UCLouvain

FACULTE DES SCIENCES

Faculté des sciences

Implémentation et exploration du Gradient Boosting pour la modélisation de quantiles conditionnels extrêmes

Auteur : Legendre Simon
Promoteur : Johan Segers
Lecteur : Christian Hafner
Année académique 2022-2023

Abstract

La régression quantile est une méthode statistique qui permet d'explorer les différents quantiles conditionnels d'une variable réponse, offrant ainsi une analyse plus détaillée de sa distribution en relation avec les variables indépendantes comparé à la régression linéaire classique qui offre une image moins exhaustive. La prédiction de quantiles extrêmes s'avère toutefois complexe en raison du faible nombre d'observations disponibles pour modéliser ces valeurs. Pour pallier ce problème, les théories des valeurs extrêmes sont une solution intéressante, offrant la possibilité de modéliser les queues de distributions à l'aide de la distribution généralisée de Pareto. Puisque la forme de cette distribution dépend d'un paramètre de forme et d'un paramètre d'échelle, il est indispensable de les estimer conditionnellement aux variables indépendantes. Ce mémoire propose l'implémentation et l'exploration de l'algorithme Gbex basé sur la méthode du Gradient Boosting, permettant l'estimation de ces paramètres conditionnels afin de prédire les quantiles extrêmes avec précision. Après l'implémentation de cet algorithme, des tests ont été réalisés sur des données simulées et empiriques pour évaluer l'efficacité du Gbex, ainsi que pour le comparer à une autre méthode de prédiction de quantiles : les forêts aléatoires généralisées. Ces analyses ont montré que le Gbex affiche des performances très intéressantes dans les conditions dans lesquelles il a été testé.

Remerciements

Tout d'abord, je voudrais adresser ma gratitude la plus sincère à mon promoteur Johan Segers, qui pendant cette aventure académique m'a été d'un soutien indéfectible. Son investissement, sa disponibilité et sa bienveillance m'ont permis de toujours garder le cap et la motivation durant l'entièreté de ce travail. Son expertise, sa rigueur et sa pédagogie ont permis la qualité et la réussite de ce projet. Je remercie également Monsieur Christian Hafner d'avoir accepté de relire ce mémoire.

Ensuite, je remercie grandement mon père, Hugues Legendre, qui m'a toujours suivi et motivé dans l'entreprise de mes bien trop longues études. Je remercie ma sœur Julie Legendre pour absolument toutes choses pour lesquelles il est pertinent de remercier quelqu'un. Je remercie ma mère Françoise Dasnois, source inépuisable de motivation quotidienne.

Mes amis méritent ensuite évidemment leurs paragraphes. Leurs camaraderies, leurs sens du mot juste et leurs goûts des bonnes choses m'ont permis de rester psychologiquement relativement stable. Merci à Antoine Labadie, Théo Richet, Steve Habib, Baptiste Najean. Merci à Jonas Robert et sa capacité à m'aider à prendre en permanence les meilleures décisions, dont l'inscription à ce master. Merci à mes collègues de master, devenus amis avec lesquelles j'ai passé tant de temps. Merci à Yousra Bahbah, Lionel Lamy, Aurèle Bartolomeo et Baudouin Martelee.

Finalement, je tiens à remercier tout particulièrement, mon "pilier" gauche Adrien Kinart et mon "pilier" droit Arthur Geeraerd, collègues et amis avec lesquels j'ai l'honneur de co-fonder Troople, entreprise en data science.

Ce mémoire est non seulement le fruit de mon travail, mais aussi le reflet de la confiance, de l'encouragement et du soutien de tous ceux mentionnés ici. Je leur dois la moindre de mes victoires. Merci.

Table des matières

Abstract i

Remerciements iii

1 Introduction 1

2 Background 3
2.1 Régression quantile . . . 3
2.1.1 Introduction et motivations . . . 3
2.1.2 Quantile . . . 3
2.1.3 Quantiles conditionnels et régression quantile . . . 6
2.1.4 Analyse et propriété de la variance . . . 10
2.1.5 Propriétés et conclusion . . . 11
2.2 Théorie des valeurs extrêmes . . . 11
2.2.1 Introduction . . . 11
2.2.2 Méthode des maximas . . . 12
2.2.3 Méthode des excès . . . 16
2.3 Forêt aléatoire pour la régression quantile . . . 21
2.3.1 Arbre décisionnel . . . 21
2.3.2 Méthodes d'agrégation d'arbres de décision . . . 25
2.3.3 Forêt aléatoire . . . 26
2.3.4 Forêt aléatoire généralisée pour la régression quantile . . . 28
2.4 Gradient Boosting . . . 32
2.4.1 Méthode générique de boosting basée sur la descente de gradient . . . 32
2.4.2 Fonctionnement Gradient Boosting . . . 33

3 Méthodologie 37
3.1 Contexte de l'algorithme . . . 37
3.2 Gradient Boosting for extreme . . . 38

4 Application du Gbex 42
4.1 Étude de simulations . . . 42
4.1.1 Génération de données . . . 42
4.1.2 Démarche pour l'utilisation du Gbex . . . 43
4.1.3 Estimation du quantile intermédiaire . . . 44
4.1.4 Estimation du quantile extrême . . . 47
4.1.5 Étude des hyper-paramètres . . . 48
4.1.6 Importance des variables dans la construction des modèles . . . 51

4.1.7 Comparaison de méthodes . . . 53
4.2 Application empirique . . . 54
4.2.1 Données . . . 54
4.2.2 Comparaison des modèles entre Gbex et GRF . . . 56

5 Discussion et conclusion **59**

Appendices **66**

A Algorithmes **67**
A.1 Algorithme CART . . . 67
A.2 Algorithme Gradient Boosting . . . 69
A.3 Algorithme Gbex pour la prédiction des paramètres de la GPD . . . 70
A.4 Algorithme de prédiction des quantiles extrêmes . . . 71

B Notions théoriques supplémentaires **72**
B.1 Approximation de Newton-Raphson . . . 72
B.2 Fonction de distribution non-dégénérée . . . 72
B.3 Descente de gradient . . . 74

C Graphiques supplémentaires **75**
C.1 Gradient Boosting . . . 75
C.2 MISE en fonction de la profondeur d'arbre . . . 78
C.3 MISE en fonction du seuil du quantile intermédiaire . . . 79
C.4 Application du Gbex . . . 80
C.4.1 Paramètres d'échelle et de forme en fonction de l'âge . . . 80

Chapitre 1

Introduction

Ces dernières années et plus particulièrement ces derniers mois, on été marqué par un essor fulgurant des méthodes d'apprentissage automatique. Beaucoup de ces méthodes sont à la frontière entre statistiques et sciences informatiques. Ce mémoire s'inscrit dans cette dynamique en proposant l'exploration et l'implémentation d'un algorithme permettant la prédiction de quantiles conditionnels extrêmes. En effet, au cours de ce travail, ces deux domaines ont été explorés en profondeur.

D'un côté, les statistiques offrent un cadre théorique solide et des méthodologies robustes pour l'analyse de données comme la régression quantile. À l'opposé de la régression linéaire classique qui se focalise sur la moyenne conditionnelle d'une variable réponse, la régression quantile permet d'explorer les différents quantiles conditionnels de cette dernière. Ceci offre une analyse plus détaillée et riche de la distribution de Y en relation avec les covariables $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, où n et p représentent respectivement le nombre d'observations et le nombre de covariables. Comme souligné par Mosteller et Tukey [1977], cette approche permet de rendre compte de toutes les caractéristiques importantes de la distribution de Y. Le modèle de régression quantile, introduit par R. Koenker et Bassett [1978], a été largement adopté dans divers domaines comme la santé publique et la météorologie, où l'analyse des comportements extrêmes de la variable réponse Y est cruciale. La régression quantile présente plusieurs atouts, tels que sa résistance aux valeurs aberrantes, son indépendance de l'hypothèse d'homoscédasticité et son insensibilité à certaines transformations monotones. Bien que la régression quantile présente de nombreux avantages, elle comporte également certaines limites. En effet, cette méthode est efficace sur l'ensemble des quantiles centraux, mais sa précision diminue à mesure que l'on s'éloigne de ceux-ci. Les quantiles extrêmes deviennent ainsi difficiles à estimer en raison de la faible densité de données dans ces régions de la distribution.

De ce fait, le deuxième domaine des statistiques qui a été exploré sont les méthodes issues des théories des valeurs extrêmes. Ces méthodes permettent d'améliorer l'estimation de quantiles extrêmes, via l'ajustement d'une distribution généralisée de Pareto. Elles permettent de modéliser des observations extrêmes dont la valeur va au-delà des maxima présents les données d'entraînements. Cette distribution généralisée de Pareto, nommée par GPD au long de ce mémoire, possède certains paramètres dont les valeurs peuvent être estimées. Encore mieux, les valeurs de ces paramètres peuvent être estimée conditionnellement à la valeur de certaines variables indépendantes.

Parallèlement, les sciences informatiques et l'apprentissage automatique proposent un large éventail de techniques, parmi lesquelles figure le gradient boosting. Cette méthode avancée a montré une efficacité remarquable dans diverses tâches de prédiction. Le principe fondamental du gradient boosting est la création d'un modèle prédictif fort à partir de plusieurs modèles prédictifs faibles, aussi appelés au long de ce travail, "apprenants faibles". Ces apprenants sont généralement des arbres de décision, mais peuvent être tout autre type de modèle. Le processus d'optimisation utilisé est basé sur la descente de gradient, d'où le nom de "gradient boosting". Plus précisément, l'algorithme optimise une fonction de coût en ajoutant successivement de nouveaux modèles qui corrigent l'erreur résiduelle du modèle cumulatif précédent.

En résumé, l'objet de ce mémoire est donc d'utiliser un algorithme de gradient boosting, pour estimer des paramètres conditionnelles d'une distribution généralisée de Pareto, afin de modéliser la relation entre des variables indépendantes et les quantiles extrêmes d'une variable cible.

Cette technique relativement récente, est basée sur les travaux de Velthoen et al. 2021 mais ne connaît pas encore d'implémentation en python. Ce mémoire a donc eu pour objectif dans un premier temps d'approfondir la compréhension de la régression quantile, ainsi que de la théorie des valeurs extrêmes, et plus particulièrement de la méthode dite "Peak-over-threshold", et du gradient boosting en abordant les fondements théoriques de ces différentes théories. Une fois cette première phase du travail terminé, une partie de ce travail a été consacrée à l'implémentation de cet algorithme en python, basé sur le package R existant. Cette implémentation était un défi intéressant autant dans l'écriture du code lui-même que dans la recherche de solution pour adapter des packages disponibles dans un langage de programmation, mais pas forcément dans l'autre. Une fois l'algorithme développé en python, et des tests de vérification de justesse du code réalisé, nous nous sommes concentrés sur une série de tests pour explorer l'algorithme en lui-même, ses forces et ses limites. Pour cela, nous avons d'abord travaillé sur des données simulées, puis nous avons réalisé des tests sur des données empiriques.

L'implémentation de cet algorithme était une chose, l'évaluation de sa performance en est une autre. Au long de nos différents tests, une comparaison des performances a été réalisée entre notre algorithme dit du Gbex et un autre algorithme de prédiction de quantile via des forêts aléatoire généralisées. Ce mémoire est donc une exploration passionnante à la frontière entre les statistiques et l'apprentissage automatique, avec l'objectif d'implémenter et d'évaluer l'utilisation du gradient boosting dans le cadre de la modélisation des quantiles conditionnels extrêmes.

Chapitre 2

Background

2.1 Régression quantile

2.1.1 Introduction et motivations

La régression quantile, point de départ de notre algorithme Gbex, est la première des méthodes qui le composent. Cette technique vise à modéliser la relation entre une ou plusieurs covariables — désignées par $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, où n représente le nombre d'observations et p le nombre de covariables — et les différents quantiles conditionnels de la variable réponse $Y \in \mathbb{R}^{n \times 1}$. Dans cette section, nous nous consacrerons à l'exposition des théories sous-jacentes à la notion de quantiles et à la présentation des méthodes pertinentes pour leur estimation. À cet égard, nous introduirons la notion de "fonction check", dont la définition sera explicitée ultérieurement. Nous nous concentrerons ensuite spécifiquement sur les quantiles conditionnels, exposant la méthode de régression quantile qui, en permettant la modélisation d'une relation entre les covariables $\boldsymbol{X}$ et la variable d'intérêt Y, offre la possibilité d'estimer les quantiles conditionnels de cette dernière.

2.1.2 Quantile

Considérons une variable aléatoire générique $Y \in \mathbb{R}$ ayant pour fonction de répartition $F_Y = P(Y < y)$ et pour fonction de densité $f(y)$. Le quantile d'ordre τ de Y est alors donné par l'inverse de la fonction de répartition F_Y, soit :

$$Q_Y(\tau) = F_Y^{-1}(\tau) = \inf \{y : F(y) > \tau\}, \quad \text{où } 0 < \tau < 1. \tag{2.1}$$

La fonction $Q_Y(\tau)$ est appelé la fonction quantile et est représentée dans la figure 2.1 qui mets en évidence les quantiles d'ordre 0.25, 0.5 et 0.95. Afin d'appréhender la régression quantile de manière claire, un parallèle sera fait dans cette première section entre moyenne et quantile, puis entre régression classique et régression quantile. Ce parallèle est assez naturel, et est souvent utilisé pour mettre en évidence l'asymétrie d'une distribution. Intéressons nous donc dans un premier temps à la moyenne, qui peut être défini comme le centre ξ de la distribution Y qui minimise la somme des distances au carré avec la distribution. On exprime alors la moyenne comme

$$\mu = \arg\min_{\xi} \mathbb{E}\left[(Y - \xi)^2\right], \quad \text{où } \xi \in \mathbb{R}.$$

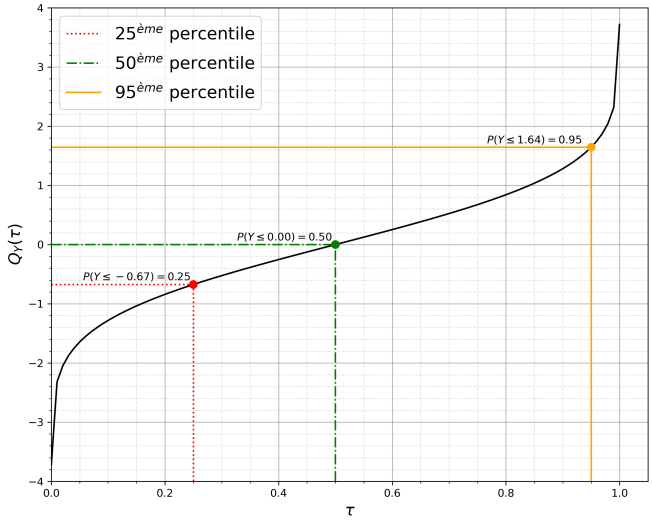


FIGURE 2.1 – Fonction inverse de la fonction de répartition de $\mathbb{N}(0,1)$ ainsi que les quantile d'ordre 0.25, 0.5 et 0.95

De la même manière, un quantile spécifique peut être déterminé par une méthode de minimisation présentée ensuite dans l'équation (2.3). Pour la médiane, on remplace simplement la distance au carré par la distance absolue entre ξ et la distribution, mais pour des quantiles d'ordres différents, il est dans un premier temps nécessaire d'introduire une autre fonction de perte qui sera utilisée, nous la désignerons par "fonction *check*".

Fonction Check

La fonction *check* est la fonction utilisée pour l'estimation d'un certain quantile. Sa forme générique est donnée par :

$$\rho_\tau(u) = \begin{cases} \tau u, & \text{si } u \geq 0 \\ (\tau - 1)u, & \text{si } u < 0 \end{cases} \tag{2.2}$$

où $0 < \tau < 1$ représente l'ordre du quantile. La forme de la fonction *check* s'adapte donc en fonction de l'ordre du quantile choisi et est asymétrique pour des valeurs de $\tau \neq 0.5$, comme présenté dans la figure 2.2. On observe en effet, une pente différente en fonction du signe de u. Cette asymétrie permet d'accorder plus de poids aux résidus qui contribuent le plus à l'estimation du quantile τ.

Estimation du quantile

En utilisant la fonction *check*, un certain quantile d'ordre τ peut être défini via le problème d'optimisation :

$$q_\tau = \arg\min_{\xi} \mathbb{E}\left[\rho_\tau(Y - \xi)\right]. \tag{2.3}$$

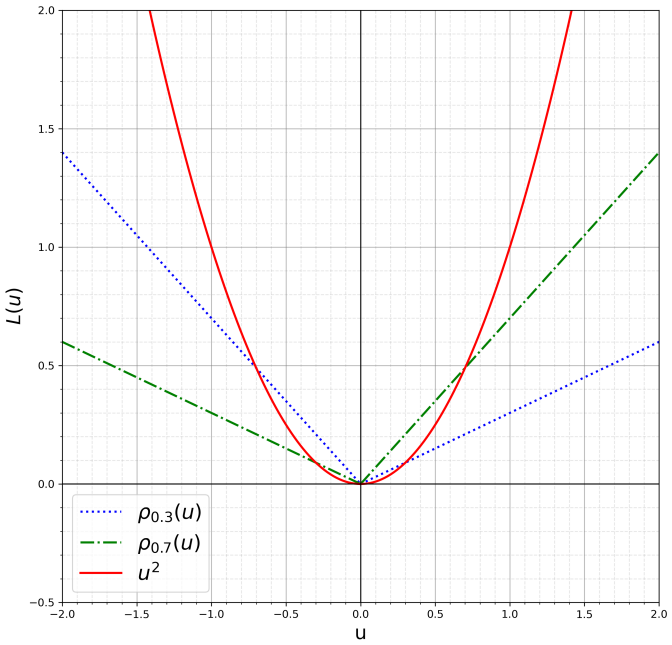


FIGURE 2.2 – Comparaison de plusieurs fonctions de perte $L(u)$. On observe la fonction *check* pour le quantile d'ordre 0.3 en pointillés et d'ordre 0.7 en traits, ainsi que l'erreur quadratique moyenne avec la ligne continue.

Dans ce problème d'optimisation, l'espérance $\mathbb{E}\left[\rho_\tau(Y-\xi)\right]$ peut être réécrite de la manière suivante :

$$\begin{aligned}\mathbb{E}(\rho_\tau(Y-\xi)) &= \int_{-\infty}^{+\infty} \rho_\tau(y-\xi)dF(y) \\ &= (1-\tau)\int_{-\infty}^{\xi}(y-\xi)dF(y) + \tau\int_{\xi}^{+\infty}(y-\xi)dF(y).\end{aligned}$$

Cette dernière expression est en fait la somme des intégrales à gauche et à droite de ξ. Pour trouver l'argument ξ, on minimise donc cette fonction en annulant sa dérivée, on a

donc :

$$
\begin{aligned}
&\frac{\partial \mathbb{E}(\rho_\tau(Y-\xi))}{\partial \xi} = 0 \\
&\iff \frac{\partial}{\partial \xi}\left[(1-\tau)\int_{-\infty}^{\xi}(y-\xi)dF(y) + \tau\int_{\xi}^{+\infty}(y-\xi)dF(y)\right] = 0 \\
&\iff \frac{\partial}{\partial \xi}\left[(\tau-1)\left(\int_{-\infty}^{\xi} y dF(y) - \xi\int_{-\infty}^{\xi} dF(y)\right)\right. \\
&\qquad\qquad \left. -\tau\left(\int_{+\infty}^{\xi} y dF(y) - \xi\int_{+\infty}^{\xi} dF(y)\right)\right] = 0 \\
&\iff (\tau-1)(-F(\xi)) - \tau(1-F(\xi)) = 0 \\
&\iff F(\xi) - \tau = 0 \\
&\iff F(\xi) = \tau
\end{aligned} \tag{2.4}
$$

La dérivée seconde de $\mathbb{E}(\rho_\tau(Y-\xi))$ étant la fonction de densité $f(\xi)$, on est certain que l'on minimise bien cette fonction étant donné qu'elle est par définition non-négative.

En résumé, minimiser la fonction *check* appliquée au résidu $(y-\xi)$ nous donne une mesure du quantile de notre variable d'intérêt Y. En général, la fonction de répartition $F(y)$ étant inconnue, on utilise la fonction de répartition empirique donnée par :

$$F_n(y) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(Y_i < y),$$

où $Y_1, \ldots, Y_n$ est un échantillon de la variable aléatoire et $\mathbb{1}(Y_i < y)$ est une fonction indicatrice qui vaut 1 si $Y_i < y$ et 0 sinon. En remplaçant dans l'équation (2.4), la fonction de répartition inconnue $F(y)$ par la fonction de répartition empirique $F_n(y)$, le problème d'optimisation devient alors :

$$
\begin{aligned}
\mathbb{E}(\rho_\tau(Y-\xi)) &= \int_{-\infty}^{+\infty} \rho_\tau(Y_i-\xi)dF_n(y) \\
&= \frac{1}{n}\sum_{i=1}^{n}\rho_\tau(Y_i-\xi).
\end{aligned} \tag{2.5}
$$

L'estimateur du quantile est alors la valeur de ξ qui minimise cette expression. En résumé, la valeur des quantiles peut être déterminée par un problème d'optimisation. Cette formulation permet une généralisation à l'estimation de quantiles conditionnels, qui est fondamentale pour la création des modèles de régression quantile.

2.1.3 Quantiles conditionnels et régression quantile

Maintenant que la méthode d'estimation du quantile a été correctement définie, les quantiles conditionnels peuvent être présentés. De la même manière que pour la partie non-conditionnelle, commençons par une comparaison avec le modèle de moyenne. Considérons $Y \in \mathbb{R}^{n\times 1}$ comme notre variable dépendante ou variable d'intérêt et $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ comme un ensemble de p variables indépendantes, où n est le nombre d'observations. À noter que $\boldsymbol{X}$ est une matrice dont chaque ligne est un vecteur de variables indépendantes, et

$\boldsymbol{x}$ représente une de ces lignes. La moyenne non-conditionnelle présentée dans la partie précédente peut être étendue à une moyenne conditionnelle. Comme précédemment, la moyenne conditionnelle peut être transformée en un problème d'optimisation. Définissons alors :

$$\mu(\boldsymbol{x}) = \arg\min_{\xi \in \mathbb{R}} \mathbb{E}\left[(Y-\xi)^2 \mid \boldsymbol{X} = \boldsymbol{x}\right]$$

Dans cette équation, $\mu(\boldsymbol{x}) = \mathbb{E}\left[Y \mid \boldsymbol{X} = \boldsymbol{x}\right]$ est la fonction de moyenne conditionnelle. En supposant que cette fonction est linéaire, nous avons $\hat{\mu}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\beta}$, où $\boldsymbol{\beta}$ est un vecteur de paramètres à estimer. Ainsi, nous pouvons évaluer la moyenne conditionnelle en résolvant le problème d'optimisation suivant :

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \mathbb{E}\left[\left(Y - \boldsymbol{x}_i^T\boldsymbol{\beta}\right)^2\right].$$

En suivant cette méthode, la minimisation des erreurs quadratiques se traduit en un problème d'algèbre linéaire qui peut être résolu de façon simple remplaçant l'espérance ($\mathbb{E}(.)$) par $\sum_{i=1}^{n}$. Cette approche, connue sous le nom de méthode des moindres carrés ordinaires (MCO), donne naissance au modèle suivant

$$Y = \boldsymbol{X}^T\boldsymbol{\beta} + \epsilon.$$

Dans ce modèle, la variable dépendante est définie comme une combinaison linéaire des variables indépendantes et du vecteur de coefficients $\boldsymbol{\beta}$. Dans ce modèle, ϵ représente le résidu et présente deux hypothèses importantes : $\mathbb{E}(\epsilon) = 0$ et $\sigma(\epsilon) = c$ ou c est une constante qui impose l'hypothèse d'homoscédasticité des résidus au modèle. La valeur du vecteur $\hat{\boldsymbol{\beta}}$ peut être alors déterminé de manière analytique. En effet si la matrice des covariance $\boldsymbol{X}$ est de rang maximum alors

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^TY.$$

En reprenant l'approche de la méthode des moindres carrés ordinaires, nous pouvons estimer les quantiles conditionnels. En s'appuyant sur l'équation (2.1), nous pouvons définir la fonction de quantile conditionnelle à $\boldsymbol{x}$ comme suit :

$$Q_{\boldsymbol{x}}(\tau) = \inf\left\{y : F_Y(y \mid X) \geq \tau\right\} = F_Y^{-1}\left(\tau \mid \boldsymbol{X} = \boldsymbol{x}\right), \tag{2.6}$$

où F_Y^{-1} est l'inverse de la fonction de répartition de Y. Ensuite, si nous adaptons l'équation du quantile non conditionnel, tel que défini dans l'équation (2.3), pour y inclure la conditionnalité de Y par rapport à $\boldsymbol{X}$, nous obtenons :

$$Q_\tau(Y \mid \boldsymbol{X}) = \arg\min_{q(X)} \mathbb{E}[\rho_\tau(Y - q(X))].$$

Comme pour la régression linéaire classique, nous pouvons approximer cette fonction par un modèle linéaire. Ainsi, nous définissons :

$$\hat{\boldsymbol{\beta}}(\tau) = \arg\min_{\boldsymbol{\beta}} \mathbb{E}[\rho_\tau(Y - \boldsymbol{X}^T\boldsymbol{\beta})],$$

où $\hat{\boldsymbol{\beta}}(\tau)$ sont les paramètres du modèle pour un certain quantile τ. Une représentation de cette modélisation est visible dans la figure 2.3 Ce modèle linéaire est une approximation pondérée des moindres carrés de la fonction inconnue des quantiles conditionnels.

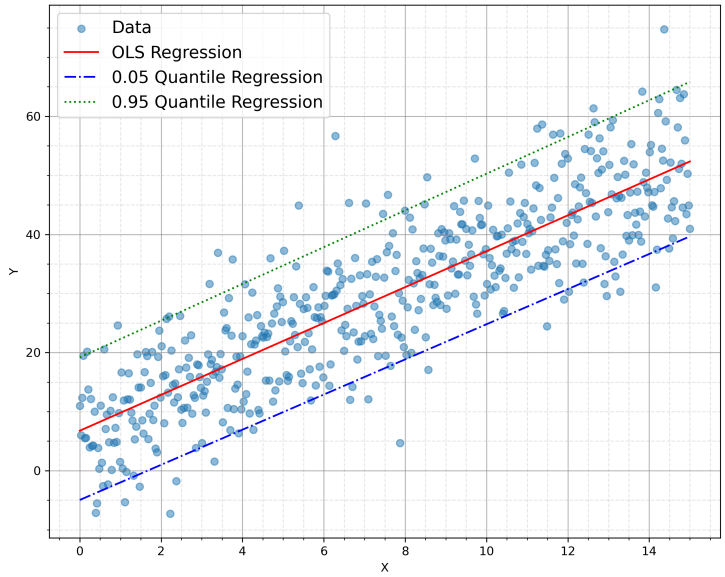


FIGURE 2.3 – Régression quantile pour des données simulées, où la variable dépendante $Y = 3X + 7 + \text{rdm}$, où rdm est un bruit provenant d'une loi normale centrée réduite. La droite représente un modèle MCO, les traits une régression quantile d'ordre 0.05 et les pointillés une régression quantile d'ordre 0.95.

Plus précisément, ANGRIST et al. 2006 montre que le vecteur $\boldsymbol{\beta}(\tau)$ de coefficients de régression quantile de la population minimise l'espérance pondérée de l'approximation de l'erreur quadratique, c'est-à-dire le carré de la différence entre la véritable fonction des quantiles conditionnels et une approximation linéaire, grâce à la fonction de pondération $w_\tau(\boldsymbol{X}, \boldsymbol{\beta})$. Ainsi, pour déterminer les coefficients de la population, nous avons le problème de minimisation suivant :

$$\boldsymbol{\beta}(\tau) = \arg\min_{\boldsymbol{\beta}} \mathbb{E}\left[\boldsymbol{w}_\tau(\boldsymbol{X}, \boldsymbol{\beta})\Delta_\tau^2(\boldsymbol{X}, \boldsymbol{\beta})\right], \quad \Delta_\tau(\boldsymbol{X}, \boldsymbol{\beta}) = \boldsymbol{X}^T\boldsymbol{\beta} - Q_\tau(Y \mid X).$$

On minimise donc le carré de la distance entre $\boldsymbol{X}^T\boldsymbol{\beta}$ et la fonction de quantile conditionnel, pondéré par des "poids d'importance". Les "poids d'importance", définis par ANGRIST et al. 2006, sont utilisés pour ajuster les estimations de la fonction de quantile conditionnel en fonction de la distance entre la vraie fonction et l'approximation linéaire. Les valeurs de $\boldsymbol{X}$ sont également pondérées en fonction de leur densité, ce qui signifie que les valeurs qui apparaissent plus fréquemment dans les données reçoivent moins de poids que celles qui apparaissent moins souvent.

Pour déterminer la valeur des $\boldsymbol{\beta}(\tau)$, on remplace, comme dans la partie précédente, l'espérance ($\mathbb{E}(.)$) par la somme de toutes les observations.

$$\hat{\boldsymbol{\beta}}(\tau) = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \rho_\tau(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta}). \tag{2.7}$$

Contrairement à la régression linéaire classique (MCO), il n'existe pas de méthode analytique pour calculer les $\boldsymbol{\beta}(\tau)$, il faut donc résoudre ce problème numériquement. La solution

consiste alors à reformuler ce problème de minimisation en un problème de programmation linéaire, qui peut être résolu à l'aide de l'algorithme du simplexe. L'intuition de ce problème de maximisation est donnée ensuite.

Résolution du problème de maximisation.

