly correlated to the viral replication as demonstrated previously (Bosinger et al., 2009). Only, two transcripts were not associated to the interferon pathway (EPSTI1 and SAMD9L). The commitment of the interferon pathway reflects the immune response to viral infection. The relevance of these results is another argument for the validation of the proposed approach.

2.6 Discussion

In this article, we introduced a new RF approach suited for the analysis of high-dimensional longitudinal data. We also generalized existing methods so they can be applied in the sto-

chastic semi-parametric mixed effects model. The simulation study revealed the superiority of both our approach and these generalizations. The proposed method has also been applied to a complex dataset coming from an HIV vaccine trial, illustrating its effectiveness and interest in such high-dimensional longitudinal context.

An important aspect highlighted by our study is the choice of the `mtry` parameter. Our advice is to choose a large value for `mtry`–roughly between $2p/3$ and $3p/4$–, not smaller than $p/2$. Indeed, as we are in an (very) high-dimensional context, the number of variables selected at each node of trees must not be too small–preventing to choose only non-informative variables too often–. Secondly, since the different approaches are based on an EM-algorithm, a too small value for `mtry` could lead to the non-convergence of the method and hence to very sub-optimal results, as illustrated by Figure 2-9. In addition, even if an automatic choice of `mtry` would obviously be appealing for users, it seems rather difficult to include it, because of the already quite high execution times of the proposed method.

Another key point about these approaches is the choice of the model, and more particularly the choice of the random effects. Driven by our application, we only use a random intercept (in addition to the stochastic process) regarding the number of individuals and the number of time-points we had in the vaccine trial. However, in a context with more individuals and/or less time points, it could be interesting to add random effects on the different time points. This should make the model more flexible and hence increase the method capacity to estimate the inter-individual variability.

Following the work of Fu and Simonoff (2015), one could also study the effect of the use of conditional inference trees (Hothorn et al., 2006) instead of CART trees in (S)REEMforest. This did not appear mandatory in our particular framework where all explanatory variables are continuous, but this could be addressed in the more general case where both continuous and categorical (with different numbers of categories) variables are available.

Finally, the theoretical analysis of such complex methods (non-parametric estimates plugged into an EM algorithm) seems rather difficult and remains, to the extend of our knowledge, an open issue.

Chapitre 3

Fréchet random forests for metric space valued regression with non euclidean predictors

Capitaine Louis[1], **Bigot Jérémie**[2], **Thiébaut Rodolphe**[1], **Genuer Robin**[1]

[1] INSERM U1219 Bordeaux Population Health Research Center, INRIA Bordeaux Sud-Ouest, SISTM Team, Bordeaux University, Bordeaux, France

[2] Bordeaux University, Bordeaux, France, Institut de Mathématiques de Bordeaux and CNRS (UMR 5251), Talence, France

Submitted

Abstract

Random forests are a statistical learning method widely used in many areas of scientific research because of its ability to learn complex relationships between input and output variables and also their capacity to handle high-dimensional data. However, current random forest approaches are not flexible enough to handle heterogeneous data such as curves, images and shapes. In this paper, we introduce Fréchet trees and Fréchet random forests, which allow to handle data for which input and output variables take values in general metric spaces (which can be unordered). To this end, a new way of splitting the nodes of trees is introduced and the prediction procedures of trees and forests are generalized. Then, random forests out-of-bag error and variable importance score are naturally adapted. A consistency theorem for Fréchet regressogram predictor using data-driven partitions is given and applied to Fréchet purely

uniformly random trees. The method is studied through several simulation scenarios on heterogeneous data combining longitudinal, image and scalar data. Finally, two real datasets from HIV vaccine trials are analyzed with the proposed method.

3.1 Introduction

Random Forests (Breiman, 2001) are one of the state-of-the-art machine learning methods. It owes its success to very good predictive performance coupled with very few parameters to tune. Moreover, as a tree-based method, it is able to handle regression and classification (2-class or multi-class) problems in a consistent manner and deals with quantitative or qualitative input variables. Finally, its non-parametric nature allows to proceed high-dimensional data where the number of input variables is very large in regards of statistical units.

The general principle of a tree predictor is to recursively partition the input space. Starting from the root node which contains all learning observations, it repeatedly splits each node into two or more child nodes until a stopping rule is reached. Let us focus, for the sake of clarity, on the case where all input variables are quantitative. For most of tree predictors, splits are binary and consist in an input variable X^j and a threshold s, leading to two child nodes containing observations that verify $\{X^j \leq s\}$ and $\{X^j > s\}$ respectively (Breiman et al., 1984). The splitting variable as well as the threshold are most of the time sought to minimize an heterogeneity criterion on child nodes. The main idea is to partition the input space into more and more homogeneous regions in terms of the output variable.

One limitation of the previously described splitting strategy is that all input variables must live in an ordered space (the method must decide if an observation of the splitting variable is less or larger than the threshold). Yet, with complex data structures, inputs can belong to unordered spaces. As an illustrative example, the real datasets to be analyzed in this paper are from HIV vaccine trials made of input and output variables which are discretely sampled curves representing repeated measurements over time. This example typically corresponds to observations collected in longitudinal studies. In such settings, a main objective is to predict for a given individual the output trajectory given the knowledge of inputs trajectories. If this objective is reached at the trajectory level, then the notion of order between explanatory variables is lost. However, ignoring the fact that measurements

are repeated observations over time generally leads to an important loss of information for prediction. Thus, one way of analyzing this kind of data is to generalize the notion of split in unordered metric spaces. Recently, random forests have been adapted to the general metric space framework but in the special case where neither the representation of the data nor the distances between data points are available (Haghiri et al., 2018). In the present work, we take into account the distances between any items of the space.

Hence, we consider the framework of a learning sample $\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, X_n)\}$ made of i.i.d. observations of a pair of random variable $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is a product of p metric spaces (which can be unordered) $(\mathcal{X}_1, d_1) \times \cdots \times (\mathcal{X}_p, d_p)$, and where $\mathcal{Y}$ is also a metric space with distance $d_{\mathcal{Y}}$. The core idea of this work is to generalize the notion of split that only uses the distance of each metric space. Furthermore, as the notion of mean in the output space $\mathcal{Y}$ is also needed to allocate predictions to terminal nodes of a tree, the Fréchet mean that generalizes the mean in general metric spaces (Fréchet, 1906) is used. This justifies the names Fréchet trees and Fréchet random forests hereafter. Once the notion of split is defined, the building of a maximal tree and the pruning of that tree to obtain an optimal tree are extended. Finally, with this generalization of CART trees, Fréchet random forests are derived in a rather standard way : a forest predictor is an aggregation of a collection of randomized trees. In our framework, the aggregation step therefore consists in taking the Fréchet mean of individual tree predictions.

The use of the Fréchet mean has now become a standard tool for statistical inference from manifold-valued data. For example, it is the key notion allowing to perform PCA for non-Euclidean data such as functional data on Riemanian manifolds (see e.g. Dai and Muller (2018), Fletcher et al. (2004), Sommer et al. (2010)) or histograms (Cazelles et al., 2018), and to analyze ensemble of complex objects with their shape, such as ECG curves (Bigot, 2013) or phylogenetic trees (Nye et al., 2017). New innovative regression methods have also emerged for the framework of a metric space valued output variable with Euclidean predictors (Petersen and Müller, 2019). The methods proposed in this paper allow to perform nonparametric regression between predictors taking their values in different metric spaces and a metric space valued output.

In this paper, we first present the Fréchet tree predictor in Section 3.2 before introducing Fréchet random forests in Section 3.3. We introduce an extremely randomized version of the Fréchet random forests method in Section 3.3.3. Section 3.4 is dedicated to the analysis of

the consistency of Fréchet regressogram estimators using data-driven partitions with output lying in metric space. We report numerical experiments using simulated longitudinal data to compare our approach with competitive methods, then we analyze two scenarios of heterogeneous data simulations involving curves, images and scalars in Section 3.5. An application of Fréchet random forests for statistical inference from longitudinal data is presented in Section 3.6. Finally, we discuss in Section 3.7 potential extensions of this work. All the numerical experiments of this paper are reproducible from our R package `FrechForest` [1].

3.2 Fréchet Trees

3.2.1 Fréchet means and Fréchet variance

The notions of mean and variance are central to the construction of regression trees (Breiman et al., 1984). We introduce in this section the notions of Fréchet empirical mean and Fréchet empirical variance (Fréchet, 1948), which are the natural generalization of mean and variance in metric spaces. Let $(z_1, \dots, z_n)$ a sample from a metric space $(\mathcal{Z}, d)$, the empirical Fréchet function is given by

$$\begin{array}{rccc} \mathcal{F}_n & : & \mathcal{Z} & \longmapsto & \mathbb{R}^+ \\ & & z & \longmapsto & \frac{1}{n}\sum_{i=1}^{n} d^2(z, z_i) \end{array}$$

the function $\mathcal{F}_n(z)$ measures the average squared distance between $z \in \mathcal{Z}$ and $z_1, \dots, z_n$. We define the empirical Fréchet means $\overline{z}_n$ of the sample $(z_1, \dots, z_n)$ as any minimizer of the empirical Fréchet function, i.e.

$$\overline{z}_n \in \underset{z \in \mathcal{Z}}{\arg\min}\ \mathcal{F}_n(z)$$

Note that the Fréchet mean can be non unique. The empirical Fréchet variance $\mathcal{V}_n$ of the sample $(z_1, \dots, z_n)$ is then given by

$$\mathcal{V}_n = \mathcal{F}_n(\overline{z}_n) = \frac{1}{n}\sum_{i=1}^{n} d^2(z_i, \overline{z}_n)$$

1. https://github.com/Lcapitaine/FrechForest

Note that even if the empirical Fréchet mean may not be a unique element of the metric space, the Fréchet variance is unique. Throughout the paper, Fréchet's mean and Fréchet variance will always refer to Fréchet empirical mean and Fréchet's empirical variance. For the sake of simplicity, we assume in the rest of the paper that the Fréchet mean is unique.

3.2.2 Splitting rule

One key ingredient in the building of a decision tree is the way its nodes are split (Breiman et al., 1984). Splitting a node t of a tree according to some variable $X^{(j)}$ amounts to find a way of grouping observations of this node into two subsets constituting the child nodes. This grouping is usually performed to maximize the differences between the two resulting child nodes according to the output variable. However, if variable $X^{(j)}$ is strongly related to the output variable Y, then it is expected that for two observations with "close" $X^{(j)}$ values in $(\mathcal{X}_j, d_j)$, associated outputs will be "close" in $(\mathcal{Y}, d_{\mathcal{Y}})$. From this idea, we introduce a way of splitting nodes in general metric spaces. Let $(\mathcal{Z}, d)$ be a metric space, a split is any couple of distinct elements (c_1, c_2) of $\mathcal{Z}$. We define the partition $\mathcal{P} = \{P_1, P_2\}$ associated with elements (c_1, c_2) by $P_1 = \{z \in \mathcal{Z}, d(z, c_1) \leq d(z, c_2)\}$ and $P_2 = \{z \in \mathcal{Z}, d(z, c_2) < d(z, c_1)\}$. Let A be a subset of the input space $\mathcal{X}$ and for any $j = 1, \ldots, p$, let $A_j = \{x^{(j)}, x = \left(x^{(1)}, \ldots, x^{(p)}\right) \in A\}$ denotes the set of the j-th coordinates of the components of A. Let $(c_{j,l}, c_{j,r})$ be a split on (A_j, d_j), denote $A_{j,r}$ and $A_{j,l}$ the right and left child nodes (*i.e.* the associated partition) obtained from the split $(c_{j,l}, c_{j,r})$.

The quality of the split $(c_{j,l}, c_{j,r})$ is then defined by the following measure of Fréchet variance decrease :

$$H_{n,j}(A, c_{j,l}, c_{j,r}) = \mathcal{V}_n(A) - \frac{N_n(A_{j,r})}{N_n(A)}\mathcal{V}_n(A_{j,r}) - \frac{N_n(A_{j,l})}{N_n(A)}\mathcal{V}_n(A_{j,l}) \tag{3.1}$$

where $N_n(A)$ is the number of observations of the learning set $\mathcal{L}_n$ belonging to A and $\mathcal{V}_n(A)$, $\mathcal{V}_n(A_{j,l})$ and $\mathcal{V}_n(A_{j,r})$ are the empirical Fréchet variances of outputs in A, $A_{j,r}$ and $A_{j,l}$ *i.e.*

$$\mathcal{V}_n(A) = \frac{1}{N_n(A)} \sum_{i:X_i \in A} d^2_{\mathcal{Y}}(Y_i, \overline{Y}_A) \qquad \text{(resp. for } A_{j,l} \text{ and } A_{j,r}).$$

$\overline{Y}_A$, $\overline{Y}_{A_{j,l}}$ and $\overline{Y}_{A_{j,r}}$ are the Fréchet means of outputs associated to observations belonging to nodes A, $A_{j,l}$ and $A_{j,r}$ *i.e.*

$$\overline{Y}_A = \underset{y \in \mathcal{Y}}{\arg\min} \sum_{i:X_i \in A} d^2_{\mathcal{Y}}(y, Y_i) \qquad \text{(resp. for } \overline{Y}_{A_{j,l}} \text{ and } \overline{Y}_{A_{j,r}}\text{)}.$$

It is worth noting that the decrease in Fréchet variance for each possible split is compared with the output space metric, which makes it possible to compare splits made on input variables from different metric spaces. At last, the split variable j_n^*, chosen for splitting the node is the one that maximizes $H_{n,j}$, that is

$$j_n^* = \underset{j \in \{1,\dots,p\}}{\arg\max}\, H_{n,j} \quad . \tag{3.2}$$

It is easy to show that $H_{n,j_n^*} \geq 0$ for all n, thanks to the use of the Fréchet mean, which means that each split leads to a decrease of Fréchet variance.

To determine the successive splits $(c_{j,l}, c_{j,r})$, the user defines, in a preliminary step, a split function *i.e.* a way to find the two representatives $(c_{j,l}, c_{j,r})$. More precisely, a split function is an application which associates a couple $(c_1, c_2) \in \mathcal{Z}$ to any sample $\{h_1, \dots, h_n\}$ from a general metric space $(\mathcal{Z}, d)$. For example, the 2-means algorithm (k-means with $k = 2$) can be used to determine the representatives. Note that for each metric space $(\mathcal{X}_j, d_j)$ we can use a different split function.

3.2.3 Tree building

Starting from the root node (associated with the whole input space $\mathcal{X}$), nodes are recursively split in order to give a partition of the input space $\mathcal{X}$. A node t of the tree is not split if it is pure, that is if the Fréchet variance of this node is null. As a first step in the building process, the tree is developed until all nodes are pure, leading to the so-called maximal tree. Then, the pruning algorithm of CART (Breiman et al., 1984) is applied, with the use of the Fréchet variance instead of the standard empirical variance. At the end of this step, a sequence of nested sub-trees of the maximal tree is obtained. Next, the sub-tree associated to the lowest prediction error (estimated by cross-validation) is selected as the final tree predictor. The way a Fréchet tree predicts new inputs is detailed in the next section.

3.2.4 Prediction

Let T_n be a Fréchet tree, we note $\widetilde{T}_n$ the set of leaves (*i.e.*, terminal nodes) of T_n. For each leaf $t \in \widetilde{T}_n$, the Fréchet mean of the outputs of observations belonging to t is associated to t. Then the prediction of the output variable associated with any $x \in \mathcal{X}$ is given by $\widehat{y} = T_n(x) = \sum_{t \in \widetilde{T}_n} \overline{Y}_t \mathbb{1}_{x \in t}$, where $\mathbb{1}_S$ denotes the indicator function of a set S and $\overline{Y}_t$ is the Fréchet mean of outputs in t

$$\overline{Y}_t = \underset{y \in \mathcal{Y}}{\arg\min} \sum_{i : X_i \in t} d^2_{\mathcal{Y}}(y, Y_i)$$

In order to determine to which leaf belongs an observation x, it is dropped down the tree as follows. Starting from the root node, the associated split variable $X^{(j_1)}$ is considered, together with its two child nodes $A_{j_1,l}$ and $A_{j_1,r}$, as well as the corresponding representatives $c_{j_1,l}$ and $c_{j_1,l}$. To decide in which child node x must fall, its d_{j_1}-distance with $c_{j_1,l}$ and $c_{j_1,l}$ must be computed and x goes to $A_{j_1,l}$ if $d_{j_1}(x^{(j_1)}, c_{j_1,l}) < d_{j_1}(x^{(j_1)}, c_{j_1,r})$ and to $A_{j_1,r}$ otherwise. This process is then repeated until x falls into a leaf. The error made by T_n on x is defined as :

$$\mathrm{err}(T_n(x)) = d^2_{\mathcal{Y}}(T_n(x), y) \ .$$

3.3 Fréchet random forests

3.3.1 An aggregation of Fréchet trees

A Fréchet random forest is derived as standard random forests (Breiman, 2001) : it consists in an aggregation of a collection of *randomized* Fréchet trees. Here, the same random perturbations as standard random forests (Breiman, 2001) are used. Let $l \in \{1, \ldots, q\}$ and consider the l-th tree. First, it is built on a bootstrap sample of the learning sample $\mathcal{L}_n^{\Theta_l}$ (n observations drawn with replacement among $\mathcal{L}_n$), and secondly, the search for the optimized split for each node is restricted to a subset of $mtry$ variables randomly drawn among the p input variables (this random subset is denoted Θ'_l hereafter). Hence, the l-th tree is denoted $T_n(., \Theta_l, \Theta'_l)$ and can be viewed as a doubly-randomized Fréchet tree. Once all randomized trees are built, the Fréchet mean is again used to aggregated them. Thus, for any $x \in \mathcal{X}$ the

prediction made by the Fréchet random forest is :

$$\widehat{y} = \underset{z \in \mathcal{Y}}{\arg\min} \sum_{l=1}^{q} d_{\mathcal{Y}}^2(z, T_n(x, \Theta_l, \Theta'_l)) \ .$$

3.3.2 OOB error and variable importance scores

Fréchet random forests inherit from standard random forest quantities : OOB (**O**ut-**O**f-**B**ag) error and variable importance scores. The OOB error provides a direct estimation of the prediction error of the method and proceeds as follows. The predicted output value, $\widehat{Y}_i^{OOB}$, of the i-th observation $(X_i, Y_i) \in \mathcal{L}_n$, is obtained by aggregating only trees built on bootstrap samples that do not contain (X_i, Y_i). The OOB error is then computed as the average squared distance between those predictions and the Y_i :

$$\text{errOOB} = \frac{1}{n} \sum_{i=1}^{n} d_{\mathcal{Y}}^2(Y_i, \widehat{Y}_i^{OOB}) \ .$$

Variable importance (VI) provides information on the use of input variables in the learning task that can be used *e.g.* to perform variable selection. For $j \in \{1, \ldots, p\}$, variable importance of input variable $X^{(j)}$, $VI(X^{(j)})$, is computed as follows. For the l-th bootstrap sample $\mathcal{L}_n^{\Theta_l}$, let us define the associated OOB_l sample of all observations that were not picked in $\mathcal{L}_n^{\Theta_l}$. First, errOOB_l, the error made by tree $T_n(., \Theta_l, \Theta'_l)$ on OOB_l is computed. Then, the values of $X^{(j)}$ in the OOB_l sample are randomly permuted, to get a disturbed sample $\widetilde{\text{OOB}}_l^j$, and the error, $\text{err}\widetilde{\text{OOB}}_l^j$, made by $T_n(., \Theta_l, \Theta'_l)$ on $\widetilde{\text{OOB}}_l^j$ is calculated. Finally, VI of $X^{(j)}$ is defined as :

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

3.3.3 Extremely randomized Fréchet random forests

The construction of a Fréchet tree is conditioned by : i) the existence of the Fréchet mean for the output space $(\mathcal{Y}, d_{\mathcal{Y}})$; ii) the use of a calculable split function for each input space. As mentioned in Section 3.2.2, in practice the 2-means algorithm can be used as the split function. However, it may not be applicable on all input spaces, for example on input spaces where the Fréchet mean does not exist. In order to have a split function applicable on all input metric spaces, we use the split function introduced by (Geurts et al., 2006) for

regression and classification trees in $\mathbb{R}^p$: let `ntry` be an integer between 1 and $n(n-1)/2$, we randomly draw `ntry` different splits *i.e.* `ntry` different couples of representatives, then we calculate the reduction of the Fréchet variance associated to each of these splits for the response variable and finally we select the split which maximizes the reduction of the Fréchet variance on the response variable. An extremely randomized Fréchet tree (ERFT) is any tree built with this random split function. An aggregation of extremely randomized Fréchet trees is called an extremely randomized Fréchet random forest (ERFRF). Note that when `ntry` $= n(n-1)/2$ the node split is no longer random. This splitting strategy has two advantages : it is applicable for any type of input and by taking a low value of `ntry`, it allows to drastically reduce calculation times while having excellent prediction capabilities (see Section 3.5.4).