Pour trouver le vecteur $\hat{\boldsymbol{\beta}}(\tau)$, on réécrit le modèle en décomposant l'erreur entre les erreurs positives et négatives. En posant $u_i = \epsilon_i \mathbb{1}(\epsilon_i > 0)$ et $v_i = |\epsilon_i| \mathbb{1}(\epsilon_i < 0)$ on obtient

$$\begin{aligned} Y_i &= \boldsymbol{x}_i^T \boldsymbol{\beta}(\tau) + \epsilon_i(\tau) \\ &= \boldsymbol{x}_i^T \boldsymbol{\beta}(\tau) + (u_i - v_i). \end{aligned}$$

Cette méthode, proposée par Roger KOENKER 2005, permet de transformer le problème de minimisation présenté dans l'équation (2.7) en un problème de minimisation sous contrainte :

$$\begin{aligned} \min_{\boldsymbol{\beta}(\tau)} \quad & \tau \mathbb{1}_n^T u + (1-\tau)\mathbb{1}_n^T v \\ \text{sous contrainte que} \quad & y - \boldsymbol{X}^T \boldsymbol{\beta}(\tau) = u - v, \\ & u \geq 0, \quad v \geq 0, \end{aligned}$$

ou encore par :

$$\min_{\boldsymbol{\beta}(\tau)} \Big\{ \mathbb{1}_n^T \boldsymbol{u} + \mathbb{1}_n^T \boldsymbol{v} \mid Y = \boldsymbol{X}\boldsymbol{\beta}(\tau) + \boldsymbol{u} - \boldsymbol{v}, \\ \{\boldsymbol{u}, \boldsymbol{v}\} \in \mathbb{R}_+^n \Big\}.$$

De plus, posons que $[\boldsymbol{a}]_+$ est la partie non-négative du vecteur $\boldsymbol{a}$, $\boldsymbol{B} = [\boldsymbol{X} - \boldsymbol{X}\boldsymbol{I} - \boldsymbol{I}]$ et posons

$$\psi = \begin{bmatrix} [\beta]_+ \\ [-\beta]_+ \\ \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix}, \boldsymbol{d} = \begin{bmatrix} 0_{[p]} \\ 0_{[p]} \\ \mathbb{1}_{[n]} \\ \mathbb{1}_{[n]} \end{bmatrix}.$$

Cette formulation matricielle permet de reformuler le problème comme un problème de programmation linéaire standard. En effet, la forme matricielle du problème de maximisation sous contrainte présenté plus haut peut être redéfinie par :

$$\begin{aligned} \underset{\psi}{\text{minimise}} \quad & \boldsymbol{d}^T \psi \\ \text{sous contrainte} \quad & \boldsymbol{B}\psi = Y \\ & \tau \geq 0 \end{aligned}$$

Cette minimisation peut facilement être transformée en un problème de maximisation :

$$\begin{aligned} \underset{\boldsymbol{d}}{\text{maximise}} \quad & Y^T \boldsymbol{z} \\ \text{sous contrainte} \quad & \boldsymbol{B}^T \boldsymbol{z} \leq \boldsymbol{d} \end{aligned}$$

Lors de la résolution d'un simplexe, les solutions recherchées devant se sur les sommets du simplexe. Ce problème peut alors être reformulé comme :

$$\max_{\boldsymbol{z}} \left\{ Y^T \boldsymbol{z} \mid \boldsymbol{X}^T \boldsymbol{z} = (1-\tau)\boldsymbol{X}^T \mathbb{1}, \boldsymbol{z} \in [0,1]^n \right\}.$$

Pour résoudre ce problème d'optimisation, la méthode du simplexe décrite par R. W. Koenker et D'Orey 1987 est la méthode la plus couramment utilisée. D'après la littérature, cette méthode s'est révélé très efficace lorsque le nombre d'observations est relativement faible ($n < 100,000$). Mais il est intéressant de noter qu'il existe d'autres méthodes pour la résolution de ce simplexe, telle que la méthode des points intérieurs proposée par Karmarkar 1984, qui est supposée plus efficace pour un grand nombre d'observations. Portnoy 1997 propose également une méthode des points intérieurs, adaptée pour les régressions quantiles et particulièrement pour les grands nombres d'observations. Enfin, C. Chen et Wei 2005 présente l'algorithme de lissage fini, particulièrement efficace lorsque le nombre de covariables est élevé.

2.1.4 Analyse et propriété de la variance

L'étude des diverses propriétés de la variance peut s'avérer intéressante pour comprendre les limites et les applications pratiques de la régression quantile. Définissons d'abord ξ_τ comme étant l'estimation d'un certain quantile d'ordre τ, c'est-à-dire $\xi_\tau = F^{-1}(\tau)$ avec une certaine densité $f(\xi)$. Autrement dit, ξ_τ est la valeur en dessous de laquelle $\tau\%$ des observations dans une distribution se situent (voir figure 2.1).

Introduisons ensuite une fonction de gradient qui mesure la différence entre la proportion d'observations en dessous de ξ et le quantile souhaité τ. Cette fonction est définie par :

$$g_n(\xi) = n^{-1} \sum_i^n \mathbb{1}(Y_i < \xi) - \tau$$

L'estimation étant basée sur un échantillon d'observations, chaque observation a une chance égale d'être au-dessus ou en dessous de l'estimation. La convergence de l'estimation est impliquée par le fait que la fonction de répartition empirique tend vers la vraie fonction de répartition. Ainsi, si les données sont identiquement et indépendamment distribuées (iid), l'estimation sera également normalement distribuée, et l'on aura :

$$\hat{\xi}_\tau \to \mathcal{N}\left(\xi_\tau, \frac{\tau(1-\tau)}{n} \times \frac{1}{f(\xi_\tau)^2}\right)$$

La distribution limite de l'estimation du quantile $\hat{\beta}(\tau)$ lorsque l'erreur n'est pas identiquement et indépendamment distribuée est donnée par la relation suivante :

$$\sqrt{n}(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)) \to \mathcal{N}(0, \tau(1-\tau)H_n^{-1}J_nH_n^{-1}) \tag{2.8}$$

où $J_n(\tau) = n^{-1}\sum_i \boldsymbol{x}_i^T\boldsymbol{x}_i$ et $H_n(\tau) = n^{-1}\sum_i \boldsymbol{x}_i'\boldsymbol{x}_i f_i(\xi_i(\tau))$.

Dans l'équation précédente, $J_n(\tau)$ désigne la matrice de variance-covariance associée aux variables explicatives, tandis que $H_n(\tau)$ représente la matrice hessienne, obtenue à partir des produits des dérivées partielles de la variable de réponse par les dérivées partielles des variables explicatives. En outre, $\xi_i(\tau)$ correspond à l'estimation du quantile pour la i-ème observation et f_i renvoie à la densité de la distribution à ce point précis.

L'estimation de cette densité, en général inconnue, peut être réalisée via deux approches distinctes : une méthode non-paramétrique ou une méthode basée sur le bootstrap, qui s'avère être extrêmement gourmande en termes de ressources computationnelles. Il convient donc de noter que l'estimation de la variance de cet estimateur peut se révéler complexe, constituant ainsi une limitation notable de ce modèle.

2.1.5 Propriétés et conclusion

En définitive, la méthode de régression quantile offre une perspective plus complète des relations statistiques entre les variables stochastiques. Elle permet d'examiner la relation entre un ensemble de covariables et les différentes parties de la distribution de la réponse, palliant ainsi certaines limites de la régression linéaire classique.

Si l'on considère un modèle $y = \boldsymbol{x}\boldsymbol{\beta} + \epsilon$, la régression quantile possède également les propriétés intéressantes suivantes :

1. Équivariance d'échelle : $\hat{\beta}(\tau; ay, \boldsymbol{X}) = a\hat{\beta}(\tau; y, \boldsymbol{X})$ et $\hat{\beta}(\tau; -ay, \boldsymbol{X}) = a\hat{\beta}(1-\tau; y, \boldsymbol{X})$.
2. Équivariance de décalage : $\hat{\beta}(\tau; \boldsymbol{X}\gamma, \boldsymbol{X}) = \hat{\beta}(\tau; y, \boldsymbol{X}) + \gamma$.
3. Équivariance de re-paramétrisation : $\hat{\beta}(\tau; y, \boldsymbol{X}\boldsymbol{A}) = \boldsymbol{A}^{-1}\hat{\beta}(\tau; y, \boldsymbol{X})$.
4. Équivariance aux transformations monotones : Soit $h(\cdot)$ une fonction monotone, $Q_{h(Y)}(\tau) = h(Q_Y(\tau))$, c'est-à-dire que, comparé à la méthode des MCO, la médiane de $\log(Y)$ est le logarithme de la médiane de Y.
5. Comparé aux MCO, la fonction d'influence de y sur la régression quantile est bornée, alors qu'elle est théoriquement infinie en MCO. La régression quantile est donc robuste face aux valeurs extrêmes.

Comme expliqué en introduction, malgré les nombreux atouts de la régression quantile, elle n'est pas sans défauts. Sa fiabilité est assurée lorsqu'elle est appliquée sur les quantiles centraux, cependant, sa justesse s'amenuise lorsqu'on s'éloigne de ceux-ci. Ceci est causé par la rareté des observations dans les zones extrêmes de la distribution. Pour ces raisons, les théories des valeurs extrêmes sont indispensables et vont être décritent dans la prochaine section.

2.2 Théorie des valeurs extrêmes

2.2.1 Introduction

La théorie des valeurs extrêmes (TVE) est une spécialisation de la statistique dédiée à l'étude des déviations extrêmes par rapport à la médiane d'une distribution donnée. Cette théorie offre des modèles permettant de caractériser ces valeurs extrêmes en se focalisant sur les queues de distribution. Bien que l'occurrence de ces valeurs extrêmes soit rare, leurs impacts peuvent être considérables lorsqu'elles surviennent.

Le développement de la théorie des valeurs extrêmes a été largement documenté par Gumbel 1958, qui a présenté des applications en hydrologie en s'appuyant sur les travaux de Fréchet 1927, Fisher et Tippett 1928 et Stamp, Mises et Kahane 1937.

Initialement utilisée dans le domaine de l'hydrologie, la théorie des valeurs extrêmes a trouvé des applications dans de nombreux autres domaines, notamment en finance. Parmi ces travaux, Embrechts, Klüppelberg et Mikosch 1997 ont démontré la portée de ses applications en finance, suivi par Daníelsson et de Vries 1997 qui ont modélisé les distributions inconditionnelles des rendements sur le marché des changes, en se concentrant en particulier sur trois devises (YEN-DM, YEN-USD et DM-USD). Par la suite, Odening et Hinrichs 2003 ont exposé les avantages et inconvénients majeurs de la TVE dans un contexte de gestion des risques. Les TVE a gagné en popularité après la crise financière de 2008.

Deux approches principales sont utilisées pour traiter les valeurs extrêmes : la méthode des "block maxima" et la méthode des excès (ou Peak over Threshold), désignée ici par POT. Bien que nous privilégierons l'approche POT dans ce travail, une présentation de la méthode des maxima sera également proposée. Il convient de mentionner l'existence d'une troisième méthode, le "r-largest order model", qui toutefois ne sera pas abordée ici.

2.2.2 Méthode des maximas

Cette méthode, initialement développée dans le livre de Gumbel 1958, consiste à diviser les observations d'une variable aléatoire en intervalles séquentiels non chevauchants et réguliers. Dans chaque intervalle, seule la valeur maximale est retenue, créant ainsi une série de maxima de blocs comme présenté dans la figue 2.4. Cette série de maxima est

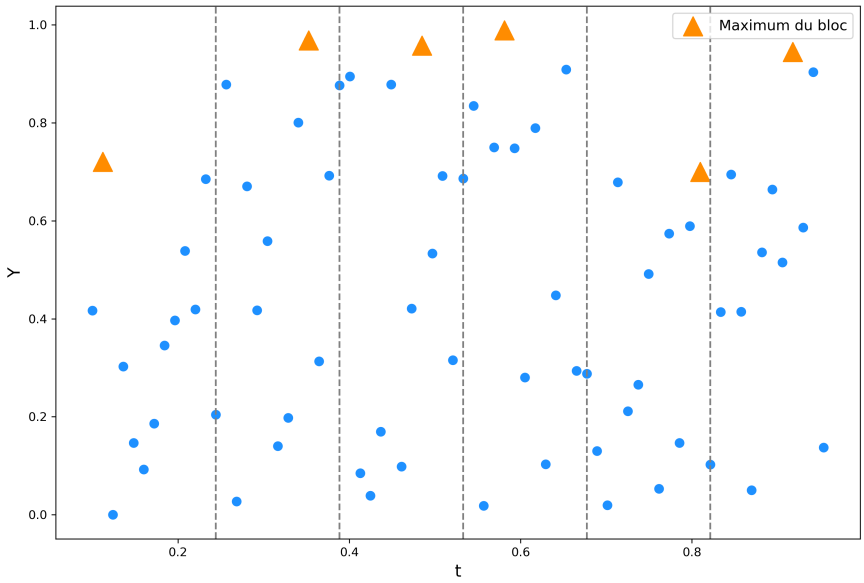


Figure 2.4 – Exemple de la méthode des maxima de blocs. Des données aléatoires suivant une loi uniforme ont été généré et représenté sur le graphique. Le graphique a ensuite été divisé en 6 régions distinctes, au sein de chaque région, le maximum est représenté par un triangle orange.

ensuite ajustée à une distribution de probabilité qui décrit les queues de la distribution de la série originale.

Pour mieux comprendre cette méthode, revenons tout d'abord au théorème central limite. Soit $(X_k)_{k\in\mathbb{N}}$, une suite de variables aléatoires indépendantes et identiquement distribuées (iid), de moyenne μ et de variance σ^2. Soit $\bar{X}n = \sum_{k=1}^{n} X_k$ la moyenne empirique de n réalisations de X. Alors, lorsque n tend vers l'infini, la distribution de $\sqrt{n}(\bar{X}_n - \mu)$ converge en probabilité vers la distribution normale centrée réduite, c'est-à-dire :

$$\frac{\sqrt{n}}{\sigma}\left(\bar{X}_n - \mu\right) \xrightarrow[n\uparrow\infty]{d} N(0,1)$$

où $\xrightarrow[n\uparrow\infty]{d}$ indique la convergence en distribution.

De la même manière, la théorie des valeurs extrêmes étudie le comportement des valeurs extrêmes d'un échantillon. Si l'on considère la même suite iid que précédemment $(X_k)_{k\in\mathbb{N}}$, avec une fonction de répartition que l'on note F, on peut s'intéresser au maximum de cette séquence : $M_n = \max(X_1, \ldots, X_n)$. La méthode des maxima par bloc étudie le comportement de M_n à mesure que la taille de l'échantillon n augmente et tend vers l'infini.

La probabilité que le maximum M_n soit inférieur ou égal à x est donnée par :

$$\begin{aligned} P\{M_n \leq x\} &= P\{X_1 \leq x, \ldots, X_n \leq x\} \\ &= \prod_{i=1}^{n} \mathbb{P}(X_i < x) \\ &= F^n(x) \end{aligned}$$

De la même manière que le théorème central limite permet d'obtenir une distribution normale pour approcher une distribution inconnue lorsque n tend vers l'infini, l'on cherche approcher la fonction F par une certaine distribution, cette distribution sera trouvée grâce au théorème de Fisher-Tippett-Gnedenko décrit ensuite.

Théorème de Fisher-Tippett-Gnedenko et distribution extrêmes généralisées

Considérons une variable aléatoire $(X_k)_k$ non bornée, c'est-à-dire que $F(x) < 1$ pour tout $x < \infty$, la probabilité que M_n soit inférieur à une certaine valeur x (c'est-à-dire $F^n(x)$) tend vers 0 lorsque n tend vers l'infini. Autrement dit : $F^n(x) \to 0$ pour tout x lorsque $n \to \infty$, et $M_n \xrightarrow{P} \infty$. Afin d'obtenir une limite non dégénérée, on introduit alors des suites a_n et b_n pour standardiser M_n. Le concept de fonction non-dégénéré est définie dans l'annexe B.2.

Theorem 1

Soit F la fonction de répartition déjà définie, G une fonction de répartition non-dégénérée, supposons alors qu'il existe une séquence $\{a_n\} > 0$ et b_n tel que :

$$P\left\{\frac{M_n - b_n}{a_n} \leq x\right\} \to G(x) \quad \text{lorsque } n \to \infty.$$

En d'autres termes, $F^n(a_n x + b_n) \to G(x)$ pour tout point x de la fonction de répartition $G(x)$. Dans une telle situation, seulement un certain nombre de distributions peuvent être considérée comme asymptotique à la distribution standardisée des maxima M_n. Cette fonction que nous appellerons "distributions fondamentales" de la théorie des valeurs extrêmes sont :

- Type I : la distribution de Gumbel

$$F_0(x) = \exp\left(-e^{-\frac{(x-\mu)}{\sigma}}\right), \quad x \in \mathbb{R}$$

- Type II : la distribution de Fréchet

$$F_1(x) = \begin{cases} 0, & x \leq 0, \\ \exp\left\{-((x-\mu)/\sigma)^{-\alpha}\right\}, & x > \mu, \alpha > 0 \end{cases}$$

- Type III : la distribution de Weibull

$$F_2(x) = \begin{cases} 1, & x > 0, \\ e^{-(-\frac{(x-\mu)}{\sigma})^{\alpha}}, & x < \mu, \alpha > 0 \end{cases}$$

où les paramètres $\sigma > 0$ et $-\infty < \mu < \infty$.

La représentation de ces différentes distributions sont données dans la figure 2.5

En résumé, le premier théorème des valeurs extrêmes indique que les maxima d'un échantillon aléatoire d'une même variable aléatoire converge forcément vers une des trois distributions décrites dans le théorème ci-dessus (Gumbel, Fréchet et Weibull). Ces trois distributions peuvent être rassemblées dans la famille des distributions généralisées des valeurs extrêmes, souvent notée GEV (Generalized Extreme Value distribution). La fonction de répartition de cette distribution est donnée par :

$$\text{GEV}(x; \mu, \sigma, \alpha) = \begin{cases} \exp\left[-\left[1 + \alpha\left(\frac{x-\mu}{\sigma}\right)\right]^{\frac{-1}{\alpha}}\right], & \text{si } \alpha \neq 0, \\ \exp\left[-e^{-(\frac{x-\mu}{\sigma})}\right] & \text{si } \alpha = 0. \end{cases}$$

Le paramètre de localisation est donné par μ (qui représente le centre de la distribution, mais ne doit pas être confondu avec la moyenne), le paramètre de forme est donné par ξ et le paramètre d'échelle est donné par σ (qui représente l'écart par rapport à μ, mais ne doit pas être confondu avec la variance).

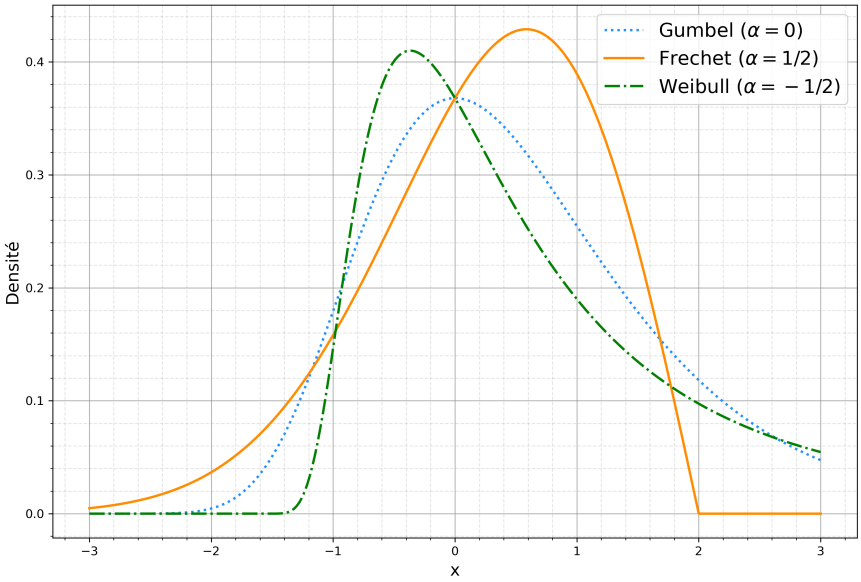


FIGURE 2.5 – Distribution GEV en fonction du paramètre α, prenant la forme d'une distribtion de Gumbel, Frechet ou Weibull

La fonction de densité de cette distribution GEV est donnée par :

$$\text{gev}(x, \mu, \sigma, \alpha) = \begin{cases} \frac{1}{\sigma} \exp\left[-\left(1+\alpha\frac{(x-\mu)}{\sigma}\right)^{-\frac{1}{\alpha}}\right]\left(1+\alpha\frac{(x-\mu)}{\sigma}\right)^{-1-\frac{1}{\alpha}} & \text{si } \alpha \neq 0, \\ \frac{1}{\sigma} \exp\left[-\exp\left(-\frac{(x-\mu)}{\sigma}\right) - \frac{(x-\mu)}{\sigma}\right] & \text{si } \alpha = 0. \end{cases}$$

À partir de cette distribution GEV, on peut retrouver facilement nos distributions fondamentales, et chacune d'entre elles est plus ou moins adaptée à des situations particulières (HOSKING, WALLIS et WOOD 1985). En effet, la distribution de Gumbel est obtenue avec $\alpha = 0$ et a beaucoup été utilisée pour des applications en hydrologie (GUMBEL 1958) ; la distribution de Fréchet est obtenue avec $\alpha > 0$ et modélise en général très bien les données de précipitations, de flux de courant ou de dommages économiques ; la distribution de Weibull est obtenue avec $\alpha < 0$ et modélise en général les données relatives aux températures, à la vitesse du vent, ou à la hauteur de la houle.

Modélisation blocs maxima Maintenant que tous les concepts importants à la compréhension de cette première méthode ont été présentés, nous allons décrire la méthode de modélisation des quantiles extrêmes. Cette méthode de modélisation des valeurs extrêmes nécessite quelques hypothèses. Premièrement, il faut que toutes les valeurs utilisées pour la modélisation proviennent de la même population qui a une certaine fonction de densité inconnue que l'on notera par exemple $f(x)$. Par exemple, si l'on s'intéresse aux températures annuelles dans un pays, $f(x)$ est la distribution des températures quotidiennes. Ensuite, la deuxième hypothèse est que les maxima sont pris indépendamment les uns des autres. La largeur des blocs choisis est importante pour assurer cette indépendance. En effet, si les blocs sont mal choisis et chevauchent deux saisons de l'année, certains maxima de périodes plus froides pourraient être négligés au profit de maxima dans des périodes plus douces.

Si ces hypothèses sont effectivement confirmées, le théorème de Fisher-Tippett-Gnedenko présenté plus haut confirme que la distribution des maxima aura une des distributions GEV. La convergence vers une des trois distributions fondamentales de la théorie des valeurs extrêmes dépend de plusieurs facteurs. Tout d'abord, cela dépend évidemment de la véritable distribution parente inconnue que l'on cherche à estimer $f(x)$. La vitesse de convergence peut dépendre de cette distribution en particulier. Ensuite, la convergence dépend du nombre d'observations qui se trouvent dans chaque bloc. Les paramètres (μ, σ, α) de la distribution GEV peuvent alors être estimé par exemple par maximum de vraisemblance afin d'ajuster la distribution aux maxima de blocs qui ont été calculé précédemment.

Une fois les paramètres estimés, les quantiles extrêmes pour un certain ordre τ peuvent être calculés en utilisant la fonction quantile inverse de la GEV donnée par

$$Q(\tau; \mu, \sigma, \alpha) = \begin{cases} \mu + \frac{\sigma}{\alpha}\left[(-\log(\tau))^{-\alpha} - 1\right] & \text{si } \alpha \neq 0, \\ \mu - \sigma \log\left(-\log(\tau)\right) & \text{si } \alpha = 0. \end{cases}$$

où $Q(\tau; \mu, \sigma, \alpha)$ est la valeur du quantile extrême pour un niveau de probabilité τ, et μ, σ, et α sont les paramètres estimés de la distribution GEV.

Conclusion

Cette méthode, bien qu'adaptée à un certain nombre de situations, comme par exemple des situations ou les données présentent une certaine saisonnalité naturelle (Ferreira et Haan 2015), elle peut tout de même présenter certains désavantages. L'un des principaux est que, dans cette méthode, on considère uniquement la valeur la plus grande d'un bloc, c'est-à-dire le maximum global du bloc, mais l'on passe à côté des maxima locaux qui pourraient avoir une grande importance pour la modélisation. Pour revenir à l'exemple de l'étude des températures, on pourrait imaginer que, dans un certain bloc, il y aurait des températures extrêmes et inédites pendant 3 jours consécutifs. Avec la méthode des maxima, une seule valeur serait alors utilisée pour la modélisation, alors que les 2 autres valeurs pourraient être bien plus élevées que le maximum de blocs adjacents. Ensuite, comme expliqué plus haut, la convergence dépend grandement du choix de la taille des blocs et donc du nombre d'observations par bloc. Finalement, elle est aussi sensible aux valeurs aberrantes. En effet, si une valeur extrême inhabituelle est incluse dans un bloc, elle pourrait avoir une influence disproportionnée sur l'estimation des valeurs extrêmes. Pour ces différentes raisons, la méthode des excès sera privilégiée et présentée dans la section suivante.

2.2.3 Méthode des excès

Introduction

Cette méthode, plus couramment connue sous le nom de "Peak Over Threshold method" et notée POT, est la méthode qui sera utilisé dans notre algorithme. Pour appliquer cette méthode, il faut tout d'abord déterminer un seuil $u \in \mathbb{R}$ au-dessus duquel une valeur est considérée comme extrême. Ensuite, on calcule les excès de la variable par rapport à ce seuil en soustrayant le seuil à chaque valeur de la variable. On peut alors modéliser la répartition des excès à l'aide d'une distribution appropriée et l'exploiter pour prédire la probabilité d'occurrence de valeurs extrêmes à l'avenir. Cette distribution est la

distribution généralisée de Pareto avec un paramètre d'échelle noté σ et un paramètre de forme noté γ. Afin de décrire au mieux cette méthode POT, rappelons d'abord certaines définitions.

Distribution généralisée de Pareto. Commençons donc par décrire la distribution généralisée de Pareto. La fonction de répartition de la distribution généralisée de Pareto est donnée par :

$$\mathrm{GPD}_{(\sigma,\gamma)}(x) = \begin{cases} 1 - \left[1 + \frac{\gamma x}{\sigma}\right]^{-\frac{1}{\gamma}} & \text{si } \gamma \neq 0, \\ 1 - e^{-\frac{x}{\sigma}} & \text{si } \gamma = 0, \end{cases} \tag{2.9}$$

et la fonction de densité de la distribution généralisée de Pareto est donnée par :

$$\mathrm{gpd}_{(\sigma,\gamma)}(x) = \begin{cases} \frac{1}{\sigma}\left[1 + \gamma\frac{x}{\sigma}\right]^{-\frac{1}{\gamma}-1} & \text{si } \gamma \neq 0, \\ \frac{1}{\sigma}e^{-\frac{x}{\sigma}} & \text{si } \gamma = 0, \end{cases}$$

pour tout $x \geq 0$ lorsque $\gamma \geq 0$, ou $x \geq 0$ et $x \leq -\frac{\sigma}{\gamma}$ lorsque $\gamma < 0$. Le paramètre $\sigma > 0$ est le paramètre d'échelle, et $\gamma \in \mathbb{R}$ est le paramètre de forme. Comme représenté dans la figure 2.6 la forme de la queue de la distribution varie en fonction de la valeur de γ et de σ. Il est intéressant de noté que lorsque $\gamma \to 0$, la GPD converge vers une distribution

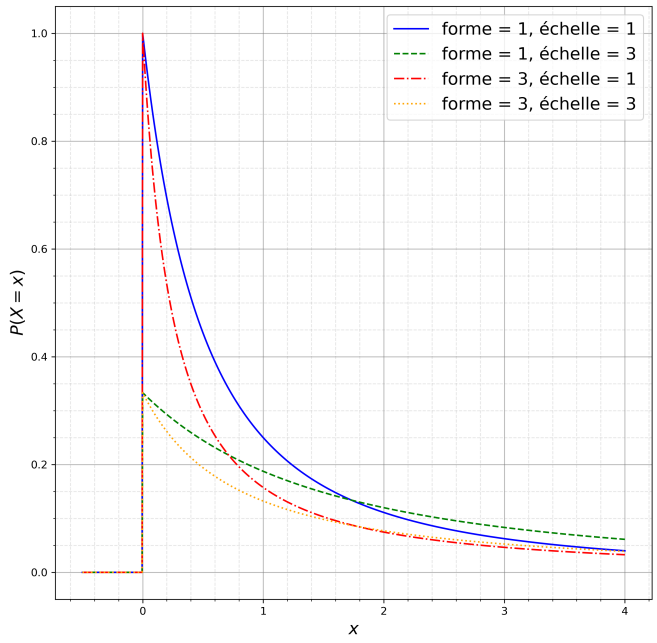


FIGURE 2.6 – Distribution généralisée de Pareto en fonction des paramètres de forme γ et d'échelle σ.

exponentielle de moyenne σ ; lorsque $\gamma = -1$, la GPD est égale à une distribution uniforme $U(0, \sigma)$; lorsque $\gamma = -1/2$, la GPD est égale à une distribution triangulaire ; et lorsque $\gamma > 0$, c'est la distribution de Pareto qui est obtenue.

Domaine Maximum d'attraction. On dit que F se situe dans le domaine maximum d'attraction de G_γ si la distribution des valeurs extrêmes de la distribution F, converge asymptotiquement vers la distribution de G_γ.

Théorème de Pickands, Balkema et De Haan. Soit $(X_k)_{k\in\mathbb{N}}$ une suite de variables aléatoires non bornées iid avec $F(x)$ comme fonction de répartition. Notons u un seuil au-delà duquel les observations sont considérées comme extrêmes. Dans ce cas, la distribution de l'excédent au-delà de u est donnée par

$$F_u(x) = \mathbb{P}(X - u \leq x \mid X > u) = \frac{F(u+x) - F(u)}{\bar{F}(u)} \tag{2.10}$$

où $\bar{F}(u) = 1 - F(u)$ est la probabilité d'excès. Sachant cela, établissons alors un bref parallèle avec la méthode des blocs maxima. Dans celle-ci, on peut montrer que les maxima de chaque bloc convergent vers une distribution GEV. Le théorème de Pickands, Balkema et de Haan, ou deuxième théorème des valeurs extrêmes, établit le lien entre la distribution GEV et distribution généralisée de Pareto décrite précédemment. En effet, Pickands 1975 montre que :

Theorem 2

Soit F une distribution contenue dans le domaine maximum d'attraction de la distribution GEV G_γ, où $\gamma \in \mathbb{R}$ est le paramètre de forme. Supposons que le paramètre d'échelle $\sigma(u) > 0$ soit mesurable en fonction du seuil u choisi. Si l'on note $\mathrm{GPD}_{(\gamma,\sigma(u))}$ la distribution généralisée de Pareto (GPD), on a alors :

$$\sup_{x\geq 0} |F_u(x) - \mathrm{GPD}_{\gamma,\sigma(u)}(x)| \to 0 \text{ for } u \to \infty$$

En d'autres termes, le théorème de Pickands, Balkema et De Haan, ou deuxième théorème des valeurs extrêmes, indique que si F se situe dans le domaine maximum d'attraction de G_γ, alors pour une valeur de u suffisamment grande, l'excédent de cette limite dont la distribution est donnée par $F_u(x)$ converge asymptotiquement vers la famille des distributions généralisées de Pareto.

Excédant

Soit $(X_k)_{k\in\mathbb{N}}$ une suite de variables aléatoires non bornées iid. On peut définir l'excédent comme étant n'importe quel $X_k > u$, où u est le seuil choisi. Une représentation de cette situation est donnée dans la figure 2.7. Posons alors $K_n(u)$ comme l'ensemble des indices parmi $1, \ldots, n$ des observations qui dépassent u, c'est-à-dire, $K_n(u) = i = 1, \ldots, n : X_i > u$. Notons $N(u)$ le nombre de valeurs dépassant u dans $X_1, \ldots, X_n$ (donc $N(u) = \#K_n(u)$). Dès lors, l'excès Z_j avec $j = 1, \ldots N(u)$ est défini comme étant :

$$\left\{Z_1, \ldots, Z_{N(u)}\right\} = \{X_i - u; i \in K_n(u)\} = \left\{X^{(1)} - u, \ldots, X^{(N(u))} - u\right\} \tag{2.11}$$

$Z_1, \ldots, Z_{N(u)}$ sont alors iid avec comme distribution F_u définie dans l'équation (2.10), étant donné que $N(u)$ est aléatoire. Ainsi, d'après le théorème 2, on peut conclure que pour un seuil u suffisamment grand, la distribution de $F_u(x)$ peut être approchée par une

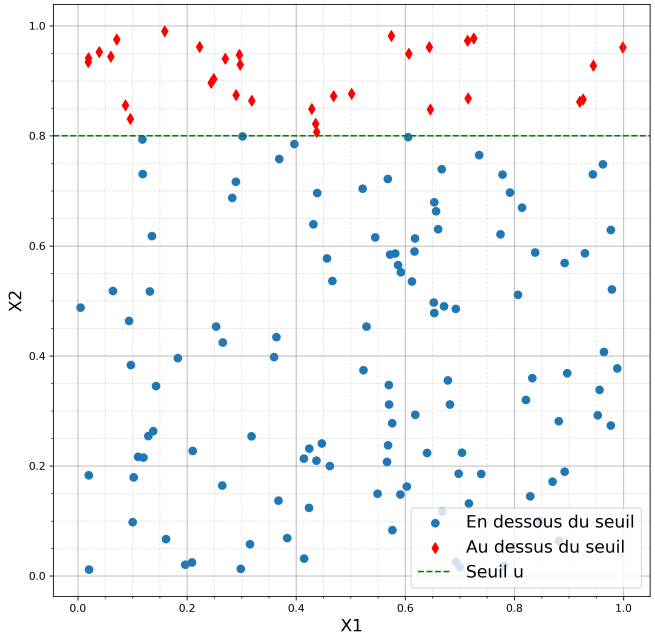


FIGURE 2.7 – Visualisation de la méthode des excédants. 150 points on été généré de manière aléatoire, puis un seuil u défini à 0.8 a été défini. Toutes valeurs plus grandes que ce seuil font partie des excédents. C'est la distribution de ces valeurs qui sera modélisée via une GPD.

$\text{GPD}_{\gamma,\sigma(u)}(x)$. Résumons donc en quelques mots le problème : On cherche à estimer la probabilité qu'une valeur soit plus grande que x, où x est très grand. Cette probabilité d'excès est notée $\bar{F}(x)$. Pour faire cette estimation, on pourrait naïvement considérer la distribution empirique $\hat{\bar{F}}_n(x)$. Cependant, lorsque x devient très grand, la fonction de distribution empirique dépendra uniquement d'un petit nombre d'observations très grandes . Cet estimateur sera donc susceptible d'avoir une grande variabilité. Pour éviter ce problème, une meilleure solution pour cet estimateur serait de déterminer une relation entre $\bar{F}(x)$ et $\bar{F}(u)$, où u serait une valeur grande, mais non-extrême, et la distribution de la GPD définie dans l'équation (2.9). La distribution de l'excès peut être alors réécrite comme :

$$\bar{F}(x) = \bar{F}(u)\bar{F}_u(x-u), \quad \text{avec } x > u.$$

Un estimateur de POT peut être trouvé en remplaçant la partie droite de cette équation par des approximations. Plus précisément, en remplaçant $F(u)$ par la fonction de distribution empirique de $F(u)$, et en utilisant l'approximation de $\bar{F}_u(x-u)$ avec l'équation (2.11), on obtient l'estimateur du POT comme défini par Embrechts 2008.

L'estimateur du POT $\hat{\bar{F}}(x)$ pour la probabilité d'excédance de $\bar{F}(x)$ pour un x très grand est :

$$\hat{\bar{F}}(x) = \frac{N(u)}{n}\hat{\text{GPD}}_{\hat{\gamma},\hat{\sigma}}(x-u) = \frac{N(u)}{n}\left\{1 + \frac{\hat{\gamma}(x-u)}{\hat{\sigma}}\right\}^{-\frac{1}{\hat{\gamma}}}$$

pour $u < x < \infty$, où $\hat{\gamma}$ et $\hat{\sigma}$ sont des estimateurs appropriés pour la GPD. Par exemple, les paramètres $\hat{\gamma}$ et $\hat{\sigma}$ peuvent être calculés par maximum de vraisemblance par rapport aux excès $Z_1, \ldots, Z_{N(u)}$ (valeurs en rouge sur la figure 2.7). En effet, prenons l'exemple où le nombre d'observations excédentaires serait égal à m et que $Z_1, \ldots, Z_m$ seraient un échantillon iid avec comme distribution une $\text{GPD}_{\gamma,\sigma}$ où $\gamma > 0$. Dans ce cas, $\text{GPD}_{\gamma,\sigma}$ est une distribution de Pareto avec une fonction de densité correspondant à :

$$\text{par}(y) = \frac{1}{\sigma}\left(1 + \frac{\gamma y}{\sigma}\right)^{-\frac{1}{\gamma}-1}, \quad x \geq 0.$$

Le log-maximum de vraisemblance est alors donné par :

$$\ell(\gamma, \sigma \mid Z_1, \ldots, Z_m) = -m\log\sigma - \left(\frac{1}{\gamma} + 1\right)\sum_{j=1}^{m}\log\left(1 + \frac{\gamma}{\sigma}Z_j\right).$$

où $Z_1, \ldots, Z_m$ sont les excès observés. En maximisant cette fonction par rapport à σ et γ, on peut facilement trouver des estimateurs $\hat{\sigma}$ et $\hat{\gamma}$. On peut également trouvé la fonction de maximum de vraisemblance pour la distribution généralisée de Pareto en acceptant que $\gamma \leq 0$. L'estimateur de quantile $\hat{x}_q$ par la méthode POT pour le quantile $x_q = F^{-1}(q)$ est la solution de l'équation $\hat{\bar{F}} = 1 - q$. C'est à dire :

$$\hat{x}_q = u + \frac{\hat{\sigma}}{\hat{\gamma}}\left[\left\{\frac{n}{N(u)}(1-q)\right\}^{-\hat{\gamma}} - 1\right].$$

Tout d'abord, l'estimateur POT dépend du seuil u choisi, qui détermine le nombre d'observations k strictement supérieures à u. On note $N(u)$ le nombre d'observations supérieures à u, et on choisit u tel que $N(u) = k > n(1-q)$ et $u = X^{(k+1)}$.

Ensuite, on calcule l'estimateur POT correspondant à ce seuil u, noté $\hat{x}_{q,k}$. Cet estimateur dépend également des estimateurs $\hat{\sigma}_k$ et $\hat{\gamma}_k$ des paramètres σ et γ, calculés par maximum de vraisemblance.

On compare cet estimateur POT à l'estimateur de quantile d'échantillon $\hat{x}_q^s$, qui correspond à la $n(1-q+1)$-ème observation.

On remarque que les deux estimateurs sont sensiblement équivalents lorsque $k = n(1-q+1)$, mais les études empiriques montrent que la valeur de seuil u qui minimise l'erreur quadratique moyenne de l'estimateur POT est largement plus grande que cette valeur.

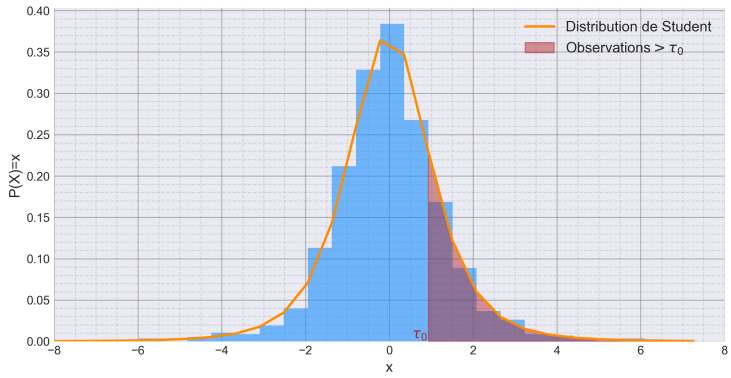
Pour trouver cette valeur optimale de seuil u, on divise l'erreur quadratique moyenne en deux composantes : la variance et le carré du biais de l'estimateur POT. On remarque que ces deux composantes évoluent dans des sens opposés lorsque u évolue. On cherche donc la valeur de seuil u qui minimise cette erreur quadratique moyenne. On se retrouve donc un dilemme : si u est trop grand, le nombre d'observation plus grand que cet excès est faible et la variance de l'estimateur est très grand, alors que si u es trop petit, l'approximation de l'excès via une distribution généralisée de Pareto n'est plus consistante, car le biais devient trop grand.

En résumé, l'estimation des quantiles extrêmes est maintenant simplifiée, puisqu'il s'agit maintenant de l'estimation d'un quantile intermédiaire, qui est relativement simple étant donné l'assez grand nombre d'observation pour utiliser des méthodes classiques empiriques (estimation de $P(Y > u)$). Cette estimation est alors corrigée via la distribution de Pareto généralisée. Une représentation de cette méthode est donnée dans la figure 2.8

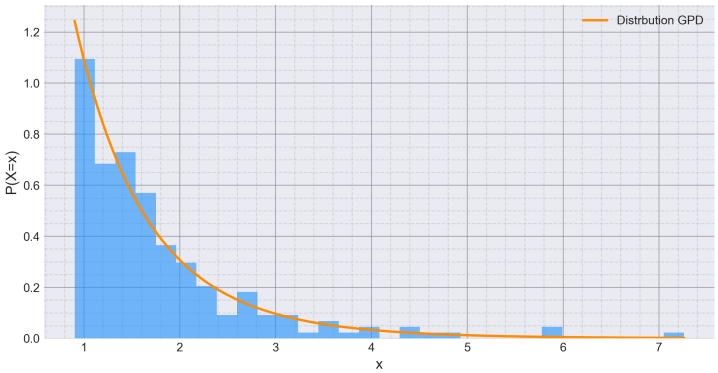
2.3 Forêt aléatoire pour la régression quantile

2.3.1 Arbre décisionnel

Afin d'appréhender les fondements des forêts aléatoires, il est primordial d'étudier d'abord leur base structurelle : les arbres de décision. Les arbres de décisions sont des modèles dont la structure est représenté comme son nom l'indique par un arbre représenté dans la figure 2.9 à gauche. Il est caractérisé par un nœud principal, appelé "racine de l'arbre" de "branches" qui divisent chaque nœud en deux, et de "feuilles" qui sont des nœuds qui ne seront plus scindé par la suite. Les arbres de décision ont été développé de manière indépendante dans le domaine des statistiques par L. Breiman et al. 1984, qui décrivent en détail le fonctionnement de cet méthode et donne également une description de l'algorithme CART 2 en se basant sur les travaux de Kass 1980 et dans le domaine du machine learning, ou Hunt, Stone et Marin 1966 pose les fondements de l'arbre décisionnel, pour ensuite être spécifiquement utilisé par Quinlan 1983 dans un contexte de résolution du jeu d'échec et enfin, Murthy 1998 centralise tous les travaux de recherches ayant été réalisé pour en donner une description approfondie dans les différents champs d'applications dans lesquelles ils ont été utilisés. Les arbres de décision représentent l'une des méthodes les plus intuitives d'effectuer des prédictions. Notamment, lors de la prédiction, ils sont essentiellement une liste de conditions "si-alors" que l'on applique en descendant le long de l'arbre. L'établissement de ces conditions est intrinsèquement lié à la tâche concernée, qu'il s'agisse d'un problème de classification ou de régression. Dans le cadre de cette thèse, la sortie étant numérique, nous sommes amenés à utiliser des arbres de régression que l'on notera T. Comme nous le verrons ultérieurement, ceci a un impact direct sur la fonction



(a) Distribution de Student, avec les quantiles supérieurs à 0.9



(b) Ajustement d'une GPD sur les observations excédants un quantile intermédiaire τ_0

FIGURE 2.8 – Exemple d'arbre de regression pour un problème en 2 dimensions

de perte que nous minimiserons.

Afin de mieux appréhender les arbres de régression, considérons l'exemple simple de la figure 2.9. Le graphique de gauche de la figure présente l'arbre de régression ainsi que la décision à prendre pour obtenir la sortie numérique finale. Le graphique de droite montre les régions de décision propre à cet arbre de décision. Pour la construction de cette arbre, un jeu de donnée a été sélectionné de manière aléatoire. Ce jeu de donnée étant composé de deux covariables et d'une variable dépendante. Un arbre décisionnel a ensuite été ajusté par rapport à ces données ce qui donne les résultats obtenus dans la figure 2.9 à droite. Un interprétation à ce graphique de décision, est que si une variable x_1 est inférieur à 0.4, alors si la variable x_2 pour cette même observation est inférieur à 0.4, alors la variable dépendente sera égal à 0.1 et sinon elle sera égal à 0.4.

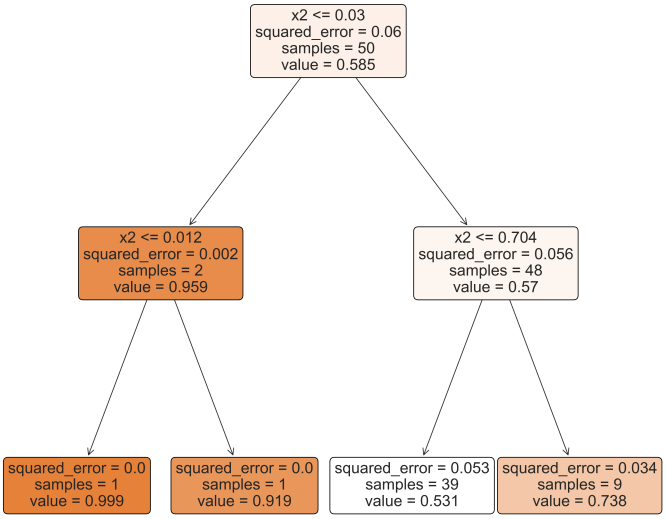
La question centrale qui se pose maintenant est celle de la méthode permettant d'effectuer ces séparations. C'est à ce niveau que la fonction de perte entre en jeu. Comme dans le cas des régressions linéaires, nous pouvons adopter la mesure de l'Erreur Quadratique Moyenne (MSE). Soit $Y_i \in \mathbb{R}$ une variable dépendante à prédire, où $i = 1, \ldots, n$ avec n le nombre d'observations à prédire, et soit $\hat{Y}_i$ la valeur prédite par le modèle pour la même observation i, la mesure de la MSE est alors donné par :

$$SSR = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i \right)^2, \tag{2.12}$$

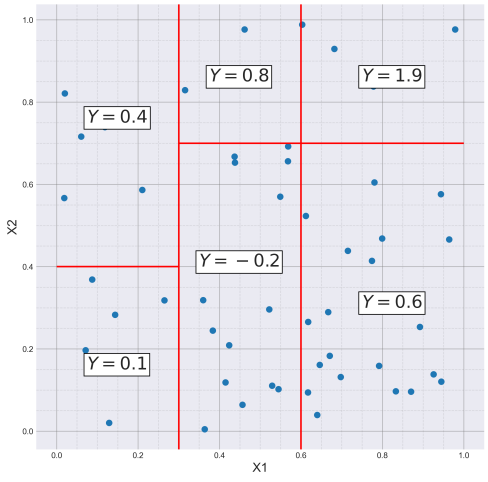
Pour chaque variable choisie aléatoirement de manière univariée, la séparation est effectuée de manière à minimiser la fonction de perte. Cependant, il est essentiel de prendre une décision concernant l'arrêt de la séparation afin d'éviter que le modèle ne soit trop ajusté aux données d'entraînement, ce qui peut conduire à un sur-apprentissage et à une diminution en la capacité de généralisation du modèle. Pour éviter cela, de nombreuses techniques d'élagage et de complexification de l'arbre existent.

En ce qui concerne l'élagage, deux approches sont généralement employées : le pré-élagage et le post-élagage. Le pré-élagage consiste à instaurer des limites à priori avant l'entraînement de l'arbre, permettant ainsi d'arrêter son développement en fonction de certains critères. Ces critères peuvent comprendre la profondeur maximale d_{max}, la taille minimale d'échantillon requis pour effectuer une séparation N_{min}, et la taille minimale d'échantillon dans un nœud feuille n_{min}. À l'opposé, le post-élagage intervient après la création de l'arbre et vise à en réduire la complexité. Deux stratégies principales sont utilisées dans ce contexte : L'une est l'élagage par réduction d'erreur : ce procédé implique de supprimer les feuilles de l'arbre et de vérifier l'impact sur la fonction d'erreur. Si la suppression de certaines feuilles ne détériore pas significativement la performance de l'arbre, alors ces feuilles sont définitivement retirées. L'autre est l'élagage basé sur la complexité : cette méthode définit la complexité de l'arbre en fonction du nombre de nœuds $|T|$, ainsi qu'un paramètre de pénalisation α. Ce dernier permet de contrôler l'équilibre entre la taille de l'arbre (que l'on cherche à minimiser) et la qualité de l'ajustement (que l'on cherche à maximiser).

Bien que les méthodes évoquées précédemment possèdent leurs avantages, elles présentent néanmoins certaines limitations. D'abord, elles requièrent souvent l'adoption d'une approche qualifiée de "gourmande", qui implique d'explorer une multitude de combinaisons de paramètres via une validation croisée avant de sélectionner la plus performante. Lorsque



(a) Arbre de régression



(b) Régions de décision

FIGURE 2.9 – Exemple d'arbre de régression et de régions de décisions pour un problème en 2 dimensions. Notez que les observations ont été générées de manière aléatoire et qu'il n'y a pas de correspondance entre les observations du graphe de gauche et de droite.

le nombre d'observations et de variables est limité, cette méthode reste praticable. Cependant, à mesure que ces nombres augmentent, la complexité computationnelle devient prohibitive. De plus, les arbres de décision présentent une grande sensibilité aux variations introduites par de nouvelles observations. Ils ont également une propension à sur-ajuster les données, un phénomène qui peut compromettre leur performance sur de nouvelles données.

C'est dans ce contexte que les méthodes d'ensemble d'arbres, telles que les forêts aléatoires ou le boosting, entrent en jeu. Le principe sous-jacent à ces techniques repose sur le concept de "sagesse des foules". En somme, il s'agit de combiner les prédictions de plusieurs modèles simples (ou "apprenants faibles") afin de constituer un modèle plus robuste (ou "apprenant fort"). Ces techniques d'ensemble permettent généralement d'améliorer la précision des prédictions tout en atténuant les problèmes d'instabilité et de surajustement.

2.3.2 Méthodes d'agrégation d'arbres de décision

Il existe différentes techniques pour combiner plusieurs arbres de décision afin de créer des modèles plus robustes, nous discuterons ici principalement de la méthode du Bootstrap Aggregating (Bagging) et de Boosting, qui feront une bonne introduction aux forêts aléatoires et au gradient boosting.

Bootstrap Aggregating (Bagging)

La technique du Bootstrap Aggregating, souvent abrégée en Bagging, a été développée par Leo Breiman 1996. Elle peut être décomposée en trois étapes principales :

1. **Bootstrapping** : Cette étape consiste à échantillonner aléatoirement n observations parmi les N observations initiales. Cet échantillonnage est effectué avec remplacement et est répété B fois, où B est un hyperparamètre du modèle.
2. **Entraînement** : Chaque ensemble échantillonné dans l'étape précédente est utilisé pour entraîner un arbre de décision distinct.
3. **Agrégation** : Dans le contexte de la régression, cette étape consiste généralement à calculer la moyenne des prédictions des différents arbres entraînés, produisant ainsi un apprenant fort à partir de plusieurs apprenants faibles.

Il convient de noter que l'ajustement des paramètres de l'arbre de décision reste possible lors de l'utilisation de la méthode de Bagging. Cela améliore généralement les performances en réduisant la variance et le surapprentissage.

Boosting

Le boosting, développé parallèlement au bagging par Freund et Schapire 1996, constitue une autre méthode robuste de sélection aléatoire d'observations et de variables. Similaires sur quelques points, ces deux méthodes ont été largement comparées empiriquement par Dietterich 1998. Structurellement, le boosting se distingue du bagging par une caractéristique majeure : dans cette architecture, si l'on considère les données d'entraînement $((\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n))$ avec $\boldsymbol{X}_i \in \mathbb{R}^p$ et $Y_i \in \mathbb{R}$ et n le nombre d'observations. La

probabilité de sélectionner la paire $(Y_i, \boldsymbol{X}_i)$ est conditionnée par la performance de prédiction du modèle précédent. Plus spécifiquement, si la qualité de prédiction d'un modèle pour une observation donnée est médiocre, la probabilité de sélection de cette observation pour la construction du modèle subséquent sera accrue. Cette stratégie confère ainsi une importance accrue aux observations délicates à prédire. À l'inverse, si une observation est prédite avec précision, elle aura moins de chances d'être intégrée dans le modèle subséquent. Les poids associés aux observations sont réajustés après chaque itération de construction d'arbre.

2.3.3 Forêt aléatoire

Le concept de forêt aléatoire a d'abord été initié par Ho 1995 qui présente l'idée de sous-espace aléatoire. Ensuite, c'est L. Breiman 2001 qui étend et formalise cette méthode d'ensemble d'arbres. L'approche des forêts aléatoires est de générer un grand nombre d'arbres, pour lesquels un sous-ensemble aléatoire de variables explicatives est choisi pour effectuer la séparation des données à chaque nœud. Le caractère aléatoire des forêts aléatoires ne se limite pas au choix des variables, car une méthode de bagging (voir section 2.3.2) est également appliquée sur les observations du jeu de données d'entraînement. En d'autres termes, un bootstrap est appliqué sur les observations. À chaque arbre, un certain nombre d'observations est sélectionné, avec remise. On notera ce nombre d'arbres B. La prédiction d'une nouvelle observation sera tout simplement la moyenne des prédictions de l'ensemble des arbres.

Plus formellement, posons θ un vecteur de paramètres qui engendre un certain arbre noté $T(\theta)$. Soit $\mathbb{B}$ l'espace de $\boldsymbol{X}$ tel que $\boldsymbol{X} : \Omega \mapsto \mathbb{B} \subseteq \mathbb{R}^p$, où $p \in \mathbb{N}_+$ est le nombre de variables explicatives. Chaque feuille $l = 1, \ldots, L$ de l'arbre correspond à un sous-espace de $\mathbb{B}$. Soit le sous-espace rectangulaire $R_l \subseteq \mathbb{B}$ pour chaque feuille $l = 1, \ldots, L$. Pour chaque $x \in \mathbb{B}$, il existe une feuille unique l tel que $x \in R_l$ qui correspond à la feuille dans laquelle se retrouve l'observation x. Cette feuille sera dès lors notée $l(x, \theta)$ pour l'arbre $T(\theta)$. Une prédiction d'un certain arbre $T(\theta)$ pour un nouveau point $X = x$ est la moyenne de tous les point observés dans la feuille $l(x, \theta)$. Notons maintenant un poids positif et constant $w_i(x, \theta)$, si l'observation X_i appartient à la feuille $l(x, \theta)$ et 0 si cette observation n'appartient pas à cette feuille. Le poids pour une certaine observation est alors donné comme :

$$w_i(x, \theta) = \frac{1_{X_i \in R_{l(x,\theta)}}}{\#\left\{j : X_j \in R_{l(x,\theta)}\right\}}$$

La prédiction d'un certain arbre $T(\theta)$, est donc la moyenne pondérée des observations originales $Y_i, i = 1, \ldots, n$:

$$\text{pour un arbre : } \hat{\mu}(x) = \sum_{i=1}^{n} w_i(x, \theta) Y_i$$

Dès lors, en appliquant maintenant une forêt aléatoire, la moyenne conditionnelle $\mathbb{E}(Y|X = x)$ est donné par la moyenne des prédictions des k arbres indépendants. Soit $w_i(x)$, la moyenne des k arbres, on a alors :

$$w_i(x) = k^{-1} \sum_{t=1}^{k} w_i(x, \theta_t).$$

La prédiction de la forêt aléatoire est alors donné par :

$$\text{Forêt aléatoire : } \hat{\mu} = \sum_{i=1}^{n} w_i(x) Y_i$$

L'approximation de la moyenne conditionnelle de Y, selon $X = x$ est donc donnée par la somme pondérée des observations. Les poids varient, et tendent à être élevés pour les observations $i \in \{i, \ldots, n\}$ pour laquelle la distribution de $Y|X = X_i$ est similaire à la distribution de $Y|X = x$ (Lin et Jeon 2006).

Maintenant que l'on a exposé comment l'algorithme de forêt aléatoire permet d'estimer l'espérance conditionnelle $\mathbb{E}(Y|X = x)$ par une moyenne pondérée des observations de la variable réponse Y. On peut maintenant s'appuyer sur cette idée et étendre cette méthode pour approximer également la distribution conditionnelle complète de Y étant donné $X = x$. La distribution conditionnelle de Y étant donné $X = x$ est définie comme :

$$F(y|X = x) = P(Y \leq y|X = x) = \mathbb{E}(1_{\{Y \leq y\}}|X = x)$$

Par analogie à l'approximation de la moyenne conditionnelle $\mathbb{E}(Y|X = x)$ donnée via la forêt aléatoire par $E(1_{Y \leq y}|X = x)$, ont peut approximer la distribution conditionnelle de Y par

$$\hat{F}(y|X = x) = \sum_{i=1}^{n} w_i(x) 1_{\{Y_i \leq y\}}$$

en utilisant les mêmes poids $w_i(x)$ précédemment décrits pour les forêts aléatoire non-conditionnelles. L'algorithme pour calculer l'estimation $\hat{F}(y|X = x)$ peut être décrit comme suit :

Algorithm 1 Forêt aléatoire pour quantiles

(a) Construire k arbres $T(\theta_t), t = 1, \ldots, k$. Dans chaque feuille, garder toutes les observations dans cette feuille, et non la moyenne.

(b) Pour une observation $X = x$, prédire le résultat de chaque arbre. Calculer le poids $w_i(x, \theta_t)$ de l'observation pour chaque arbre. Calculer le poids $w_i(x)$ pour chaque observation comme étant la moyenne des poids $w_i(x, \theta_t)$.

(c) Déterminer l'estimation de la fonction de distribution pour chaque $y \in \mathbb{R}$, en utilisant les poids de l'étape b).

.

L'estimation de $\hat{Q}_\alpha(x)$ du quantile conditionnel $Q_\alpha(x)$ est obtenu en replacant $F(y|X = x)$ par $\hat{F}(y|X = x)$ dans la formule de la fonction de distribution du quantile α : $Q_\alpha(x) = \inf\{y : F(y|X = x) \geq \alpha\}$. La différence principale entre la forêt aléatoire pour la regression quantile et la forêt aléatoire classique est que dans le cas de la regression quantile, chaque arbre garde les valeurs de toutes les observations de chaque feuille et non uniquement leur moyenne, et évalue la distribution conditionnelle basée sur cette information. La consistance de cette méthode a été montrée par Meinshausen 2006.

2.3.4 Forêt aléatoire généralisée pour la régression quantile

Dans le présent travail, nous faisons appel à une méthode de forêt aléatoire généralisé, suggérée par Wager et Athey 2017, qui introduit des fonctions de perte spécifiques à la régression quantile, permettant ainsi une estimation plus précise des quantiles conditionnels. La méthodologie proposée est très semblable à celle décrite précédemment, mais inclut quelques modifications essentielles. La procédure de modélisation comprend les trois étapes suivantes :

1. Un rééchantillonnage de l'ensemble de données initial est effectué pour obtenir divers échantillons distincts.
2. Sur chaque échantillon, un arbre de décision est construit en appliquant une division binaire récursive pour segmenter l'espace en plusieurs régions, chaque région étant représentée par une feuille
3. Pour une nouvelle observation x, et pour un nombre prédéterminé d'arbres B, le modèle génère une prédiction en calculant d'abord les poids basés sur les partitions des arbres $w_i(x) = \frac{1}{B}\sum_{b=1}^{B} w_{i,b}(x)$ et ensuite en minimisant la fonction de perte pondérée :

$$\hat{Q}_x(\tau) = \arg\min_q \sum_{i=1}^{n} w_i(x)\rho_\tau(Y_i - q).$$

La nuance principale se situe à cette étape. En effet, contrairement à la régression linéaire classique qui minimise la somme des carrés des erreurs pondérées, cette approche se concentre sur la minimisation d'une fonction qui intègre la fonction check présentée dans la sous-section 2.1.2. Plus de détails concernant chacune des étapes sont donnés ci-dessous.

Ré-échantillonnage

Pour le ré-échantillonnage, l'on utilise dans la méthode de la forêt aléatoire généralisée le sous-échantillonnage plutôt que le bootstrap, ce qui facilite l'analyse théorique du modèle et garantit une normalité asymptotique pour un faible taux de sous-échantillonnage. Le sous-échantillonnage signifie qu'un ensemble de données plus petit est produit en sélectionnant aléatoirement s observations parmi n (sans remplacement contrairement au bootstrap). Le rapport $\frac{s}{n}$ est appelé la fraction de sous-échantillonnage.

En effet, une efficacité computationnelle accrue dans les techniques de sous-échantillonnage a été théoriquement démontrée par Bühlmann et B. Yu 2002, tout en conservant une précision comparable à celle des méthodes de bootstrap. De surcroît, il a été mis en lumière par Mentch et Hooker 2015 que les ensembles construits à partir de sous-échantillons peuvent être traités comme des statistiques uniformes, ouvrant la voie au calcul de distributions limites pour les prédictions.

L'efficacité de cette approche, particulièrement lors de l'examen de données de grande taille et complexes, est attestée par les travaux de Scornet, Biau et Vert 2015. De plus, Strobl et al. 2007 a constaté l'efficacité des méthodes de sélection de variables lors de l'utilisation de techniques de sous-échantillonnage, et ce, même lorsque les variables d'intérêt présentent une grande variabilité dans le nombre de catégories pour les variables catégorielles, ou lorsque les échelles sont très différentes entre différentes variables continues.

Finalement, Wager et Athey 2017 établit dans son premier théorème que, par l'intermédiaire d'une méthode de sous-échantillonnage, la forêt aléatoire est asymptotiquement Gaussienne. Pour saisir pleinement la portée de ce théorème, présenté dans le théorème 2.3.4, il est essentiel d'appréhender plusieurs notions. Supposons que nous ayons n observations indépendantes et identiquement distribuées $Z_i = (X_i, Y_i) \in [0,1]^d \times \mathbb{R}$. Posons alors plusieurs définitions relatives aux arbres décisionnels :

Definition 1. Honnêteté

Un arbre est dit *honnête* si, pour chaque observation Z_i utilisée durant la phase d'apprentissage, la variable réponse Y_i est soit employée pour établir les poids, soit pour déterminer le seuil de scission d'un nœud. Notons toutefois que Y_i n'est jamais utilisé simultanément pour ces deux tâches.