3.4 Theory

In this section we study the consistency of Fréchet regressogram using data-driven partitions. First, we recall the notions of specific risk and global risk in a general framework before recalling the notion of Fréchet function. Then we remind the notion of family of partitions on $\mathbb{R}^p$. Finally we give the definition of Fréchet regressogram using data-driven partition and a result of its consistency in the case where the input space is $\mathbb{R}^p$ and the output space is a metric space.

3.4.1 Problem

In this section we present some notations in the general framework where $\mathcal{X}$ is any separable space and $(\mathcal{Y}, d)$ is a sparable metric space. Consider the pair of random variables $(X, Y) \in \mathcal{X} \times (\mathcal{Y}, d)$. The task is to learn a mapping $\phi : \mathcal{X} \longrightarrow \mathcal{Y}$.
For any mapping $\phi : \mathcal{X} \longrightarrow \mathcal{Y}$ the loss function L is given by

$$L\left(y, \phi(x)\right) = d^2\left(y, \phi(x)\right) \qquad y \in \mathcal{Y},\ x \in \mathcal{X}$$

The global risk associated with the mapping ϕ is defined by

$$R(\phi) = \mathbb{E}\left[L(Y, \phi\left(X\right)\right] = \mathbb{E}\left[d^2(Y, \phi\left(X\right)\right] \tag{3.3}$$

The bayes optimal mapping ϕ^* is any minimizer of the global risk function *i.e.*

$$\phi^* \in \underset{\phi:\mathcal{X}\longrightarrow\mathcal{Y}}{\arg\min}\ R(\phi) \tag{3.4}$$

When $\mathcal{X}$ and $\mathcal{Y}$ are separable, according to (Blackwell and Maitra, 1984) the global risk can be factorized as

$$R(\phi) = \mathbb{E}_X\left(\mathbb{E}_Y\left[d^2(Y,X)|X\right]\right) \tag{3.5}$$

We define the point risk function of ϕ by

$$r\left(x,\phi(x)\right) = \mathbb{E}_Y\left[L(Y,\phi(X))|X=x\right] = \mathbb{E}_Y\left[d^2(Y,\phi(X))|X=x\right] \tag{3.6}$$

The Bayes optimal point-risk mapping ϕ^* is defined by

$$\phi^*(x) \in \underset{y\in\mathcal{Y}}{\arg\min}\ r\left(x,y\right), \quad \text{where} \quad r\left(x,y\right) = \mathbb{E}_Y\left[d^2(Y,y)|X=x\right]. \tag{3.7}$$

This mapping introduced in (Petersen and Müller, 2019) is called Fréchet regression function.

3.4.2 Family of partitions

Let $\mathcal{X} = \mathbb{R}^p$, denote $\mathcal{Z} = \mathbb{R}^p \times \mathcal{Y}$ and let π_n be a partitioning rule of $\mathbb{R}^p$ *i.e* a function that associates a measurable partition of $\mathbb{R}^p$ to any vector $(z_1,\ldots,z_n) \in \mathcal{Z}^n$. We note $\mathcal{A}_n$ the family of all the partitions we can obtain with π_n :

$$\mathcal{A}_n := \left\{\pi_n\left(z_1,\ldots,z_n\right), (z_1,\ldots,z_n) \in \mathcal{Z}^n\right\} \tag{3.8}$$

We denote $\mathcal{C}\left(\mathcal{A}_n\right) = \sup_{\pi\in\mathcal{A}_n} |\pi|$ the maximal number of cells for the partitions family $\mathcal{A}_n$. Finally, let $\mathcal{A}$ be a family of partitions, let $x_1,\ldots,x_n$ n points of $\mathbb{R}^p$ and let $B = \{x_1,\ldots,x_n\}$. We note $\Delta\left(\mathcal{A}, x_1^n\right)$ the number of distinct partitions

$$\{A_1 \cap B, A_2 \cap B, \ldots, A_n \cap B\}$$

induced by the partitions $\{A_1,\ldots,A_n\} \in \mathcal{A}$. The growing function of the partitions family $\mathcal{A}$ is defined by

$$\Delta_n^*\left(\mathcal{A}\right) = \max_{x_1^n\in\mathbb{R}^{d.n}} \Delta\left(\mathcal{A}, x_1^n\right) \tag{3.9}$$

Let $(X_1, \ldots, X_n)$ a sample made of independent observations with the same distribution as X. Denote μ the distribution of X and μ_n the empirical distribution of the sample $(X_1, \ldots, X_n)$. The following Lemma can be found in (Lugosi and Nobel, 1996, lemma 1).

Lemma 1. Let $\mathcal{A}$ be any collection of partitions of $\mathbb{R}^p$. For every $n \geq 1$ and every $\epsilon > 0$,

$$\mathbb{P}\left(\sup_{\pi \in \mathcal{A}} \sum_{A \in \pi} |\mu(A) - \mu_n(A)| > \epsilon\right) \leq 4\Delta_n^*(\mathcal{A})\, 2^{\mathcal{C}(\mathcal{A})} \exp\left(-n\epsilon^2/32\right) \tag{3.10}$$

3.4.3 Fréchet regressogram

Let $\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be a learning sample made of independent observations with same distribution as (X, Y). Let π_n a partitioning rule, we define the Fréchet regressogram estimator by

$$T_n(x) = \underset{y \in \mathcal{Y}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} d^2(Y_i, y) \mathbb{1}\{X_i \in \pi_n[x]\} \tag{3.11}$$

where $\pi_n[x]$ denotes the unique cell containing x. The goal is then to show that under certain assumptions on the metric space $(\mathcal{Y}, d)$, on the distribution of (X, Y) and on the partitioning rule, this estimator is consistent for the punctual risk as well as for the global risk.

We recall the definitions of doubling dimension and covering numbers given in (Gottlieb et al., 2016).

Definition 1 (Doubling dimension)**.** Let $(\mathcal{Y}, d)$ be a metric space, let $\lambda_{\mathcal{Y}} > 0$ be the smallest positive integer such that every ball in $\mathcal{Y}$ can be covered by $\lambda_{\mathcal{Y}}$ balls of half its radius. The doubling dimension of $(\mathcal{Y}, d)$ is then defined as $\text{ddim}(\mathcal{Y}) := \log_2(\lambda_{\mathcal{Y}})$.

Definition 2 (Covering numbers)**.** The ϵ-covering number $\mathcal{N}(\epsilon, \mathcal{Y}, d)$ of a metric space $(\mathcal{Y}, d)$ is defined as the smallest number of balls of radius ϵ that suffices to cover $\mathcal{Y}$.

The diameter of a metric space $(\mathcal{Y}, d)$, denoted $\text{diam}(\mathcal{Y})$ is defined by $\text{diam}(\mathcal{Y}) = \sup_{y_1, y_1 \in \mathcal{Y}} d(y_1, y_2)$. When both the diameter and doubling dimension of the metric space $(\mathcal{Y}, d)$ are finite, according to (Gottlieb et al., 2016), the following lemma allows to bound the $\epsilon-$covering number.

Lemma 2. Let $(\mathcal{Y}, d)$ be a metric space with finite diameter $\text{diam}(\mathcal{Y}) < \infty$ and finite doubling dimension $\text{ddim}(\mathcal{Y}) < \infty$. Then, for every $0 < \epsilon \leq \text{diam}(\mathcal{Y})$

$$N(\epsilon, \mathcal{Y}, d) \leq \left(\frac{2\,\text{diam}(\mathcal{Y})}{\epsilon} \right)^{\text{ddim}(\mathcal{Y})} \tag{3.12}$$

We now state the main result of our analysis.

Theorem 1. Let $(\mathcal{Y}, d)$ with finite diameter $\text{diam}(\mathcal{Y})$ and finite doubling dimension $\text{ddim}(\mathcal{Y})$. Let π_n be a partitioning rule on $\mathbb{R}^p$, Π_n be the family of partitions of $\mathbb{R}^p$ obtained from π_n and $\mathcal{V}_n[x] = \mathbb{E}(\text{Vol}(\pi_n[x]))$ be the expected volume of the cell containing x. Assume that the following properties hold :

- **P1.** We assume that (X, Y) has uniformly continuous and bounded density ρ and the marginal ρ_X verifies $0 < \rho_{\min} \leq \rho_X$
- **P2.** $\frac{\mathcal{C}(\Pi_n)}{n} \to 0$
- **P3.** $\frac{\log(\Delta_n^*(\Pi_n))}{n} \to 0$
- **P4.** $\frac{\log \mathcal{V}_n[x]}{n} \to 0$
- **P5.** $\frac{1}{\mathcal{V}_n[x]} = o(\frac{n}{\log n})$
- **P6.** $\text{diam}(\pi_n[x]) \to 0$ almost surely

then

$$\lim_{n\to\infty} \left| r\left(x, T_n(x)\right) - \min_{y \in \mathcal{Y}} r\left(x, y\right) \right| = 0, \qquad \text{a.s.} \tag{3.13}$$

Furthermore,

$$\lim_{n\to\infty} R(T_n) - R(\phi^*) = 0, \qquad \text{a.s} \tag{3.14}$$

Démonstration. The proof can be found in Appendix 3.8. □

3.4.4 Fréchet purely uniformly random trees

In this (sub)section the input space considered is $\mathcal{X} = [0, 1]$. As several theoretical works on regression trees, we consider a simplified version of Fréchet trees. Hence, we study a variant of the purely random trees introduced in Genuer (2012), denoted Fréchet purely random tree.

Definition 3 (Fréchet purely uniformly random tree)**.** Let $\mathcal{L}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be a learning sample of i.i.d measurements in $[0,1] \times (\mathcal{Y}, d)$. Let k_n be a positive integer and $U_1, \ldots, U_{k_n}$ be k_n i.i.d uniformly drawn random variables on $[0,1]$. Denote $U_{(1)}, \ldots, U_{(k_n)}$ the order statistics, the Fréchet purely random tree predictor $FPURT_n$ is given by

$$FPURT_n(x) = \arg\min_{y \in \mathcal{Y}} \frac{1}{n} \sum_{j=0}^{k_n} \sum_{i=1}^{n} d^2(y, Y_i) \mathbb{1}\{U_{(j)} \leq x \leq U_{(j+1)}\} \qquad \forall x \in [0,1] \tag{3.15}$$

with $U_{(0)} = 0$ and $U_{(k_n+1)} = 1$

Corollary 1. Let $(\mathcal{Y}, d)$ with finite diameter $\mathrm{diam}(\mathcal{Y})$ and finite doubling dimension $\mathrm{ddim}\,(\mathcal{Y})$. Let k_n be an integer depending on n. Assume consider the following assumptions :

- **A1.** We assume that (X, Y) has uniformly continuous and bounded density ρ and the marginal ρ_X verifies $0 < \rho_{\min} \leq \rho_X$
- **A2.** $k_n \to \infty$ as $n \to \infty$ and $k_n = o(n/\log n)$

if the assumptions **A1** and **A2** hold then the Fréchet purely uniformly random tree estimator is consistent for the global risk i.e

$$\lim_{n \to \infty} R(FPURT_n) - R(\phi^*) = 0, \qquad \text{a.s} \tag{3.16}$$

Démonstration. Let π_n be the partitioning rule used to build $FPURT_n$ and let Π_n the family of partitions associated with π_n. The interval $[0,1]$ is partitioned into $k_n + 1$ intervals, then $\mathcal{C}(\Pi_n) = k_n + 1$ which implies

$$\frac{\mathcal{C}(\Pi_n)}{n} = \frac{k_n + 1}{n} \underset{n \longrightarrow \infty}{\to} 0$$

It is easy to show that $\Delta_n^*(\Pi_n) \leq n^{k_n}$, then we deduce from $k_n = o(n/\log n)$ that

$$\frac{\log \Delta_n^*(\Pi_n)}{n} \leq \frac{k_n \log n}{n} \underset{n \to \infty}{\longrightarrow} 0$$

From (Arlot and Genuer, 2014) (page 34-36) we have that the expected volume (diameter in dimension one) of the interval containing x is :

$$\mathcal{V}_n[x] = \frac{2 - x^{k_n+1} - (1-x)^{k_n+1}}{k_n + 1} \qquad \forall x \in [0,1] \tag{3.17}$$

Hence, $\mathcal{V}_n[x] \leq \frac{2}{k_n+1}$, then

$$\frac{\log \mathcal{V}_n[x]}{n} \leq \frac{\log 2 - \log(k_n+1)}{n} \underset{n\to\infty}{\longrightarrow} 0$$

Finally, for $x \in \{0,1\}$

$$\frac{\log n}{n\mathcal{V}_n[x]} = \frac{(k_n+1)\log n}{n} \underset{n\to\infty}{\longrightarrow} 0$$

and for every $0 < x < 1$

$$\frac{\log n}{n\mathcal{V}_n[x]} = \frac{(k_n+1)\log n}{(2 - x^{k_n+1} - (1-x)^{k_n+1})n} \underset{n\to\infty}{\sim} \frac{(k_n+1)\log n}{2n} \underset{n\to\infty}{\longrightarrow} 0$$

We demonstrated that the properties **P1-P5** of Theorem 1 are verified. We thus conclude that the the $FPURT_n$ estimator is consistent point-wise consistent as well as consistent for the global risk. □

3.5 Simulation study

In this section, we study the behavior of Fréchet random forests through three simulation scenarios.

3.5.1 First scenario, longitudinal data

The first scenario, inspired by our real data applications, deals with the analysis of longitudinal data where inputs and outputs are curves. We simulate $n = 100,\ 200,\ 400$ and 1000 observations of $p = 6$ input variables according to the following model for any $i = 1, \ldots, n$ and for any $j \in \{1, \ldots, 6\}$:

$$X_i^{(j)}(t) = \begin{cases} \beta_i \left(f_{j,1}(t)\mathbb{1}_{\{G_i^j=0\}} + f_{j,2}(t)\mathbb{1}_{\{G_i^j=1\}} \right) + W_i^1(t) & \text{if } j \in \{1,2\} \\ \beta_i' \left(f_{j,1}(t)\mathbb{1}_{\{G_i'^j=0\}} + f_{j,2}(t)\mathbb{1}_{\{G_i'^j=1\}} \right) + W_i^1(t) & \text{if } j \in \{3,4,5,6\} \end{cases} \tag{3.18}$$

where $X_i^{(j)}(t)$ is the observation of the jth input variable at time t for the ith curve (individual/patient) ; t browses a regular subdivision of $[0,1]$ with a step size of 0.05, G_i^j and $G_i'^j \sim B(0.5)$, β_i and $\beta_i' \sim \mathcal{N}(1, 0.3)$, $W_i^1(t)$ is a Gaussian white noise with standard devia-

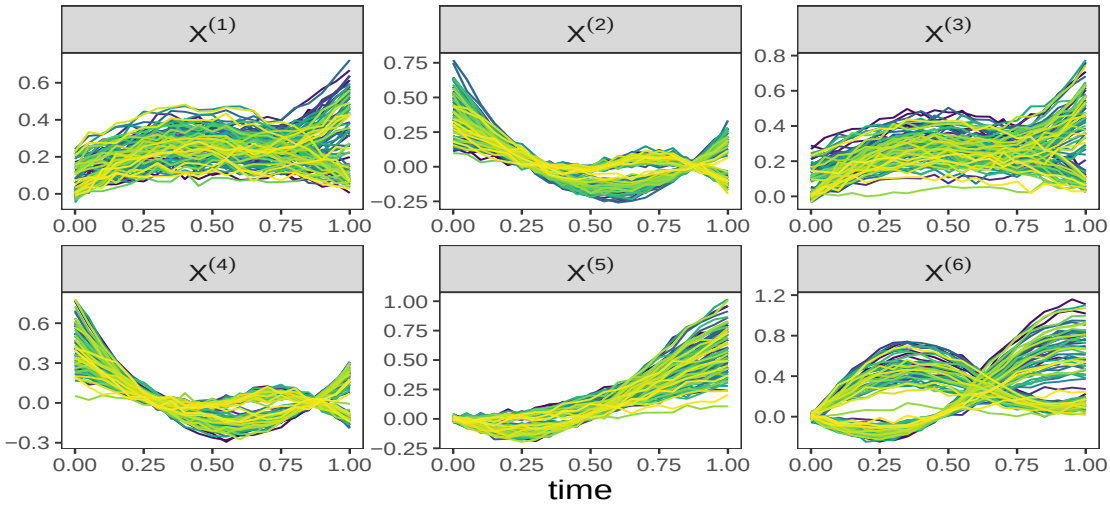

FIGURE 3-1 – Dynamics of $n = 100$ simulated input trajectories according to the model (3.18)

.

tion 0.02 and $f_{j,1}$ and $f_{j,2}$ are defined as follows :

$$\begin{cases} f_{1,1}(t) = 0.5t + 0.1\sin(6t) \\ f_{1,2}(t) = 0.3 - 0.7(t-0.45)^2 \\ f_{2,1}(t) = 2(t-0.5)^2 - 0.3t \\ f_{2,2}(t) = 0.2 - 0.3t + 0.1\cos(8t) \\ f_{3,1}(t) = f_{1,1}(t) \\ f_{3,2}(t) = f_{1,2}(t) \end{cases} \quad \begin{cases} f_{4,1}(t) = f_{2,1}(t) \\ f_{4,2}(t) = f_{2,2}(t) \\ f_{5,1}(t) = 0.5t^2 - 0.15\sin(5t) \\ f_{5,2}(t) = 0.5t^2 \\ f_{6,1}(t) = 0.6\log(t+1) - 0.3\sin(5t) \\ f_{6,2}(t) = 0.6\log(t+1) + 0.3\sin(5t) \end{cases}$$

The terms G_i^j and $G_i'^j$ allow to randomly affect typical temporal behaviors, defined by $f_{j,1}$ and $f_{j,2}$ functions, to observations. The β_i and β_i' are dilatation/shrinkage terms of $f_{j,1}$ or $f_{j,2}$, while $W_i^1(t)$ corresponds to an additive noise. As illustrated in Figure 3-1, for each input variable, the observed trajectories are variations of the typical temporal behavior functions. The observations are divided into two groups of trajectories.

Output variable Y is simulated in a similar way. The pair (G_i^1, G_i^2) is used to determine

a trajectory for the output variable, this is the primary link between X_i and Y_i

$$Y_i(t) = \beta_i \sum_{j=1}^{2} \sum_{k=1}^{2} g_{j,k}(t) \mathbb{1}_{\{G_i^j = j-1\}} \mathbb{1}_{\{G_i^k = k-1\}} + W_i^2(t) \tag{3.19}$$

where $Y_i(t)$ is the ith output curve measured at time t ; t browses the same subdivision as in (3.18), β_i are the same coefficients used in (3.18), $W_i^2(t)$ is a Gaussian white noise with standard deviation 0.05 and $g_{j,k}$ are given by :

$$\begin{cases} g_{1,1}(t) = t + 0.3 \sin(10\,(t+1)) \\ g_{1,2}(t) = t + 2(t-0.7)^2 \\ g_{2,1}(t) = 1.5 \exp\left(-\frac{(t-0.5)^2}{0.5}\right) - 0.1\,(t+1)\cos(10t) \\ g_{2,2}(t) = \frac{\log(13(t+0.2))}{1+t} \end{cases} \tag{3.20}$$

The response curves are distributed according to four different trajectory shapes, one for each pair of possible trajectory shapes for the first two input curve variables $X^{(1)}$ and $X^{(2)}$. Of note, the variables $X^{(3)}$ and $X^{(4)}$ are simulated using the same temporal functions as variables $X^{(1)}$ and $X^{(2)}$; however, the trajectories of variables $X^{(3)}$ and $X^{(4)}$ are simulated from G'^j and not from G^j and thus have no relation with the output variable Y.

3.5.2 Second scenario, predict curves with images, scalars and curves

In this scenario, we want to predict output curves from inputs that are curves, scalars and images to illustrate the flexibility of the Fréchet RF method, in particular its ability to learn about different types of inputs and outputs. The input curve variables are simulated according to the model (3.18) of the first scenario with β_i and β'_i drawn according to $\mathcal{N}(1,1)$ in order to have large variations of the curves around their average temporal behavior. Similarly, the output curves are simulated according to the model (3.19) of the first scenario. Let $\left(\mathcal{M}_i^1\right)_i$ and $\left(\mathcal{M}_i^2\right)_i$ two sequences of handwritten images of numbers 1 (for $\mathcal{M}_i^1$) and 2 (for $\mathcal{M}_i^2$) randomly drawn from the MNIST dataset (LeCun and Cortes, 2010). We simulate two input image variables $I^{(1)}$ and $I^{(2)}$ according to the following model :

$$I_i^{(j)} = \mathcal{M}_i^1 \mathbb{1}_{\{G_i^j = 0\}} + \mathcal{M}_i^2 \mathbb{1}_{\{G_i^j = 1\}} \quad \text{for } j \in \{1,2\};\ i \in \{1,\dots,n\} \tag{3.21}$$

where G_i^j are the same draws as those used to simulate the input and output curves in model (3.18) and model (3.19). Finally, consider the two real input variables $R_i^{(1)} = \beta_i$ and $R_i^{(2)} = \beta_i'$, where the β_i and β_i' are the same as those used to simulate the input and output curves. The first variable $R^{(1)}$ determines the intensity of the contraction/expansion of the $X^{(1)}$ and $X^{(2)}$ response curves. It is important to note that the link between the output curves and the input variables is entirely contained in the pairs (G_i^1, G_i^2) which determine the general shape of the output curve as well as the β_i which determine the compression/expansion of the output curves. The pairs (G_i^1, G_i^2) as well as the β_i are used to simulate the first two curve input variables $X^{(1)}$ and $X^{(2)}$. However the two image variables are constructed only from the pairs (G_i^1, G_i^2) and the scalar variables are β_i and β_i'.

3.5.3 Third scenario, predict images with curves, a toy example

The purpose of this scenario is to illustrate the ability of the Fréchet RF method to predict images from input curves. We simulate a dataset of $n = 500$ observations, the input curve variables are simulated according to the model (3.18) of the first scenario. As in the second scenario, the output images are taken from the MNIST dataset (LeCun and Cortes, 2010). For any $k = 1, \ldots, 8$ and for any $i = 1, \ldots, n$ we note $\mathcal{M}_i^k$ the random draw of the handwritten k digit in the MNIST dataset for the ith observation. Let the pair (G_i^1, G_i^2) used to attribute their shape to the curves of the first two input variables for the ith observation and β_i the expansion/contraction parameter of these same curves, then the output images are drawn according to the combinations summarized in the Table 3.1.

	$G_i^1 = 0,\ G_i^2 = 0$	$G_i^1 = 1,\ G_i^2 = 0$	$G_i^1 = 0,\ G_i^2 = 1$	$G_i^1 = 1,\ G_i^2 = 1$
$\beta_i > 1$	$\mathcal{M}_i^1$	$\mathcal{M}_i^3$	$\mathcal{M}_i^5$	$\mathcal{M}_i^7$
$\beta_i \leq 1$	$\mathcal{M}_i^2$	$\mathcal{M}_i^4$	$\mathcal{M}_i^6$	$\mathcal{M}_i^8$

TABLE 3.1 – Random draws in the MNIST dataset of the output images from the realizations $G_i^1,\ G_i^2$ and β_i used to simulate the input curves.

As in the first scenario, the output images depend only on the first two input variables, the link between the images and the curves is entirely contained in the pairs (G_i^1, G_i^2) as well as in the β_i. This means that the handwritten number images then depend both on the shape of the curves of the first two input variables and their amplitude.

3.5.4 Results

First scenario

First, we need to determine a metric for each input and output space. In the case of longitudinal data *i.e.* when repeated measurements of quantitative variables are available over time. The observations for p input and one output variables can thus be represented by time-dependent curves. In this case, the i-th observation X_i is a curve from $\mathcal{I}_1 \times \cdots \times \mathcal{I}_p \subset \mathbb{R}_+^p$ to $\mathbb{R}^p$ (where $\mathcal{I}_1 = [0, 1]$ and $\mathcal{I}_2 = [0, 1]$ in the first scenario), and Y_i is a curve from $\mathcal{J} \subset \mathbb{R}_+$ to $\mathbb{R}$. We choose to equip the resulting curves spaces with the Fréchet distance $d_{\mathcal{F}}$ introduced in (Fréchet, 1906) defined for two real-valued curves f and g with support in $\mathcal{I} \subset \mathbb{R}_+$ as

$$d_{\mathcal{F}}(f, g) = \inf_{\alpha,\beta} \max_{t \in \mathcal{I}} |f(\alpha(t)) - g(\beta(t))|$$

where α and β are any re-parameterizations of $\mathcal{I}$. The definition is the same in the discrete case (polygonal curves), except that t takes values on $\mathcal{I}$ by intervals, see Alt and Godeau (1995) for a full description of Fréchet distance for discretely sampled curves. This distance is a natural measure of similarity between the shapes of curves and has been widely used in various applications such as signature authentication (Zheng et al., 2008), path classification (Genolini et al., 2016) and speech recognition (Kwong et al., 1998). Note that, unlike several classical distances, the calculation of the Fréchet distance does not require the same number of measurements, nor the same observations times on the two trajectories. Once we have determined the metrics used for the different spaces we need to define the split function used to cut on the input spaces. Finally, the 2-means algorithm for longitudinal data using Fréchet distance and Fréchet mean introduced in (Genolini et al., 2016) is chosen on each input space to determine the different competing splits. This split function called `kmlShape` is an adaptation of the k-means method tailored to one-dimensional curves. It allows to find groups of trajectories based on their shapes (which are usually not found by conventional methods, *e.g.* based on Euclidean distance).

Fréchet trees and Fréchet random forests were compared on simulated datasets to standard CART trees (Breiman et al., 1984) and standard random forests (Breiman, 2001) as well as standard existing methods for longitudinal data analysis such as linear mixed effects model (LMEM) with a random intercept and a random effect on time and the boosting

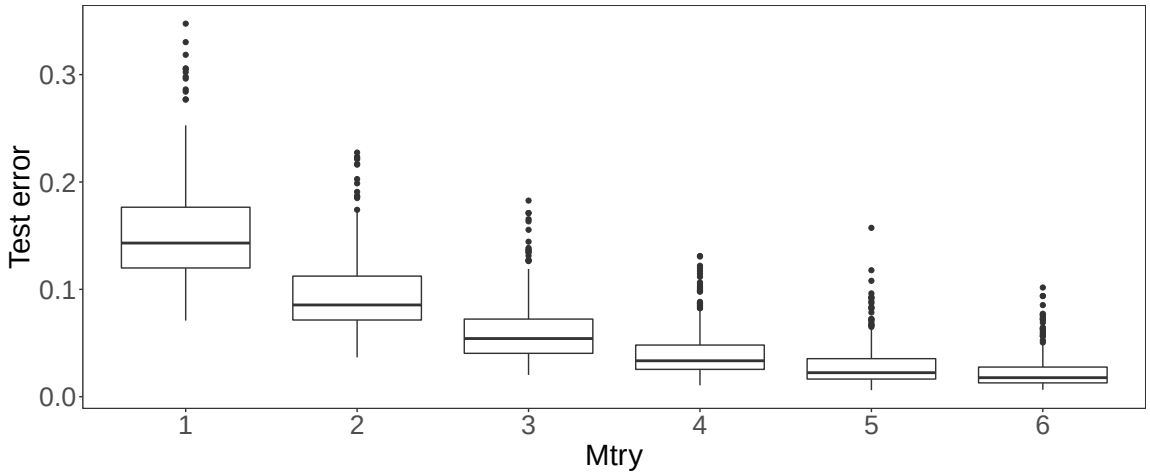

FIGURE 3-2 – Boxplots of the prediction error of the Fréchet random forests method according to the `mtry` parameter. Prediction errors are calculated on 100 datasets of size $n = 100$ simulated according to models (3.18) and (3.19) of the first scenario.

functional regression method `FDboost` (Brockhaus et al., 2017) with optimized number of iterations.

The prediction errors (mean squared error) of all the methods are estimated on several sample sizes $n = 100,\ 200,\ 400$ and 1000 using for each sample size, 100 datasets simulated according to models (3.18) and (3.19). For each simulated dataset $\mathcal{L}_n$, we randomly divide $\mathcal{L}_n$ into a training set (with $0.8n$ observations) and a test set (made of the remaining $0.2n$ observations). The Fréchet distance is used on the curved input and output spaces to build Fréchet trees and Fréchet random forests, however in order not to advantage our method, prediction errors are calculated with the usual L^2 Euclidean distance (time by time) which benefits to the standard approaches like CART trees, RF and `FDboost`.

The number of randomly drawn variables `mtry` at each node has usually a strong impact on random forests performance : if `mtry` is too small, individual trees would give too poor predictions, and if `mtry` is too high, the collection of trees could be not diverse enough ((Díaz-Uriarte and Alvarez De Andres, 2006) ;(Genuer et al., 2008)). As illustrated in Figure 3-2 the prediction error (MSE) of the Fréchet random forest decreases as the value of the `mtry` increases. In all our experiments in the first scenario, we chose `mtry`=5. We set the number of trees q to 250 (justified by the fact that, in this experiment, the OOB error stabilizes as soon as 100 trees are included in the forest). The standard random forest was composed of 500 trees and the `mtry` parameter was otpimized to 2. The number of iterations for `FDboost`

is selected between 1 and 500 through the internal procedure of the package.

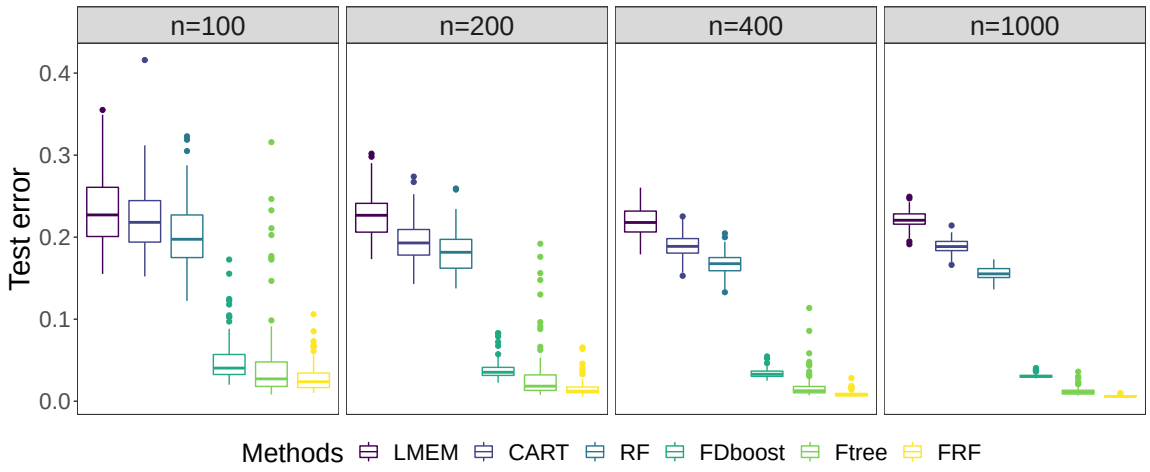

FIGURE 3-3 – Boxplots of the prediction error (MSE) of the Linear mixed effects model (LMEM), CART tree, random forests (RF), FDboost, Fréchet tree (Ftree) and Fréchet random forest (FRF) methods estimated on 100 datasets simulated according to the simulation scheme of the first scenario for $n = 100,\ 200,\ 400$ and 1000 sample sizes.

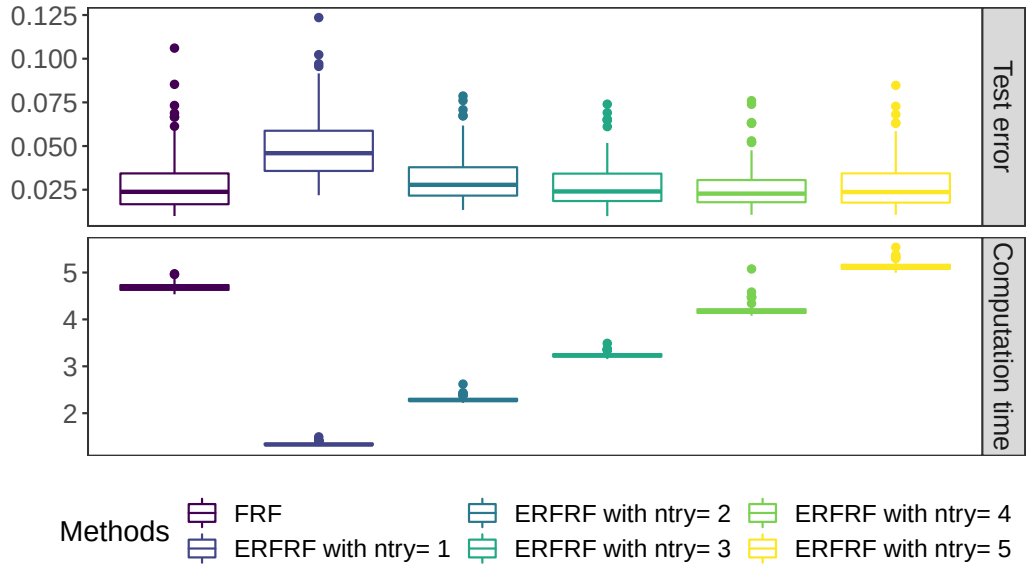

FIGURE 3-4 – Boxplots of the prediction error (MSE) and computation times estimated over 100 datasets of sample size $n = 100$ simulated under models (3.18) and (3.19) for Fréchet RF (FRF) method and Extremely Randomized Fréchet RF (ERFRF) method with different values of `ntry`.

As illustrated in Figure 3-3 for any sample size, `FDboost`, Fréchet tree, and Fréchet random forests clearly outperform the standard LMEM, CART and RF methods. Not surprisingly, the transition from a Fréchet tree to a Fréchet RF greatly improves predictive capacity by reducing prediction error and error variance. For instance, when $n = 100$, the estimated MSE obtained with a Fréchet tree is 0.047 while the one obtained with a Fréchet RF is 0.028 which is a 40% decrease in prediction error. Even though `FDboost` (our principal competitor) shows very good performances, Fréchet tree and Fréchet RF are the methods that obtain the lowest prediction errors for all sample sizes. More precisely, for small dataset ($n = 100$) `FDboost` obtains an estimated MSE of 0.05 while Fréchet tree and Fréchet RF obtain respectively 0.047 and 0.028 while for large dataset ($n = 1000$) the estimated MSE of `FDboost` is 0.031 and the Fréchet tree and Fréchet RF estimated MSE are respectively 0.012 and 0.006. Finally, note that the prediction error of the FDboost, Fréchet tree and Fréchet RF methods decreases as the sample size n increases which is not the case with other methods that keep a stable prediction error. Additionally, this decrease is much larger with the Fréchet tree and Fréchet RF methods than with the FDboost method. Moreover, the error prediction of the Fréchet RF converges to zero as n tends to infinity.

The extremely randomized version of Fréchet random forests introduced in Section 3.3.3 has some advantages over the Fréchet RF method. In particular, they are easy to implement, can be used for any type of data and reduce calculation times. In order to verify this claim we calculate the prediction error obtained by extremely randomized Fréchet forests (ERFRF) for different values of `ntry` on 100 data sets of size $n = 100$ simulated according to the first scenario. As shown in Figure 3-4, the prediction error of the ERFRF method decreases as the value of the `ntry` increases. When `ntry` is large enough (here `ntry`=3), the error obtained by ERFRF is similar to that obtained by Fréchet RF. Moreover, the execution time of an ERFRF is much lower than that of a Fréchet RF. For example, the build time of an Fréchet RF is 281 seconds while the build time of an ERFRF with ntry=3 is 191 seconds which is 30% lower. Similar results are obtained on larger datasets (not shown here).

As mentioned in the presentation of the Fréchet distance at the beginning of this section, using the Fréchet distance allows to calculate the distance between two curves measured at different times. Thus, having missing observation times for some curves does not prevent

the construction of the trees, as long as not all observation times are missing for a given curve. In order to study the robustness of Fréchet RF to missing observations, we simulate new datasets with $n = 100$ individuals according to models (3.18) and (3.19) by randomly removing 10%, 20% and 30% of the observation times for each curve. It is important to note that the removed observation times are different for each curve. For example, the observations removed for the first variable of the first individual will not necessarily be the same as those removed for the second or third variable or even the output curve of the same individual. It is then impossible to use the standard LMEM, RF and FDboost methods (it is always possible to use the CART method by removing the missing observations for the output curves). As shown in Figure 3-5, the prediction error obtained by Fréchet RF increases slightly as the percentage of missing observations increases. Moreover, the prediction error obtained with Fréchet RF on simulated data sets with 30% missing data remains competitive with that obtained by the FDboost method on datasets without missing observations.

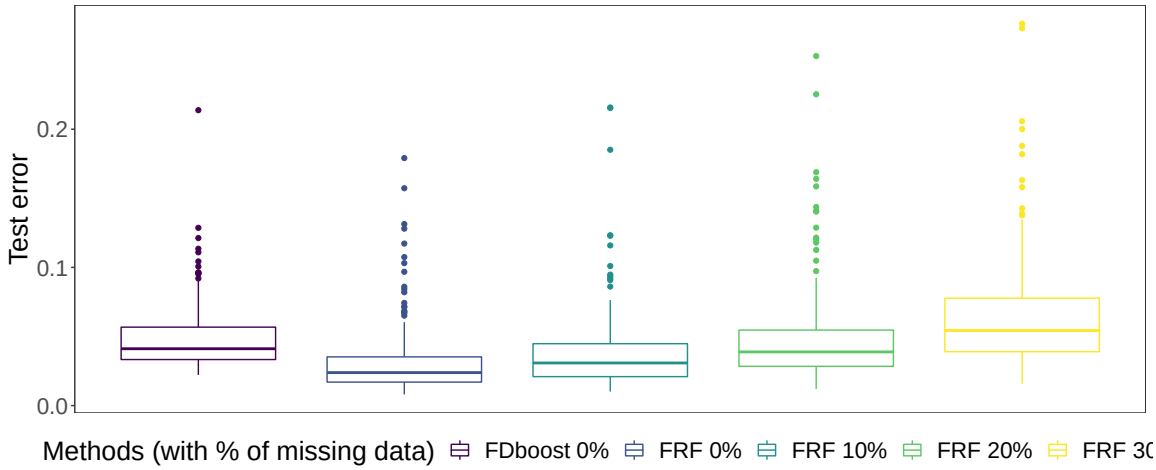

FIGURE 3-5 – Boxplots of the estimated prediction error over 100 datasets of sample size n=100 simulated under models (3.18) and (3.19) for FDboost and Fréchet RF (FRF) methods based on the number of missing observations.