Definition 2. Division aléatoire

Un arbre est dit *random-split* (ou "arbre à divisions aléatoires") si à chaque étape de sa construction, en marginalisant sur ξ (un terme de bruit aléatoire), la probabilité que la prochaine division s'opère selon la $j^{\text{ème}}$ variable est au moins $\frac{\pi}{p}$ pour un certain $0 < \pi \leq 1$, pour tout $j = 1, ..., p$, où p est le nombre de variables indépendantes.

Definition 3. α-régularité

Un arbre est dit *α-régulier* si chaque division d'un nœud conserve au moins une fraction α des observations d'apprentissage disponibles de chaque côté de la division. Par ailleurs, pour une profondeur d'arbre déterminée k, on retrouve au minimum entre k et $2k-1$ observations dans chaque nœud terminal. Cette notion contribue à générer des arbres de décision à la fois robustes et résistants au sur-apprentissage.

Definition 4. Symétrie

Un arbre est dit *symétrique* si la sortie de celui-ci est indépendante de l'ordre ($i = 1, 2 \ldots$) dans lequel les exemples d'apprentissage sont indexés. Autrement dit, indépendamment de l'ordre d'entrée des données d'apprentissage, la prédiction sera toujours la même.

Maintenant que ces différents concepts ont été défini, voici le théorème démontré par Wager et Athey 2017.

Theorem 3

Soit n observations indépendantes et identiquement distribuées $Z_i = (X_i, Y_i) \in [0,1]^d \times \mathbb{R}$. Supposons que les covariables X_i suivent une distribution uniforme $X_i \sim U\left([0,1]^d\right)$. Supposons en outre que

$$\mu(x) = \mathbb{E}[Y \mid X = x],$$
$$\mu_2(x) = \mathbb{E}\left[Y^2 \mid X = x\right],$$

soient des fonctions Lipschitz-continues. Enfin, supposons que

$$\mathrm{Var}[Y \mid X = x] > 0,$$
$$\mathbb{E}\left[|Y - \mathbb{E}[Y \mid X = x]|^{2+\delta} \mid X = x\right] \leq M,$$

pour certaines constantes $\delta, M > 0$, uniformément pour tous les $x \in [0,1]^d$.

Soit T un arbre aléatoire à division aléatoire honnête, α-régulier avec $\alpha \leq 0.2$, et symétrique. Considérons $\hat{\mu}_n(x)$ comme l'estimation de $\mu(x)$ fournie par une forêt aléatoire avec un apprenant de base T et une taille d'échantillon s_n. Supposons que la taille d'échantillon s_n évolue de telle sorte que

$$s_n \asymp n^{\beta} \text{ pour } \beta_{\min} := 1 - \left(1 + \frac{d}{\pi} \frac{\log\left(\alpha^{-1}\right)}{\log\left((1-\alpha)^{-1}\right)}\right)^{-1} < \beta < 1. \tag{2.13}$$

Dans ce cas, les prévisions de la forêt aléatoire sont asymptotiquement gaussiennes :

$$\frac{\hat{\mu}_n(x) - \mu(x)}{\sigma_n(x)} \Rightarrow \mathcal{N}(0,1) \text{ pour une séquence } \sigma_n(x) \to 0. \tag{2.14}$$

Construction de l'arbre

L'obtention de tous les quantiles pour chaque scission imposant une charge de calcul trop importante, comme montré par Athey, Tibshirani et Wager 2018, une stratégie différente de celle de la régression linéaire classique est employée :

Au préalable, la variable Y est transformée en une variable catégorielle comprenant K catégories, basées sur l'ordre des quantiles $(\tau_1 \leq \tau_2, \ldots, \tau_{K-1})$. Soit P le nœud parent d'un autre nœud que l'on envisage de scinder, et soit $\hat{Q}_{Y|\boldsymbol{X} \in P}(\tau_k)$ le quantile empirique de Y_i pour un certain $\boldsymbol{X}_i \in P$. On définit alors :

$$S_i = \sum_{k=1}^{K} \mathbb{1}(Y_i \leq \hat{Q}_{Y|\boldsymbol{X} \in P}(\tau_k)).$$

La scission de P en deux nœuds enfants, désignés par C_1 et C_2, est déterminée de manière à maximiser le critère multiclasse :

$$L\left(C_1, C_2\right) = \frac{\sum_{k=1}^{K}\left[\sum_{\boldsymbol{X}_i \in C_1} \mathbb{1}\left(S_i = k\right)\right]^2}{n_{C_1}} + \frac{\sum_{k=1}^{K}\left[\sum_{\boldsymbol{X}_i \in C_2} \mathbb{1}\left(S_i = k\right)\right]^2}{n_{C_2}},$$

où n_{C_l} est le nombre d'observations $\boldsymbol{X}_i \in C_l, l = 1, 2$.

Cette méthode permet d'obtenir des nœuds enfants qui sont aussi purs que possible, c'est-à-dire des nœuds où une seule catégorie domine, tout en gardant un bon équilibre entre les tailles des deux nœuds enfants.

Prédiction

Pour une entrée spécifique désignée $\boldsymbol{x}$, la prédiction correspondante, $\hat{Q}_{\boldsymbol{x}}(\tau)$, est déterminée par la minimisation de la fonction de perte ci-dessous :

$$\hat{Q}_{\boldsymbol{x}}(\tau) = \arg\min_{q} \sum_{i=1}^{n} w_i(\boldsymbol{x}) \rho_\tau(Y_i - q),$$

où $w_i(\boldsymbol{x}) = \frac{1}{B} \sum_{b=1}^{B} w_{i,b}(\boldsymbol{x})$ est une fonction de poids, définie comme la moyenne des poids $w_{i,b}(\boldsymbol{x})$ établis sur l'ensemble des arbres b de la forêt aléatoire.

Cette approche produit ce que l'on appelle une forêt aléatoire non honnête. La notion d'honnêteté dans le contexte des forêts aléatoires est intrinsèquement liée à celle des arbres de décision, tel que décrit dans la définition 2.3.4. Pour construire une forêt aléatoire honnête, Wager et Athey 2017 décrit la démarche suivante : soit le sous-échantillon D_b de taille s doit être divisé en deux ensembles $\mathcal{I}_b$ et $\mathcal{J}_b$, chacun de taille $\frac{s}{2}$. Dans cette configuration, l'arbre T_b est construit à partir de l'ensemble $\mathcal{J}_b$, tandis que les poids sont ajustés en fonction de l'ensemble $\mathcal{I}_b$ comme suit :

$$w_{i,b}(\boldsymbol{x}) = \frac{\mathbb{1}\left\{\boldsymbol{X}_i \in L_b(\boldsymbol{x}) \;\&\; i \in \mathcal{I}_b\right\}}{\sum_{k \in \mathcal{I}_b} \mathbb{1}\left\{\boldsymbol{X}_k \in L_b(\boldsymbol{x})\right\}},$$

où, $L_b(\boldsymbol{x})$ représente la feuille dans T_b qui contient $\boldsymbol{x}$ et par convention, on considère que $\frac{0}{0} = 0$. Les poids de la forêt aléatoire sont obtenus en calculant la moyenne des poids sur les B arbres de la forêt :

$$w_i(\boldsymbol{x}) = \frac{1}{B} \sum_{i=1}^{B} w_{i,b}(\boldsymbol{x}).$$

Dans le cadre de la prédiction, nous faisons appel à la stratégie dite *out-of-bag* (OOB). Cette approche consiste à calculer les poids uniquement pour les arbres de la sous-forêt $\mathcal{F}_i$ qui n'ont pas utilisé l'observation $(\boldsymbol{X}_i, Y_i)$ dans la construction de leur structure.

Le principe sous-jacent est le suivant : lorsque $\boldsymbol{x} = \boldsymbol{X}_i$, les poids $(w_k(\boldsymbol{x}))_{1 \leq k \leq n}$ peuvent conférer une importance disproportionnée à l'observation i, ce qui entraîne un impact significatif de la valeur Y_i sur l'estimation de $\hat{Q}_{\boldsymbol{x}_i}(\tau)$.

Pour clarifier, considérons $\mathcal{D}b$ comme le sous-échantillon utilisé pour la construction d'un arbre T_b. Une observation est dite "out-of-bag" par rapport au sous-échantillon $\mathcal{D}b$ si $(\boldsymbol{X}_i, Y_i) \notin \mathcal{D}_b$. Dans cette optique, les poids sont alors calculés en exploitant uniquement la sous-forêt $\mathcal{F}_i$, composée exclusivement des arbres T_b pour lesquels $(\boldsymbol{X}_i, Y_i)$ est "out-of-bag". Plus formellement, nous avons :

$$\mathcal{F}_i = \left\{T_b : (\boldsymbol{X}_i, Y_i) \notin \mathcal{D}_b\right\}.$$

Les poids ainsi obtenus sont désignés par $(w_k^{\text{oob}}(\boldsymbol{X}_i))_{1 \leq k \leq n}$ avec $w_k^{\text{oob}}(\boldsymbol{X}_i) = 0$. Par la suite, nous procédons au calcul de l'estimateur du quantile, noté $\hat{Q}^{\text{oob}}_{\boldsymbol{X}_i}(\tau)$.

2.4 Gradient Boosting

Nous abordons maintenant le gradient boosting, une variante de la méthode de boosting présentée dans la sous-section 2.3.2. La différence réside dans le fait qu'à chaque itération, la cible de prédiction du nouvel arbre de décision devient le résidu de l'arbre précédent. Autrement dit, le gradient boosting est une combinaison linéaire d'arbres de décision, chaque arbre étant pondéré par un certain paramètre que nous noterons β et qui sera ensuite décrit par la suite. Pour introduire la méthode du gradient boosting, nous allons commencer par décrire une méthode générique de boosting basée sur la descente de gradient.

2.4.1 Méthode générique de boosting basée sur la descente de gradient

Soit $X = \{x_1, \ldots, x_n\}$ un ensemble de variables indépendantes dans des données d'entraînement. Pour n'importe quel $h \in \mathcal{H}$, on va projeter h dans un espace vectoriel et le voir sous la forme $(h(x_1), \ldots, h(x_n)) \in \mathcal{Y}^n$. Dès lors, nous pouvons voir le boosting comme une technique de descente de gradient dans l'espace $\mathcal{Y}^n$. Posons alors ℓ comme une fonction de perte concave et différentiable (par exemple le carré des erreurs), nous définissons alors la fonction de perte comme :

$$\ell(f) = n\hat{\mathcal{R}}(f) = \sum_{i=1}^{n} \ell(y_i, f(x_i)).$$

Admettons que l'on a déjà formé b itération de la méthode de boosting, nous offrant un prédicteur $f_b(x)$. De ce fait, à l'itération $t+1$, nous allons ajouter un prédicteur faible suplémentaire noté h_{t+1}. On cherche donc un prédicteur faible qui minimise la fonction de perte :

$$h_{t+1} = \arg\min_{h \in \mathcal{H}} \ell(f_b + \alpha h),$$

permettant de trouver le nouveau prédicteur fort : $f_{t+1} := f_t + \alpha h$.

Approximation de Taylor

Concentrons nous maintenant sur le problème d'optimisation posé dans l'équation ci-dessus. Supposons un α fixe, alors pour n'importe quel f, on peut évaluer l'approximation de Taylor pour $\ell(f + \alpha h)$:

$$\ell(f + \alpha h) \approx \ell(f) + \alpha \langle \nabla \ell(f), h \rangle.$$

La fonction linéaire ci-dessus est une bonne approximation uniquement au voisinage de $\ell(f)$. Il est donc primordiale de choisir une petite valeur pour α. Grâce à cette approximation linéaire, on peut donc trouver une apprenant faible h qui résoud :

$$\arg\min_{h \in \mathcal{H}} \langle \nabla \ell(f), h \rangle = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} \frac{\partial \ell}{\partial \left[f(x_i) \right]} h(x_i),$$

où $\frac{\partial \ell}{\partial [f(x_i)]}$ est la dérivée de la fonction de perte $\ell(y_i, f(x_i))$ en fonction de la prédiction $f(x_i)$.

En remettant cela dans le contexte d'optimisation à l'étape $b+1$, on résout maintenant :

$$h_{b+1} = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} \underbrace{\frac{\partial \ell}{\partial \left[f\left(x_i\right)\right]}}_{r_i} h\left(x_i\right).$$

2.4.2 Fonctionnement Gradient Boosting

Maintenant que la base du principe de la méthode du gradient boosting a été expliquée, nous allons rentré plus en profondeur dans cette méthode. Le processus d'apprentissage du gradient boosting peut être décomposé en plusieurs étapes. Afin de mieux appréhender et de mieux visualiser chacune de ces étapes, des graphiques sont présentés dans dans l'annexe C.1 décrivant chacune de celles-ci.

La première étape du gradient boosting est de déterminer une première prédiction, constante que l'on note $F_0(x)$. Formellement, on note cette phase d'initialisation par

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma),$$

où L est une fonction de perte. Pour simplifier nos explications, on utilisera comme fontion de perte, l'erreur au carré notée $L = (y_i - \gamma)^2$, mais il est important de noté que d'autres fonctions de perte pourrait être utilisée. On cherche donc la valeur de γ qui minimise la somme des erreurs au carrés. Pour résoudre ce problème d'optimisation, nous allons tout donc calculer la dérivée de $\sum_{i=1}^{n} L(y_i, \gamma)$, soit

$$\begin{aligned}
\frac{\partial}{\partial \gamma} L &= \frac{\partial}{\partial \gamma} \sum_{i=1}^{n} (y_i - \gamma)^2 \\
&= -2 \sum_{i=1}^{n} (y_i - \gamma) \\
&= -2 \sum_{i=1}^{n} y_i + 2n\gamma,
\end{aligned}$$

puis déterminer la valeur de γ qui annule cette dérivée, soit

$$\begin{aligned}
&-2 \sum_{i=1}^{n} y_i + 2n\gamma = 0 \\
\iff\ & n\gamma = \sum_{i=1}^{n} y_i \\
\iff\ & \gamma = \frac{1}{n} \sum_{i=1}^{n} y_i = \bar{y}.
\end{aligned}$$

En utilisant la somme des erreurs au carré comme fonction de perte, la valeur qui minimise la fonction de perte est donc la moyenne de y. C'est pourquoi la moyenne sera utilisée pour la prédiction du modèle initiale $F_0(x)$. Donc $F_0(x) = \bar{y}$. Cette première prédiction est montrée dans la figure C.1.

Une fois que la première prédiction naïve a été réalisée, nous allons répéter plusieurs opération B fois, où B représente le nombre d'arbres qui seront construits durant cet algorithme. Les arbres sont des arbres CART décrit dans l'annexe A.1. Tout d'abord, admettons que nous trouvons à l'itération b, nous allons calculer le résidus noté $r_{i,b}$ en calculant la dérivée de la fonction de perte, en fonction de la prédiction faite à l'étape précédente F_{b-1}. Nous prendrons ensuite l'inverse de cette valeur pour se diriger à l'inverse du gradient, comme expliqué dans l'annexe B.3. Cette erreur est calculée pour chaque observation i et est en fait le gradient négatif qui indique la direction (signe de la dérivée) et la magnitude avec laquelle la fonction de perte peut être minimisée. Pour déterminer les valeurs de $r_{i,b}$ on résout donc :

$$\begin{aligned} r_{ib} &= -\left[\frac{\partial L\left(y_i, F(x_i)\right)}{\partial F(x_i)}\right]_{F(x)=F_{b-1}(x))} \\ &= -\frac{\partial\left(y_i - F_{b-1}\right)^2}{\partial F_{b-1}} \\ &= 2\left(y_i - F_{b-1}\right). \end{aligned}$$

Le facteur 2 étant constant et positif, il n'impactera jamais l'erreur $r_{i,b}$ on peut donc la retirer de l'équation. On obtient donc $r_{i,b} = y_i - F_{b-1}$. Le gradient est donc en fait donné par le résidu.

Une fois les résidus obtenus, un nouvel arbre sera entraîné à ajuster au mieux le résidu r grâce aux variables indépendantes comme présenté dans la figure C.2. Des régions $R_{j,b}$ sont alors créées, où j est une feuille d'un arbre T_b, comme représentées dans la figure 2.9b.

L'arbre b construit, on recherche ensuite la valeur qui minimise la fonction de perte de chaque feuille de l'arbre. On note cette valeur γ_{jb} où j représente une certaine feuille de l'arbre T_b. Pour cela, on va agréger la perte de toutes les observations x_i qui appartiennent à la région R_{jm}. Formellement, on recherche :

$$\begin{aligned} \gamma_{jb} &= \arg\min_{\gamma} \sum_{x_i \in R_{jb}} L\left(y_i, F_{b-1}(x_i) + \gamma\right) \\ &= \arg\min_{\gamma} \sum_{x_i \in R_{jb}} \left(y_i - F_{b-1}(x_i) - \gamma\right). \end{aligned}$$

Pour déterminer la valeur qui minimise ce problème, on égalise donc la dérivée de cette équation à 0, ce qui donne :

$$\begin{aligned} &\frac{\partial}{\partial\gamma} \sum_{x_i \in R_{jb}} \left(y_i - F_{b-1}(x_i) - \gamma\right)^2 = 0 \\ \iff &-2 \sum_{x_i \in R_{jb}} \left(y_i - F_{b-1}(x_i) - \gamma\right) = 0 \\ \iff &n_j\gamma = \sum_{x_i \in R_{jb}} \left(y_i - F_{b-1}(x_i)\right) \\ \iff &\gamma = \frac{1}{n_j} \sum_{x_i \in R_{jb}} r_{ib}, \end{aligned}$$

où n_j est le nombre d'observation dans la feuille j. La valeur optimale de γ_{jb} qui minimise la fonction de perte est donc la moyenne des résidus r_{ib} dans la région R_{bm}. En d'autres termes, γ_{bm} est la valeurs de prédiction régulières des arbres de régression qui sont la moyenne des résidus dans chaque feuille. La moyenne d'une certaine feuille j est donc utilisée comme prédiction pour toutes les observations qui tombent dans cette feuille.

La dernière étape, pour chaque itération b est de mettre à jour le modèle de la combinaison des prédiction F_b par :

$$F_b(x) = F_{b-1}(x) + \sum_{j=1}^{J_b} \gamma_{jb} \mathbb{1}(x \in R_{jb}),$$

où $\gamma_{jb}\mathbb{1}(x \in R_{jb})$ indique que l'on ne considère uniquement les valeurs de γ_{jb} si une observation x se trouve dans la région R_{jb}. Chaque région R_{jb} étant exclusive, chaque x qui se retrouve dans une certaine feuille avec un certain γ_{jb} correspondant est ajouté à la prédiction précédente, ce qui met à jour F_b. Nous observons ces étapes dans la figure C.4 et C.5.

Finalement, en fonction des données d'entraînement, ce modèle peut présenter le risque d'avoir une variance assez élevée dans ses prédictions, causée par un certain sur-ajustement des données. Pour gérer ce problème potentiel, un taux d'apprentissage ($v \in (0; 1)$) est introduit pour équilibrer la contribution du nouvel arbre à la prédiction finale. L'étape de mise à jour est donc données par

$$F_b(x) = F_{b-1}(x) + v \sum_{j=1}^{J_b} \gamma_{jb} \mathbb{1}(x \in R_{jb}).$$

Ainsi, à chaque étape, seule une légère correction de l'erreur de l'étape précédente est apportée, permettant une estimation finale avec une variance significativement réduite. Comme l'objectif du gradient boosting est de réduire l'erreur entre l'estimation et la vraie valeur à chaque itération un choix trop grand du nombre d'arbres peut entraîner un sur-ajustement. Le nombre d'arbres doit donc être considéré comme un hyper-paramètre et être choisi par validation croisée. Comme déjà exprimé en introduction du gradient boosting, la fonction de perte L ne doit pas forcément être la somme des erreurs au carré, et peut être remplacé par bien d'autres fonctions, présentant tout de même certaines conditions de continuité et de dérivabilité. Dans notre algorithme, ce sera le maximum de vraisemblance par rapport à une GPD qui sera choisie.

Plusieurs implémentations du gradient boosting ont été développée avec chacun ses propres spécificités. Les algorithmes les plus populaires comprennent l'eXtreme Gradient Boosting (XGBoost), le Light Gradient Boosting Machine (LightGBM) et le CatBoost, chacun ayant ses propres forces et faiblesses. Le XGBoost a été développé par T. Chen et Guestrin 2016 et est une optimisation du Gradient boosting, il a en effet été développé pour être parallélisable. Il a également d'autres avantages comme la possibilité de gérer les valeurs manquantes et une évaluation intégrée de l'importance des variables. Le LightGBM développé par Ke et al. 2017 est surtout apprécié pour sa capacité à gérer les grandes dimensions et les grands ensembles de données via une intégration de sélection de variables performante. Il utilise également une technique novatrice d'échantillonnage unilatéral qui contribuent à augmenter l'efficacité du modèle et à réduire l'utilisation de la mémoire. De son côté, CatBoost développé par Prokhorenkova et al. 2018 est

particulièrement adapté aux tâches de régression impliquant de nombreuses variables catégorielles. Récemment, MÄRZ 2020 ont développé le CATBOOSTLSS pour la prédiction de paramètres d'un ensemble de distributions, mais la performance n'a pas encore été montrée.

Chapitre 3

Méthodologie

3.1 Contexte de l'algorithme

L'algorithme Gbex est décrit en 2 étapes : D'abord, la prédiction des paramètres de la GPD, puis la prédiction des quantiles extrêmes. Dans ce mémoire, nous travaillerons dans un contexte où la variable indépendante notée $Y_i \in \mathbb{R}$, a une certaine dépendance avec un ensemble de covariables notées $\boldsymbol{X}_i \in \mathbb{R}^d$. L'objectif est donc de développer un estimateur pour le quantile conditionnel $Q_{\boldsymbol{x}}(\tau_n)$ défini dans la sous-section 2.1.3 par

$$Q_{\boldsymbol{x}}(\tau) = F_Y^{-1}(\tau \mid \boldsymbol{X} = \boldsymbol{x}).$$

À partir de cette équation, nous pouvons définir plusieurs scénarios dépendant de l'ordre du quantile $\tau = \tau_n$:

- la situation classique où $\tau_n \equiv \tau_0 < 1$, pour laquelle les méthodes de régression quantile traditionnelle fonctionne parfaitement.
- la situation intermédiaire où $\tau_n \to 1$ et $n(1-\tau_n) \to \infty$. Dans ce cas, $n(1-\tau_n)$ est l'espérance du nombre de points dans un échantillon de taille n plus grand que le quantile τ_n. Dans le cas intermédiaire, on a encore une espérance du nombre de points au-delà du quantile que nous étudions qui tend vers l'infini. Les méthodes empiriques d'évaluation du quantile conditionnel peuvent encore être utilisés.
- le cas extrême où $\tau_n \to 1$ et $n(1-\tau_n) \to 0$. Dans ce cas extrême, plus le quantile étudié est grand, plus l'espérance du nombre de points qui se trouve au-delà du quantile étudié est petit et tend même vers 0. Étant donné le manque d'observations dépassant ce quantile $Q_x(\tau_n)$, une estimation empirique n'est pas réalisable (estimation par maximum de vraisemblance par exemple), et une extrapolation au-delà des observations est nécessaire.

Dans ce contexte, lorsque l'on s'intéresse à un quantile d'ordre τ proche de 1 que l'on note y_τ, d'une certaine distribution Y, au lieu de s'intéresser au quantile y directement, nous pouvons introduire un quantile intermédiaire que l'on notera u. Ce u correspond au seuil défini dans la sous-section 2.2.3. En utilisant ce seuil, ainsi que la loi multiplicative nous nous intéressons maintenant à :

$$P(Y > y_\tau) = P(Y > u) \times P(Y > y_\tau \mid Y > u), \text{ où } y_\tau \geq u.$$

Grâce à cette méthode, nous augmentons le nombre d'observations que nous étudions. En effet, il y a beaucoup plus d'observations pour lesquelles $Y - u > 0$ que d'observations pour lesquelles $Y - y_\tau > 0$. Il est donc plus facile d'estimer cette quantité.

Ensuite, c'est ici que l'on introduit la théorie des valeurs extrêmes. En effet, on peut approcher la partie droite de l'équation précédente en introduisant la fonction de répartition de la distribution généralisée de Pareto avec les paramètres d'échelle et de forme $\sigma > 0$ et $\gamma \in \mathbb{R}$ définie dans l'équation (2.9) :

$$P(Y > y) \approx P(Y > u) \times (1 - \mathrm{GPD}_{\sigma,\gamma}(y - u)_+), \text{ où } y_\tau \geq u.$$

La forme développée de cette formule est donnée par :

$$P(Y > y) \approx P(Y > u) \times \left(1 + \gamma \frac{(y-u)}{\sigma}\right)^{\frac{-1}{\gamma}}, \qquad \text{où } (y-u) > 0.$$

L'estimation des quantiles extrêmes est maintenant simplifiée, puisqu'il s'agit maintenant de l'estimation d'un quantile intermédiaire, qui est relativement simple étant donné l'assez grand nombre d'observations pour utiliser des méthodes classiques empiriques (estimation de $P(Y > u)$). Cette estimation est alors corrigée par la distribution de Pareto généralisée.

Dans ce contexte, une approximation pour le quantile extrême conditionnel aux covariables, en fonction d'un certain quantile intermédiaire, et des paramètres d'échelle et de dispersion (γ, σ) de la GPD présentée dans figure 2.6 peut être trouvée. En effet, en considérant que $P(Y > y_\tau) = 1 - \tau$ et $P(Y > u) = 1 - \tau_0$, où τ_0 est le niveau de quantile intermédiaire, avec$\tau_0 \leq \tau$, on obtient une approximation pour le quantile d'ordre τ de Y donné par :

$$Q(\tau) \approx Q(\tau_0) + \sigma \frac{\left(\frac{1-\tau}{1-\tau_0}\right)^{-\gamma} - 1}{\gamma}. \tag{3.1}$$

On peut donc modéliser la queue de distribution de Y sachant $\boldsymbol{X} = \boldsymbol{x}$ via le quantile intermédiaire $Q_{\boldsymbol{x}}(\tau), \sigma(\boldsymbol{x}), \gamma(\boldsymbol{x})$. Un estimateur des quantiles conditionnels extrêmes $(\hat{Q}_{\boldsymbol{x}}(\tau))$ peut être obtenu en utilisant les estimateurs $\hat{Q}_{\boldsymbol{x}}(\tau), \hat{\sigma}(\boldsymbol{x}), \hat{\gamma}(\boldsymbol{x})$ dans l'équation (3.1). Dans la section suivante, nous expliquerons comment obtenir ces estimateurs. On a donc finalement :

$$Q_{\boldsymbol{x}}(\tau) \approx Q_{\boldsymbol{x}}(\tau_0) + \sigma(\boldsymbol{x}) \frac{\left(\frac{1-\tau}{1-\tau_0}\right)^{-\gamma(\boldsymbol{x})} - 1}{\gamma(\boldsymbol{x})}, \qquad \tau > \tau_0. \tag{3.2}$$

3.2 Gradient Boosting for extreme

Afin d'estimer les quantiles extrêmes conditionnels, nous devons donc trouver des estimateurs pour les 3 quantités $(Q_{\boldsymbol{x}}(\tau_0), \sigma(\boldsymbol{x}), \gamma(\boldsymbol{x}))$. Tout d'abord, le quantile intermédiaire $Q_{\boldsymbol{x}}(\tau_0)$, sera déterminé via une forêt aléatoire généralisée comme expliquée dans la sous-section 2.3.4. L'estimateur du quantile étant évalué via la méthode "out-of-bag", nous noterons donc cet estimateur $\hat{Q}^{\mathrm{oob}}_{\boldsymbol{x}_i}(\tau_0)$.

Pour les estimateurs de σ et γ, nous utiliserons la méthode du Gbex proposée par Velthoen et al. 2021. Commençons par décrire cette méthode. Comme expliqué dans

la section 3.1, nous devons tout d'abord déterminer les excédents de nos données, c'est à dire les observations $Z_i = \left(Y_i - \hat{Q}^{\text{oob}}_{\boldsymbol{x}_i}(\tau_0)\right), i = 1, \ldots, n$. Nous ne concentrerons ensuite uniquement sur les observations pour lesquelles $Z_i > 0$ pour nous concentrer effectivement sur les excédents. L'ordre du quantile intermédiaire sera à priori fixé à 0.8, bien que sa valeur sera discutée dans la section consacrée aux simulations. Ce choix de 0.8 est choisi suivant les recommandations de Velthoen et al. 2021 et permet de s'assurer qu'il y ai suffisamment de données excédantes afin qu'ils soit modélisés par une GPD. Pour modéliser cette GPD, l'objectif est donc d'estimer le paramètre conditionnel $\theta(\boldsymbol{x}) = (\sigma(\boldsymbol{x}), \gamma(\boldsymbol{x}))$ basé sur l'excédant $Z_1, \ldots Z_n$ basé sur un gradient boosting.