It is rather common in biostatistical applications and more particularly in clinical trials to have a response variable observed after the measurement times of the input variables, this is the case of our application on DALIA-I vaccine trial (see Section 3.6). In order to study the stability of Fréchet RF method to time shifts, we transform the output curves by shifting them : i) by the same time shift of 1 for all the curves, i.e., the output curves are observed on windows $[1, 2]$ instead of $[0, 1]$ (keeping the same shapes) ; ii) by randomly

shifting each of them according to a uniform $\mathcal{U}([0,1])$, making the windows of observation of the output curves all different in this case (see Figure 3-6 for the simulated dynamics according to the time shifts). As illustrated in Figure 3-7, the constant time shift for the response curves has no influence on the Fréchet RF prediction error. When the offsets are randomly drawn for each output curve, the prediction error increases slightly to an average error of 0.039, however note that this error is still 20% smaller than the average error of 0.05 obtained by FDBoost on simulated data without time shifts.

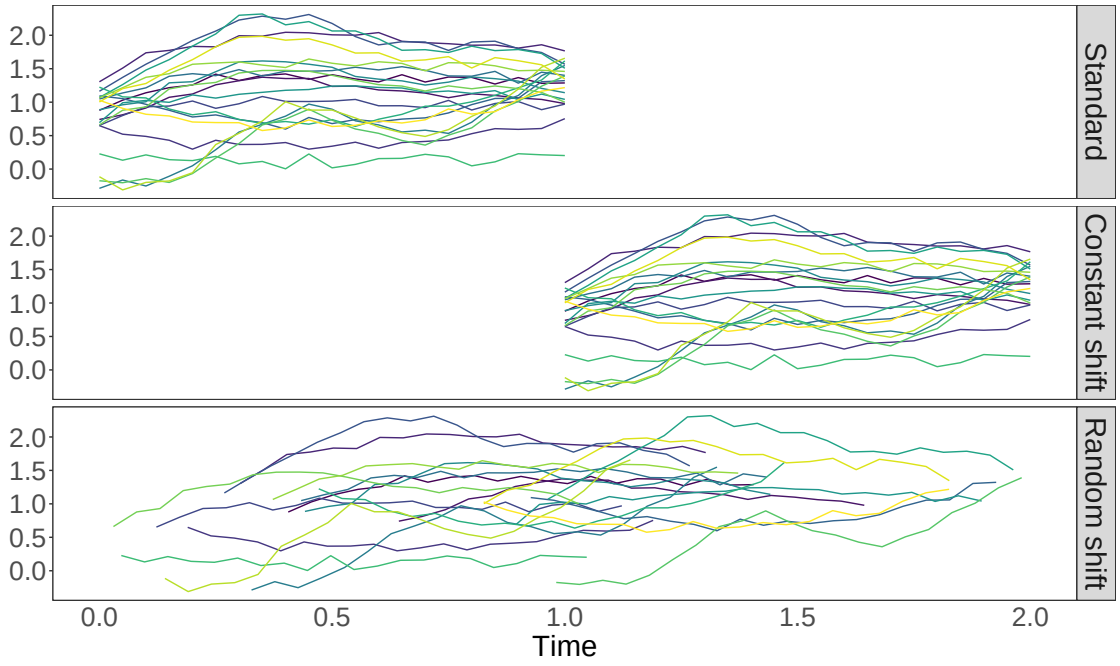

FIGURE 3-6 – Dynamics of the output variable curves simulated according to the model (4) in the standard case (i.e. without time shift), with a constant time shift equal to 1 ; with a uniform time shift $\mathcal{U}([0,1])$.

Finally, Figure 3-8 gives the importance scores of variables calculated with the Fréchet RF method on 4 datasets of size $n = 100$ simulated according to models (3.18) and (3.19) :

1. With no time shifts on the output curves and no missing observation times.
2. With random time shifts according to a uniform $\mathcal{U}([0,1])$ on the output curves but with no missing measurement times.
3. With no time shifts but with 30% missing observation times.
4. With 30% missing observations and time shifts on the output curves.

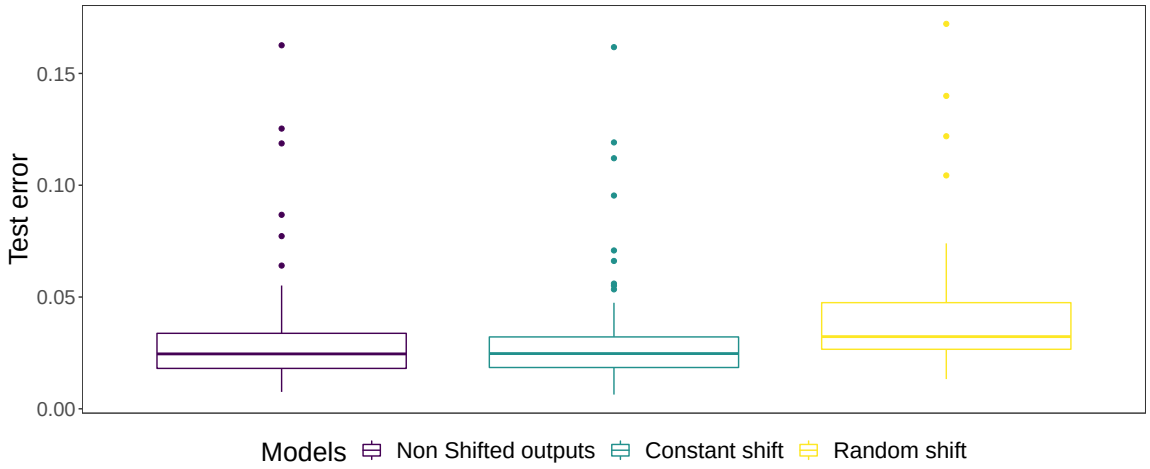

FIGURE 3-7 – Boxplots of the estimated prediction error over 100 data sets of size n=100 simulated under the second scenario for the FDboost methods based on the time shift applied to the output curves.

This graph shows that neither time shifts nor missing observation times have an impact on the importance of the variables. In fact, the first two variables (those related to the output variable) are always the ones with the highest importance scores. The other four variables (unrelated to the output variable) have extremely low importance scores compared to the first two variables.

As a conclusion, we illustrate the superiority on longitudinal data (in terms of prediction error) of the Fréchet trees and Fréchet RF methods compared to the standard LMEM, CART, RF methods as well as the longitudinal boosting method FDboost. In addition, we illustrate the great robustness of the method to missing data and time shifts, both in terms of prediction error and the importance of the variables. Lastly, we show that the extremely randomized variant ERFRF can obtain a prediction error similar to that of Fréchet RF while having lower computation times, making it a method of choice for analyzing very large datasets.

Second scenario

The Fréchet distance is used on curve spaces while the standard Euclidean distance is used on scalar spaces and image variables. Since there is no comparison with other methods in this scenario, the OOB error will be used as a measure of the performance of the Fréchet

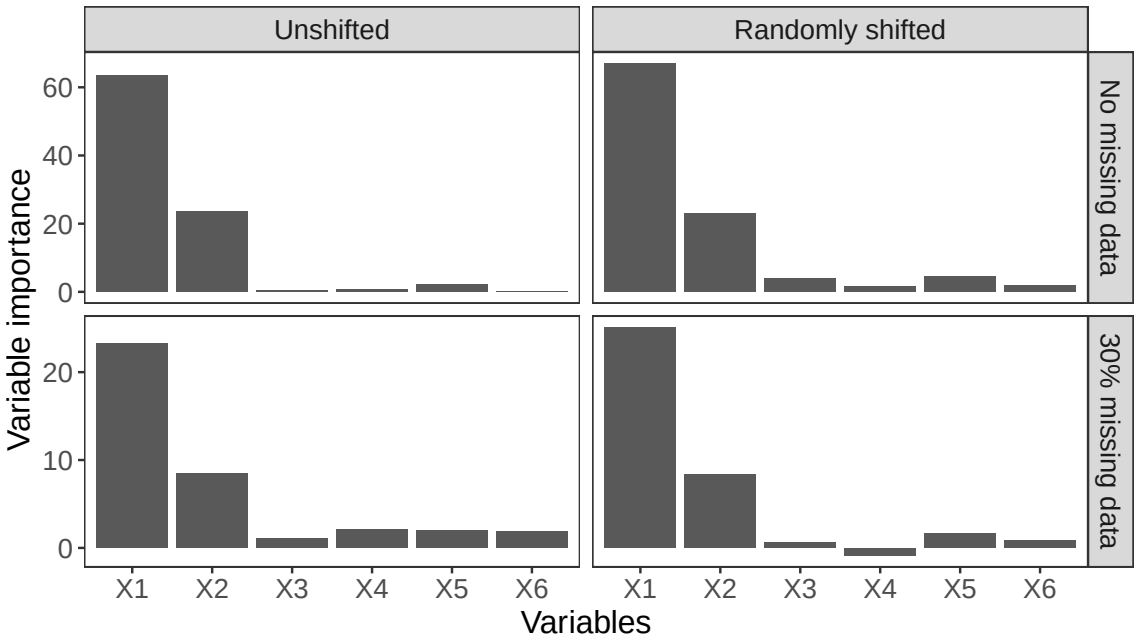

FIGURE 3-8 – Barplots of the Fréchet RF variable importance scores, obtained on 4 datasets simulated according to model (3.18) and model (3.19). The results in the left-hand column are obtained on the simulated datasets without time shift while the right-hand column contains those obtained with a random time shift on the output curves. The results on the first row are those obtained on the simulated data sets without missing data while those on the second row are those obtained on the simulated data sets with 30% missing data on the input and output curves.

RF. Throughout this section we study the ERFRF method, the version implemented in our package `FrechForest` that can handle images and shapes as inputs.

We study the OOB error obtained by ERFRF according to the types of input variables (images, curves or scalars) on 100 datasets of size n =100 simulated according to the second scenario. We consider the following models :

1. Only scalar variables $R^{(1)}$ and $R^{(2)}$ are used to predict output curves.
2. Only curve variables $X^{(1)}, \ldots, X^{(6)}$ are used to predict output curves.
3. Image variables $I^{(1)}$ and $I^{(2)}$ and scalar variables are used.
4. all variables *i.e.* curves, scalars and images are used to predict output curves.

Note that case 2 corresponds to the first simulation scenario. Figure 3-9 shows an example of an extremly randomised Fréchet tree of depth 2 (only the first three splits are shown here)

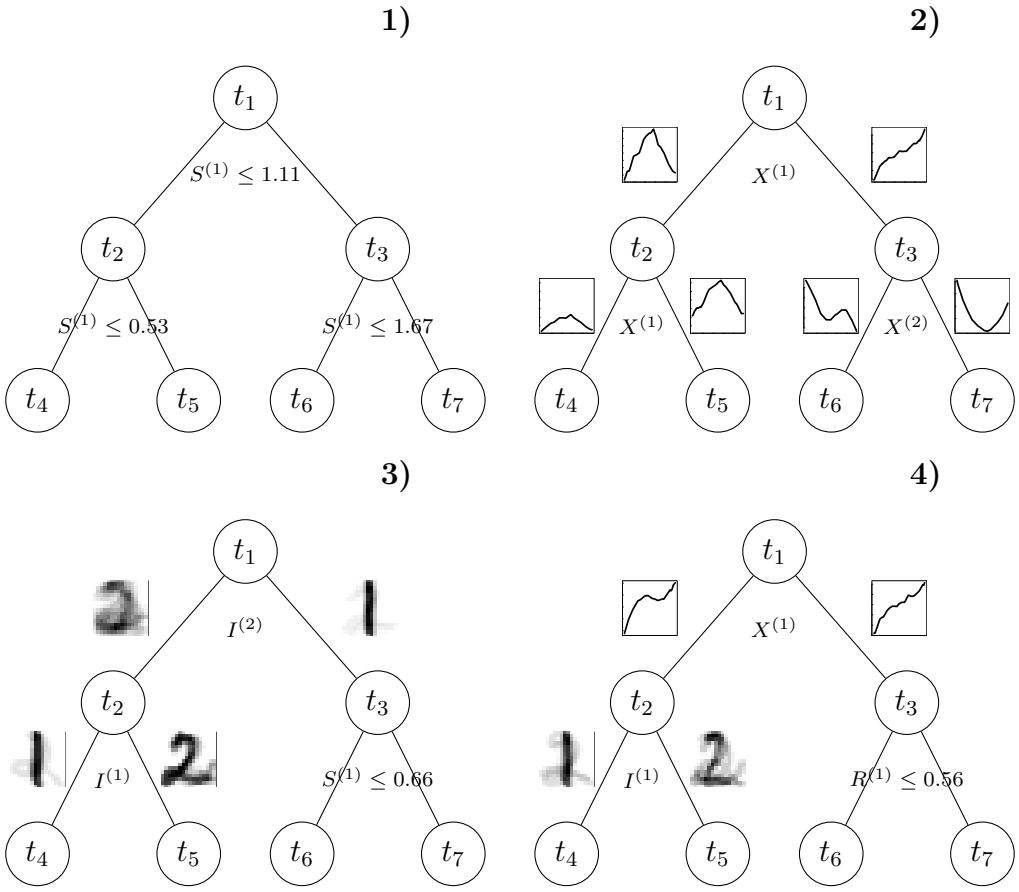

FIGURE 3-9 – Examples of 4 extremely randomized trees of depth 2 built on $n = 100$ simulated observations according to the second scenario. The 4 trees are constructed from the input variables of : 1) scalars only ; 2) curves only ; 3) images and scalars ; 4) curves, images and scalars. Below each node is indicated the split variable. To the left and right of each node are indicated the representative elements of the right and left child nodes for the split variable in question. For example for model 3) the split variable of the root node is $I^{(2)}$, the images of the variable $i^{(2)}$ which are closer to the image on the left (for the Euclidean distance), a blurred 2, go into the node t_2 while those closer to 1 go into the node t_3.

for each model above. When the models incorporate different types of inputs, in the case of models 3) and 4), the constructed trees are mixed in the sense that they can alternate the split spaces. For example, in the case of model 4), Figure 9 shows an example of a tree with the first three splits in the three different types of input spaces : curves, scalars and images. The parameters `mtry` and `ntry` are selected for each model to minimize the OOB error of the ERFRF, and are therefore different for each model.

As shown in Figure 3-10, the highest OOB error is obtained when only scalar variables

are used. When the image variables are added to the scalar variables, the OOB error is the same as the one obtained on the model using only the input curves. This was expected since the input curve variables provide the same information as the image and scalar variables combined. More precisely, the input curve variables provide both information on the shape of the output curves as well as on their amplitude, whereas the information on the shape is only provided by the images and the ones on the amplitude is only provided by the scalars. Individually, the input variables of images or scalars provide only part of the information that is provided by the input variables of curves. Finally, when the image and scalar variables are added to the curve variables, the OOB error of the ERFRF decreases. This is explained by the fact that in some cases, when the contraction or dilation of the input curves is too large, the dilated or contracted curves may have a very different shape than their initial shape and thus lose the information they brought due to their shape. Thus the addition of image variables allows to always have access to information on the shapes of the output curves. Finally, the results of these simulations emphasizes the main strength of the ERFRF method, which is to handle heterogeneous data, i.e. input and output variables of different natures.

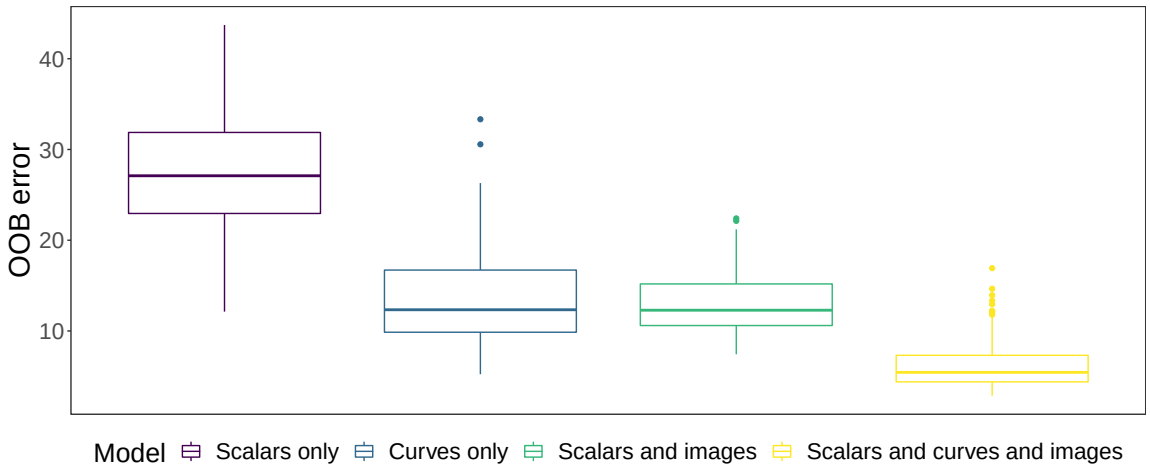

FIGURE 3-10 – OOB errors of the ERFRF method according to the types of input variables. The OOB errors are obtained on 100 data sets of size $n = 100$ simulated according to the second scenario.

Third scenario

The Fréchet distance is used on the curve spaces, i.e. on the 6 input variables. The distance used on the output space is the standard Euclidean distance. A Fréchet RF is constructed with $q = 500$ trees (justified by the fact that the OOB error of the Fréchet RF becomes stable as long as 350 trees compose the forest). Similarly, the `mtry` parameter is set to 5. As shown in Figure 3-11, OOB predictions of output images always give the correct written digit. However, ghosting can be seen on some digit predictions. This is due to the simulation scheme itself. The input curves only give information about the written number, and do not provide any information about its individual characteristics such as the width of the number, its height, the presence or not of a loop (for writing a 2 for example). More precisely, there is within the same group of numbers (for example the set of numbers 4 drawn) a variability in the written numbers that is not explained by the input curves. By introducing variables that provide information on the fine characteristics of each written number (such as its height, width, etc.) we would get even more accurate predictions. Moreover, it is noticeable that this phenomenon of ghosting is not present for numbers that have a very low variability in their writing such as the number 1. In order to highlight this point a Fréchet RF is constructed on the same simulated dataset and the images are replaced by factors indicating what the written number is. When the outputs are images, the percentage of explained variance is 20%, which was expected since there is a large variability between the same numbers that is not explained by the input curves. When the outputs are factors expressing the written numbers, the percentage of explained variance is 98%. It is therefore clear that the link between the output images and the input curves relates only to the digits and not to its individual characteristics. So even if the explained variance percentage is only 20% for the images, the Fréchet RF (almost) always predicts the right digit.

3.6 Application to real data

3.6.1 DALIA vaccine trial

DALIA is a therapeutic vaccine trial including 17 HIV-infected patients who received an HIV vaccine candidate before stopping their antiretroviral treatment. For a full description of the DALIA vaccine trial we refer to (Lévy et al., 2014). At each harvest time before

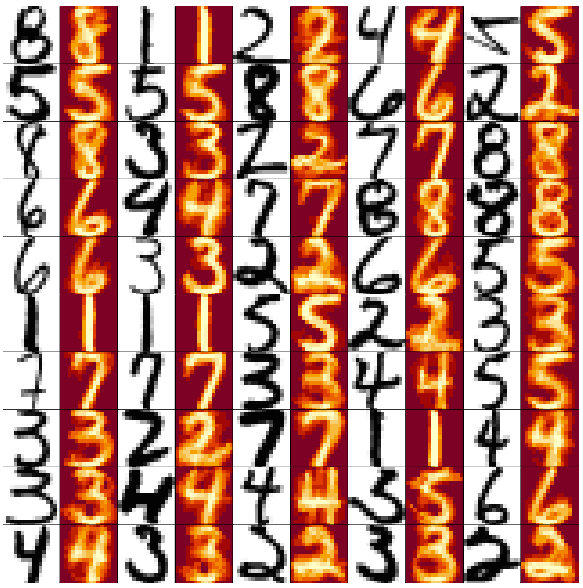

FIGURE 3-11 – True output images and OOB predictions. In black and white (grayscale), 50 output images from the dataset of $n = 500$ observations simulated according to the third scenario are displayed. The redscale image to the right of each grayscale image is the OOB prediction given by the trained Fréchet RF.

stopping their treatment, 5,399 gene transcripts which significantly vary over time during the vaccination phase were selected by (Hejblum et al., 2015) among more than 32000. The plasma HIV viral load (which was log-transformed) for every patient was measured at each harvest time after the antiretroviral treatment interruption (called HAART interruption). In this application the measurement times of the inputs (gene transcripts) differ from the ones of the output (HIV viral load). The objective is to be able to predict the HIV viral load dynamics after antiretroviral treatment interruption for a patient given the evolution of his/her gene expression during the vaccination phase (Thiébaut et al., 2019). Figure 3-12 illustrates the design of the DALIA vaccine trial and the dynamics of the viral replication after antiretroviral treatment interruption with a large between-individuals variability. The analysis with Fréchet random forest was performed on the 17 patients. The *mtry* parameter was fixed to 1500 and the number of trees, q, was set to 500. The OOB error of the Fréchet random forest converged and stabilized for almost 100 trees composing the forest. Figure 3-13 illustrates both the OOB predictions and the predictions on the learning samples (fits) of the evolution of the viral load after the HAART interruption for 4 patients of the vaccine trial.

The predictions of the Fréchet forest on the learning sample were close to the observed

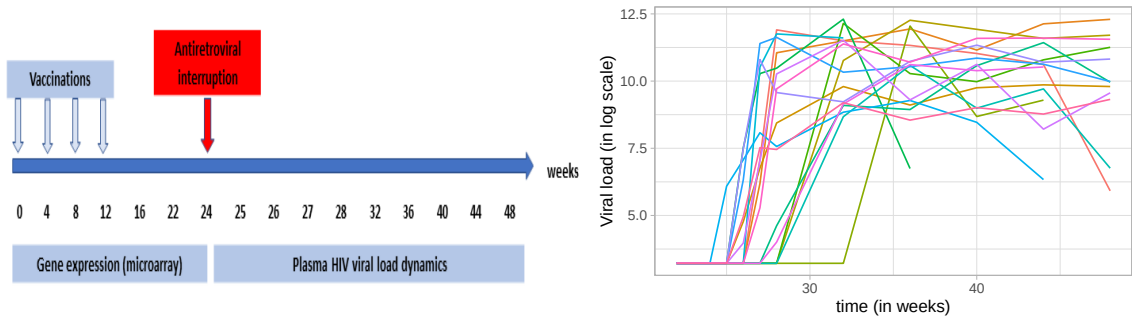

FIGURE 3-12 – On the left, the vaccine trial design. On the right, dynamics of plasma HIV viral load (one curve per patient) after antiretroviral treatment interruption (from week 24 to week 48), DALIA vaccine trial.

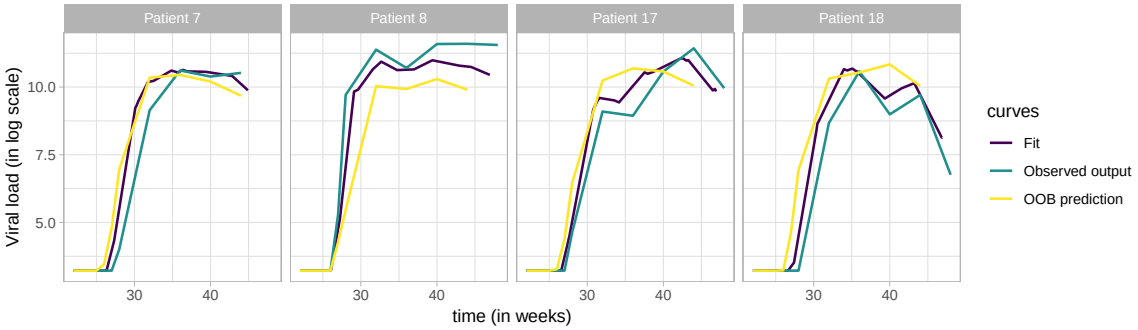

FIGURE 3-13 – Viral load after antiretroviral treatment interruption as a function of time, for four patients, together with both OOB predictions and fits (predictions on learning samples) obtained by Fréchet random forests, DALIA vaccine trial.

viral load curves. Moreover, despite a very small number of individuals, the OOB predictions obtained with this forest are quite close *in shape* to the true curves.

Among the 100 most important variables, many belong to the groups of genes (modules) that were selected in (Thiébaut et al., 2019) because i) their dynamics was influenced by the vaccine ii) their abundance after vaccination were associated with the maximum of the observed viral load. For instance, five genes from the inflammation module 3.2 and three genes from the T cell module 4.1 were selected with the current approach. Both groups of genes were extremely relevant in the context of such vaccine that generates strong T cell response.

Thus, the Fréchet random forests method applied on the complex example of the DALIA vaccine trial is extremely effective both for its capacity to predict the output variable as well as for its ability to find relevant genes in order to explain the evolution of the viral load

after the treatment interruption. Previous analyses performed in (Thiébaut et al., 2019) were done by looking at the association between one time of measurement for the input gene expressions and only one characteristic of the viral load dynamics : its maximum value. The Fréchet random forest allows a direct analysis of the whole longitudinal information available. It should be noted that standard CART trees and random forests methods cannot be used on such an application. Indeed, both the number and the observation times of the input and output variables were different.

3.6.2 LIGHT vaccine trial

LIGHT is a therapeutic vaccine trial including 97 HIV-infected patients. Using the data available in this trial, the objective of the present analysis was to assess the capacity of predicting the abundance of CD4 T cells using gene expression data as measured by RNA sequencing in whole blood, dealing with high-dimensional longitudinal data with extremely unbalanced trajectories. The dataset is composed by 1150 input variables of genes abundance. Those 1150 input genes were preselected using `dearseq` (Gauthier et al., 2019), a method for differential expression analysis. These are found to be differentially expressed overtime among more than 17000 genes. A targeted control rate for the False Discovery Rate was used at a nominal level of 5%.

Furthermore, there were 234 observations in the dataset : one to four measurements were available for each patient. In Section 3.5.4 we showed that the Fréchet RF method was very robust to unbalanced experimental designs. However, in the LIGHT dataset there is a significant proportion of patients with only one observation time, which is an extreme case that can cause some problems in predicting trajectories with three or four measurement times. We then consider the output as scalars, *i.e.* all the output observations are independent. We used 4 different RF models :

1. A standard RF.
2. A Fréchet RF considering the inputs as scalars.
3. A Fréchet RF considering the inputs as curves.
4. A Fréchet RF considering both the inputs as curves and as scalars.

As in our simulations, the Fréchet distance is used for curve spaces while the Euclidean distance is used for scalar spaces. For each random forest model the mtry parameters as well

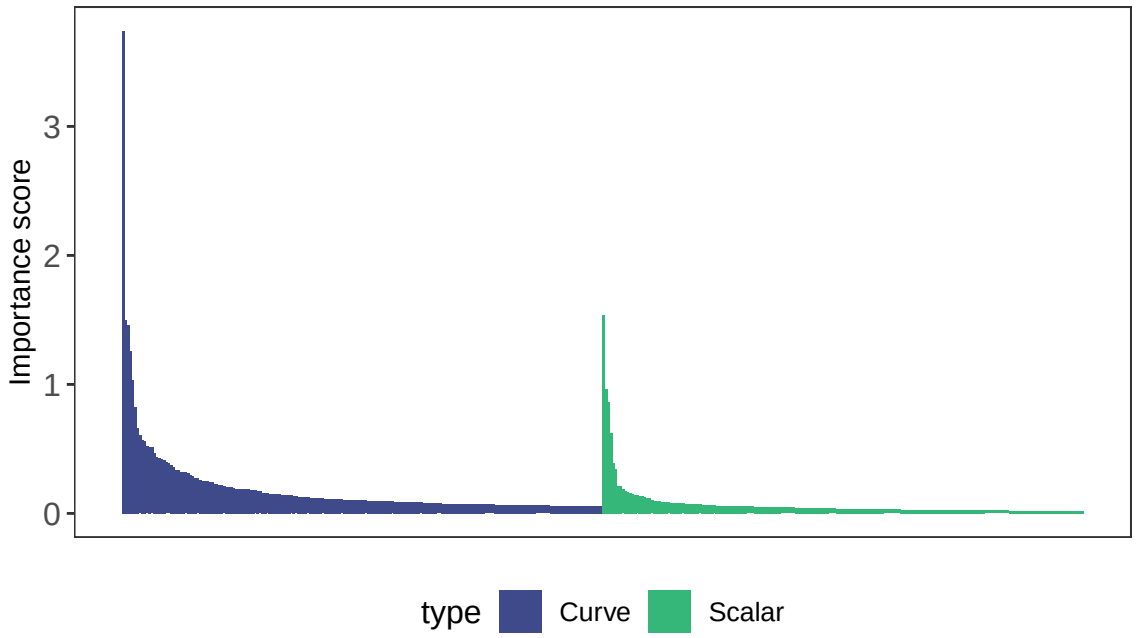

FIGURE 3-14 – Importance scores of the 200 most important input genes calculated with the RF Fréchet, in LIGHT vaccine trial.

as the number of trees composing the forest are optimized to minimize the OOB error. The prediction error estimated on OOB samples for the different RF models are compared. The standard RF method as well as the Fréchet RF method with inputs and outputs treated as scalars obtain the highest OOB errors with 22.65 and 22.61 respectively. When the input is treated as curves, the Fréchet RF method obtains an OOB error of 17.59 which represents a decrease of more than 20% of the OOB error compared to the first two models. Unsurprisingly, and in line with our simulations, considering repeated observations from the same patient as curves and partitioning the input space according to the shape of these curves greatly improves predictive performance. When the inputs are treated as both curves and scalars, the prediction error obtained from the Fréchet RF is 17.36. Thus, adding the same input variables considered as scalars does not improve the OOB error already obtained by the Fréchet RF on the curve variables only. Genes as scalars (i.e. when all observations are independent) do not therefore provide additional information compared to genes as curves, because integrating these genes in the form of scalars does not improve the prediction error.

Finally, the importance scores of the variables are considered. Figure 3-14 shows the importance scores of the 200 most important input genes calculated with the RF Fréchet built

on the genes considered as curves as well as scalars. The genes as curves globally obtain much higher importance scores than those obtained by the genes as scalars. A closer analysis of the importance scores of genes according to their type (curve or scalar) shows that the importance score for a gene as curve is systematically higher than that obtained by the same gene as a scalar. Moreover, of the 200 genes in curve form that obtain the highest importance scores, less than 47% are common to the 200 most important genes for genes in scalar form. To go further, only 42% of the 200 genes in the form of curves that obtain the highest importance scores are common to the 200 most important genes found with the standard RF method. Thus, in the LIGHT vaccine trial, considering the observations from the same patient as a trajectory not only greatly improves the prediction error but also allows us to find genes that were not previously found with the standard RF method. From the importance scores of the variables we selected the 100 most important genes for the genes considered as curves. Among these 100 genes, many of them (e.g. CD8A, CD40, TLR3, TRBV1, TRBV18...) were completely related to T cell pathways as expected because they are associated to the abundance of CD4+ T cells. Hence, Th1 pathway, Th2 pathway, T cell exhaustion pathways were among the most enriched pathways among the 100 selected genes (Ingenuity Pathway Analysis). Interestingly, the genes and pathways associated to the analysis considering inputs as scalar were less relevant with for instance the absence of the genes associated to communication between immune cells (CCL4, CCL5, CCL7) that were selected when taking into account the curves. Finally, the RF Fréchet considering the repeated measures as curves provided very relevant gene selection in this application.

3.7 Discussion

Two new tree-based methods, Fréchet trees and Fréchet random forests, for general metric spaces-valued data were introduced. Let us emphasize that the proposed methods are very general. Indeed, input variables can thus all be of different kinds, each one having its own metric, and the kind of the output variable can also be a different one.

The example of learning curve shapes in the context of longitudinal data was presented to illustrate the capacity of the methods to learn from data in unordered metric spaces. A simulation study in this framework demonstrated the superiority of Fréchet trees and forests over the existing classical methods, both in terms of prediction error as well as robustness

and flexibility. An important aspect highlighted in our study is the great robustness of Fréchet trees and Fréchet random forests. Indeed, our simulations illustrated the ability to handle missing data as well as different observation times for the different variables, which is common in longitudinal datasets. Two other simulation scenarios demonstrated the capacity of the methods to simultaneously handle data of different natures such as curves, images, scalars, factors, shapes etc. This great flexibility allows the construction of more efficient predictors while being able to compare the information provided by each of these variables of different natures thanks to the importance score. In this article, we analyzed two high-dimensional longitudinal datasets from vaccine trials. For the first time and due to the very high flexibility of Fréchet RF, it was possible to associate the entire evolution of the transcriptome during the vaccination phase with the entire evolution of the immune response after interruption of treatment on DALIA-I dataset. We have thus highlighted the groups of genes that best explain the different immune response after interruption of treatment. Finally, within the framework of LIGHT vaccine trial on which the classical methods stumbled, we highlighted the superiority of the Fréchet RF method. We illustrated that regression on curve shapes could greatly improve the prediction error while selecting new variables.

However, there are two main limitations to Fréchet trees and forests : the first one is that the Fréchet mean has to exist in the output space (Le Gouic and Loubes, 2017) and has to be fairly approximated. The second concerns the computation time. Indeed, even if the proposed approaches have been fully coded in the R package `FrechForest` for the trajectories case, Fréchet random forests can still be computationally intensive. This problem can be alleviated by the fact that, as all forests methods, they are easily parallelized (the different trees can be built in parallel).

Finally, we are working on the rates of convergence of Fréchet trees and Fréchet random forests. In parallel, we are working on the efficient implementation of metrics adapted to the image problem, such as the Wassertein distance (Vallender, 1974), in order to apply the Fréchet RF method to large brain imaging databases.

3.8 Proof of Theorem 1

First, we demonstrate the point-wise consistency given by (3.13). We introduce the following quantity

$$r_n(x,y) = \frac{\frac{1}{n}\sum_{i=1}^{n} d^2(Y_i,y)\mathbb{1}\{X_i \in \pi_n[x]\}}{\mathbb{P}(X \in \pi_n[x])} \tag{3.22}$$

From 3.11 we have $T_n(x) = \underset{y\in\mathcal{Y}}{\arg\min}\ r_n(x,y)$. First, we use the following classical upper bound in M-estimation :

$$\begin{aligned} r(x,T_n(x)) - \min_{y\in\mathcal{Y}} r(x,y) &= r(x,T_n(x)) - r_n(x,T_n(x)) + r_n(x,T_n(x)) - \min_{y\in\mathcal{Y}} r(x,y) \\ &= r(x,T_n(x)) - r_n(x,T_n(x)) + r_n(x,T_n(x)) - r(x,\phi^*(x)) \\ &\leq r(x,T_n(x)) - r_n(x,T_n(x)) + r_n(x,\phi^*(x)) - r(x,\phi^*(x)) \\ &\leq 2\sup_{y\in\mathcal{Y}} |r_n(x,y) - r(x,y)| \end{aligned} \tag{3.23}$$

We are going to decompose the above supremum in several terms that we are going to appropriately upperbound to obtain their decay to zero under the assumptions Theorem 1. Consider a δ covering of $\mathcal{Y}$ with centers $\{y_\alpha\}_{\alpha=1}^{Q}$ where $Q = N(\delta,\mathcal{Y},d)$. Thus, for every $y \in \mathcal{Y}$, there is $\alpha = \alpha_y \in \{1,\dots,Q\}$ such as $d(y,y_\alpha) < \delta$. We introduce the following quantity

$$r^E(x,y) = \frac{\mathbb{E}\left(d^2(Y,y)\mathbb{1}\{X \in \pi_n[x]\}\right)}{\mathbb{P}(X \in \pi_n[x])} \tag{3.24}$$

Then, the following decomposition is used

$$\begin{aligned} r_n(x,y) - r(x,y) = &\underbrace{r_n(x,y) - r_n(x,y_\alpha)}_{(i)} + \underbrace{r_n(x,y_\alpha) - r^E(x,y_\alpha)}_{(ii)} \\ &+ \underbrace{r^E(x,y_\alpha) - r^E(x,y)}_{(iii)} + \underbrace{r^E(x,y) - r(x,y)}_{(iv)} \end{aligned} \tag{3.25}$$

We are now going to derive upper bounds for each of the four terms above that do not depend on y. Let us start with the term (i) of (3.25), we introduce the following event

$$\mathcal{E}_n = \left\{ \left| \frac{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{X_i \in \pi_n[x]\}}{\mathbb{P}(X \in \pi_n[x])} - 1 \right| < \frac{1}{2} \right\}.$$

We can upper bound the probability of the complementary of the event $\mathcal{E}_n$ (denoted $\mathcal{E}_n^c$) as

$$\mathbb{P}\left(\mathcal{E}_n^c\right) = \mathbb{P}\left(\left|\frac{\frac{1}{n}\sum_{i=1}^n \mathbb{1}\{X_i \in \pi_n[x]\}}{\mathbb{P}\left(X \in \pi_n[x]\right)} - 1\right| > \frac{1}{2}\right) \tag{3.26}$$

$$\leq \mathbb{P}\left(\sup_{\pi\in\Pi_n}\sum_{A\in\pi}\left|\frac{\frac{1}{n}\sum_{i=1}^n \mathbb{1}\{X_i \in A\}}{\mathbb{P}\left(X \in A\right)} - 1\right| > \frac{1}{2}\right) \tag{3.27}$$

Then, we upper bound the last probability using Lemma 1

$$\mathbb{P}\left(\sup_{\pi\in\Pi_n}\sum_{A\in\pi}\left|\frac{\frac{1}{n}\sum_{i=1}^n \mathbb{1}\{X_i \in A\}}{\mathbb{P}\left(X \in A\right)} - 1\right| > \frac{1}{2}\right) \leq 4\Delta_n^*(\Pi_n)2^{\mathcal{C}(\Pi_n)}\exp -\frac{n}{128} \tag{3.28}$$

On the event $\mathcal{E}_n$, one has that $\left|\frac{\frac{1}{n}\sum_{i=1}^n \mathbb{1}\{X_i\in\pi_n[x]\}}{\mathbb{P}(X\in\pi_n[x])}\right| < \frac{3}{2}$ which implies that

$$\begin{aligned}
\left|r_n\left(x,y\right) - r_n\left(x,y_\alpha\right)\right| &= \left|\frac{\frac{1}{n}\sum_{i=1}^n \left(d^2(Y_i,y) - d^2(Y_i,y_\alpha)\right)\mathbb{1}\{X_i \in \pi_n[x]\}}{\mathbb{P}\left(X \in \pi_n[x]\right)}\right| \\
&= \left|\frac{\frac{1}{n}\sum_{i=1}^n \left(d(Y_i,y) - d(Y_i,y_\alpha)\right)\left(d(Y_i,y) + d(Y_i,y_\alpha)\right)\mathbb{1}\{X_i \in \pi_n[x]\}}{\mathbb{P}\left(X \in \pi_n[x]\right)}\right| \\
&\leq 2\,\mathrm{diam}\left(\mathcal{Y}\right) d\left(y,y_\alpha\right)\left|\frac{\frac{1}{n}\sum_{i=1}^n \mathbb{1}\{X_i \in \pi_n[x]\}}{\mathbb{P}\left(X \in \pi_n[x]\right)}\right| \leq 3\,\mathrm{diam}\left(\mathcal{Y}\right)\delta
\end{aligned} \tag{3.29}$$