En l'absence de covariables, une méthode courante pour estimer les paramètres GPD $\theta = (\sigma, \gamma)$ est la méthode du maximum de vraisemblance, qui fournit des estimateurs asymptotiquement normaux dans le cas non-conditionnel avec $\gamma > -1/2$ (Smith 1987). De même, nous utilisons le logarithme négatif de la vraisemblance, ou de façon équivalente, la déviance de la distribution GPD comme fonction objectif de la procédure de gradient boosting. Plus précisément, la déviance pour une observation Z_i provenant d'une distribution GPD avec des paramètres $\theta(\boldsymbol{X}_i) = (\sigma(\boldsymbol{X}_i), \gamma(\boldsymbol{X}_i))$ est donnée par :

$$\ell_{Z_i}(\theta(\boldsymbol{X}_i)) = \left[(1 + \frac{1}{\gamma(\boldsymbol{X}_i)}) \log\left(1 + \gamma(\boldsymbol{X}_i)\frac{Z_i}{\sigma(\boldsymbol{X}_i)}\right) + \log \sigma(\boldsymbol{X}_i)\right] \mathbb{1}_{Z_i > 0}. \tag{3.3}$$

Comme expliqué dans la section 2.4 consacrée au gradient boosting, le point de départ de cet algorithme est une estimation initiale dite"naïve". Pour cette estimation naïve, c'est une estimation des paramètres non-conditionnel aux covariables via un maximum de vraisemblance qui sera faite, en remplaçant $\theta(\boldsymbol{X}_i)$ par θ dans la formule du maximum de vraisemblance donné dans l'équation (3.3), ce qui donne :

$$\theta_0(\boldsymbol{x}) \equiv \theta_0 = \arg\min_{\theta} \sum_{i=1}^{n} \ell_{Z_i}(\theta)$$

Cette première estimation des paramètres est donc la même pour l'ensemble des observations Z_i. Le modèle est ensuite amélioré étape par étape de manière additive (les prédictions des erreurs sont ajoutées à chaque étape à l'estimation finale). En démarrant de l'estimation $\theta_0 = (\sigma_0, \gamma_0)$, un nombre B de paire d'arbres vont être construit de manière récursive, l'un se concentrant sur le paramètre γ, l'autre sur le paramètre σ. A l'étape b de l'algorithme, le but est d'améliorer le modèle $\theta_{b-1}(\boldsymbol{x}) = (\sigma_{b-1}(\boldsymbol{x}), \gamma_{b-1}(\boldsymbol{x}))$ en ajoutant une paire d'arbres notés $(T_b^{\sigma}(\boldsymbol{x}), T_b^{\gamma}(\boldsymbol{x}))$ pour corriger les pseudo-résidus engendré par $\theta_{b-1}(\boldsymbol{x})$. Pour $b = 1, \ldots, B$, on suit donc les étapes suivantes :

1. Soit $s \in (0, 1]$ une fraction de sous-échantillonnage, on créé aléatoirement un sous ensemble de taille $[sn]$ noté S_b de l'ensemble des observations en excédent $(\boldsymbol{X}_i, Z_i)_{1 \leq i \leq n}$.
2. Sur ce sous-ensemble S_b, les pseudos-résidus sont calculés en fonction des valeurs de σ et γ estimées à l'itération précédente. Plus précisément, les résidus sont donnés par :
$$r^{\sigma}_{b,i} = \frac{\partial \ell_{Z_i}}{\partial \sigma}(\theta_{b-1}(\boldsymbol{X}_i)) \quad \text{et} \quad r^{\gamma}_{b,i} = \frac{\partial \ell_{Z_i}}{\partial \gamma}(\theta_{b-1}(\boldsymbol{X}_i)), i \in S_b.$$
3. Afin de mieux ajuster ces pseudo-résidus, deux arbres de régression, à savoir $(T_b^{\sigma}(\boldsymbol{x}), (T_b^{\gamma}(\boldsymbol{x}))$, sont construits. Ces arbres s'adaptent respectivement aux paires $(\boldsymbol{X}_i, r^{\sigma}_{b,i})_{i \in S_b}$ et

$(\boldsymbol{X}_i, r_{b,i}^{\gamma})_{i \in S_b}$. La construction de ces arbres suit l'algorithme "Classification and Regression Trees" (CART) comme détaillé dans l'algorithme 2.

Une fois l'arbre $T_b^{\sigma}(\boldsymbol{x})$ établi, il est nécessaire d'actualiser les valeurs des paramètres σ et γ pour chaque feuille de manière à minimiser la pseudo-erreur. Théoriquement, cela peut être accompli via une recherche linéaire. Pour la feuille $L_{b,j}^{\sigma}$, par exemple, la valeur mise à jour $\tilde{\xi}_{b,j}^{\sigma}$ est obtenue en minimisant la déviance, soit :

$$\xi_{b,j}^{\sigma} = \arg\min_{\xi} \sum_{\boldsymbol{X}_i \in L_{b,j}^{\sigma}} \ell_{Z_i}(\theta_{b-1}(\boldsymbol{X}_i) + \xi e_{\sigma}), \quad j = 1, \ldots, J_b^{\sigma}, \tag{3.4}$$

où $e_{\sigma} = (1, 0)$ et $e_{\gamma} = (0, 1)$ indiquent respectivement les directions de la recherche linéaire pour les paramètres σ et γ.

En pratique, la ligne de recherche peut être coûteux en puissance de calcul, mais une approximation peut être trouvée via une étape de Newton-Raphson. La méthode de Newton-Raphson est expliquée dans la sous-section B.1. Il en résulte que :

$$\tilde{\xi}_{b,j}^{\sigma} = -\frac{\sum_{\boldsymbol{X}_i \in L_{b,j}^{\sigma}} \frac{\partial \ell_{Z_i}}{\partial \sigma}(\theta_{b-1}(\boldsymbol{X}_i))}{\sum_{\boldsymbol{X}_i \in L_{b,j}^{\sigma}} \frac{\partial^2 \ell_{Z_i}}{\partial \sigma^2}(\theta_{b-1}(\boldsymbol{X}_i))}.$$

Le passage du problème de minimisation par recherche linéaire à l'approximation de Newton-Raphson se fait en utilisant la première et la deuxième dérivée de la fonction de coût par rapport au paramètre σ. La recherche linéaire est un processus coûteux en termes de puissance de calcul, car elle nécessite une évaluation répétée de la fonction de coût pour différentes valeurs de ξ. En revanche, l'approximation de Newton-Raphson utilise les dérivées de la fonction de coût pour estimer la valeur optimale de ξ.

Ainsi, pour chaque feuille $L_{b,j}^{\sigma}$ de l'arbre $T_b^{\sigma}(\boldsymbol{x})$, la valeur mise à jour $\tilde{\xi}_{b,j}^{\sigma}$ est obtenue en utilisant l'approximation de Newton-Raphson.

La première dérivée $\frac{\partial \ell_{Z_i}}{\partial \sigma}(\theta_{b-1}(\boldsymbol{X}_i))$ est l'estimation de la pente de la fonction de coût pour un point $\boldsymbol{X}_i$ dans la feuille $L_{b,j}^{\sigma}$, tandis que la deuxième dérivée $\frac{\partial^2 \ell_{Z_i}}{\partial \sigma^2}(\theta_{b-1}(\boldsymbol{X}_i))$ est l'estimation de la courbure de la fonction de coût pour le même point.

En utilisant ces estimations pour tous les points dans la feuille $L_{b,j}^{\sigma}$, nous obtenons la valeur $\tilde{\xi}_{b,j}^{\sigma}$ qui minimise la déviance pour cette feuille.

Pour le paramètre γ, l'approximation pour la feuille $L_{b,j}^{\gamma}$ est la valeur $\tilde{\xi}_{b,j}^{\gamma}$ et est donnée par la même équation que l'équation précédente, à la seule différence que l'on remplace σ par γ. Les gradients des arbres sont finalement donnés par

$$T_b^{\sigma}(\boldsymbol{x}) = \sum_{j=1}^{J_b^{\sigma}} \tilde{\xi}_{b,j}^{\sigma} \mathbb{1}_{\{\boldsymbol{x} \in L_{b,j}^{\sigma}\}} \text{ et } T_b^{\gamma}(\boldsymbol{x}) = \sum_{j=1}^{J_b^{\gamma}} \tilde{\xi}_{b,j}^{\gamma} \mathbb{1}_{\{\boldsymbol{x} \in L_{b,j}^{\gamma}\}}$$

4. Finalement, le modèle $\theta_{b-1}(\boldsymbol{x}) = (\sigma_{b-1}(\boldsymbol{x}), \gamma_{b-1}(\boldsymbol{x}))$ est mis à jour par :

$$\begin{aligned} \theta_b(\boldsymbol{x}) &= (\sigma_b(\boldsymbol{x}), \gamma_b(\boldsymbol{x})) \\ &= (\sigma_{b-1}(\boldsymbol{x}) + \lambda^{\sigma} T_b^{\sigma}(\boldsymbol{x}) \gamma_{b-1}(\boldsymbol{x}) + \lambda^{\gamma} T_b^{\gamma}(\boldsymbol{x})), \end{aligned} \tag{3.5}$$

où les paramètres $\lambda^{\gamma}, \lambda^{\sigma}$ sont les taux d'apprentissages déjà décrit dans la sous-section 2.4.2.

L'output final pour l'estimation des paramètres est donné par :

$$\hat{\sigma}(\boldsymbol{x}) = \sigma_0 + \sum_{b=1}^{B} T_b^{\sigma}(\boldsymbol{x}), \qquad \hat{\gamma}(\boldsymbol{x}) = \gamma_0 + \sum_{b=1}^{B} T_b^{\gamma}(\boldsymbol{x})$$

Il est à noter que les auteurs suggèrent l'introduction d'un "gradient clipping" pour éviter une instabilité numérique due au fait que la fonction de perte négative de la GPD n'est pas Lipschitzienne. Cela provoque une explosion potentielle du gradient.

Dire que la fonction de perte négative de la GPD n'est pas Lipschitzienne signifie qu'il n'existe pas de constante Lipschitz qui puisse majorer la variation de la fonction de perte entre deux points quelconques de son domaine. Autrement dit, des variations de la perte peuvent être arbitrairement grandes, ce qui peut causer des problèmes de stabilité numérique lors de l'optimisation de la fonction de perte. En particulier, cela peut causer une explosion du gradient de la perte, qui peut rendre l'optimisation difficile ou même impossible.

Le gradient clipping est une technique utilisée en apprentissage automatique pour résoudre le problème de gradient explosif (Mai et Johansson 2021). Le gradient explosif se produit lorsque les valeurs du gradient deviennent extrêmement grandes et peuvent perturber la convergence de l'algorithme d'optimisation. Cela consiste en une limitation des valeurs du gradient pour éviter qu'elles ne deviennent trop grandes. Cette technique est souvent utilisée en conjonction avec une fonction d'activation qui peut amplifier les valeurs du gradient, comme la fonction d'activation ReLU. L'implémentation est faite en calculant le gradient, puis en vérifiant sa norme. Si la norme du gradient dépasse un certain seuil prédéfini, toutes les valeurs du gradient sont réduites proportionnellement de manière à ce que la norme du gradient soit inférieure au seuil. Afin d'assurer la robustesse et la convergence de l'algorithme d'apprentissage, il est souvent bénéfique d'ajuster l'amplitude du gradient tout en conservant sa direction. Cette technique, connue sous le nom de "gradient clipping", peut contribuer à améliorer la stabilité du processus d'apprentissage et prévenir le sur-apprentissage, un problème courant en apprentissage automatique (Velthoen et al. 2021).

Dans le cadre particulier de notre travail, l'approximation de Newton-Raphson a été limitée en valeur absolue. Cette approche permet de restreindre l'influence des observations extrêmes sur le gradient. Nous obtenons ainsi les expressions suivantes :

$$T_b^{\sigma}(\boldsymbol{x}) = \sum_{j=1}^{J_b^{\sigma}} \operatorname{sign}(\tilde{\xi}_{b,j}^{\sigma}) \min\left(|\tilde{\xi}_{b,j}^{\sigma}|, 1\right) \mathbb{1}_{\left\{\boldsymbol{x} \in L_{b,j}^{\sigma}\right\}},$$

$$T_b^{\gamma}(\boldsymbol{x}) = \sum_{j=1}^{J_b^{\gamma}} \operatorname{sign}(\tilde{\xi}_{b,j}^{\gamma}) \min\left(|\tilde{\xi}_{b,j}^{\gamma}|, 1\right) \mathbb{1}_{\left\{\boldsymbol{x} \in L_{b,j}^{\gamma}\right\}}.$$

Selon Velthoen et al. 2021, la stabilité et la performance de l'algorithme sont améliorées en utilisant le gradient clipping. L'ensemble de cette procédure est résumé dans l'annexe à l'algorithme 4

Chapitre 4

Application du Gbex

Au terme de l'implémentation du Gbex, nous avons exploré des applications numériques pour apprécier la performance et la fiabilité de cet algorithme. Pour mener cette évaluation de manière précise et exhaustive, nous avons effectué deux séries de tests distincts.

Initialement, nous avons testé l'algorithme sur des données simulées, dont les quantiles extrêmes pouvaient être déduits avec une certaine précision. Cette étape a permis d'examiner l'influence des divers hyper-paramètres sur la performance du Gbex et d'identifier ceux qui ont un impact significatif ou non sur les résultats. En adoptant le même protocole de génération de données que celui utilisé dans VELTHOEN et al. 2021, nous avons pu faire une comparaison éclairée de nos résultats afin de vérifier leur cohérence. Par ailleurs, ce test nous a donné l'occasion d'évaluer la qualité du Gbex par rapport à une autre méthode.

Dans un second temps, après les tests sur les données simulées, nous avons employé notre algorithme sur une base de données réelles, qui sera détaillée ultérieurement. Cette phase était d'une importance cruciale, puisqu'elle a révélé le comportement de l'algorithme dans des situations plus complexes et non-prédictibles. Lors de ces tests, la forme de la fonction de quantile conditionnel ainsi que la dépendance des variables explicatives étaient inconnues à l'avance. Cet essai nous a confrontés à des contextes où le nombre d'observations n'était pas nécessairement contrôlable, fournissant un cadre d'évaluation plus réaliste et pertinent pour des conditions proches de la réalité.

4.1 Étude de simulations

4.1.1 Génération de données

Dans cette première étape consacrée aux données simulées, nous commençons par générer des échantillons indépendants à partir d'une distribution uniforme. Soit les variables explicatives, notées $\boldsymbol{X} = (X_1, \ldots, X_d)$ aussi appelés caractéristiques, sont distribuées indépendamment selon une distribution uniforme sur l'intervalle $[-1, 1]$, où $X_i \in \mathbb{R}^n$ où n est le nombre d'observation et $i = 1, \ldots, d$ où d est le nombre de caractéristiques d'un certain jeu de données, donc $\boldsymbol{X} \in \mathbb{R}^{n \times d}$. Plusieurs jeux de données sont alors générés pour représenter différents scénarios de complexité. Ces jeux de données se distinguent d'abord par le nombre d'observations choisi, on aura effectivement soit $n = 2000$ ou $n = 5000$. Les variables indépendantes sont générées en utilisant la fonction `uniform` du module

`numpy.random` dans Python, qui génère des échantillons à partir d'une distribution uniforme continue. Les jeux de données se distinguent ensuite par le choix de la variable dépendante ou variable cible. Comme expliqué plus tôt, par soucis de comparabilité, ces variables ont été construites suivant le même modèle que celui proposé par VELTHOEN et al. 2021. Plus précisément, on a pour la variable dépendante $Y \in \mathbb{R}^n$ deux scénarios possibles :

— Y suit une distribution t-Student avec 4 degrés de liberté et d'échelle

$$\sigma(\boldsymbol{x}) = 1 + \mathbb{1}(x_1 > 0),$$

pour un certain $\boldsymbol{X} = \boldsymbol{x} \in \mathbb{R}^{40}$. Le paramètre d'échelle sera donc égal à 1 ou à 2 en fonction du signe de la variable x_1. L'avantage de ce choix de modèle est qu'il génère une distribution de Y dont la queue se comporte de la même manière qu'une distribution généralisée de Pareto ayant d'échelle dépendant de x_1 et un paramètre de forme $\gamma = 1/4$ et donc constant. Dès à présent, nous nommerons les jeux de données générés de cette manière par "Modèle 1".

— Y suit une distribution Student-t avec un degré de liberté (df) dépendant de x_1, plus précisément :

$$\mathrm{df}(\boldsymbol{x}) = 7(1 + \exp(4x_1 + 1.2))^{-1} + 3,$$

et un paramètre d'échelle $\sigma(\boldsymbol{x})$ dépendant de x_1 et de x_2, plus précisément :

$$\sigma(\boldsymbol{x}) = 1 + \varphi(x_1, x_2),$$

pour un certain $\boldsymbol{X} = \boldsymbol{x} \in \mathbb{R}^{10}$. Le paramètre φ représente une fonction de densité normale bivariée, avec des paramètres de marges avec des marges $\mathbb{N}(0, 1)$ et un paramètre de corrélation de 0.9. De tels paramètres de degré de liberté et d'échelle sont choisis telle que la queue de distribution de la Student-t puisse être approchée par une distribution généralisée de Pareto, de paramètres de formes $\frac{1}{\gamma(\boldsymbol{x})} \in [0.10, 0.33]^{10}$. Dès à présent, nous nommerons les jeux de données générés de cette manière par "Modèle 2".

En résumé, nous avons donc 2 jeux de données pour le Modèle 1, l'un où $n = 2000$ l'autre où $n = 5000$, et de la même manière, nous avons 2 jeux de données pour le Modèle 2. Le nombre de caractéristiques de chaque modèle est par contre fixé à 40 pour le Modèle 1, et 10 pour le Modèle 2.

4.1.2 Démarche pour l'utilisation du Gbex

Afin de prédire au mieux les différents quantiles extrêmes, nous avons procédé en plusieurs étapes. D'abord, une étape d'optimisation des hyper-paramètres de nos divers algorithmes, ainsi qu'une évaluation de l'impact de ceux-ci sur la qualité de prédiction des quantiles extrêmes. Ensuite, nous avons procédé à l'identification des variables ayant le plus d'impact dans la construction du gradient boosting afin d'évaluer l'impact des variables sur la prédiction des paramètre de la GPD, cela dans une optique d'étude de l'interprétabilité du Gbex. Ces différentes étapes seront exposées dans les sections suivantes. Avant ceci, il est préalablement nécessaire d'introduire certaines notations pour faciliter la compréhension de notre démarche.

Comme décrit dans le chapitre 3 consacré à la méthodologie, notre algorithme fonctionne en deux étapes. Initialement, un quantile intermédiaire est prédit via une forêt aléatoire

généralisée que l'on désignera par GRF-QI. Ensuite, nous calculons l'excédent par rapport à notre variable dépendante en utilisant cette première prédiction. Puis, nous mettons en œuvre un gradient boosting afin d'ajuster une distribution généralisée de Pareto à ces excédants (Gbex). Enfin, nous prédisons la valeur des quantiles extrêmes en utilisant (3.2).

Comme expliqué dans la sous-section 4.1.1, nous avons généré des données selon deux modèles différents, avec divers nombres d'observations et de variables dépendantes. Pour s'assurer de la qualité de notre modèle, éviter les problèmes de sur-ajustement ou que l'on cherche à prédire certaines observations ayant servi à la phase d'entraînement, nous avons adopté une procédure rigoureuse : pour chaque combinaison modèle/nombre d'observations, nous avons généré trois jeux de données provenant du même modèle de création de données (Modèle 1 ou Modèle 2). Le premier ensemble, que nous nommerons désormais DE-QI (Données d'Entraînement du Quantile intermédiaire), est exclusivement destiné à l'entraînement pour l'évaluation du quantile intermédiaire via GRF-QI. Deux autres ensembles de données sont ensuite créés : DE-GB (Données d'Entraînement du Gradient Boosting) et DT-GB (Données de tests Gradient Boosting). Ces deux ensembles de même dimension que DE-QI sont utilisés pour prédire le quantile intermédiaire via GRF-QI entraîné. Notez que DE-GB $\equiv$ DT-QI où DT-QI sont les données de tests pour le quantile intermédiaire. L'une ou l'autre notation est utilisée en fonction du contexte.

Le calcul de l'excédent précédemment décrit est ensuite effectué sur ces deux ensembles. DE-GB est utilisé pour entraîner notre Gbex et DT-GB est utilisé pour prédire les paramètres de la GPD, déterminer les quantiles extrêmes via (3.2)et pour évaluer la qualité du modèle.

Pour récapituler, GRF-QI est entraîné sur DE-QI, il prédit les quantiles intermédiaires de DE-GB et DT-GB. Gbex est entraîné sur DE-GB, puis les prédictions et l'évaluation de la performance du modèle sont réalisées sur DT-GB. Pour mieux comprendre la génération de données, une représentation est données dans la figure De cette manière, on s'assure de l'indépendance de chaque observation entre les différents modèles et l'on évite ainsi tout problème de sur-ajustement. Cette méthode est en fait équivalente à la prédiction out-of-bag décrite dans la sous-section prédiction de 2.3.4, puisqu'on s'assure qu'aucune des observations sur lesquelles ont fait des prédictions n'ont servi à la construction du modèle fort. Cette stratégie a été suivie, car il n'existe pas encore de package en python qui permette une prédiction "out-of-bag" pour GRF. En pratique, plusieurs solutions sont possibles pour entraîner les différents algorithmes en fonction du nombre d'observations disponibles. Ce point sera décrit dans la partie consacrée aux données empiriques.

4.1.3 Estimation du quantile intermédiaire

Pour estimer le quantile intermédiaire, dont la valeur sera discutée ultérieurement, nous avons employé un GRF comme décrit dans la sous-section 2.3.4. Initialement, nous nous sommes concentrés sur la détermination des hyper-paramètres optimaux de ce GRF, en procédant à une validation croisée sur DE-QI en utilisant comme fonction objectif l'intégrale de l'erreur au carré (ISE).

L'intégrale de l'erreur au carré est définie comme suit :

$$\mathrm{ISE}_i = \int_{[-1,1]^d} \left(\hat{Q}^{(i)}_{\boldsymbol{x}}(\tau) - Q_{\boldsymbol{x}}(\tau) \right)^2 d\boldsymbol{x},$$

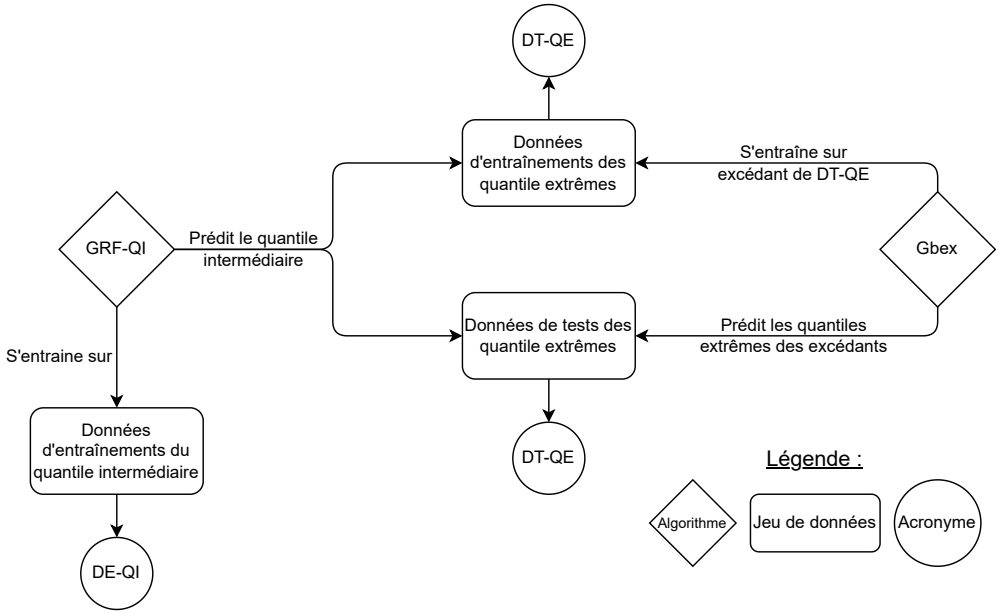


FIGURE 4.1 – Représentation des différents jeux de données utilisés pour les différents étapes du Gbex.

où $\hat{Q}_{\boldsymbol{x}}^{(i)}(\tau)$ est la prédiction du quantile intermédiaire obtenue via GRF-QI, et $Q_{\boldsymbol{x}}(\tau)$ est la valeur attendue du quantile. Pour déterminer cette dernière, nous avons simplement identifié le quantile d'une distribution Student-t correspondant à la valeur du paramètre d'échelle de l'observation en particulier en suivant le schéma donné dans 4.1.1. Une représentation graphique de cette distribution de Student est présentée dans la Figure 4.2. La valeur de l'ISE étant à la fois dépendante des données présentes dans DE-QI et des

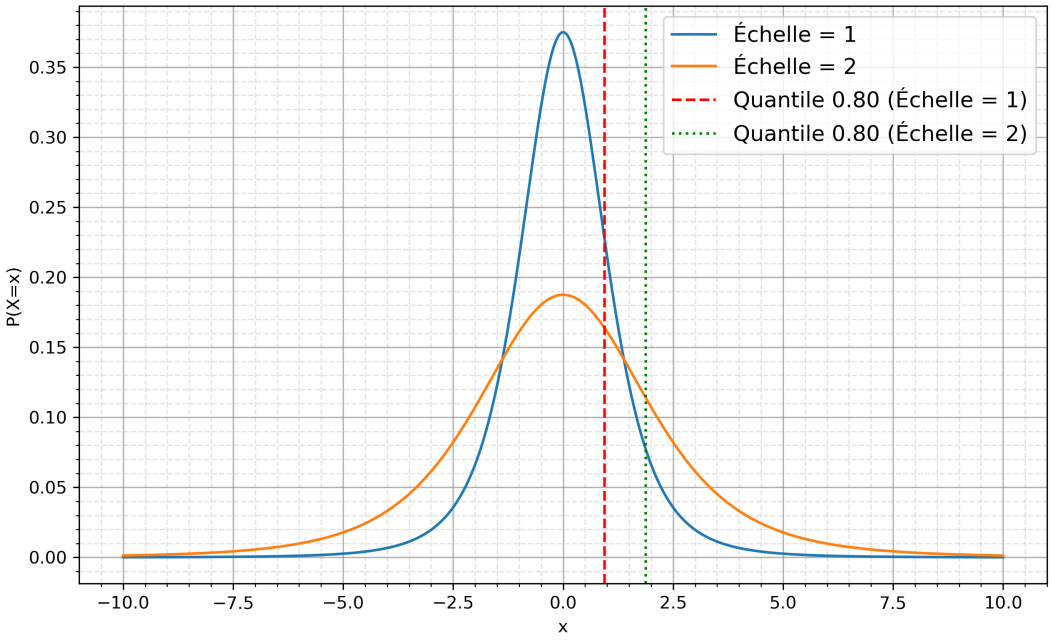


FIGURE 4.2 – Distribution de Student, avec un degré de libérté $df = 4$

données présentes dans DT-QI, nous avons donc généré 500 ensembles de données DE-QI, calculant ensuite l'ISE pour chaque DT-QI. En moyennant ensuite l'ISE pour chaque génération, nous obtenons le MISE. On a donc :

$$\text{MISE} = \frac{1}{M} \sum_{m=1}^{M} \text{ISE}_m. \tag{4.1}$$

En pratique, ne connaissant pas à priori la distribution de nos variables explicatives, cette méthode ne pourra pas être utilisé. Un autre critère d'évaluation sera alors utilisé et sera expliqué dans la section consacré au données empiriques.

Réglage des paramètres

Étant donné le grand nombre de paramètres à tester pour le GRF-QI, il aurait été compliqué de tester toutes les combinaisons, car cela est gourmand en puissance de calcul et par extension en temps. Pour optimiser cette sélection d'hyper-paramètres, on a donc réalisé un petit algorithme qui choisissait aléatoirement une valeur des paramètres parmi le domaine possible. Un ensemble de 100 ISE était alors calculé pour chacune des combinaisons testées. Cette évaluation aléatoire a été réalisée sur 30 combinaisons de paramètres réduisant largement le nombre de combinaisons total à tester.

Cette première étape a permis de déterminer une région dans laquelle la valeur de chaque hyper-paramètre serait la plus correcte. Une fois cette région déduite, une recherche par grille a été effectuée, c'est-à-dire que toutes les combinaisons possibles dans chaque région ont été testées permettant un travail de précision pour ces paramètres. Étant donné que le temps de calcul était encore long, un algorithme en multi-thread a été réalisé, permettant de diviser le temps de calcul par 4. Pour le GRF, les paramètres qui ont été étudiés pendant cette phase d'ajustement des paramètres ont été :

1. **n.estimator** : Ce paramètre spécifie le nombre d'arbres dans la forêt aléatoire. Il est crucial, car un nombre trop faible d'arbres peut causer un sous-ajustement des données, tandis qu'un trop grand nombre d'arbres pourrait entraîner un problème de sur-ajustement.
2. **min.node.size** : Il s'agit de la taille minimale d'un nœud. Cela contrôle le degré de profondeur de l'arbre, influençant ainsi la complexité du modèle. Un nœud de taille minimale plus petite peut conduire à un modèle plus complexe, qui peut être plus précis mais aussi plus susceptible de sur-ajustement.
3. **mtry** : Ce paramètre représente le nombre de variables à considérer à chaque scission d'un nœud. Il est essentiel, car il contrôle la diversité des arbres dans le modèle. Une valeur trop faible peut conduire à des arbres trop corrélés, réduisant l'efficacité du modèle, tandis qu'une valeur trop élevée peut conduire à des arbres trop complexes.
4. **sample.fraction** : Il détermine la fraction des données utilisée pour entraîner chaque arbre. Un fractionnement adéquat permet de garantir la diversité parmi les arbres du modèle et de prévenir le sur-ajustement.
5. **honesty.fraction** : Ce paramètre déjà défini dans une section précédente 2.3.4 représente la fraction des données utilisée pour la division des sous-ensembles.
6. **alpha** : Ce paramètre également déjà défini dans la section 2.3.4 représente le niveau maximal d'inégalité de chaque scission. Ce paramètre a une influence directe sur la balance des arbres dans le modèle. Un alpha trop élevé peut conduire à des arbres déséquilibrés, ce qui peut entraîner une précision moindre dans la prédiction.

Notez que ce travail de réglage des paramètres a été effectué pour le modèle 1 et le modèle 2 indépendamment, le nombre d'observations étant fixé à 2000 car aucune différence significative n'a été observée lors des tests avec $n = 5000$. Cependant, les hyper-paramètres ont été évalués pour prédire le quantile intermédiaire d'ordre 0.8 mais ils n'ont pas été adaptés à tous les autres quantiles intermédiaires qui ont été testés et discutés ensuite. Il est probable que ce point soit une piste d'amélioration de notre modélisation.

4.1.4 Estimation du quantile extrême

Pour l'estimation des quantiles extrêmes, la même procédure que la section précédente a été appliquée. Comme expliqué précédemment, GRF-QI entrainé sur DE-QI a permis de faire les prédictions du quantile intermédiaire sur DT-QI, et c'est sur cet ensemble de données que l'on a entraîné Gbex. Pour cet algorithme, la même technique de recherche des hyper-paramètres qu'expliqué précédemment a été appliquée. Certains hyper-paramètres similaires à la section précédente ont été recherchés, c'est le cas pour **n.estimator**, **nmin.node.size**, et **sample.fraction**. En revanche, d'autres hyper-paramètres ont été pris en compte tels que :

1. **depth** : Ce paramètre détermine la profondeur maximale de chaque arbre dans le modèle. En contrôlant la profondeur des arbres, on peut influencer la complexité

du modèle. Un arbre plus profond sera capable de modéliser des interactions plus complexes, mais aussi plus susceptible de sur-ajuster les données d'apprentissage. En revanche, un arbre moins profond peut contribuer à une meilleure généralisation, mais au risque de sous-ajuster les données en cas de relations complexes entre les caractéristiques et la variable cible.

2. **lambda** : Ce paramètre, aussi appelé taux d'apprentissage, contrôle la vitesse à laquelle le modèle apprend. Un taux d'apprentissage plus élevé permet au modèle d'apprendre plus rapidement, mais à un risque potentiel d'ignorer des détails importants qui pourraient améliorer sa performance. En revanche, un taux d'apprentissage plus faible augmente la chance de trouver une solution optimale en permettant au modèle d'explorer soigneusement l'espace de recherche, mais au coût d'un temps de calcul plus long. Des précisions sur ce paramètre ont été données dans la section consacrée au Gradient boosting 2.4.