On the complementary of the event $\mathcal{E}_n$, we use similar arguments and the upper bound $\left|\frac{\frac{1}{n}\sum_{i=1}^n \mathbb{1}\{X_i\in\pi_n[x]\}}{\mathbb{P}(X\in\pi_n[x])}\right| \leq \frac{1}{\mathbb{P}(X\in\pi_n[x])}$ to derive that

$$\left|r_n\left(x,y\right) - r_n\left(x,y_\alpha\right)\right| \leq 2\,\mathrm{diam}\left(\mathcal{Y}\right)\delta\frac{1}{\mathbb{P}\left(X \in \pi_n[x]\right)}.$$

Therefore, we finally obtain that

$$\left|r_n\left(x,y\right) - r_n\left(x,y_\alpha\right)\right| \leq \mathrm{diam}\left(\mathcal{Y}\right)\delta\left(3\mathbb{P}\left(\mathcal{E}_n\right) + 2\frac{1-\mathbb{P}\left(\mathcal{E}_n\right)}{\mathbb{P}\left(X \in \pi_n[x]\right)}\right)$$

Now we consider the term (ii) in (3.25). To this end, we propose to bound the following probability

$$\mathbb{P}\left(\max_{\alpha=1,\dots,Q} \left|r_n\left(x,y_\alpha\right)-r^E\left(x,y_\alpha\right)\right|>\epsilon\right)=\mathbb{P}\left(\cup_{\alpha=1}^{Q}\left\{\left|r_n\left(x,y_\alpha\right)-r^E\left(x,y_\alpha\right)\right|>\epsilon\right\}\right)$$
$$\leq \sum_{\alpha=1}^{Q}\mathbb{P}\left(\left|r_n\left(x,y_\alpha\right)-r^E\left(x,y_\alpha\right)\right|>\epsilon\right) \qquad (3.30)$$

For a fixed $\alpha\in\{1,\dots,Q\}$, we define $W_i=\frac{d^2(Y_i,y_\alpha)\mathbb{1}\{X_i\in\pi_n[x]\}}{\mathbb{P}(X\in\pi_n[x])}$, and we thus have that

$$r_n\left(x,y_\alpha\right)-r^E\left(x,y_\alpha\right)=\frac{1}{n}\sum_{i=1}^{n}W_i-\mathbb{E}\left(W_i\right),$$

which can be controlled thanks to Bernstein's inequality by finding upper bounds on $|W_i|$ and $\mathrm{var}(W_i)$. To this end, we first derive a lower bound on $\mathbb{P}\left(X\in\pi_n[x]\right)$.

$$\begin{aligned}\mathbb{P}\left(X\in\pi_n[x]\right)&=\mathbb{E}\left(\mathbb{P}\left(X\in\pi_n[x]\ |\mathcal{L}_n\right)\right)\\&=\mathbb{E}\left(\int_{\pi_n[x]}\rho_X(t)dt\right)\geq\rho_{\min}\mathbb{E}\left(\mathrm{Vol}(\pi_n[x])\right)\\&=\rho_{\min}\mathcal{V}_n[x]\qquad\text{with}\quad\mathcal{V}_n[x]=\mathbb{E}\left(\mathrm{Vol}(\pi_n[x])\right).\end{aligned}$$

Therefore, we obtain that

$$|W_i|\leq\frac{d^2(Y_i,y_\alpha)\mathbb{1}\{X_i\in\pi_n[x]\}}{\rho_{\min}\mathcal{V}_n[x]}\leq\frac{\mathrm{diam}^2\left(\mathcal{Y}\right)}{\rho_{\min}\mathcal{V}_n[x]}$$

Moreover,

$$\mathrm{var}(W_i)\leq\mathbb{E}\left(W_i^2\right)=\frac{\mathbb{E}\left(d^4(Y_i,y_\alpha)\mathbb{1}^2\{X_i\in\pi_n[x]\}\right)}{\mathbb{P}\left(X\in\pi_n[x]\right)^2}\leq\frac{\mathrm{diam}^4(\mathcal{Y})\mathbb{P}(X\in\pi_n[x])}{\mathbb{P}(X\in\pi_n[x])^2}\leq\frac{\mathrm{diam}^4(\mathcal{Y})}{\rho_{\min}\mathcal{V}_n[x]}$$

Then, by Bernstein's inequality, we have for every $\alpha\in\{1,\dots,Q\}$

$$\begin{aligned}\mathbb{P}\left(\left|r_n(x,y_\alpha)-r^E(x,y_\alpha)\right|>\epsilon\right)&\leq 2\exp\left(\frac{-n\epsilon^2}{\frac{2\,\mathrm{diam}^4(\mathcal{Y})}{\rho_{\min}\mathcal{V}_n[x]}+\frac{2\,\mathrm{diam}^2(\mathcal{Y})\epsilon}{\rho_{\min}\mathcal{V}_n[x]}}\right)\\&=2\exp\left(\frac{-n\epsilon^2\rho_{\min}\mathcal{V}_n[x]}{2\,\mathrm{diam}^2(\mathcal{Y})(\mathrm{diam}^2(\mathcal{Y})+\epsilon)}\right)\end{aligned}$$

For $\epsilon < 1$ we have

$$\mathbb{P}\left(\left|r_n(x, y_\alpha) - r^E(x, y_\alpha)\right| > \epsilon\right) \leq 2\exp\left(-Cn\epsilon^2\mathcal{V}_n[x]\right) \quad \text{with} \quad C = \frac{\rho_{\min}}{2\,\mathrm{diam}^2(\mathcal{Y})(1+\mathrm{diam}^2(\mathcal{Y}))} \tag{3.31}$$

We deduce from Equation (3.30) and Lemma 2

$$\mathbb{P}\left(\max_{\alpha=1,\ldots,Q} \left|r_n\left(x, y_\alpha\right) - r^E\left(x, y_\alpha\right)\right| > \epsilon\right) \leq 2\left(\frac{2\,\mathrm{diam}(\mathcal{Y})}{\delta}\right)^{\mathrm{ddim}(\mathcal{Y})} \exp\left(-Cn\epsilon^2\mathcal{V}_n[x]\right) \tag{3.32}$$

Let us now bound the term (iii) in (3.25) as follows

$$\begin{aligned}
\left|r^E\left(x, y_\alpha\right) - r^E\left(x, y\right)\right| &= \frac{\mathbb{E}\left[\left(d^2(Y, y_\alpha) - d^2(Y, y)\right)\mathbb{1}\{X \in \pi_n[x]\}\right]}{\mathbb{P}\left(X \in \pi_n[x]\right)} && (3.33)\\
&\leq \frac{\mathbb{E}\left[\left|\left(d(Y, y) - d(Y, y_\alpha)\right)\left(d(Y, y) + d(Y, y_\alpha)\right)\right|\mathbb{1}\{X \in \pi_n[x]\}\right]}{\mathbb{P}\left(X \in \pi_n[x]\right)} && (3.34)\\
&\leq 2\,\mathrm{diam}\left(\mathcal{Y}\right)\delta. && (3.35)
\end{aligned}$$

We combine inequalities (3.29), (3.32) and (3.33) such that with probability, we have : $1 - 2\exp\left(\mathrm{ddim}(\mathcal{Y})\log\frac{2\,\mathrm{diam}(\mathcal{Y})}{\delta} - Cn\epsilon^2\mathcal{V}_n[x]\right)$.

$$\begin{aligned}
\sup_{y\in\mathcal{Y}}\left|r_n(x, y) - r^E(x, y)\right| &\leq \mathrm{diam}(\mathcal{Y})\delta\left(3\mathbb{P}(\mathcal{E}_n) + 2\frac{1-\mathbb{P}(\mathcal{E}_n)}{\mathbb{P}(X \in \pi_n[x])}\right) + \epsilon + 2\,\mathrm{diam}(\mathcal{Y})\delta\\
&\leq \mathrm{diam}(\mathcal{Y})\delta\left(3 + 2\frac{1-\mathbb{P}(\mathcal{E}_n)}{\rho_{\min}\mathcal{V}_n[x]}\right) + \epsilon + 2\,\mathrm{diam}(\mathcal{Y})\delta\\
&= \mathrm{diam}(\mathcal{Y})\delta\left(5 + 2\frac{1-\mathbb{P}(\mathcal{E}_n)}{\rho_{\min}\mathcal{V}_n[x]}\right) + \epsilon\\
&\leq \mathrm{diam}(\mathcal{Y})\delta\left(5 + 8\frac{\Delta_n^*(\Pi_n)2^{\mathcal{C}(\Pi_n)}\exp -n/128}{\rho_{\min}\mathcal{V}_n[x]}\right) + \epsilon. \qquad (3.36)
\end{aligned}$$

Thanks to the assumptions $\frac{\mathcal{C}(\Pi_n)}{n} \to 0$, $\frac{\log(\Delta_n^*(\Pi_n))}{n} \to 0$ and $\frac{\log\mathcal{V}_n[x]}{n} \to 0$ of Theorem 1, the term $\frac{\Delta_n^*(\Pi_n)2^{\mathcal{C}(\Pi_n)}\exp -n/128}{\rho_{\min}\mathcal{V}_n[x]}$ appearing in the right hand side of the Inequality (3.36) converges to zero. Hence, there is a constant D such that

$$\frac{\Delta_n^*(\Pi_n)2^{\mathcal{C}(\Pi_n)}\exp -n/128}{\rho_{\min}\mathcal{V}_n[x]} \leq D$$

for every n. Thus we deduce the following inequality that holds with probability $1 -$

$$2\exp\left(\mathrm{ddim}(\mathcal{Y})\log\tfrac{2\,\mathrm{diam}(\mathcal{Y})}{\delta} - Cn\epsilon^2\mathcal{V}_n[x]\right)$$

$$\sup_{y\in\mathcal{Y}}\left|r_n(x,y) - r^E(x,y)\right| \leq B\delta + \epsilon \tag{3.37}$$

with $B = \mathrm{diam}(\mathcal{Y})(5+8D)$.

Let $s > 0$ and $\delta = n^{-s}$, for s large enough $B\delta$ is bounded by ϵ. Thus, for s large enough we deduce that

$$\mathbb{P}\left(\sup_{y\in\mathcal{Y}}\left|r_n(x,y) - r^E(x,y)\right| > 2\epsilon\right) \leq 2\exp\left(\mathrm{ddim}(\mathcal{Y})\log\frac{2\,\mathrm{diam}(\mathcal{Y})}{\delta} - Cn\epsilon^2\mathcal{V}_n[x]\right) \tag{3.38}$$

Under the assumption on $\frac{1}{\mathcal{V}_n[x]} = o\left(\frac{n}{\log n}\right)$, the probability upper bound on the right hand side in Inequality (3.38) becomes summable over n. We thus conclude the almost sure convergence of $\sup_{y\in\mathcal{Y}}\left|r_n(x,y) - r^E(x,y)\right|$ towards zero by the Borel-Cantelli Lemma.

Finally, we analyze the term (iv) in (3.25) $\left|r\left(x,y\right) - r^E\left(x,y\right)\right|$ For fixed $x_0 \in \mathbb{R}^p$ and $y_0 \in \mathcal{Y}$, we have

$$r\left(x_0,y_0\right) = \mathbb{E}(d^2(Y,y_0)|X = x_0) = \int_{\mathcal{Y}} d^2(y,y_0)\frac{\rho(x_0,y)}{\rho_X(x_0)}dy \tag{3.39}$$

and

$$\begin{aligned} r^E\left(x_0,y_0\right) &= \mathbb{E}(d^2(Y,y_0)|X\in\pi_n[x_0]) \\ &= \int_{\mathcal{Y}} d^2(y,y_0)\left(\int_{\pi_n[x_0]}\frac{\rho(x,y)}{\mathbb{P}(X\in\pi_n[x_0])}dx\right)dy \\ &= \int_{\pi_n[x_0]\times\mathcal{Y}} d^2(y,y_0)\rho(x,y)dxdy \times \frac{1}{\mathbb{P}(X\in\pi_n[x_0])} \end{aligned} \tag{3.40}$$

Moreover,

$$\int_{\pi_n[x_0]}\frac{\rho(x,y)}{\mathbb{P}(X\in\pi_n[x_0])}dx = \frac{\int_{\mathbb{R}}\mathbb{1}\left\{x\in\pi_n[x_0]\right\}\rho(x,y)dx}{\int_{\mathbb{R}}\mathbb{1}\left\{x\in\pi_n[x_0]\right\}\rho_X(x)dx} \tag{3.41}$$

Since ρ is uniformly continuous, for every $(x_0,y) \in \mathbb{R}^p\times\mathcal{Y}$, $\forall\epsilon > 0, \exists\delta^1_\epsilon > 0$ such that

$||x_0 - x|| \leq \delta_\epsilon^1 \Rightarrow |\rho(x_0, y) - \rho(x, y)| \leq \epsilon$. Thus, there exists $\delta_\epsilon^1 > 0$ such that

$$\begin{aligned}
&\left| \int_{\mathbb{R}} \mathbb{1}\left\{x \in \pi_n[x_0]\right\} (\rho(x, y) - \rho(x_0, y)) dx \right| \\
&\leq \int_{\pi_n[x_0]} |\rho(x, y) - \rho(x_0, y)| dx \\
&= \int_{B(x_0, \delta_\epsilon^1) \cap \pi_n[x_0]} |\rho(x, y) - \rho(x_0, y)| dx + \int_{\pi_n[x_0]/B(x_0, \delta_\epsilon^1)} |\rho(x, y) - \rho(x_0, y)| dx \\
&\leq \epsilon \operatorname{Vol}\left(\pi_n[x_0] \cap B(x_0, \delta_\epsilon^1)\right) + 2||\rho(., y)||_\infty \operatorname{Vol}\left(\pi_n[x_0] \backslash B(x_0, \delta_\epsilon^1)\right) \\
&\leq \epsilon \operatorname{Vol}\left(\pi_n[x_0]\right) + 2||\rho(., y)||_\infty \operatorname{Vol}\left(\pi_n[x_0] \backslash B(x_0, \delta_\epsilon^1)\right) \qquad (3.42)
\end{aligned}$$

Using the same argument of continuity on the density ρ_X, for all ϵ, there is δ_ϵ^2 such that

$$\int_{\mathbb{R}} \mathbb{1}\left\{x \in \pi_n[x_0]\right\} (\rho_X(x) - \rho_X(x_0)) \, dx \leq \epsilon \operatorname{Vol}(\pi_n[x_0]) + 2||\rho_X||_\infty \operatorname{Vol}\left(\pi_n[x_0] \backslash B(x_0, \delta_\epsilon^2)\right) \qquad (3.43)$$

We define $\delta_\epsilon = \min(\delta_\epsilon^1, \delta_\epsilon^2)$. We will apply the dominated convergence theorem to conclude. To this end, we remark that for every sequence of functions $(f_n)_n$, $(g_n)_n$ and for every functions f and g we have

$$\begin{aligned}
\left| \frac{f_n}{g_n} - \frac{f}{g} \right| &= \left| \frac{f_n}{g_n} - \frac{f}{g_n} + \frac{f}{g_n} - \frac{f}{g} \right| \\
&= \left| \frac{f_n - f}{g_n} - f \frac{g - g_n}{g g_n} \right| \\
&\leq \frac{|f_n - f|}{g_n} + f \frac{|g - g_n|}{g g_n}
\end{aligned}$$

We take

$$f_n(x_0, y) = \int_{\pi_n[x_0]} \rho(x, y) dx; \qquad f(x_0, y) = \rho(x_0, y);$$

$$g_n(x_0) = \int_{\pi_n[x_0]} \rho_X(x) dx; \qquad g(x_0) = \rho_X(x_0).$$

We deduce the following upper bound

$$\begin{aligned}\frac{|f_n - f|}{g_n} &= \frac{\left|\int_{\pi_n[x_0]} \rho(x,y)dx - \rho(x_0,y)\right|}{\int_{\pi_n[x_0]} \rho_X(x)dx} \\ &\leq \frac{\left|\int_{\pi_n[x_0]} \rho(x,y)dx - \rho(x_0,y)\right|}{\rho_{\min} \operatorname{Vol}(\pi_n[x_0])} \\ &\leq \frac{\epsilon \operatorname{Vol}(\pi_n[x_0]) + 2||\rho(.,y)||_\infty \operatorname{Vol}\left(\pi_n[x_0] \backslash B(x_0,\delta_\epsilon)\right)}{\rho_{\min} \operatorname{Vol}(\pi_n[x_0])} \quad \text{using (3.42)} \\ &= \frac{\epsilon}{\rho_{\min}} + \frac{2||\rho(.,y)||_\infty \operatorname{Vol}\left(\pi_n[x_0] \backslash B(x_0,\delta_\epsilon)\right)}{\rho_{\min} \operatorname{Vol}(\pi_n[x_0])} \end{aligned} \tag{3.44}$$

with the same arguments we also get using (3.43)

$$\frac{|g_n - g|}{g_n} = \frac{\left|\int_{\pi_n[x_0]} \rho_X(x)dx - \rho_X(x_0)\right|}{\int_{\pi_n[x_0]} \rho_X(x)dx} \leq \frac{\epsilon}{\rho_{\min}} + \frac{2||\rho_X||_\infty \operatorname{Vol}\left(\pi_n[x_0] \backslash B(x_0,\delta_\epsilon)\right)}{\rho_{\min} \operatorname{Vol}(\pi_n[x_0])} \tag{3.45}$$

From the assumptions of Theorem 1, we have that $\operatorname{diam}(\pi_n[x_0])$ converges towards zero almost surely. Hence, with probability 1, for every δ_ϵ, there is $N_\epsilon > 0$ such that for every $n \geq N_\epsilon$, $\operatorname{diam}(\pi_n[x_0]) \leq \delta_\epsilon/2$. Thus, for every $n \geq N_\epsilon$, $\operatorname{Vol}(\pi_n(x_0) \backslash B(x_0,\delta_\epsilon)) = 0$ almost surely. Then from (3.44) and (3.45) we deduce that for every $n \geq N_\epsilon$ the following inequalities hold almost surely

$$\frac{|f_n - f|}{g_n} \leq \frac{\epsilon}{\rho_{\min}} \quad \text{and} \quad \frac{|g_n - g|}{g_n} \leq \frac{\epsilon}{\rho_{\min}} \tag{3.46}$$

Finally, we deduce from (3.46)

$$\left|\frac{f_n}{g_n} - \frac{f}{g}\right| \leq \frac{|f_n - f|}{g_n} + f\frac{|g - g_n|}{g g_n} \leq \frac{(f+g)\epsilon}{g\rho_{\min}} \qquad \text{a.s} \tag{3.47}$$

Moreover

$$\left|\frac{f_n}{g_n}\right| \leq \frac{||\rho||_\infty \operatorname{Vol}(\pi_n[x])}{\rho_{\min} \operatorname{Vol}(\pi_n[x])} = \frac{||\rho||_\infty}{\rho_{\min}} < \infty \quad \text{from } \mathbf{P1} \tag{3.48}$$

Using the dominated convergence theorem we thus get

$$\lim_{n \to +\infty} r^E(x_0, y_0) = \int_{\mathcal{Y}} d^2(\omega, y_0) \frac{\rho(x_0,\omega)}{\rho(x_0)} d\omega = r\left(x_0, y_0\right) \qquad \text{with probability 1.} \tag{3.49}$$

Finally, we demonstrate the weak consistency given by (3.14). The proof uses the arguments from (Hein, 2009). Under the assumptions of Theorem 1, for every $x \in \mathbb{R}^p$, one has that,

$\lim_{n\to\infty} r\left(x, T_n(x)\right) = r\left(x, \phi^*(x)\right)$ almost surely. Now, remark that

$$R(T_n) - R(\phi^*) \leq \mathbb{E}\left(\left|r\left(X, T_n(X)\right) - r\left(X, \phi^*(X)\right)\right|\right).$$

As $\operatorname{diam}(\mathcal{Y}) < \infty$, we have that $\mathbb{E}\left(r\left(X, T_n(X)\right)\right) < +\infty$ and $\mathbb{E}\left(r\left(X, \phi^*(X)\right)\right) < +\infty$. Therefore, an extension of the dominated convergence theorem given in (Glick, 1974) allows to conclude.