Une fois une région pour les hyper-paramètres de Gbex évaluée, nous sommes passés à l'étude de certains de ces hyper-paramètres, afin de comprendre leurs influences sur la prédiction des paramètres de la GPD et par extension des quantiles extrêmes. Pour cela, c'est de nouveau la MISE décrite dans l'équation (4.1) qui sera utilisée. Le premier paramètre qui a été étudié est le nombre d'arbres B utilisé pour la construction du gradient boosting. C'est un paramètre très important, car il affecte directement la capacité d'apprentissage du modèle et sa complexité. Plus précisément, le nombre d'arbres, aussi appelé dans ce mémoire "nombre d'estimateurs", influe sur le compromis entre biais et variance du modèle. Un nombre élevé d'arbres peut permettre de réduire le biais en augmentant la capacité d'apprentissage du modèle, c'est-à-dire sa capacité à capter des relations complexes dans les données. Cependant, un nombre trop important d'arbres peut provoquer une augmentation de la variance et conduire à un sur-ajustement, où le modèle s'adapte trop aux données d'apprentissage au détriment de sa capacité à généraliser à partir de nouvelles données. Inversement, un nombre trop faible d'arbres peut entraîner un sous-ajustement, où le modèle n'est pas capable de saisir pleinement les relations dans les données, conduisant à une performance médiocre ou sous-ajustement. Ainsi, le réglage adéquat du nombre d'arbres est essentiel pour équilibrer le compromis entre le biais et la variance, et atteindre une performance optimale avec le gradient boosting. Décrit par Friedman 2001 et Velthoen et al. 2021 comme étant l'hyper-paramètre le plus important du gradient boosting, le nombre d'arbres sera à chaque fois étudié en fonction des autres hyper-paramètres afin d'évaluer son impact sur la construction de nos modèles.

4.1.5 Étude des hyper-paramètres

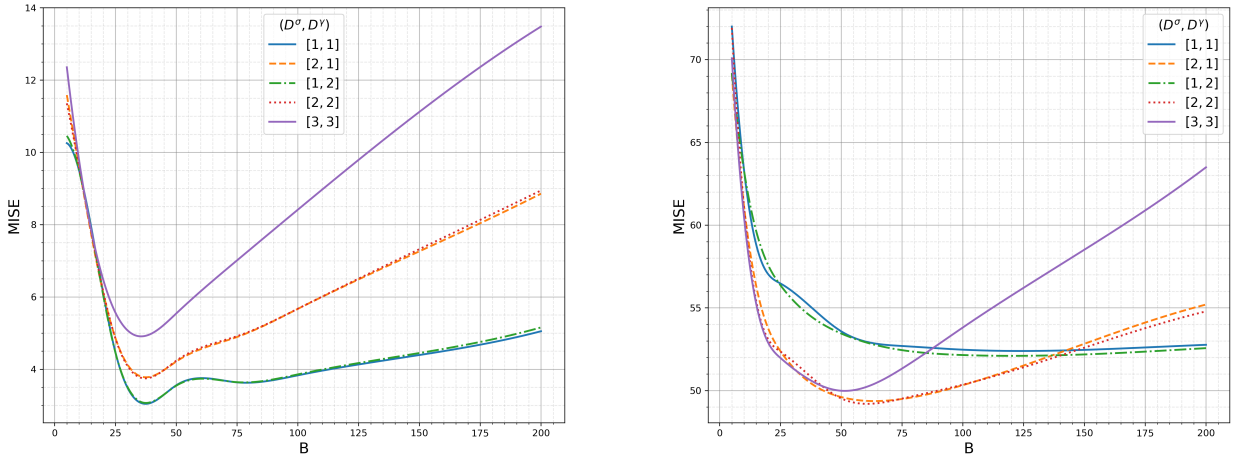
L'étude des hyper-paramètres a été réalisée sur les modèles 1 et 2 et pour $n = 2000$ et $n = 5000$. Les résultats donnés ci-dessous seront uniquement les résultats pour $n = 2000$, les résultats pour $n = 5000$ étant disponibles en annexe. Le choix du quantile extrême sera tout au long de cette section d'ordre 0.9995.

Profondeur de l'arbre

La profondeur de l'arbre qui détermine la complexité du modèle est le premier paramètre discuté. Comme expliqué dans le chapitre 3 consacré à la méthodologie, deux gradient boosting sont ajustés en parallèle, l'un pour estimer le paramètre d'échelle σ, l'autre pour

estimer le paramètre de forme γ de la GPD. Dès lors, un choix de profondeur d'arbre doit être choisi pour chacun de ces modèles, notons les alors (D^σ, D^γ).

Dans la figure 4.3, on visualise donc l'évolution de la MISE en fonction du nombre d'arbres noté B et en fonction de la profondeur de l'arbre pour chacun des paramètres d'échelle et de forme. Pour le modèle 1, on observe dans la figure 4.3a, que les courbes sont quasiment



(a) Modèle 1

(b) Modèle 2

FIGURE 4.3 – Évolution de la MISE en fonction du nombre d'arbre et de la profondeur maximale de chaque arbre pour le modèle 1 à gauche et le modèle 2 à droite, avec $n = 5000$. La MISE a été calculé comme la moyenne de 100 ISE pour chaque valeur des hyper-paramètres.

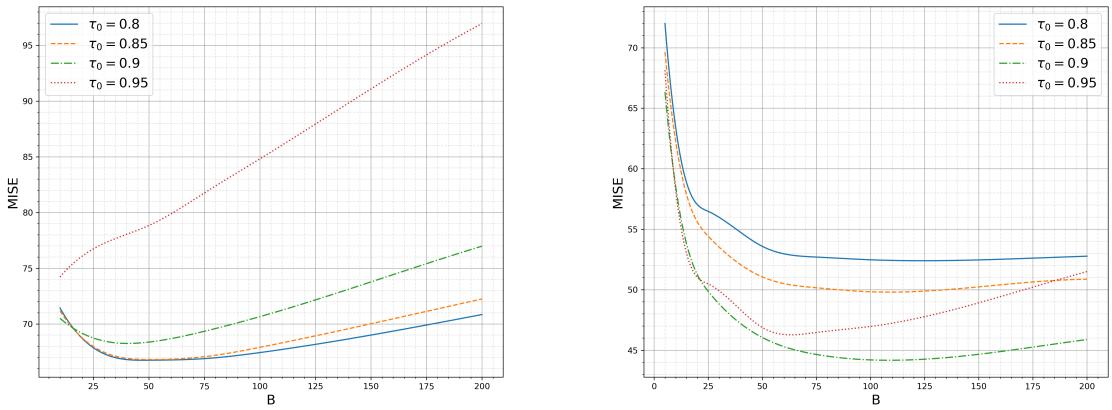
superposées pour des valeurs de D^σ similaire. Cette observation n'est pas étonnante, étant donné que comme expliqué précédemment, le modèle 1 génère une queue de distribution dont l'approximation par la GPD possède un paramètre de forme $\gamma \equiv 1/4$ ainsi qu'un paramètre σ dépendant de x_1. De plus, on observe que les valeurs des profondeurs sont relativement faibles. L'interprétation de ces faibles paramètres de profondeur est que le gradient boosting est composé d'arbres très simples, et que c'est effectivement la combinaison de ces apprenants faibles qui apporte la qualité de notre modèle. Ces mêmes graphiques pour $n = 2000$ sont disponibles dans l'annexe C.6b. On remarque en revanche que pour le modèle 2, la profondeur $[1, 1]$ n'est pas adéquate, les arbres ne sont pas assez complexe pour ajuster ce genre de données, nous obligeant à nous diriger vers des modèles avec des profondeurs d'arbres de 2 ou de 3. Pour le modèle 1, des modèles simples sont suffisants pour modéliser les données, mais l'on remarque qu'à partir d'un B suffisamment grand, la MISE augmente de manière linéaire. Cela est sûrement justifié par un certain sur-ajustement du modèle. Pour la suite de l'analyse, nous garderons donc pour le modèle 1 une profondeur $(D^\sigma, D^\gamma) = (1, 1)$ pour le modèle 1, et $(D^\sigma, D^\gamma) = (2, 2)$ pour le modèle 2.

Étude du MISE en fonction du seuil choisi

La prochaine étape de notre analyse repose sur l'étude de l'ordre du quantile intermédiaire que l'on a choisi. Comme précisé plus tôt dans ce travail, nous avons fixé un seuil de 0.8

en adéquation avec les expériences réalisées par VELTHOEN et al. 2021. Étant donné que ce seuil n'a pas été testé dans leurs travaux, nous proposons ici d'étudier la MISE en fonction de différents seuils de quantiles intermédiaires. Comme expliqué plus tôt, les hyper-paramètres nécessaires à la prédiction de ce quantile intermédiaire sont restés fixe en fonction des seuils, et pourraient être une limitation à cette analyse.

Pour cette analyse, nous comparerons le modèle 2 pour un nombre d'observations $n = 2000$ et $n = 5000$. La comparaison pour le modèle 1 est présenté en annexe C.7. Sur ces



(a) Modèle 2, $n = 2000$

(b) Modèle 2, $n = 5000$

FIGURE 4.4 – Évolution de la MISE en fonction du nombre d'arbre et du quantile intermédiaire

graphiques, il est intéressant de constater que pour un nombre d'observations relativement faible ($n = 2000$), le modèle 2 assez complexe ne s'améliore pas en fonction de l'ordre du quantile intermédiaire τ_0, et plus l'ordre du quantile intermédiaire est élevé, plus la MISE est élevée. La MISE se détériore également en fonction du nombre d'arbres B sélectionné, ce qui indique sûrement un problème de sur-ajustement. Au contraire, lorsque $n = 5000$, on trouve pour le modèle 2 un ordre de quantile optimal à $\tau_0 = 0.9$. Cela peut se justifier par le fait que dans lorsque $n = 2000$, le nombre d'observations pour ajuster une GPD correctement est trop faible, provoquant une dégradation de la MISE en fonction de l'ordre du quantile intermédiaire τ_0. Cette explication est corroborée par le fait que l'ordre du quantile τ_0 optimal pour chaque nombre d'observations n induit un nombre d'observations excédantes d'approximativement $80\% \times 2000 = 400$ dans un cas et de $90\% \times 5000 = 450$ dans l'autre cas. Environ 400 observations semblent nécessaires à un ajustement correct de la distribution généralisée de Pareto.

Finalement, la taille minimale des feuilles a également été testée, mais l'impact de cet hyper-paramètre sur la MISE n'avait rien de significatif. Cela n'est pas étonnant vu la profondeur très faible des arbres, et du paramètre α de régularisation, imposant une certaine homogénéité dans la taille des feuilles. Passons donc maintenant à l'évaluation de l'importance des variables dans la construction des modèles.

4.1.6 Importance des variables dans la construction des modèles

Une fois les hyper-paramètres optimaux des algorithmes évalués, un autre point tout aussi important est l'interprétabilité du modèle. Comme expliqué dans la section 2.3.1, un des principaux avantages de l'arbre de régression est son interprétabilité (voir 2.9). Cet avantage est perdu lorsque l'on réalise des ensembles d'arbres tels que le gradient boosting, étant donné que l'on se retrouve avec une somme pondérée d'un très grand nombre d'arbres. Un point important de ce travail a donc été de réussir à évaluer quelles variables avaient un impact sur la prédiction de nos paramètres de la GPD. La qualité d'interprétabilité peut facilement être évaluée étant donné la construction de nos jeux de données. Pour le modèle 1, la variable dépendante est directement liée à la première variable indépendante que l'on note x_1, pour le modèle 2, la variable dépendante est liée à la fois à la variable x_1 et x_2. Pour évaluer l'importance des variables permettant la prédiction des quantiles extrêmes, la somme de l'importance de chaque variable pour chaque arbre constituant le gradient boosting, pondéré par le poids de l'arbre en particulier a été réalisée. Plus en détail, notons FI_j l'importance de la variable j pour un certain arbre T. Notons également $CR_{\{t,j\}}$, la réduction de l'erreur au carré apportée par la variable j à un certain arbre t, calculé comme la somme de la réduction de l'erreur au carré pour chaque nœud d'un arbre pour lesquel la variable j est utilisé. Dans ce contexte, l'importance de la variable j est donnée par :

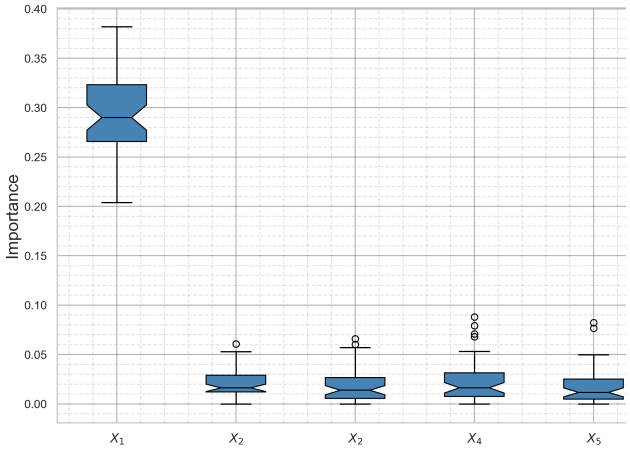
$$FI_j = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}(t, j) \cdot CR_{t,j},$$

où $CR_{t,j}$ est la réduction de l'erreur au carré provoquée par la scission par une variable j dans un arbre t. Cette somme est alors normalisée par la réduction totale du critère de somme des erreurs pour obtenir une importance relative à l'apport de toutes les autres variables. On a donc

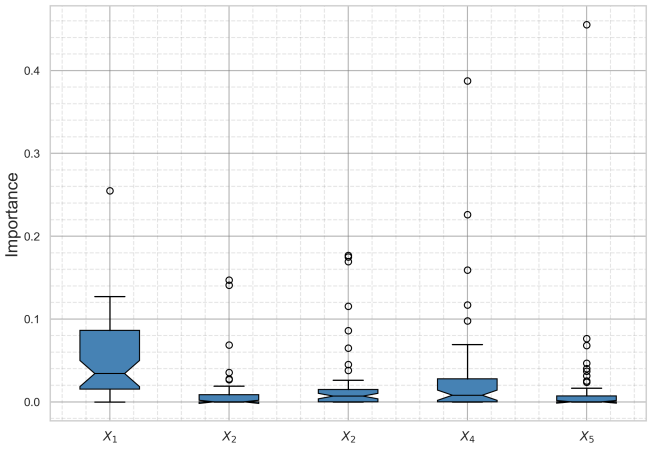
$$FI_{j,norm} = \frac{FI_j}{\sum_{j=1}^{p} FI_j},$$

où p est le nombre de variables. De la même manière que pour les hyper-paramètres, l'importance d'une variable en particulier pourrait être dépendante du jeu de données utilisé. Il est donc nécessaire pour évaluer l'importance des variables de répéter ce procédé un grand nombre de fois, pour avoir la distribution de l'importance de chacune des variables. Pour les deux modèles, on va donc afficher les boxplots donnant une idée de la distribution de l'importance des variables pour la prédiction de chacun des paramètres. Pour le modèle 1,on retrouve les boxplots des 5 premières variables dans la Figure 4.5 à gauche pour le paramètre d'échelle σ et à droite pour le paramètre de forme γ. On remarque que la variable x_1 a un impact clair sur la prédiction de σ, mais qu'aucune variable ne se distingue pour la prédiction du paramètre γ.

La même analyse peut être faite pour le modèle 2. On retrouve les boxplots des 5 premières variables dans la Figure 4.6 à gauche pour le paramètre σ et à droite pour le paramètre γ. On remarque que pour le deuxième modèle, les variables x_1 et x_2 ont un impact sur la prédiction du paramètre d'échelle, ce qui est tout a fait logique quant à la simulation des données. En ce qui concerne le paramètre de forme, c'est la variable x_1 qui a un impact évident sur ces variables. Les mêmes résultats sont obtenus pour $n = 2000$, les résultats n'ont donc pas été présentés, car ils n'apportent pas d'information supplémentaire.

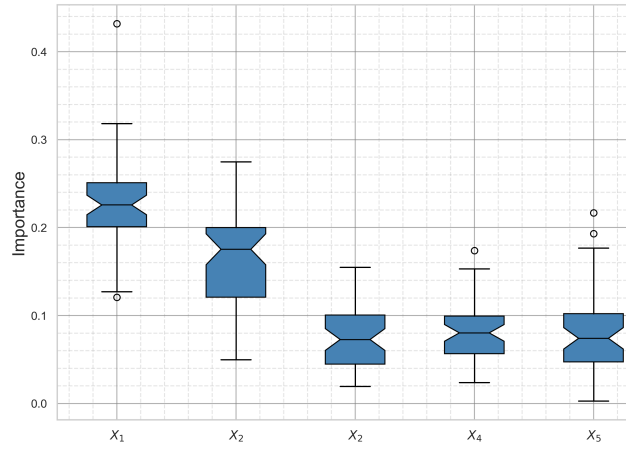


(a) Pour le paramètre Sigma

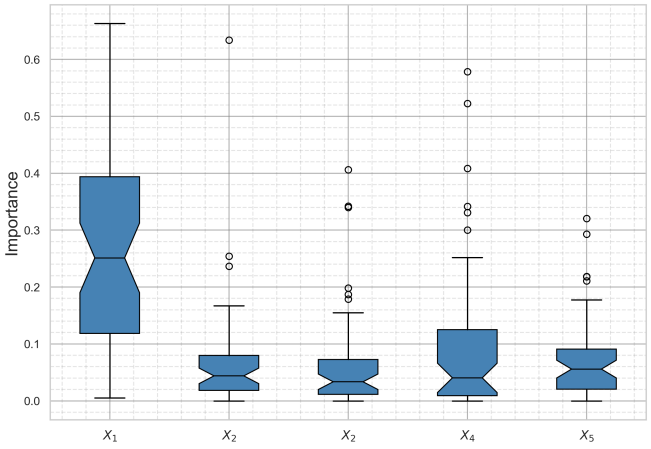


(b) Pour le paramètre Gamma

FIGURE 4.5 – Importance des paramètres en fonction des paramètres d'échelle et de forme pour le modèle 1



(a) Pour le paramètre σ.

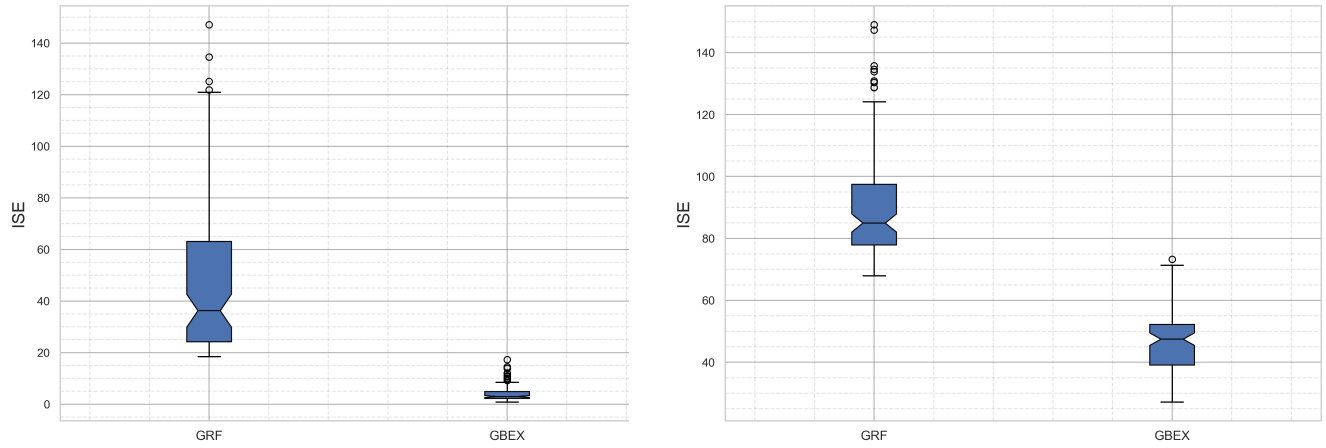


(b) Pour le paramètre γ.

FIGURE 4.6 – Importance des paramètres en fonction des paramètres d'échelle et de forme pour le modèle 2

4.1.7 Comparaison de méthodes

La dernière étape de cette analyse vise à comparer notre algorithme Gbex avec la forêt aléatoire généralisée présentée dans la sous-section 2.3.4, afin de comparer les performances de ces deux modèles. Pour cette comparaison des deux modèles, nous nous sommes concentrés sur les deux modèles de données, avec $n = 5000$. Pour le Gbex, nous avons utilisé les hyper-paramètres optimisés décrits dans la section précédente, alors que pour la méthode de forêt aléatoire généralisée, la même procédure de sélection des hyper-paramètres décrite dans la section 4.1.3 a été opérée. Afin de comparer les 2 modèles, nous avons créé 150 jeux de données de manière indépendante afin d'appliquer chaque algorithme de manière indépendante. L'ISE a ensuite été calculé pour chacun de ces jeux de données. On peut retrouver une visualisation pour chacun des modèles dans la figure 4.7. Notez que pour



(a) Modèle 1 (b) Modèle 2

FIGURE 4.7 – Comparaison du GRF et du Gbex

le modèle 1, les ISE calculés par le GRF avaient une très grande variabilité. En effet, la moyenne des ISE est d'environ 128.33, avec un écart-type de 370, alors qu'à l'inverse le Gbex à une MISE de 4, avec un écart-type 3. Cependant, la moyenne et la variance élevée du GRF sont provoquées par un faible nombre de jeux de données pour lesquels le GRF n'a pas réussi à prédire les quantiles extrêmes correctement, influençant énormément la moyenne. Pour améliorer la visualisation de la figure 4.7a, les valeurs aberrantes ont donc été retirée (environ 10% des jeux de données), correspondant au ISE supérieur à 150.

Lorsque l'on compare les performances des deux algorithmes sur le modèle 2, on observe que le Gbex est également meilleur que le GRF avec également une variabilité plus faible. En effet, la MISE du GRF est d'environ 91, avec un écart-type de 19, alors la MISE du Gbex est de 49, avec un écart-type d'environ 12.

En conclusion, on peut affirmer sur ces deux simulations de données, ayant des niveaux de complexités et un nombre de variables indépendantes différentes, que le Gbex a non seulement une meilleure performance que le GRF, mais est également meilleur dans la généralisation des données, réussissant à prédire correctement des observations que le GRF n'arrive pas à prédire correctement.

4.2 Application empirique

4.2.1 Données

Maintenant qu'une analyse a été faite sur des données simulées, nous allons passer sur des données réelles. Pour ce faire, et pour permettre une comparaison de nos résultats avec d'autres résultats, nous avons décidé de nous concentrer sur des données relatives aux salaires horaires américains, comme l'on fait auparavant Angrist et al. 2006 et Gnecco, Terefe et Engelke 2022. Ces données portent sur des données de salaires horaires de 65023 hommes, de couleurs de peau blanche et noire, âgés de 40 à 49 ans et ayant fait entre 5 et 20 ans d'études. Les nombres d'années d'études sont regroupés par catégories, le détail est donné dans le tableau 4.1. La variable dépendante est donc le salaire hebdomadaire exprimé en dollar (valeur du dollar en 1989). Nous utiliserons comme données les 3 variables présentent dans le jeu de données (couleur de peau, âge et éducation), et nous ajouterons également 10 variables aléatoires pour augmenter la dimension de notre jeu de données et pour rendre l'étude de ces données plus complexe. Ces variables proviennent d'une distribution uniforme sur l'intervalle $[-1, 1]$. Ce jeu de données est particulièrement adapté, étant donné son nombre d'observations ainsi que la distribution des salaires que l'on sait très inégalement répartis aux états-unis, comme l'atteste l'indice de Gini qui était de 0.43 selon WorldBank 2023.

Catégorie	**Niveau d'éducation**
8	5e, 6e, 7e ou 8e année
9	9e année
10	10e année
11	11e ou 12e année, pas de diplôme
12	Diplômé du secondaire
13	Quelques années dans le supérieur, pas de diplôme
14	Diplôme supérieur, programme professionnel
15	Diplôme supérieur, programme académique
16	Diplôme univsersitaire, non scolarisé actuellement
17	Diplôme univsersitaire, actuellement inscrit
18	Master obtenue
19	Diplôme professionnel obtenu
20	Doctorat obtenu

Table 4.1 – Correspondance entre les années d'études et le niveau d'éducation

La première étape de cette analyse a été une brève analyse des données dont nous disposions. Nous observons que les données ne sont pas vraiment balancées sur la variable couleur de peau, en effet, il n'y a que 7% de personne de couleur noire dans ce jeu de données. La variable éducation n'est également pas très balancée puisque près de 40% des observations ont comme étude maximum un diplôme secondaire.

Ensuite, de la même manière que précédemment, une prédiction doit dans un premier temps être faite pour déterminer le quantile intermédiaire d'ordre 0.8. Pour ce faire, nous profiterons du fait que la taille du jeu de données soit grande, pour procéder à la même

technique que dans la partie consacrée aux simulations. Dans cette optique, notre jeu de données sera divisé en 3 parties : une première partie notée DE-QI est utilisée pour entraîner l'algorithme à prédire au mieux le quantile intermédiaire, DT-QI $\equiv$ DE-QE permettra une prédiction du quantile intermédiaire, ainsi qu'à calculer l'excédant déjà plusieurs fois défini et noté par Z. Ce nouveau jeu de données servira ensuite à entraîner le Gbex. Finalement, le jeu de données DT-QI sera utilisé pour tester le Gbex ainsi qu'à évaluer sa performance.

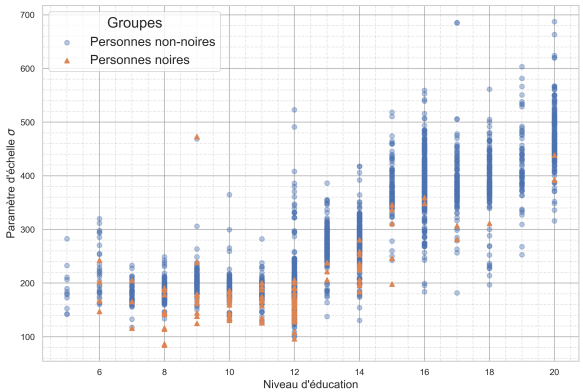
La prochaine problématique est de déterminer comment évaluer la qualité de notre prédiction. Contrairement aux données simulées, on ne sait pas à priori comment se comporte la queue de la distribution de notre variable dépendante. On ne peut donc pas utiliser le vrai quantile pour étudier la performance de notre modèle. La solution qui sera alors utilisée est une méthode d'évaluation proposée par Wang et Li 2013 et qui est donnée par

$$R_n\left(\hat{Q}_\cdot(\tau)\right) := \frac{\sum_{i=1}^{n} \mathbb{1}\left\{Y_i - \hat{Q}_{X_i}(\tau)\right\} - n\tau}{\sqrt{n\tau(1-\tau)}}, \tag{4.2}$$

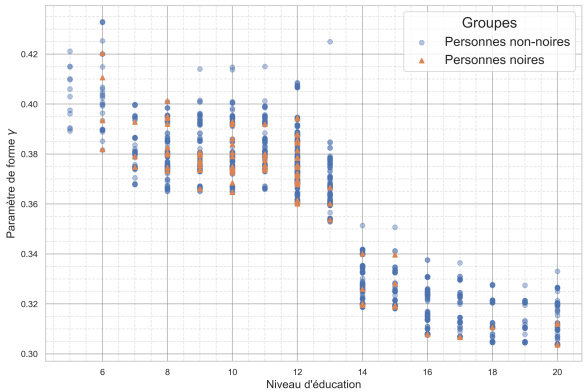
où n est le nombre d'observations tests et $\hat{Q}_\cdot(\tau)$ est le quantile $\tau^{\text{ème}}$ estimé sur les observations d'entraînement. Cette métrique compare l'estimation de la proportion normalisée des observations où $Y_i < \hat{Q}_{X_i}(\tau)$ avec le niveau théorique de τ. En utilisant la vraie fonction quantile $Q_\cdot(\tau)$, la variable aléatoire $\mathbb{1}\{Y_i < Q_{X_i}(\tau)\}$ suit alors une distribution de Bernoulli de paramètre d'espérance τ et de variance $\tau(1-\tau)$.

Maintenant que nous avons tout à disposition, nous avons procédé à plusieurs analyses.

Tout d'abord, nous allons nous intéresser à l'influence des variables d'intérêts. Pour ce faire, nous allons comme précédemment diviser notre jeu de données en trois et de tailles égales afin de construire nos différents modèles, puis nous étudierons la relation entre la valeur des paramètres d'échelle et de forme et différentes variables d'intérêt. Concernant



(a) Paramètre d'échelle en fonction de l'éducation, segmenté par la couleur de peau



(b) Paramètre d'échelle en fonction de l'éducation, segmenté par la couleur de peau

Figure 4.8 – Comportement des paramètres de la GPD en fonction de variables d'intérêts

l'éducation, on observe une nette croissance entre le niveau d'éducation et le paramètre

d'échelle, on remarque également une certaine tendance pour les personnes noires, qui semblent légèrement sur les parties basses de chaque segment. L'interprétation, en se référant à l'analyse de la GPD dans la figure 2.6 est que plus le niveau d'éducation est grand, plus la queue de distribution sera épaisse. On remarque également sur ce graphique 4.8a que le niveau de ce paramètre semble constant pour des niveaux d'éducation inférieurs à la catégorie 12, et clairement croissant à partir de ce seuil de niveau d'éducation. La même observation peut être faite pour le paramètre de forme γ. Bien que la décroissance semble plus constante en fonction du niveau d'étude, une véritable marche est observée au voisinage de cette catégorie d'étude.

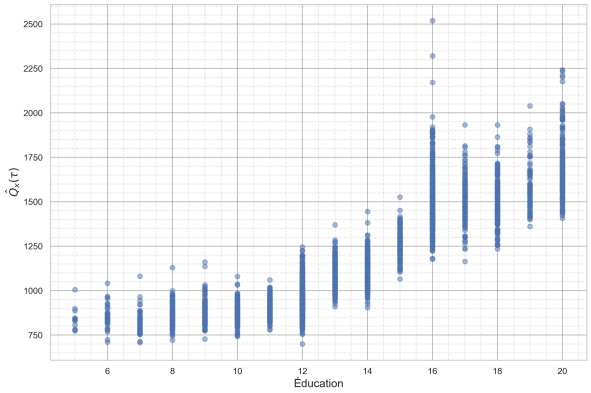
La même analyse a été faite pour étudier la valeur des paramètres en fonction de l'âge des individus. Ces graphiques, disponibles dans l'annexe C.8, ne montrent pas de relation claire, entre soit le paramètre d'échelle soit le paramètre de forme et cette variable d'intérêt.

La prochaine analyse vise à évaluer la capacité du Gbex à modéliser les quantiles extrêmes de nos données. On observe effectivement que lorsque l'éducation augmente, la prédiction du quantile extrême augmente également avec la méthode du Gbex, alors qu'elle est constante pour la méthode GRF. En d'autres termes, pour un quantile extrême, le Gbex arrive à capturer la variabilité du quantile conditonnel, là où le GRF n'y parvient pas. Cette analyse est d'autant plus intéressante, que l'on remarque que pour un quantile d'ordre 0.9, le GRF arrive très bien à capturer cette variabilité. Cette analyse met donc en avant la capacité du Gbex à extrapoler la tendance de la fonction quantile pour les quantiles d'ordre extrême.