Chapitre 4

Conclusion et perspectives

Dans cette thèse, nous avons apporté deux solutions très différentes pour répondre à la problématique de l'analyse de données longitudinales de grande dimension par forêts aléatoires.

4.1 Approche par modèles mixtes

La première méthode proposée s'inscrit dans la directe continuité des travaux débutés par Hajjem et al. (2011) et Sela and Simonoff (2012). Dans ce travail nous avons proposé à la fois une extension des méthodes de Hajjem et al. (2014), en y introduisant un processus stochastique, ainsi qu'une extension de la méthode introduite par Sela and Simonoff (2012) au cadre des forêts aléatoires. Les méthodes déjà existantes ainsi que celles introduites ont toutes été codées dans un paquet R appelé `LongituRF`. Dans ce paquet, nous avons laissé une liberté totale à l'utilisateur pour choisir le processus stochastique utilisé. Il peut définir lui-même la fonction de variance-covariance du processus stochastique qu'il veut intégrer. Il est également possible de ne pas en mettre. Ce paquet R permettra à la communauté de tester les différentes méthodes sur les données longitudinales dont ils disposent.

Nous avons comparé les différentes approches entres elles sur un ensemble de simulations inspirées du schéma de l'essai vaccinal DALIA-I. Nous avons ainsi pu ainsi mettre en évidence la supériorité de la méthode introduite. A l'instar de Genuer et al. (2008) pour les forêts aléatoires classiques, nous avons montré que le paramètre du `mtry` était crucial pour la convergence de ces algorithmes vers un maximum de la vraisemblance. Comme dans

toutes les approches basées sur un modèle, une question essentielle est de savoir comment se comportent les méthodes lorsque le modèle est mal spécifié, c'est-à-dire déterminer l'impact de la mauvaise spécification sur les performances des méthodes autant en termes de capacité prédictive que de sélection de variables. Dans nos simulations, nous avons montré que les méthodes introduites sont stables à la mauvaise spécification du processus stochastique dans tous les cas de figure considérés. Leur robustesse en fait donc des outils mathématiques de choix.

Enfin, nous avons utilisé les méthodes développées pour analyser les données de l'essai DALIA-I sur la phase post traitement. Nous avons mis en évidence l'avantage de pouvoir utiliser un processus stochastique. En effet, dans le cadre de la grande dimension, on ne connait a priori pas les variables susceptibles d'être en effets aléatoires. Usuellement, on met chaque temps de mesure en effet aléatoires. Cependant, sur des bases de données avec très peu de patients, ce qui est le cas dans DALIA-I, et un nombre de temps de mesures relativement élevé, il n'est pas raisonnable de procéder de la sorte. Une solution à ce problème est d'utiliser un intercept aléatoire auquel on ajoute un processus stochastique pour capter les variations individuelles non captées par l'intercept. Les gènes sélectionnés par la méthode VSURF de Genuer et al. (2015) se sont révélés être totalement en adéquation avec la littérature.

4.2 Approche métrique

Dans la deuxième approche, nous avons changé de cadre de régression en nous plaçant dans des espaces métriques. La principale difficulté dans ces espaces est l'absence potentielle de relation d'ordre, et donc, par extension de construire des arbres CART sur ces derniers. Afin de contourner ce problème, nous avons utilisé la notion de fonction de *split* qui permet de proposer des découpages sur les différents espaces d'entrées. Puis, en faisant usage des notions de moyenne et de variance de Fréchet, nous avons par suite modifié le critère de découpage à ces espaces. Nous avons ensuite, à partir de ces notions très simples, introduit les arbres de Fréchet qui peuvent être vus comme l'adaptation naturelle des arbres CART aux espaces métriques non ordonnés. A partir des arbres de Fréchet, nous avons introduit les forêts aléatoires de Fréchet.

Nous avons mis en exergue comment ces nouvelles méthodes pouvaient être utilisées dans le cadre des données longitudinales. En considérant les observations provenant d'un même

individu comme une courbe discrète, on a montré qu'il était possible d'analyser des jeux de données où la variable explicative est observée dans une fenêtre de temps disjointe de la fenêtre des temps d'observation des variables explicatives. En effet, dans ce paradigme, une observation pour une variable donnée n'est pas une mesure à un temps donné pour cette variable mais bien toute sa trajectoire pour un individu donné. Il n'est donc plus question d'apparier des mesures singulières entre elles comme pouvaient le faire les méthodes classiques d'apprentissage comme les forêts aléatoires de Breiman mais bien des courbes entières. Cela nous permet de considérer des jeux de données dans lesquels toutes les variables sont observées sur des fenêtres différentes.

Comme toutes les méthodes dépendantes de la notion de distance, il est nécessaire d'équiper nos espaces de courbes d'une métrique. Dans le cadre particulier de la régression sur trajectoires discrètes, nous avons considérée la distance de Fréchet. Cette dernière est intégralement basée sur la proximité en forme et non sur la proximité en temps (c'est-à-dire sur une distance temps-par-temps) entre les courbes. Cette métrique, utilisée dans les travaux de Genolini et al. (2016) pour adapter la méthode des k-means au clustering de trajectoires à permis de mettre en évidence de nouveaux groupes de trajectoires qui n'étaient pas trouvés avec les distances Euclidiennes. A cela s'ajoute le fait qu'elle permet de calculer la distance entre deux courbes qui seraient mesurées à des temps d'observation différentes. Ainsi, l'utilisation de la distance de Fréchet sur les espaces de courbes permet encore d'augmenter la flexibilité de nos arbres et forêts aléatoires de Fréchet. En effet, ces dernières peuvent être utilisées dans des études longitudinales où, à la fois, les variables ont toutes des fenêtres d'observations disjointes et où toutes les courbes d'une même variable sont observées à des temps différents. Ajoutons à cela que le fait de considérer les observations d'un même individu comme une courbe discrète permet une certaine gestion des données manquantes. Dans ce cadre, une donnée manquante est une "courbe non observée" *i.e.* l'absence totale d'observations pour une variable et un individu donnés. Si quelques temps d'observations sont manquants mais que l'on a toujours des observations pour cette variable et cet individu à plusieurs temps de mesures, on peut alors toujours calculer sa distance de Fréchet aux autres trajectoires. Nous avons, dans une étude de simulations, illustré la résilience des forêts aléatoires de Fréchet aux données manquantes dans le cadre des données longitudinales. Toutes ces caractéristiques font des arbres et forêts aléatoires une méthode d'apprentissage extrêmement flexible qui peut être utilisée dans n'importe quel design longitudinal.

Dans notre étude de simulations, nous avons exposé la supériorité des performances prédictives des arbres et forêts aléatoires de Fréchet par rapport à la méthode de Boosting fonctionnel introduite par Brockhaus et al. (2017) (seul compétiteur à notre connaissance sur de la régression fonction-fonction). Nous avons ensuite montré la très grande flexibilité de notre approche par sa résilience aux décalages temporels ainsi qu'aux données manquantes. Nous avons enfin mis en lumière que le calcul de l'importance des variables associé à cette nouvelle méthode de forêt permet bien de retrouver les variables impliquées dans l'explication de la variable de sortie.

Nous avons appliqué la méthode des forêts aléatoires de Fréchet à l'essai vaccinal DALIA-I. Pour la première fois nous avons pu mettre en relation l'intégralité des trajectoires de gènes pendant la phase de traitement avec l'intégralité de la concentration plasmatique après l'interruption du traitement. Malgré le très faible nombre de patients, les prédictions OOB s'avèrent très proches des véritables trajectoires de concentration plasmatique. Il apparaît aujourd'hui possible de prédire l'intégralité de la charge plasmatique après l'arrêt du traitement, seulement à partir du comportement des gènes pendant le traitement. Sur cet exemple nous avons aussi pu isoler, à partir de l'importance des variables, les gènes qui expliquent au mieux par leur trajectoires pendant le traitement les variations futures de la charge virale. Les gènes sélectionnés se sont avérés être en adéquation avec la littérature.

Le deuxième jeu de données que nous avons analysé avec la méthode des forêts aléatoires de Fréchet est l'essai vaccinal LIGHT. Ce dernier représentait un défi différent : le plan d'experience était extrêmement déséquilibré et les trajectoires individuelles étaient composées que de très peu d'observations. Nous avons également extrait un ensemble de gènes déjà présent dans la littérature.

L'analyse des données longitudinales, par une régression sur des courbes, est un exemple d'utilisation des méthodes par arbres et forêts aléatoires de Fréchet. Cependant, ces méthodes s'inscrivent dans un cadre plus général encore. En effet, les découpages étant tous comparés sur la même variable réponse, les différentes variables peuvent toutes être de natures différentes. Nous avons illustré dans deux schémas de simulations cette capacité à construire des prédicteurs performants sur des données hétérogènes. Dans le premier schéma, nous avons montré qu'il était possible de prédire des courbes à partir d'images, de courbes et de scalaires. Dans un deuxième schéma, nous avons montré qu'il était aussi possible de prédire des images à partir de courbes. L'intégration de données hétérogènes ne s'arrête pas

à la seule construction de prédicteurs efficaces mais aussi à la comparaison des variables de natures différentes par le calcul de l'importance des variables qui est toujours possible dans ce cadre.

En nous inspirant des idées de Geurts et al. (2006), nous avons introduit une variante de nos forêts aléatoires de Fréchet, appelée "*Extremely randomized Fréchet random forests*". Cette méthode possède deux avantages. Le premier est que la construction de la forêt ne nécessite que de pouvoir calculer une moyenne de Fréchet sur l'espace de la variable à expliquer. Le deuxième est qu'elle permet une accélération drastique des temps de calculs tout en conservant d'excellentes capacités prédictives.

Nous avons obtenu un premier résultat de consistance sur des prédicteurs par regressogramme à valeurs dans un espace métrique de diamètre et de dimension de doublement finis sur des partitions de $\mathbb{R}^p$. Les hypothèses de ce théorème sont celles classiquement retrouvées dans les travaux théoriques sur les arbres et forêts aléatoires (c.f. Nobel et al. (1996), Haghiri et al. (2018)). Ce résultat a été appliqué au cas des arbres de Fréchet purement uniformément aléatoires qui sont une adaptation des arbres de Genuer (2012) lorsque la sortie prend ses valeurs dans un espace métrique.

Enfin, nous avons implémenté toutes les méthodes d'arbres et de forêts aléatoires de Fréchet dans un paquet R appelé `FrechForest`. Ce paquet permet à l'utilisateur d'appliquer ces méthodes sur des images, des courbes, des formes, des scalaires et des facteurs, que ce soit en variables explicatives ou en variable à expliquer. Dans ce paquet, tous les types de variables supportés peuvent être intégrés conjointement. Il est tout à fait possible de prédire des courbes de lacets jetés sur une table à partir de formes d'oeufs et d'images de villes en même temps. Néanmoins, nous ne garantissons pas les performances de l'algorithme sur cet exemple particulier. Nous avons aussi intégralement implémenté le calcul de l'importance des variables pour tous les types de données supportés. Toutes les procédures de construction des arbres ainsi que le calcul des scores d'importance des variables ont été parallélisées dans ce paquet R.

4.3 Perspectives

En ce qui concerne la première approche, nous sommes allés au bout d'une logique débutée par Hajjem et al. (2011) et Sela and Simonoff (2012). Il nous semble avoir donné toutes

les informations pratiques au sujet de ces méthodes. L'analyse théorique de ces algorithmes itératifs restent cependant un problème ouvert.

Pour la deuxième approche, l'élaboration des arbres et forêts aléatoires de Fréchet nous a amenés à considérer de nouveaux problèmes d'analyse de données. Notre étude de simulations a mis en évidence la capacité de ces dernières à bien gérer les données hétérogènes. Il serait intéressant de tester ces nouvelles méthodes sur des jeux de données réelles complexes qui peuvent coupler images, courbes, formes, scalaires et facteurs. Le domaine médical semble le pourvoyeur de données hétérogènes toutes désignées. On peut par exemple, penser à coupler de l'imagerie cérébrale avec des courbes d'activité cérébrale (EEG), ainsi que des caractéristiques individuelles telles que la glycémie, le cholestérol, etc. Plus généralement, la méthode des forêts aléatoires de Fréchet est un nouvel outil qui répond à un véritable besoin d'intégration des données en santé. Une perspective est alors de l'appliquer à de nouveaux jeux de données, comme par exemple à l'analyse de courbes épidémiques.
Un pan de nos développements futurs concerne le paquet `FrechForest`. Dans ce paquet, contrairement aux courbes, la métrique utilisée pour les images est la distance Euclidienne. Il paraît évident que cette distance n'est pas optimale pour traiter ce genre de données. La distance de Haussdorf (Huttenlocher et al., 1993) semblerait bien plus adaptée à l'analyse d'images. Cependant, cette métrique est très coûteuse à calculer (en prenant en compte le fait qu'elle doit être calculée un très grand nombre de fois). Un développement futur est alors l'intégration optimale de cette distance (ou d'autres métriques comme la distance de Wasserstein), en passant par exemple par un langage de plus bas niveau et/ou une adaptation de la méthode sur GPU pour accelérer la méthode avec une parallélisation massive.

Enfin, Scornet et al. (2015) nous ont prouvé qu'il était possible d'analyser ces mastodontes théoriques que sont les forêts aléatoires. L'analyse théorique des arbres et forêts aléatoires de Fréchet reste un défi stimulant qu'il convient de relever. La principale difficulté réside dans le fait de considérer un espace métrique à la fois en entrée et en sortie. Dans ce sens, le travail de Hein (2009) montre que le cadre de l'analyse sur des variétés Riemanniennes semble le plus adapté.

De par son importante versatilité, les arbres et les forêts de Fréchet ouvrent un champ

des possibles non négligeable dans le domaine de l'intégration des données. En définitive, il semble que les limites des forêts aléatoires de Fréchet soient nos propres limites de conception des objets d'étude.

Annexe A

Valorisations scientifiques

A.1 Publications scientifiques

— **Capitaine L**, Genuer R, Thiébaut R. Random forests for high-dimensional longitudinal data. *Statistical Methods in Medical Research*. August 2020. https://doi:10.1177/0962280220946080

— **Capitaine L**, Bigot J, Thiébaut R, Genuer R. Fréchet random forests for metric space valued regression with non euclidean predictors. Submitted.

A.2 Communications orales

— **Capitaine L**, Bigot J, Thiébaut R, Genuer R. Fréchet random forests. *13th International Conference of the ERCIM WG on Computational and Methodological Statistics (CMStatistics 2020)*, conférence virtuelle, (A venir) Décembre 2020.

— **Capitaine L**, Thiébaut R, Genuer R. Fréchet random forests. *12th International Conference of the ERCIM WG on Computational and Methodological Statistics (CMStatistics 2019)*, Londres, Royaume-Uni, Décembre 2019.

— **Capitaine L**, Thiébaut R, Genuer R. Arbres et forêts aléatoires de Fréchet. *51èmes Journées de Statistique de la Société Française de Statistique (SFdS)*, Nancy, France, Juin 2019.

— **Capitaine L**, Thiébaut R, Genuer R. Random forests for high-dimensional longitudinal data. *29th IBS Conference*, Barcelone, Espagne, Juillet 2018.

— **Capitaine L**, Thiébaut R, Genuer R. Forêts aléatoires pour données longitudinales de grande dimension. *50èmes Journées de Statistique de la Société Française de Statistique (SFdS)*, Paris, France, Mai 2018.

A.3 Paquets R

— `LongituRF` : un paquet R pour l'analyse de données longitudinales de grande dimension par forêts aléatoires. Disponible sur le CRAN, version de développement sur GitHub.

— `FrechForest` : un paquet R pour l'analyse de données hétérogènes par forêts aléatoires de Fréchet. Version de développement sur GitHub.

Bibliographie

Alam, M. S. and Vuong, S. T. (2013). Random forest classification for detecting android malware. In *2013 IEEE international conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing*, pages 663–669.

Alt, H. and Godeau, M. (1995). Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 05 :75–91.

Anderson, T. (2008). *The theory and practice of online learning*. Athabasca University Press.

Arlot, S. and Genuer, R. (2014). Analysis of purely random forests bias. *arXiv preprint arXiv :1407.3939*.

Arlot, S. and Genuer, R. (2016). Comments on : A random forest guided tour. *Test*, 25(2) :228–238.

Arlot, S. and Lerasle, M. (2016). Choice of v for v-fold cross-validation in least-squares density estimation. *The Journal of Machine Learning Research*, 17(1) :7256–7305.

Audigier, V., Husson, F., and Josse, J. (2016). A principal component method to impute missing values for mixed data. *Advances in Data Analysis and Classification*, 10(1) :5–26.

Bellman, R. E. (2015). *Adaptive control processes : a guided tour*, volume 2045. Princeton university press.

Biau, G. (2012a). Analysis of a random forests model. *The Journal of Machine Learning Research*, 13(1) :1063–1095.

Biau, G. (2012b). Analysis of a random forests model. *J. Mach. Learn. Res.*, 13(1) :1063–1095.

Biau, G., Devroye, L., and Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *The Journal of Machine Learning Research*, 9(66) :2015–2033.

Biau, G. and Scornet, E. (2016). A random forest guided tour. *Test*, 25(2) :197–227.

Bigot, J. (2013). Fréchet means of curves for signal averaging and application to ECG data analysis. *The Annals of Applied Statistics*, 7(4) :2384–2401.

Blackwell, D. and Maitra, A. (1984). Factorization of probability measures and absolutely measurable sets. *Proceedings of the American Mathematical Society*, 92(2) :251–254.

Booth, A., Gerding, E., and McGroarty, F. (2015). Performance-weighted ensembles of random forests for predicting price impact. *Quantitative Finance*, 15(11) :1823–1835.

Bosch, A., Zisserman, A., and Munoz, X. (2007). Image classification using random forests and ferns. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8.

Bosinger, S. E., Li, Q., Gordon, S. N., Klatt, N. R., Duan, L., Xu, L., Francella, N., Sidahmed, A., Smith, A. J., Cramer, E. M., et al. (2009). Global genomic analysis reveals rapid control of a robust innate response in siv-infected sooty mangabeys. *The Journal of clinical investigation*, 119(12) :3556–3572.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2) :123–140.

Breiman, L. (2000a). Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40(3) :229–242.

Breiman, L. (2000b). Some infinity theory for predictor ensembles. Technical report, Technical Report 579, Statistics Dept. UCB.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1) :5–32.

Breiman, L. and Cutler, A. (2005). Random forests. *Software available at : http ://stat-www. berkeley. edu/users/breiman/RandomForests.*

Breiman, L., Friedman, J., Olshen, R., and Stone, Charles, J. (1984). *Classification and Regression Trees.* Chapman & Hall, New York.

Brockhaus, S., Melcher, M., Leisch, F., and Greven, S. (2017). Boosting flexible functional regression models with a high number of functional historical effects. *Statistics and Computing*, 27(4) :913–926.

Calhoun, P., Levine, R. A., and Fan, J. (2020). Repeated measures random forests (RMRF) : Identifying factors associated with nocturnal hypoglycemia. *Biometrics*, pages 1–9.

Caruana, R., Karampatziakis, N., and Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, page 96–103.

Casanova, R., Saldana, S., Chew, E. Y., Danis, R. P., Greven, C. M., and Ambrosius, W. T. (2014). Application of random forests methods to diabetic retinopathy classification analyses. *PLOS ONE*, 9(6) :1–8.

Cazelles, E., Seguy, V., Bigot, J., Cuturi, M., and Papadakis, N. (2018). Log-PCA versus Geodesic PCA of histograms in the Wasserstein space. *SIAM Journal on Scientific Computing*, 40(2) :B429–B456.

Chaussabel, D. and Baldwin, N. (2014). Democratizing systems immunology with modular transcriptional repertoire analyses. *Nature Reviews Immunology*, 14(4) :271.

Chen, X. and Ishwaran, H. (2012). Random forests for genomic data analysis. *Genomics*, 99(6) :323–329.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian cart model search. *Journal of the American Statistical Association*, 93(443) :935–948.

Commenges, D. and Jacqmin-Gadda, H. (2015). *Modèles biostatistiques pour l'épidémiologie*. De Boeck Superieur.

Cutler, D. R., Edwards, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., and Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, 88(11) :2783–2792.

Dai, X. and Muller, H.-G. (2018). Principal component analysis for functional data on riemannian manifolds and spheres. *The Annals of Statistics*, 46 :3334–3361.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society : Series B (Methodological)*, 39(1) :1–22.

Devroye, L., Györfi, L., and Lugosi, G. (2013). *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media.

Díaz-Uriarte, R. and Alvarez De Andres, S. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1) :3.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15.

Diggle, P. J. and Hutchinson, M. F. (1989). On spline smoothing with autocorrelated errors. *Australian & New Zealand Journal of Statistics*, 31(1) :166–182.

Donoho, D. L. et al. (2000). High-dimensional data analysis : The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000) :32.

Eo, S.-H. and Cho, H. (2014). Tree-structured mixed-effects regression modeling for longitudinal data. *Journal of Computational and Graphical Statistics*, 23(3) :740–760.

Feelders, A. (1999). *Handling Missing Data in Trees : Surrogate Splits or Statistical Imputation ?*

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems ? *The Journal of Machine Learning Research*, 15(1) :3133–3181.

Firth, D. (1988). Multiplicative errors : Log-normal or gamma ? *Journal of the Royal Statistical Society : Series B (Methodological)*, 50(2) :266–268.

Fletcher, P. T., Conglin Lu, Pizer, S. M., and Sarang Joshi (2004). Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8) :995–1005.

France, S. L., Douglas Carroll, J., and Xiong, H. (2012). Distance metrics for high dimensional nearest neighborhood recovery : Compression and normalization. *Information Sciences*, 184(1) :92 – 110.

Francois, D., Wertz, V., Verleysen, M., et al. (2005). About the locality of kernels in high-dimensional spaces. In *International Symposium on Applied Stochastic Models and Data Analysis*, pages 238–245.

Fréchet, M. (1948). Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310.

Fréchet, M. M. (1906). Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1) :1–72.

Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer.

Fu, W. and Simonoff, J. S. (2015). Unbiased regression trees for longitudinal and clustered data. *Computational Statistics & Data Analysis*, 88 :53 – 74.

Gauthier, M., Agniel, D., Thiébaut, R., and Hejblum, B. P. (2019). dearseq : a variance component score test for rna-seq differential analysis that effectively controls the false discovery rate. *bioRxiv*.

Genolini, C., Ecochard, R., Benghezal, M., Driss, T., Andrieu, S., and Subtil, F. (2016). kmlshape : An efficient method to cluster longitudinal data (time-series) according to their shapes. *PLOS ONE*, 11(6) :1–24.

Genuer, R. (2012). Variance reduction in purely random forests. *Journal of Nonparametric Statistics*, 24(3) :543–562.

Genuer, R. and Poggi, J.-M. (2017). Arbres CART et Forêts aléatoires, Importance et sélection de variables. hal-*01387654*.

Genuer, R. and Poggi, J.-M. (2019). *Les forêts aléatoires avec R*. Presses universitaires de Rennes.

Genuer, R., Poggi, J.-M., and Tuleau, C. (2008). Random forests : some methodological insights. *arXiv preprint arXiv :0811.3619*.

Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14) :2225–2236.

Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2015). VSURF : an R package for variable selection using random forests. *The R Journal*, 7(2) :19–33.

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

Gey, S. and Nedelec, E. (2005). Model selection for cart regression trees. In *IEEE Transactions on Information Theory*, volume 51, pages 658–670.

Giraud, C. (2014). *Introduction to high-dimensional statistics*, volume 138. CRC Press.

Glick, N. (1974). Consistency conditions for probability estimators and integrals of density estimators. *Utilitas Mathematica*, 6 :61–74.

Goldstein, B. A., Hubbard, A. E., Cutler, A., and Barcellos, L. F. (2010). An application of random forests to a genome-wide association dataset : methodological considerations & new findings. *BMC genetics*, 11(1) :49.

Gottlieb, L.-A., Kontorovich, A., and Krauthgamer, R. (2016). Adaptive metric dimensionality reduction. *Theoretical Computer Science*, 620 :105 – 118.

Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. (2006). *A distribution-free theory of nonparametric regression*. Springer Science & Business Media.

Haghiri, S., Garreau, D., and Von-Luxburg, U. (2018). Comparison-based random forests. In *International Conference on Machine Learning*, pages 1866–1875.

Hajjem, A., Bellavance, F., and Larocque, D. (2011). Mixed effects regression trees for clustered data. *Statistics & probability letters*, 81(4) :451–459.

Hajjem, A., Bellavance, F., and Larocque, D. (2014). Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation*, 84(6) :1313–1328.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning : data mining, inference, and prediction*. Springer Science & Business Media.

Hein, M. (2009). Robust nonparametric regression with metric-space valued output. In *Advances in Neural Information Processing Systems*, pages 718–726.

Hejblum, B. P., Skinner, J., and Thiébaut, R. (2015). Time-course gene set analysis for longitudinal gene expression data. *PLOS Computational Biology*, 11(6) :e1004310.

Hothorn, T., Bühlmann, P., Dudoit, S., Molinaro, A., and Van Der Laan, M. J. (2005). Survival ensembles. *Biostatistics*, 7(3) :355–373.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning : A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3) :651–674.

Husson, F., Josse, J., Narasimhan, B., and Robin, G. (2019). Imputation of mixed data with multilevel singular value decomposition. *Journal of Computational and Graphical Statistics*, 28(3) :552–566.

Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J. (1993). Comparing images using the hausdorff distance. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15, pages 850–863.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., Lauer, M. S., et al. (2008). Random survival forests. *The Annals of Applied Statistics*, 2(3) :841–860.

Ishwaran, H., Kogalur, U. B., Gorodeski, E. Z., Minn, A. J., and Lauer, M. S. (2010). High-dimensional variable selection for survival data. *Journal of the American Statistical Association*, 105(489) :205–217.

Koerner, T. K. and Zhang, Y. (2017). Application of linear mixed-effects models in human neuroscience research : A comparison with pearson correlation in two auditory electrophysiology studies. *Brain Sciences*, 7(3).

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conferences on Artificial Intelligence*, volume 14, pages 1137–1145. Montreal, Canada.

Kundu, M. G. and Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. *Biostatistics & Epidemiology*, 3(1) :1–22.

Kwong, S., He, Q., Man, K.-F., Tang, K., and Chau, C. (1998). Parallel genetic-based hybrid pattern matching algorithm for isolated word recognition. In *International Journal of Pattern Recognition and Artificial Intelligence*, volume 12, pages 573–594.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, 38(4) :963–974.

Lakshminarayanan, B., Roy, D. M., and Teh, Y. W. (2014). Mondrian forests : Efficient online random forests. In *Advances in Neural Information Processing Systems 27*, pages 3140–3148.

Le Gouic, T. and Loubes, J.-M. (2017). Existence and consistency of Wasserstein barycenters. *Probability Theory and Related Fields*, 168 pages 901-917.

LeCun, Y. and Cortes, C. (2010). The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

Lévy, Y., Thiébaut, R., Montes, M., Lacabaratz, C., Sloan, L., King, B., Pérusat, S., Harrod, C., Cobb, A., Roberts, L. K., et al. (2014). Dendritic cell-based therapeutic vaccine elicits polyfunctional hiv-specific t-cell immunity associated with control of viral load. *European journal of immunology*, 44(9) :2802–2810.

Li, D., Deogun, J., Spaulding, W., and Shuart, B. (2004). Towards missing data imputation : A study of fuzzy k-means clustering method. In *Rough Sets and Current Trends in Computing*, pages 573–579.

Liaw, A. and Wiener, M. (2002). The randomforest package. *R news*, 2(3) :18–22.

Lin, J. (1991). Divergence measures based on the shannon entropy. In *IEEE Transactions on Information Theory*, volume 37, pages 145–151.

Linero, A. R. (2018). Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522) :626–636.

Lugosi, G. and Nobel, A. (1996). Consistency of data-driven histogram methods for density estimation and classification. *The Annals of Statistics*, 24(2) :687–706.

McLachlan, G. J. and Krishnan, T. (1997). *The EM Algorithm and Extensions*. John Wiley & Sons.

Mentch, L. and Hooker, G. (2016). Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *The Journal of Machine Learning Research*, 17(1) :841–881.

Menze, B. H., Kelm, B. M., Splitthoff, D. N., Koethe, U., and Hamprecht, F. A. (2011). On oblique random forests. In *Machine Learning and Knowledge Discovery in Databases*, pages 453–469.

Mourtada, J., Gaïffas, S., and Scornet, E. (2020). Minimax optimal rates for mondrian trees and forests. *The Annals of Statistics*, 48(4) :2253–2276.

Nobel, A. et al. (1996). Histogram regression estimation using data-dependent partitions. *The Annals of Statistics*, 24(3) :1084–1105.

Noroozi, F., Sapiński, T., Kamińska, D., and Anbarjafari, G. (2017). Vocal-based emotion recognition using random forests and decision tree. *International Journal of Speech Technology*, 20(2) :239–246.

Nye, T. M. W., Tang, X., Weyenberg, G., and Yoshida, R. (2017). Principal component analysis and the locus of the Fréchet mean in the space of phylogenetic trees. *Biometrika*, 104(4) :901–922.

Petersen, A. and Müller, H.-G. (2019). Fréchet regression for random objects with euclidean predictors. *The Annals of Statistics*, 47(2) :691–719.

Poggi, J.-M. and Tuleau, C. (2006). Classification supervisée en grande dimension. application à l'agrément de conduite automobile. *Revue de Statistique Appliquée*, 54(4) :41–60.

Prasad, A. M., Iverson, L. R., and Liaw, A. (2006). Newer classification and regression tree techniques : bagging and random forests for ecological prediction. *Ecosystems*, 9(2) :181–199.

R Core Team (2019). *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Radovic, A., Williams, M., Rousseau, D., Kagan, M., Bonacorsi, D., Himmel, A., Aurisano, A., Terao, K., and Wongjirad, T. (2018). Machine learning at the energy and intensity frontiers of particle physics. *Nature*, 560(7716) :41–48.

Rodriguez, J. D., Perez, A., and Lozano, J. A. (2010). Sensitivity analysis of k-fold cross validation in prediction error estimation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 32, pages 569–575.

Schafer, J. L. and Olsen, M. K. (1998). Multiple imputation for multivariate missing-data problems : A data analyst's perspective. *Multivariate Behavioral Research*, 33(4) :545–571.

Schrider, D. R. and Kern, A. D. (2018). Supervised machine learning for population genetics : A new paradigm. *Trends in Genetics*, 34(4) :301 – 312.

Scornet, E., Biau, G., and Vert, J.-P. (2015). Consistency of random forests. *The Annals of Statistics*, 43(4) :1716–1741.

Segal, M. R. (1992). Tree-structured methods for longitudinal data. *Journal of the American Statistical Association*, 87(418) :407–418.

Sela, R. J. and Simonoff, J. S. (2012). RE-EM trees : a data mining approach for longitudinal and clustered data. *Machine learning*, 86(2) :169–207.

Sommer, S., Lauze, F., Hauberg, S., and Nielsen, M. (2010). Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In *Computer Vision – ECCV 2010*, pages 43–56.

Steingrimsson, J. A., Diao, L., and Strawderman, R. L. (2019). Censoring unbiased regression trees and ensembles. *Journal of the American Statistical Association*, 114(525) :370–383.

Svetnik, V., Liaw, A., Tong, C., and Wang, T. (2004). Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. In *Multiple Classifier Systems*, pages 334–343.

Thiébaut, R., Hejblum, B. P., Hocini, H., Bonnabau, H., Skinner, J., Montes, M., Lacabaratz, C., Richert, L., Palucka, K., Banchereau, J., and Lévy, Y. (2019). Gene expression signatures associated with immune and virological responses to therapeutic vaccination with dendritic cells in hiv-infected individuals. *Frontiers in Immunology*, 10 :874.

Thiébaut, R., Hejblum, B. P., and Richert, L. (2014). L'analyse des " Big Data " en recherche clinique. *Epidemiology and Public Health / Revue d'Epidémiologie et de Santé Publique*, 62(1) :1–4.

Tin Kam Ho (1998). The random subspace method for constructing decision forests. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20, pages 832–844.

Vallender, S. S. (1974). Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability & Its Applications*, 18(4) :784–786.

Verbeke, G. and Molenberghs, G. (2009). *Linear Mixed Models for Longitudinal Data*. Springer.

Verikas, A., Gelzinis, A., and Bacauskiene, M. (2011). Mining data with random forests : A survey and results of new tests. *Pattern Recognition*, 44(2) :330–349.

Wager, S. (2014). Asymptotic theory for random forests. *arXiv preprint arXiv :1405.0352*.

Wagner, M., Helmer, C., Tzourio, C., Berr, C., Proust-Lima, C., and Samieri, C. (2018). Evaluation of the Concurrent Trajectories of Cardiometabolic Risk Factors in the 14 Years Before Dementia. *JAMA Psychiatry*, 75(10) :1033–1042.

Wei, Y., Liu, L., Su, X., Zhao, L., and Jiang, H. (2020). Precision medicine : Subgroup identification in longitudinal trajectories. *Statistical Methods in Medical Research*, 29(9) :2603–2616.

Wu, H. and Zhang, J.-T. (2006). *Nonparametric regression methods for longitudinal data analysis : mixed-effects modeling approaches*. John Wiley & Sons.

Yoo, P. D., Kim, M. H., and Jan, T. (2005). Machine learning techniques and use of event information for stock market prediction : A survey and evaluation. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, volume 2, pages 835–841.

Zhang, D. and Davidian, M. (2001). Linear mixed models with flexible distributions of random effects for longitudinal data. *Biometrics*, 57(3) :795–802.

Zhang, D., Lin, X., Raz, J., and Sowers, M. (1998). Semiparametric stochastic mixed models for longitudinal data. *Journal of the American Statistical Association*, 93(442) :710–719.

Zheng, J., Gao, X., Zhan, E., and Huang, Z. (2008). Algorithm of on-line handwriting signature verification based on discrete fréchet distance. In *Advances in Computation and Intelligence*, pages 461–469.

Zhu, R., Zeng, D., and Kosorok, M. R. (2015). Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512) :1770–1784.

Forêts aléatoires pour données longitudinales de grande dimension

Résumé : Introduites par Leo Breiman en 2001, les forêts aléatoires sont une méthode d'apprentissage statistique largement utilisée dans de nombreux domaines de recherche scientifiques tant pour sa capacité à décrire des relations complexes entre des variables explicatives et une variable réponse que pour sa faculté à traiter des données de grande dimension. Dans de nombreuses applications en santé, on dispose de mesures répétées au cours du temps. On parle alors de données longitudinales. Les corrélations induites entre les mesures d'un même individu à différents temps doivent être prises en compte, ce qui n'est pas le cas dans la méthode classique des forêts aléatoires. L'objectif de cette thèse est d'adapter cette méthode à l'analyse des données longitudinales dans un contexte de grande dimension. Pour ce faire, deux approches sont proposées. La première s'appuie sur l'utilisation d'un modèle semi-paramétrique à effets mixtes qui permet de prendre en compte la structure de covariance intra-individuelle dans la construction de la forêt aléatoire. Cette méthode a été appliquée à un essai vaccinal contre le VIH et a permis de sélectionner 21 transcrits de gènes pour lesquels l'association avec la charge virale du VIH était en adéquation avec les résultats observés lors de l'infection primaire. La seconde se place dans le cadre plus général de la régression sur des espaces métriques. Dans ce contexte, les données répétées sont traitées comme des courbes. Nous introduisons alors le concept de forêts aléatoires de Fréchet qui permet d'apprendre des relations entre des variables de natures diverses, comme des courbes, des images ou des formes, dans des espaces métriques non ordonnés. Nous décrivons une nouvelle manière de découper les noeuds des arbres constituant la forêt de Fréchet puis nous détaillons la procédure de prédiction pour une variable de sortie à valeurs dans un espace non euclidien. Les notions classiques d'erreur OOB ainsi que d'importance des variables sont adaptées aux forêts aléatoires de Fréchet. Un théorème de consistance pour les régressogrammes de Fréchet utilisant des partitions données-dépendantes est énoncé puis appliqué aux arbres de Fréchet purement uniformément aléatoires. Une étude de simulations est ensuite menée pour étudier le comportement de cette nouvelle méthode dans le cadre de la régression sur courbes, images et scalaires. Enfin, la méthode des forêts aléatoires de Fréchet est appliquée à l'analyse de deux essais vaccinaux de grande dimension sur le VIH.

Mots-clés : Forêts aléatoires, Arbres de régression, Grande dimension, Données longitudinales, Essais vaccinaux, VIH, Génomique, Modèle semi-paramétrique à effets mixtes, Données hétérogènes, Données complexes, Régression non-paramétrique

---

Random forests for high dimensional and longitudinal data

Abstract: Introduced by Leo Breiman in 2001, random forests are a statistical learning method that is widely used in many fields of scientific research both for its ability to describe complex relationships between explanatory variables and a response variable as well as for its ability to handle high dimensional data. In many health applications, repeated measurements over time are available. These are referred to as longitudinal data. The correlations induced by the measurements of the same individual at different times must be taken into account, which is not the case in the classical random forests method. The aim of this thesis is to adapt this method to the analysis of longitudinal data in a high dimensional context. To do so, two approaches are proposed. The first one is based on a semi-parametric mixed-effects model which allows the intra-individual covariance structure to be taken into account in the construction of the random forest. This method was applied to an HIV vaccine trial and enabled to select 21 gene transcripts for which the association with the HIV viral load was in line with the results observed during the primary infection. The second method takes place in the more general framework of regression on metric spaces. In this context, repeated data are treated as curves. We then introduce the concept of Fréchet random forests, which allows to learn relationships between heterogeneous variables, such as curves, images or shapes, in unordered metric spaces. We describe a new way of splitting the nodes of the trees composing the Fréchet random forest and then we detail the prediction procedure for a non-Euclidean output vari. The classical notions of OOB error as well as the variable importance are adapted to the Fréchet random forest. A consistency theorem for Fréchet regressogram predictor using data-dependent partitions is stated and then applied to Fréchet purely uniformly random trees. A simulation study is then carried out to study the behaviour of this new method within the framework of regression on curves, images and scalars. Finally, Fréchet random forest is applied to the analysis of two high dimensional HIV vaccine trials.

Keywords: Random forests, Regression trees, High dimension, Longitudinal data, Vaccine trials, HIV, Genomics, Semi-parametric mixed-effects model, Heterogeneous data, Complex data analysis, Non-parametric regression

---

Discipline : Santé publique – option : Biostatistiques
Laboratoire : Unité INSERM U1219, Bordeaux Population Health center - INRIA - Université de Bordeaux
146 rue Léo Saignat 33076 Bordeaux, FRANCE