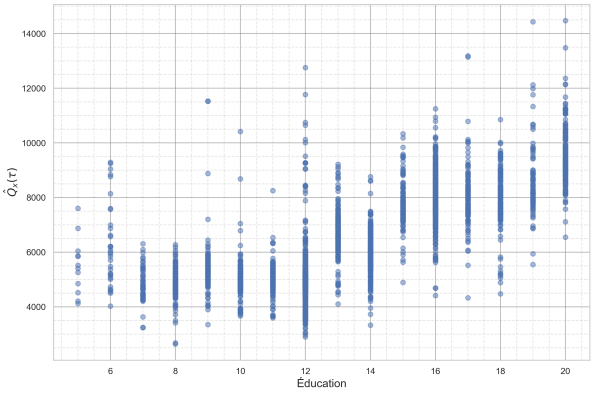
Il est important de noter que pour le graphique 4.9a, les quantiles supérieurs à 2500 ont été supprimés pour une meilleure visualisation des données. De la même manière pour le graphique 4.9b où les quantiles plus grands que 12000 ont été supprimé. Pour 4.9c les quantiles supérieurs à 2500 ont été supprimés et finalement pour le graphique 4.9d, les quantiles au delà de 6000 ont été supprimés.

4.2.2 Comparaison des modèles entre Gbex et GRF

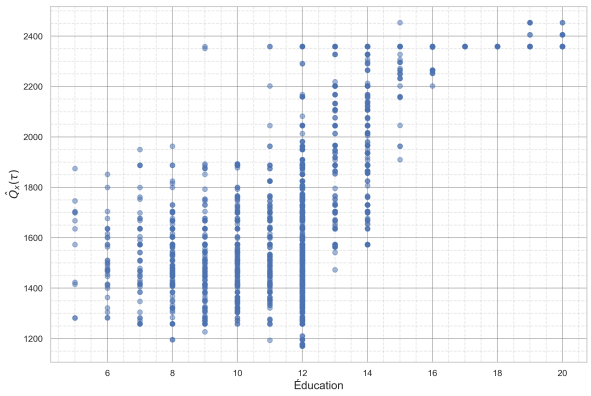
La dernière étape de notre analyse a été de comparer les modèles Gbex et GRF de la même manière que ça a été fait dans la partie consacrée aux simulations. La stratégie a donc été la suivante : choisir aléatoirement des sous-ensembles du jeu de données global pour former les différents sous-échantillons nécessaires à la construction de nos différents modèles. Chaque sous-échantillon a à chaque fois été tiré de manière aléatoire afin d'observer les performances en fonction de données aléatoires. Sur chaque sous-ensemble, le critère d'évaluation donné dans l'équation (4.2) a été calculé noté R_n. Puis l'ensemble des R_n trouvés pour les 100 sous-échantillons sont repris dans la figure 4.10. Les performances sur des données réelles sont similaires aux données simulées. En effet, on observe non seulement que le Gbex a une meilleure performance que le GRF étant donné la moyenne des ISE, mais aussi, on observe que le Gbex a une plus grande stabilité, les ISE ayant une moins grande variabilité que le GRF.



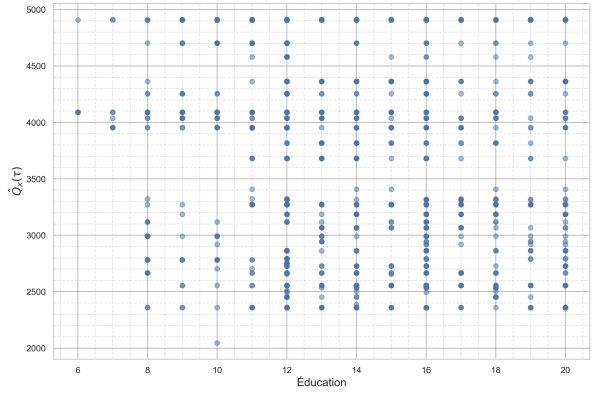
(a) Évolution de la prédiction du quantile d'ordre 0.9 en fonction de l'éducation. La prédiction est basée sur la méthode du Gbex



(b) Évolution de la prédiction du quantile d'ordre 0.9995 en fonction de l'éducation. La prédiction est basée sur la méthode du Gbex



(c) Évolution de la prédiction du quantile d'ordre 0.9 en fonction de l'éducation. La prédiction est basée sur la méthode de GRF



(d) Évolution de la prédiction du quantile d'ordre 0.9995 en fonction de l'éducation. La prédiction est basée sur la méthode de GRF

FIGURE 4.9 – Comportement des paramètres de la GPD en fonction de variables d'intérêts

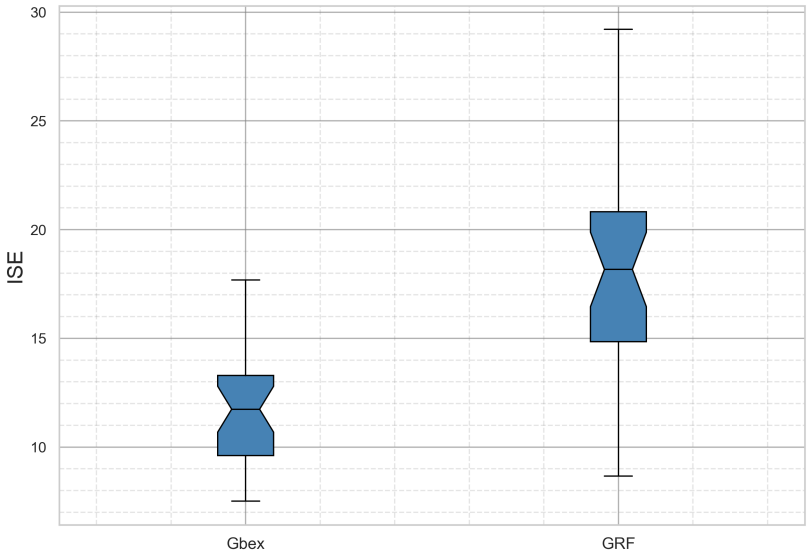


FIGURE 4.10 – Comparaison de la valeur du critère 4.2 du Gbex et du GRF

Chapitre 5

Discussion et conclusion

Dans ce travail, basé sur la description du Gbex par Velthoen et al. 2021, nous avons eu l'opportunité d'explorer et d'approfondir plusieurs concepts clé en statistiques et en apprentissage automatique. Nos études successives ont démontré l'efficacité du gradient boosting comme méthode robuste pour la prédiction des quantiles extrêmes, surtout lorsqu'elle est combinée avec les théories des valeurs extrêmes. Plusieurs points intéressants sont à mettre en avant à la suite de ces différentes études. Tout d'abord, lors de l'évaluation des performances de l'algorithme Gbex, nous avons constaté qu'il surpassait significativement la méthode GRF. Cette supériorité s'est révélée tant sur le plan de la précision de la prédiction que sur celui de variance des prédictions. Cela suggère que le Gbex est une méthode de prédiction robuste et fiable pour l'estimation des quantiles extrêmes. Une grande partie de travail ayant été l'implémentation de l'algorithme en python, puisqu'aucun package n'existait à ce jour pour cet algorithme, les résultats trouvés ont été comparé avec ceux trouvés par Velthoen et al. 2021 pour s'assurer que les résultats étaient dans le bon ordre de grandeur, et donc que l'algorithme avait été correctement implémenté.

Malgré les bons résultats obtenus lors de nos analyses, plusieurs points doivent tout de même être souligné, représentant soit des limites, soit des pistes d'améliorations ou de recherches concernant le Gbex. Le premier point qui devrait être mis en avant est le travail de pré-traitement de données. Comme présentée pendant ce travail, l'utilisation du Gbex nécessite l'implémentation en amont d'une forêt aléatoire afin de déterminer une prédiction pour un quantile intermédiaire. Cette forêt aléatoire nécessitant elle-même un certain travail d'ajustement des hyper-paramètres, et donc un travail de validation croisée qui peut être gourmand en temps. Ce point nous emmène directement à plusieurs autres limites de notre travail : dans les différentes applications, une exploration du seuil du quantile intermédiaire a été discutée et testée. Ces tests ont permis de montrer que dans certains cas, utiliser un ordre de quantile intermédiaire supérieur à 0.8% pouvait être bénéfique pour la précision du modèle. En revanche, les hyper-paramètres de la forêt aléatoire n'ont pas été ajustés pour ces seuils supérieurs à 80%. La prédiction des quantiles extrêmes étant complètement liée à la qualité de prédiction du quantile intermédiaire, ce travail supplémentaire ne pourrait qu'améliorer la qualité de nos prédictions de quantiles extrêmes.

Ensuite, une autre limite de notre travail est la taille des échantillons utilisés. Comme montré dans la partie consacrée aux simulations, on a pu montrer que la taille des jeux de données sur lesquelles on travaillait avait un impact direct sur la qualité d'ajustement

de notre modèle. Ce point est tout à fait prévisible dans la mesure où l'on cherche en fait à ajuster une distribution généralisée de Pareto sur un excédent de données. Si le jeu de données de base est petit, le nombre d'observations excédantes est faible et donc le nombre d'observations utilisées pour l'ajustement de la GPD est elle aussi petite, ce qui peut mener à une moins bonne précision. On a montré dans nos différentes expériences qu'environ 400 observations permettait une très bonne qualité d'ajustement. Là est donc une limite de ce travail, nous avons travaillé soit sur des données simulées, permettant un choix arbitraire du nombre d'observation, et nous ayant donc permis de montrer toute la puissance de notre modèle, nous avons également travaillé sur des données réelles donc le jeu de données était composé de plus de 60.000 observations, facilitant donc l'implémentation du modèle. Il serait intéressant de tester notre algorithme sur des données plus limitées, des solutions pourraient être trouvée pour améliorer la qualité d'ajustement dans ces cas limite.

La synthèse des limites présentées ci-dessus nous emmène à la dernière limite de notre travail. Nous avons en effet un certain nombre limité d'observations qui doit être utilisé pour à la fois entraîner le GRF nécessaire à la prédiction du quantile intermédiaire, mais aussi à l'entraînement du Gbex nécessaire à la prédiction des paramètres de la distribution généralisée de Pareto. Dans une situation dans laquelle le nombre d'observations pourraient être limité, il pourrait être compliqué de ne pas sur-ajuster notre modèle, et de pouvoir l'évaluer de manière consistante. En effet, on doit être certain que les données d'entraînements ne sont jamais utilisées dans les données de tests. Dans notre travail, cela a pu être réalisé grâce au grand nombre d'observations que nous possédions, mais en réalité, ce point pourrait clairement être une limite. Une solution déjà discutée dans ce travail serait l'utilisation de prédiction "out-of-bag" tant pour le GRF que pour le Gbex. Pour le GRF, aucun package n'a été développé à ce jour en python permettant ce type de prédiction, cet algorithme pourrait être développé et serait une réelle solution à ce problème. À ce jour, les méthodes de prédiction out-of-bag en python n'existe uniquement pour les forêt aléatoire classique, mais il a été montré que ces méthodes étaient moins performantes que les forêts aléatoires généralisées pour la prédiction de quantile relativement grand (typiquement les quantiles intermédiaires).

Finalement, pour conclure ce travail, il est intéressant de souligner que depuis le commencement de la rédaction de ce mémoire, d'autres algorithmes ont été développé pour la prédiction de quantile extrême et on montré des résultats très intéressants. En effet, Gnecco, Terefe et Engelke 2022 on développé un algorithme très similaire au Gbex, mais ils utilisent une forêt aléatoire pour prédire les paramètres de la GPD. Les résultats semblent meilleurs que le Gbex sur les données simulées, mais ces deux méthodes devraient encore été comparé étant donné les avantages et désavantages connus des forêts aléatoires comparées au gradient boosting. En effet, on pourrait s'attendre à ce que le gradient boosting puisse détecter des relations plus complexe que les forêts aléatoires. À l'inverse, le risque de sur-ajustement du gradient boosting peut être plus grand. Ensuite, Pasche et Engelke 2023 ainsi que Richards et Huser 2022 ont développé une méthodologie similaire, qui utilise les réseaux de neurones pour la prédiction des paramètres de la GPD. Bien que ces travaux montrent une meilleure performance que le Gbex, certains avantages non-négligeables de notre implémentation sont à souligner. D'abord, la puissance de calcul et la structure nécessaire au gradient boosting est bien plus efficace et simple que les réseaux de neurones, ensuite, comme nous l'avons montré dans nos différentes applications, la méthode du Gbex permet une assez bonne interprétabilité de modèle. En effet, on a pu mettre en lumière tant dans les applications simulées que réelles

l'impact des différentes variables indépendantes sur la prédiction de notre modèle, chose que les réseaux de neurones ne permettent pas, causés par la structure de "black-box" intrinsèque à leurs implémentations.

Quoi qu'il en soit, les recherches récentes et les découvertes successives concernant la régression de quantiles extrêmes montrent que ces techniques sont assez indispensables. La méthode du Gbex se trouve à la frontière entre statistiques et apprentissage automatique se trouve donc au cœur du domaine des sciences des données en pleine expansion.

Bibliographie

Gnecco, Nicola, Edossa Merga Terefe et Sebastian Engelke (2022). « Extremal Random Forests ». In : arXiv : 2201.12865 [stat.ME] (pages 54, 60).

Shannon, Claude Elwood (1948). « A Mathematical Theory of Communication ». In : *The Bell System Technical Journal* 27, p. 379-423. URL : http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf (visité le 22/04/2003).

Yu, Keming, Zudi Lu et Julian Stander (2003). « Quantile Regression : Applications and Current Research Areas ». In : *Journal of the Royal Statistical Society. Series D (The Statistician)* 52.3, p. 331-350. ISSN : 00390526, 14679884. URL : http://www.jstor.org/stable/4128208.

Koenker, R. et G. Bassett (1978). « Regression Quantiles ». In : *Econometrica* 46.1, p. 33. DOI : 10.2307/1913643 (page 1).

Powell, James L. (1986). « Censored regression quantiles ». In : *Journal of Econometrics* 32.1, p. 143-155. ISSN : 0304-4076. DOI : https://doi.org/10.1016/0304-4076(86)90016-3. URL : https://www.sciencedirect.com/science/article/pii/0304407686900163.

A. Colin Cameron, Pravin K. Trivedi (2005). *Microeconometrics - Methods And Applications*. Cambridge University Press. ISBN : 9780521848053. URL : libgen.li/file.php?md5=5fc6c86d38b339dae4ee7d2b36f53106.

Hosking, J. R. M., J. R. Wallis et E. F. Wood (1985). « Estimation of the Generalized Extreme-Value Distribution by the Method of Probability-Weighted Moments ». In : *Technometrics* 27.3, p. 251-261. DOI : 10.1080/00401706.1985.10488049, eprint : https://www.tandfonline.com/doi/pdf/10.1080/00401706.1985.10488049. URL : https://www.tandfonline.com/doi/abs/10.1080/00401706.1985.10488049 (page 15).

Pickands, James (1975). « Statistical Inference Using Extreme Order Statistics ». In : *The Annals of Statistics* 3.1, p. 119-131. DOI : 10.1214/aos/1176343003. URL : https://doi.org/10.1214/aos/1176343003 (page 18).

Paul Embrechts Jürgen Franke, Wolfgang K. Härdle et Christian M. Hafner (août 2008). « Statistics of Financial Markets : An Introduction, 2nd Edition by Jürgen Franke, Wolfgang K. Härdle, Christian M. Hafner ». In : *International Statistical Review 2008-aug vol. 76 iss. 2* 76 (2). DOI : 10.1111/j.1751-5823.2008.00054_15.x. URL : libgen.li/file.php?md5=dda7493eb0944a42fb4d2ac58052413a.

Embrechts, Paul (2008). « Statistics of Financial Markets : An Introduction ». In : *International Statistical Review* 76.2, p. 313-314. DOI : 10.1111/j.1751-5823.2008.00054_15.x (page 20).

M. R. Leadbetter G. Lindgren, H. Rootzen (1983). *Extremes and Related Properties of Random Sequences and Processes*. 1re éd. Springer Series in Statistics. Springer. URL : libgen.li/file.php?md5=1e7b56831be29d1363fc2e8c0dba90e5.

LIN, Yi et Yongho JEON (2006). « Random Forests and Adaptive Nearest Neighbors ». In : *Journal of the American Statistical Association* 101.474, p. 578-590. DOI : 10.1198/016214505000001230. eprint : https://doi.org/10.1198/016214505000001230. URL : https://doi.org/10.1198/016214505000001230 (page 27).

MEINSHAUSEN, Nicolai (2006). « Quantile Regression Forests ». In : *Journal of Machine Learning Research* 7.35, p. 983-999. URL : http://jmlr.org/papers/v7/meinshausen06a.html (page 27).

VELTHOEN, Jasper, Clément DOMBRY, Juan-Juan CAI et Sebastian ENGELKE (2021). « Gradient boosting for extreme quantile regression ». In : DOI : 10.48550/ARXIV.2103.00808. URL : https://arxiv.org/abs/2103.00808 (pages 2, 38, 39, 41–43, 48, 50, 59).

MOSTELLER, Frederick et John W. TUKEY (1977). *Data analysis and regression. A second course in statistics* (page 1).

FERREIRA, Ana et Laurens de HAAN (2015). « On the block maxima method in extreme value theory : PWM estimators ». In : *The Annals of Statistics* 43.1, p. 276-298. DOI : 10.1214/14-AOS1280. URL : https://doi.org/10.1214/14-AOS1280 (page 16).

DANÍELSSON, Jón et Casper G. DE VRIES (1997). « Tail index and quantile estimation with very high frequency data ». In : *Journal of Empirical Finance* 4.2. High Frequency Data in Finance, Part 1, p. 241-257. ISSN : 0927-5398. DOI : https://doi.org/10.1016/S0927-5398(97)00008-X. URL : https://www.sciencedirect.com/science/article/pii/S092753989700008X (page 11).

ODENING, Martin et Jan HINRICHS (mai 2003). « Using extreme value theory to estimate Value-at-Risk ». In : *Agricultural Finance Review* 63, p. 55-73. DOI : 10.1108/00215000380001141 (page 11).

KOENKER, Roger, Victor CHERNOZHUKOV, Xuming HE et Limin PENG, éd. (oct. 2017). *Handbook of Quantile Regression*. Chapman et Hall/CRC. DOI : 10.1201/9781315120256. URL : https://doi.org/10.1201%5C%2F9781315120256.

WAGER, Stefan et Susan ATHEY (2017). *Estimation and Inference of Heterogeneous Treatment Effects using Random Forests*. arXiv : 1510.04342 [stat.ME] (pages 28, 29, 31).

BÜHLMANN, Peter et Bin YU (2002). « Analyzing bagging ». In : *The Annals of Statistics* 30.4, p. 927-961. DOI : 10.1214/aos/1031689014. URL : https://doi.org/10.1214/aos/1031689014 (page 28).

MENTCH, Lucas et Giles HOOKER (2015). *Quantifying Uncertainty in Random Forests via Confidence Intervals and Hypothesis Tests*. arXiv : 1404.6473 [stat.ML] (page 28).

SCORNET, Erwan, Gé rard BIAU et Jean-Philippe VERT (août 2015). « Consistency of random forests ». In : *The Annals of Statistics* 43.4. DOI : 10.1214/15-aos1321. URL : https://doi.org/10.1214%5C%2F15-aos1321 (page 28).

STROBL, Carolin, Anne-Laure BOULESTEIX, Achim ZEILEIS et Torsten HOTHORN (jan. 2007). « Bias in random forest variable importance measures : Illustrations, sources and a solution ». In : *BMC Bioinformatics* 8.1, p. 25. ISSN : 1471-2105. DOI : 10.1186/1471-2105-8-25. URL : https://doi.org/10.1186/1471-2105-8-25 (page 28).

ATHEY, Susan, Julie TIBSHIRANI et Stefan WAGER (2018). *Generalized Random Forests*. arXiv : 1610.01271 [stat.ME] (page 30).

KOENKER, Roger (2005). *Quantile Regression*. Cambridge University Press. URL : https://EconPapers.repec.org/RePEc:cup:cbooks:9780521608275 (page 9).

KOENKER, Roger W. et Vasco D'OREY (1987). « Algorithm AS 229 : Computing Regression Quantiles ». In : *Journal of the Royal Statistical Society. Series C (Applied*

Statistics) 36.3, p. 383-393. ISSN : 00359254, 14679876. URL : http://www.jstor.org/stable/2347802 (visité le 23/03/2023) (page 10).

ANGRIST, Joshua et al. (2006). « Quantile Regression under Misspecification, with an Application to the U.S. Wage Structure ». In : *Econometrica* 74.2. Accessed : 5 June 2023, p. 539-563. URL : http://www.jstor.org/stable/3598810 (pages 8, 54).

KARMARKAR, Narendra (1984). « A new polynomial-time algorithm for linear programming ». In : *Combinatorica* 4, p. 373-395 (page 10).

PORTNOY, Stephen (1997). « On Computation of Regression Quantiles : Making the Laplacian Tortoise Faster ». In : *Lecture Notes-Monograph Series* 31, p. 187-200. ISSN : 07492170. URL : http://www.jstor.org/stable/4355977 (visité le 24/03/2023) (page 10).

CHEN, Colin et Ying WEI (2005). « Computational Issues for Quantile Regression ». In : *Sankhyā : The Indian Journal of Statistics (2003-2007)* 67.2, p. 399-417. ISSN : 09727671. URL : http://www.jstor.org/stable/25053439 (visité le 24/03/2023) (page 10).

GUMBEL, E. J. (1958). *Statistics of Extremes*. New York Chichester, West Sussex : Columbia University Press. ISBN : 9780231891318. DOI : doi:10.7312/gumb92958. URL : https://doi.org/10.7312/gumb92958 (pages 11, 12, 15).

EMBRECHTS, Paul, Claudia KLÜPPELBERG et Thomas MIKOSCH (1997). *Modelling extremal events*. T. 33. Applications of Mathematics (New York). For insurance and finance. Berlin : Springer-Verlag, p. xvi+645. ISBN : 3-540-60931-8 (page 11).

FRÉCHET, M. (1927). *Sur la loi de probabilité de l'écart maximum*. T. 6, p. 93-116 (page 11).

FISHER, R. A. et L. H. C. TIPPETT (1928). « Limiting forms of the frequency distribution of the largest or smallest member of a sample. » English. In : *Proc. Camb. Philos. Soc.* 24, p. 180-190. ISSN : 0008-1981. DOI : 10.1017/S0305004100015681 (page 11).

STAMP, J. C. (1937). In : *Economica* 4.14, p. 223-225. ISSN : 00130427, 14680335. URL : http://www.jstor.org/stable/2548660 (visité le 08/05/2023).

STAMP, J. C., L. von MISES et J. KAHANE (1937). « Socialism-An Economic and Sociological Analysis ». In : *Economica* 4.14, p. 223. DOI : 10.2307/2548660 (page 11).

HO, Tin Kam (1995). « Random Decision Forests ». In : ICDAR '95, p. 278-282 (page 26).

BREIMAN, L. (2001). « Machine Learning ». In : *Machine Learning* 45.1, p. 5-32. DOI : 10.1023/a:1010933404324 (page 26).

BREIMAN, L., J. H. FRIEDMAN, R. A. OLSHEN et C. J. STONE (1984). *Classification and Regression Trees*. Chapman et Hall/CRC (page 21).

KASS, G. V. (1980). « An Exploratory Technique for Investigating Large Quantities of Categorical Data ». In : *Applied Statistics* 29.2, p. 119. DOI : 10.2307/2986296 (page 21).

HUNT, Earl B., Philip J. STONE et Janet MARIN (1966). *Experiments in induction*. New York : Academic Press (page 21).

QUINLAN, J. R. (1983). « Learning Efficient Classification Procedures and Their Application to Chess End Games ». In : p. 463-482. DOI : 10.1007/978-3-662-12405-5_15 (page 21).

MURTHY, S. K. (1998). « Automatic Construction of Decision Trees from Data : A Multi-Disciplinary Survey ». In : *Data Mining and Knowledge Discovery* 2.4, p. 345-389. DOI : 10.1023/A:1009744630224. URL : https://doi.org/10.1023/A:1009744630224 (page 21).

BREIMAN, Leo (1996). « Bagging predictors ». In : *Machine Learning* 24.2, p. 123-140. DOI : 10.1007/BF00058655 (page 25).

FREUND, Y. et R. SCHAPIRE (1996). « Experiments with a new boosting algorithm ». In : p. 148-156 (page 25).

DIETTERICH, T. (1998). « An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting and randomization ». In : *Machine Learning*, p. 1-22 (page 25).

WANG, H. J. et D. LI (2013). « Estimation of Extreme Conditional Quantiles Through Power Transformation ». In : *Journal of the American Statistical Association* 108.503, p. 1062-1074. DOI : 10.1080/01621459.2013.8201 (page 55).

FRIEDMAN, Jerome H. (oct. 2001). « Greedy function approximation : A gradient boosting machine ». In : *Ann. Statist.* 29.5, p. 1189-1232. DOI : 10.1214/aos/1013203451. URL : https://doi.org/10.1214/aos/1013203451 (pages 48, 69).

CHEN, T. et C. GUESTRIN (août 2016). « XGBoost : A Scalable Tree Boosting System ». In : p. 785-794. DOI : 10.1145/2939672.2939785 (page 35).

KE, Guolin, Qi MENG, Thomas FINLEY, Taifeng WANG, Wei CHEN, Weidong MA, Qiwei YE et Tie-Yan LIU (2017). « LightGBM : A Highly Efficient Gradient Boosting Decision Tree ». In : 30. Sous la dir. d'I. GUYON, U. Von LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN et R. GARNETT. URL : https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf (page 35).

PROKHORENKOVA, Liudmila, Gleb GUSEV, Aleksandr VOROBEV, Anna Veronika DOROGUSH et Andrey GULIN (2018). « CatBoost : unbiased boosting with categorical features ». In : 31. Sous la dir. de S. BENGIO, H. WALLACH, H. LAROCHELLE, K. GRAUMAN, N. CESA-BIANCHI et R. GARNETT. URL : https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf (page 35).

MÄRZ, Alexander (2020). « CatBoostLSS – An extension of CatBoost to probabilistic forecasting ». In : arXiv : 2001.02121 [stat.ML] (page 36).

SMITH, R. L. (1987). « Estimating Tails of Probability Distributions ». In : *The Annals of Statistics* 15.3, p. 1174-1207. DOI : 10.1214/aos/1176350499 (page 39).

MAI, Vien V. et Mikael JOHANSSON (2021). « Stability and Convergence of Stochastic Gradient Clipping : Beyond Lipschitz Continuity and Smoothness ». In : arXiv : 2102.06489 [math.OC] (page 41).

WORLDBANK (2023). *Gini Index (World Bank Estimate) - United States*. URL : https://data.worldbank.org/indicator/SI.POV.GINI?locations=US (visité le 20/05/2023) (page 54).

PASCHE, Olivier C. et Sebastian ENGELKE (2023). « Neural Networks for Extreme Quantile Regression with an Application to Forecasting of Flood Risk ». In : arXiv : 2208.07590 [stat.ME] (page 60).

RICHARDS, Jordan et Raphaël HUSER (2022). « Regression modelling of spatiotemporal extreme U.S. wildfires via partially-interpretable neural networks ». In : arXiv : 2208.07581 [stat.ML] (page 60).

Appendices

Annexe A

Algorithmes

A.1 Algorithme CART

L'algorithme CART (Classification And Regression Trees) est un algorithme d'apprentissage automatique supervisé qui permet de construire des arbres de décision pour la classification et la régression.

Le principe de cet algorithme consiste à diviser successivement l'ensemble de données d'apprentissage en sous-ensembles en se basant sur les valeurs de certaines variables, jusqu'à ce que chaque sous-ensemble soit homogène en termes de classe ou de valeur de réponse. Les variables utilisées pour la division sont choisies de manière à minimiser une fonction de coût qui mesure l'homogénéité des sous-ensembles résultants.

Pour construire l'arbre, on part d'une racine qui correspond à l'ensemble de données d'apprentissage initial, puis on divise cette racine en deux sous-ensembles en utilisant la variable qui minimise la fonction de coût. On répète ensuite le processus de division sur chaque sous-ensemble obtenu, jusqu'à atteindre un critère d'arrêt (par exemple, un nombre maximal de niveaux de l'arbre, un nombre minimal d'observations dans une feuille, ou une réduction de la fonction de coût inférieure à un certain seuil).

L'algorithme s'arrête en fonction de plusieurs règles : La profondeur maximale (D^σ, D^γ), la taille minimale des feuilles (L^σ, L^γ). Les feuilles de l'arbre T^σ et T^γ sont respectivement notée $(L^\sigma_{b,j})_{1\leq j\leq J^\sigma_b}$ et $(L^\gamma_{b,j})_{1\leq j\leq J^\gamma_b}$.

L'arbre ainsi construit peut être utilisé pour prédire la classe ou la valeur de réponse d'un nouvel exemple en le faisant descendre dans l'arbre en suivant les règles de division correspondantes aux valeurs de ses variables. La classe ou la valeur de réponse prédite correspond à la feuille atteinte à la fin du parcours dans l'arbre.

Algorithm 2 Algorithme CART pour la régression

Require: Un ensemble de données d'apprentissage $\mathcal{D} = (x_i, y_i)_{i=1}^n$, où $x_i \in \mathbb{R}^p$ et $y_i \in \mathbb{R}$ pour $i = 1, \ldots, n$, où n est le nombre d'observations.
Require: Le nombre maximal de niveaux de l'arbre, noté $T_{\max}$
Require: Le nombre minimal d'observations dans une feuille, noté $n_{\min}$
Ensure: Un arbre de régression $T(x)$

1: Initialiser l'arbre T à une racine, avec l'ensemble de données d'apprentissage $\mathcal{D}$ attribué à cette racine.
2: **for** chaque nœud de l'arbre T **do**
3: **if** la profondeur maximale de l'arbre a été atteinte **then**
4: convertir le nœud en une feuille et attribuer la valeur moyenne des réponses dans l'ensemble de données d'apprentissage à cette feuille.
5: **else**
6: sélectionner la variable j et la valeur de seuil s qui divisent l'ensemble de données du nœud en deux sous-ensembles, tels que la fonction de coût suivante soit minimale :

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]$$

 où $R_1(j, s) = x_i : x_{ij} \leq s$ et $R_2(j, s) = x_i : x_{ij} > s$ sont les deux sous-ensembles obtenus par la division du nœud selon j et s, et c_1 et c_2 sont les valeurs constantes qui minimisent la somme des carrés des écarts à la moyenne des réponses pour les deux sous-ensembles.
7: **if** la réduction de la fonction de coût est inférieure à un certain seuil prédéfini **then**
8: convertir le nœud en une feuille et attribuer la valeur moyenne des réponses dans l'ensemble de données d'apprentissage à cette feuille.
9: **else**
10: diviser le nœud en deux sous-nœuds en utilisant la variable j et la valeur de seuil s trouvées à l'étape précédente, et attribuer les sous-ensembles $R_1(j, s)$ et $R_2(j, s)$ à ces sous-nœuds.
11: **end if**
12: **end if**
13: **end for**

A.2 Algorithme Gradient Boosting

Algorithm 3 Algorithme Gradient booting de FRIEDMAN 2001

Require: Un ensemble de données d'apprentissage $\mathcal{D} = (x_i, y_i)_{i=1}^n$, où $x_i \in \mathbb{R}^p$ et $y_i \in \mathbb{R}$ pour $i = 1, \ldots, n$, où i est le nombre d'observations.
Require: Le nombre d'itération B
Require: Une fonction de perte $\ell(y, f)$
Require: Un modèle apprenant faible $h(x, \theta)$

1: Initialiser $\hat{F}_0$ avec une constante
2: **for** $b = 1$ à B **do**
3: Calculer le gradient négatif $g_b(x)$ {Ce calcul permet de déterminer la direction de la plus grande diminution de la fonction de perte}
4: Ajuster un nouvel apprenant faible $h(x, \theta_b)$ {L'apprenant faible est ajusté sur les résidus de la fonction de perte}
5: Trouver le meilleur pas de progression ρ_b de la descente de gradient :

$$\rho_b = \arg\min_{\rho} \sum_{i=1}^{n} \ell \left[y_i, \hat{F}_{b-1}(x_i) + \rho h(x_i, \theta_b) \right]$$

{Le pas de progression détermine combien nous avançons dans la direction du gradient négatif}
6: Mettre à jour la fonction d'estimation de la cible :

$$\hat{F}_b \leftarrow \hat{F}_{b-1} + \lambda \rho_b h(x, \theta_t)$$

{La fonction d'estimation est mise à jour en ajoutant le produit du pas de progression et de l'apprenant faible ajusté, et λ est un taux d'apprentissage pour éviter le sur-ajustement.}
7: **end for**

A.3 Algorithme Gbex pour la prédiction des paramètres de la GPD

Algorithm 4 Gbex boosting algorithm for GPD modeling

• **Input :**

— θ_0 : Valeurs initiales des paramètres déterminée via le quantile intermédiaire.

— $(\boldsymbol{X}_i, Z_i)_{1\leq i\leq n}$: l'échantillon en excedance.

— $\boldsymbol{B}$: nombre d'arbre pour le gradient.

— $\boldsymbol{D}^\sigma, \boldsymbol{D}^\gamma$: la profondeur maximal de chaque arbre

— $\boldsymbol{\lambda}^\sigma, \boldsymbol{\lambda}^\gamma$: taux d'apprentissage respectivement pour les paramètres σ et γ de la GPD

— $\boldsymbol{s}$: fraction de sous-échantillonage

— $\boldsymbol{L}^\sigma_{\min}, \boldsymbol{L}^\gamma_{\min}$: nombre d'éléments minimum de chaque feuille.

• **Algorithme :** pour $b = 1, \ldots, B$:

1. Sélectionner un échantillon aléatoire $S_b \subset \{1, \ldots, n\}$ of size $[sn]$.
2. Calculer la dérivée de la déviance sur le sous échantillon S_b :

$$r^\sigma_{b,i} = \frac{\partial l_{Z_i}}{\partial \sigma}(\theta_{b-i}(\boldsymbol{X}_i)) \text{ and } r^\gamma_{b,i} = \frac{\partial l_{Z_i}}{\partial \gamma}(\theta_{b-i}(\boldsymbol{X}_i)) \ , \ i \in S_b$$

3. Ajuster les arbres de régression T^σ_b, T^γ_b qui prédisent les gradients $r^\sigma_{b,i}$ et $r^\gamma_{b,i}$ comme une fonction des covariables $\boldsymbol{X}_i$ pour l'échantillon $i \in S_b$; Les arbres sont construits avec une profondeur maximal $(\boldsymbol{D}^\sigma, \boldsymbol{D}^\gamma)$ et une taille de feuille minimal $(\boldsymbol{L}^\sigma_{\min}, \boldsymbol{L}^\gamma_{\min})$.
4. Mettre à jour les paramètres $\theta_b(\boldsymbol{x}) = (\hat{\sigma}_b(\boldsymbol{x}), \hat{\gamma}_b(\boldsymbol{x}))$ avec les taux d'apprentissages $(\lambda^\sigma, \lambda^\gamma)$, c'est à dire :

$$\hat{\sigma}_b(\boldsymbol{x}) = \hat{\sigma}_{b-1}(\boldsymbol{x}) + \lambda^\sigma T^\sigma_b(\boldsymbol{x}) \text{ et } \hat{\gamma}_b(\boldsymbol{x}) = \hat{\gamma}_{b-1}(\boldsymbol{x}) + \lambda^\gamma T^\gamma_b(\boldsymbol{x})$$

• **Output :** les paramètres conditionnels de la GPD : $(\hat{\sigma}(\boldsymbol{x}), \hat{\gamma}(\boldsymbol{x})) = (\hat{\sigma}_{\boldsymbol{B}}(\boldsymbol{x}), \hat{\gamma}_{\boldsymbol{B}}(\boldsymbol{x}))$

A.4 Algorithme de prédiction des quantiles extrêmes

Algorithm 5 gbex algorithm for extreme quantile prediction

- **Input :**
  - $(\boldsymbol{X}_i, Y_i)_{1\leq i\leq n}$: l'échantillon de données.
  - τ_0 : la valeur du quantile intermédiaire.
  - τ : la valeur du quantile extrême tel que $\tau > \tau_0$
  - Les paramètres de la GPD évalués dans l'algorithme 4
- **Algorithme :**
  1. Ajuster une régression quantile via RF à l'échantillon $(\boldsymbol{X}_i, Y_i)_{1\leq i\leq n}$ pour obtenir l'estimation $\hat{Q}_{\boldsymbol{x}}(\tau_0)$ du quantile conditionnel d'ordre τ_0
  2. Calculer l'excédant $Z_i = (Y_i - \hat{Q}^{\text{oob}}_{\boldsymbol{X}_i}(\tau))_+$, $1 \leq i \leq n$ où $\hat{Q}^{\text{oob}}_{\boldsymbol{X}_i}(\tau)$ est l'estimation "out-of-bag" du quantile conditionnel $\boldsymbol{x} = \boldsymbol{X}_i$
  3. Déterminer $I = \{i : Z_i > 0\}$ est l'ensemble des indexes des excédants positifs et appliquer l'algorithme 4 sur le data set $(\boldsymbol{X}_i, Z_i)_{i\in I}$ pour estimer les paramètres $(\hat{\sigma}(\boldsymbol{x}), \hat{\gamma}(\boldsymbol{x}))$.
- **Output :** L'estimation du quantile conditionnel extrême :

$$\hat{Q}_{\boldsymbol{x}}(\tau) = \hat{Q}_{\boldsymbol{x}}(\tau_0) + \hat{\sigma}(\boldsymbol{x})\frac{(\frac{1-\tau}{1-\tau_0})^{-\hat{\gamma}(\boldsymbol{x})} - 1}{\hat{\gamma}(\boldsymbol{x})}$$

Annexe B

Notions théoriques supplémentaires

B.1 Approximation de Newton-Raphson

L'approximation de Newton-Raphson est une méthode numérique utilisée pour trouver les racines d'une fonction non linéaire. Elle consiste à utiliser une approximation locale de la fonction à l'aide de sa tangente en un point donné et à itérer pour trouver une meilleure approximation de la racine.

Plus précisément, si l'on cherche une racine de la fonction $f(x) = 0$, la méthode de Newton-Raphson utilise la formule suivante :

$x_{(n+1)} = x_n - \frac{f(x_n)}{f'(x_n)}$

où x_n est une approximation initiale de la racine, $f'(x_n)$ est la dérivée de la fonction f en x_n et $x_{(n+1)}$ est la nouvelle approximation de la racine obtenue en utilisant la formule.

La méthode de Newton-Raphson converge généralement rapidement vers une solution précise, à condition que l'approximation initiale soit suffisamment proche de la racine et que la fonction soit suffisamment régulière. Cependant, elle peut également converger vers des solutions incorrectes ou diverger si ces conditions ne sont pas remplies.

La méthode de Newton-Raphson est largement utilisée en analyse numérique et en optimisation pour résoudre des problèmes non linéaires dans une variété de domaines, y compris la physique, l'ingénierie et les sciences sociales.

B.2 Fonction de distribution non-dégénérée

La fonction de distribution dégénérée est une fonction de distribution pour laquelle l'intégralité de la densité est concentrée en un point. En d'autres termes, c'est une distribution dont la fonction de répartition est de la forme :

$$F(x) = \begin{cases} 0 & \text{if } x < c, \\ 1 & \text{if } x \geq c, \end{cases}$$

où $-\infty < c < +\infty$. Une fonction de distribution non dégénérée implique donc certaines conditions. En effet, la fonction de répartition $F(x)$ doit être non décroissante et continue à droite en tout point de son domaine de définition. La probabilité associée à une fonction de

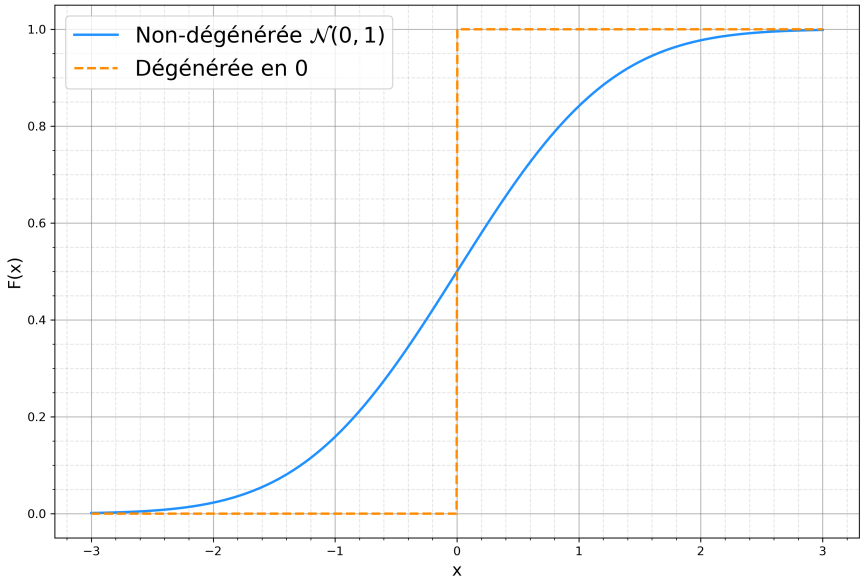


FIGURE B.1 – Exemple de fonction de répartition dégénérée et non-dégénérée

répartition non dégénérée ne doit pas être concentrée en un seul point ou un petit ensemble de valeurs. Ceci est équivalent à demander qu'il existe un $x \in \mathbb{R}$ tel que $0 < F(x) < 1$. En résumé, pour qu'une fonction de répartition $F(x)$ soit non dégénérée, elle doit satisfaire les conditions suivantes :

1. $F(x)$ doit être non décroissante.
2. $F(x)$ doit être continue à droite en tout point de son domaine de définition.
3. Il doit exister un $x \in \mathbb{R}$ tel que $0 < F(x) < 1$.

Ainsi, une fonction de répartition non dégénérée garantit que la probabilité associée à cette distribution ne soit pas concentrée en un seul point. Une illustration de cette notion se trouve dans la figure B.1

B.3 Descente de gradient

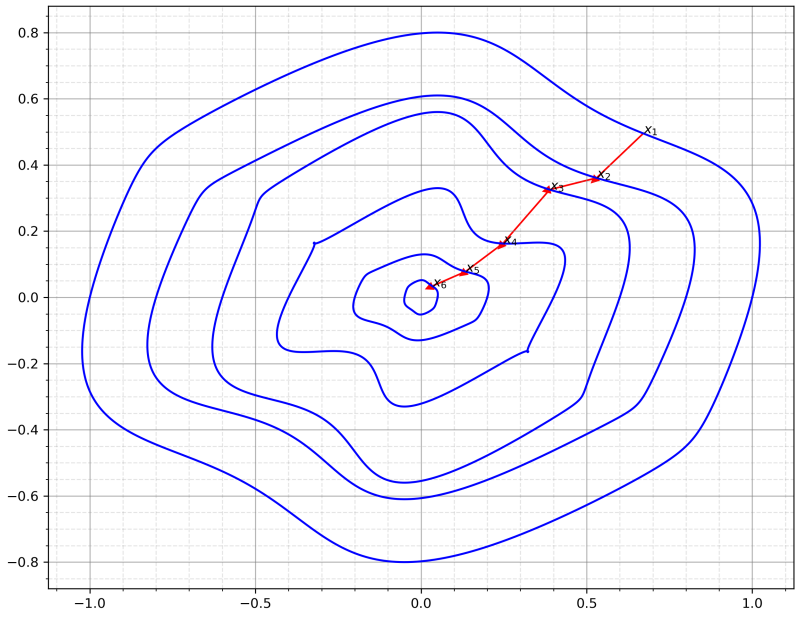


FIGURE B.2 – Distribution GEV en fonction du paramètre α

Annexe C

Graphiques supplémentaires

C.1 Gradient Boosting

Les différentes étapes du gradient boosting sont représentées dans ces différents graphes. Une méthode détaillé est données dans la section correspondantes.

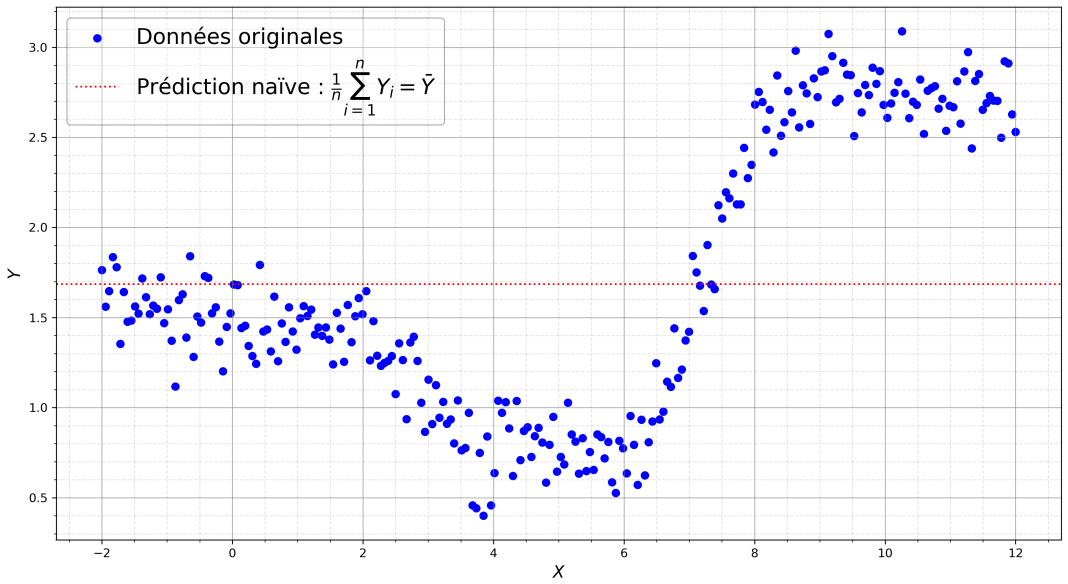


FIGURE C.1 – 1-ère étape du Gradient Boosting : Prédiction naïve

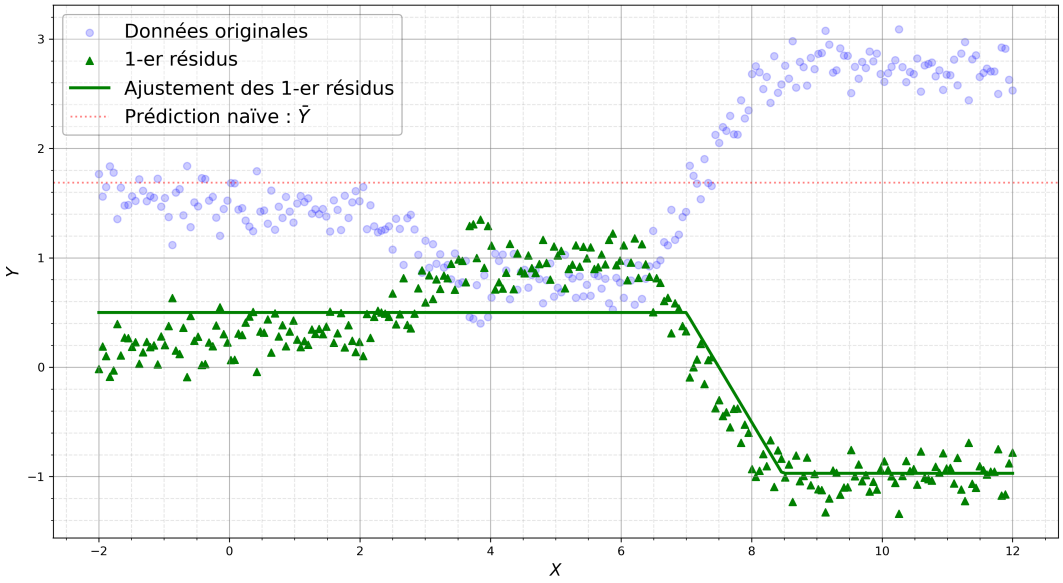


FIGURE C.2 – 2-ème étape du Gradient Boosting : Calcul et ajustement des résidus de la prédiction naïve

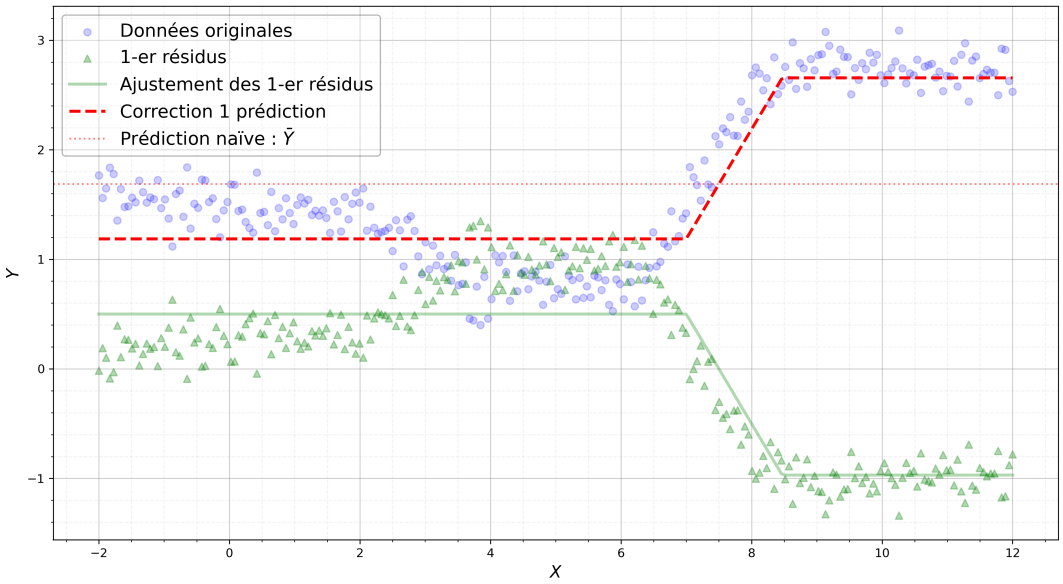


FIGURE C.3 – 3-ème étape du Gradient Boosting : Correction de la prédiction de Y, via les résidus ajustés

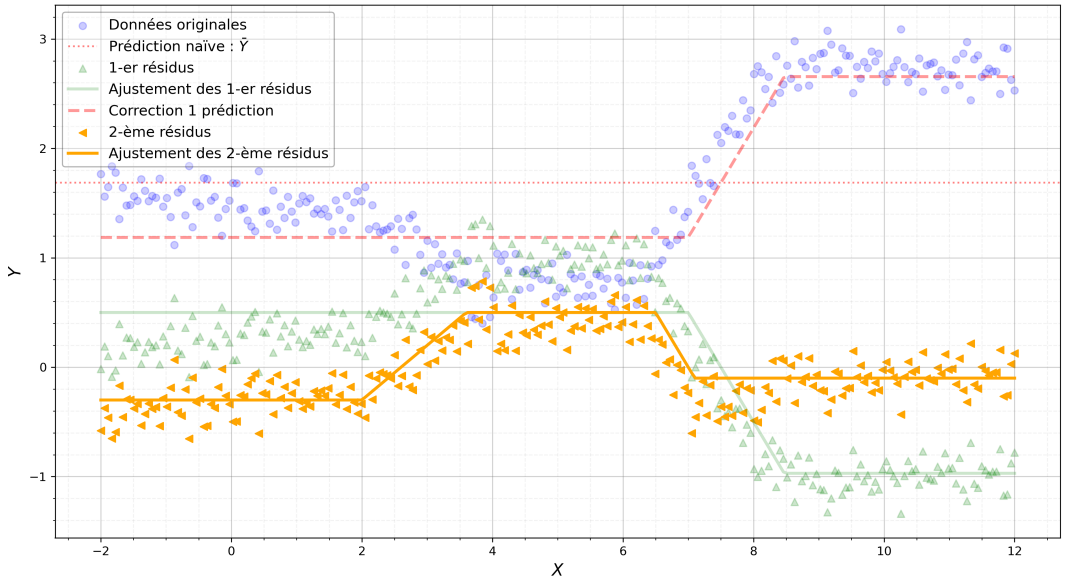


FIGURE C.4 – 4-ème étape du Gradient Boosting : Calcul et ajustement des résidus du dernier modèle

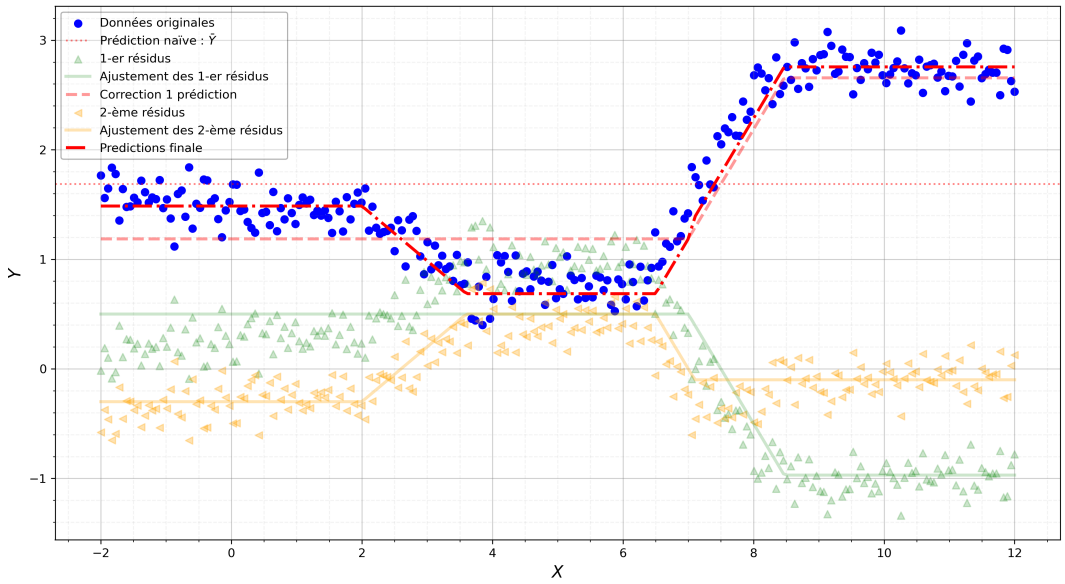
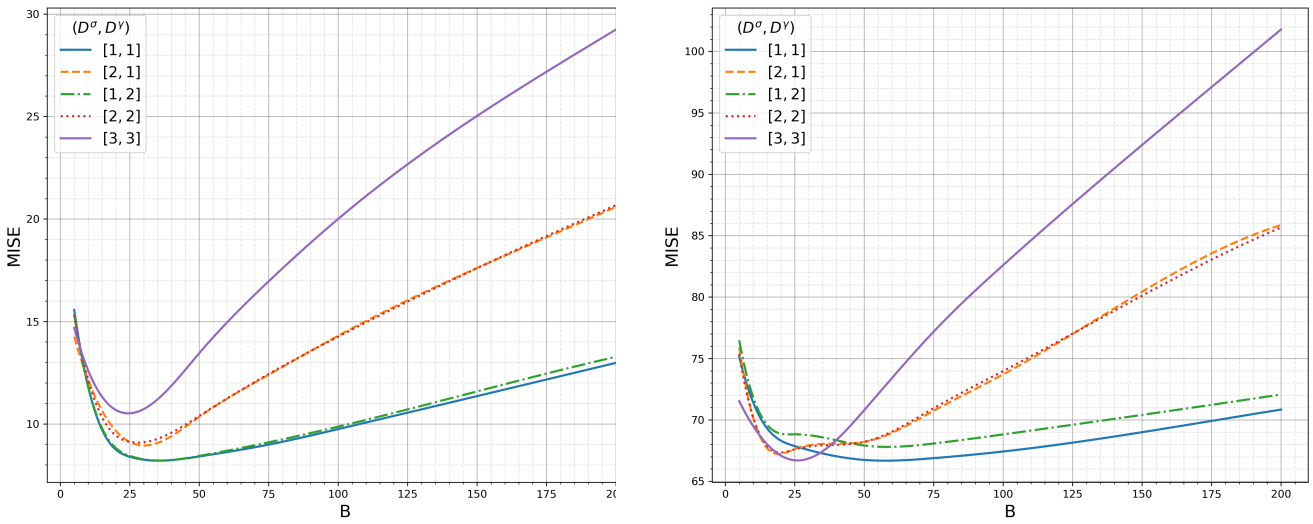


FIGURE C.5 – B-ème étape du Gradient Boosting : Calcul et ajustement des résidus du dernier modèle

C.2 MISE en fonction de la profondeur d'arbre

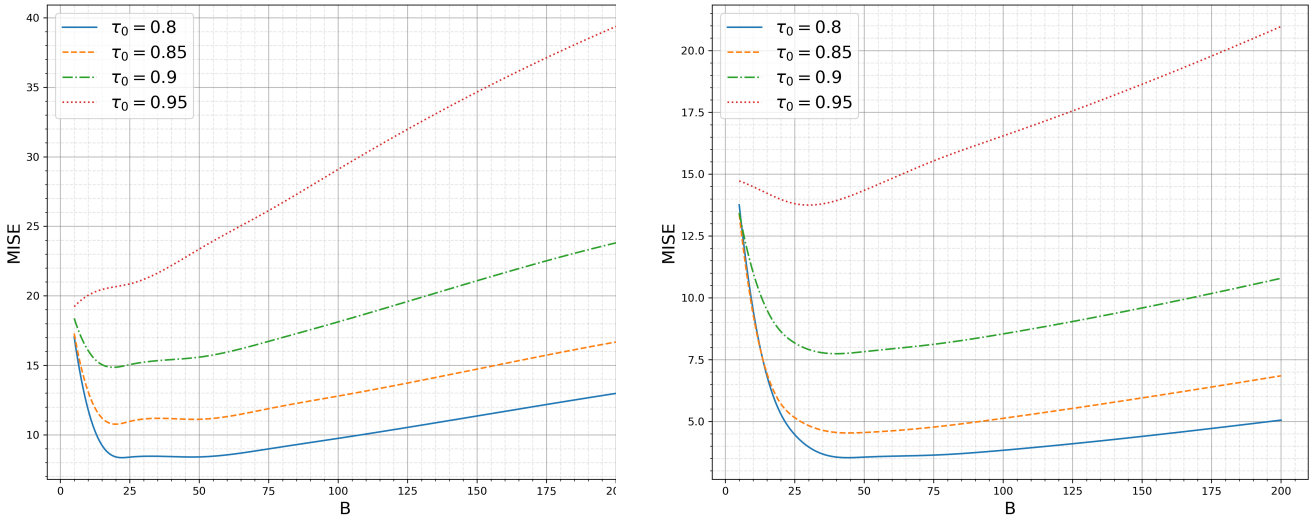


(a) Modèle 1

(b) Modèle 2

FIGURE C.6 – Évolution de la MISE en fonction du nombre d'arbre et de la profondeur maximale de chaque arbre pour le modèle 1 à gauche et le modèle 2 à droite, avec $n = 2000$. Dans la légende, le premier élément de chaque liste correspond à D^σet le deuxième élément correspond à D^γ

C.3 MISE en fonction du seuil du quantile intermédiaire



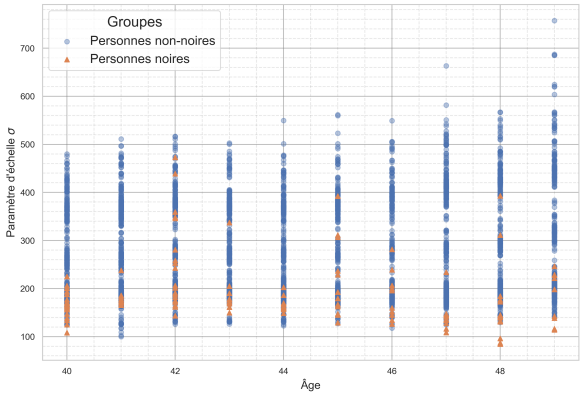
(a) Modèle 1

(b) Modèle 2

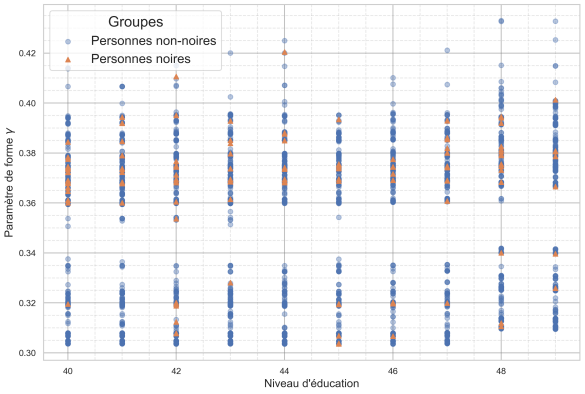
FIGURE C.7 – Évolution de la MISE en fonction du nombre d'arbre et de la profondeur maximale de chaque arbre pour le modèle 1 à gauche et le modèle 2 à droite, avec $n = 5000$. Dans la légende, le premier élément de chaque liste correspond à D^{σ}et le deuxième élément correspond à D^{γ}

C.4 Application du Gbex

C.4.1 Paramètres d'échelle et de forme en fonction de l'âge



(a) Paramètre d'échelle en fonction de l'âge, segmenté par la couleur de peau



(b) Paramètre d'échelle en fonction de l'âge, segmenté par la couleur de peau

FIGURE C.8 – Comportement des paramètres de la GPD en fonction de variables d'intérêts

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences
Place des sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/sc**