Interpretable Algorithms for Regression : Theory and Applications

Vincent Margot

Thèse de doctorat

Spécialité **: Mathématiques**

Option **: Statistiques**

présentée et soutenue publiquement par

Vincent MARGOT

le 02 octobre 2020

pour obtenir le titre de

Docteur de Sorbonne Université

Sujet de la thèse :

Interpretable Algorithms for Regression: Theory and Applications.

Directeur de thèse : **Olivier Wintenberger**
Responsable industriel : **Christophe Geissler**
Co-encadrants de thèse : **Jean-Patrick Baudry & Frédéric Guilloux**

Jury

M. Biau Gérard,	Professeur, Sorbonne Université	Présient
M. Fortin-Stoltz Gilles,	Directeur de recherche, CNRS	Rapporteur
M. Garivier Aurélien,	Professeur, ENS Lyon	Rapporteur
M. Geissler Christophe,	CEO, Advestis	Responsable industriel
Mme. Olteanu Madalina,	Professeur, Dauphine	Examinateur
M. Wintenberger Olivier,	Professeur, Sorbonne Université	Directeur
M. Baudry Jean-Patrick,	Maître de conférences, Sorbonne Université	Invité
M. Guilloux Frédéric,	Maître de conférences, Sorbonne Université	Invité

Sorbonne Université - LPSM
Campus Pierre et Marie Curie
4, Place Jussieu
75005 Paris

A ma femme, mon fils et ma famille
avec tout mon amour...

et à tous ceux qui pensaient que je n'aurais pas mon BAC
avec tout mon orgeuil.

Remerciements

Enfin les remerciements, sans doute la partie la plus lue d'un manuscrit de thèse (à moins que je ne sois le seul à trouver cette littérature fascinante). Les gens qui m'ont entouré pendant cette thèse peuvent imaginer le soulagement que j'ai à écrire ces lignes. Ça y est, c'est terminé !

Je voulais d'abord grandement remercier mon directeur de thèse Olivier Wintenber et mon responsable industriel Christophe Geissler sans qui cette thèse n'aurait pas eu lieu. Merci à Olivier pour sa bienveillance, son soutien, ses encouragements et ses nombreuses relectures. Il a su me pousser lorsque c'était nécessaire et me rassurer dans les moments de doute. Merci à Christophe pour son soutien dans les creux de vagues et pour son enthousiasme dans les bons moments.

Je remercie également mes co-encradents Jean-Patrick Baudry et Frédéric Guilloux pour leur patience, leur disponibilité et leurs conseils. Merci à Jean-Patrick pour son sans du détail qui m'a poussé à beaucoup plus de rigueur. Et merci à Frédéric pour sa pédagogie pour m'expliquer en quoi ces détails étaient toujours importants.

Merci à Aurélien Garivier et Gilles Stoltz qui ont bien voulu rapporter cette thèse et qui, grâce à leurs remarques et questions, m'ont permis d'améliorer ce manuscrit et de préparer la soutenance dans d'excellentes conditions. Je tiens particulièrement à remercier Gilles pour m'avoir renvoyé mon manuscrit rempli d'annotations, remarques et questions ouvertes qui m'ont grandement servi.

Merci également à Gerard Biau et Madalina Olteanu d'avoir accepté de participer au jury de thèse. Merci pour leurs questions et remarques. J'espère qu'il y en aura d'autres.

J'aimerais, bien évidemment, remercier ma famille pour leurs encouragements et leurs soutiens sans failles et plus particulièrement ma femme, Camille, qui m'a soutenu tous les jours pendant cette thèse et sans qui je n'en serais pas là aujourd'hui.

Un très grand merci a tous mes amis. Merci à Clément et à Florian pour être toujours là en toutes circonstances et même le dimanche soir. Merci à François, Peupeu (non ce n'est pas son vrai nom), Pierre, Philippe-Alexandre (lui en revanche c'est son vrai nom) pour m'avoir donné le goût des mathématiques. Merci à Clémence et Anastasia pour leurs soutiens lors de ma soutenance, même si les mathématiques ne sont pas leur domaine.

Merci également à tous mes collègues d'Advestis Noureddine, Nicolas, David, William, Philippe et Adrien pour leurs encouragements. Un remerciement spécial pour Noureddine qui m'a supporté, sans jamais se plaindre, râler sans cesse sur le bureau d'en fasse.

Je voulais également remercier les personnes de l'ombre, ceux qui m'ont permi, sans même le savoir, d'entreprendre cette thèse. Julien Valletoux, mon responsable de stage de fin d'études, qui a cru en moi et m'avait conseillé d'aller chercher une thèse. Et Salim Khelifa, mon professer de topologie et de théorie de la mesure en L3, qui a été le premier enseignant à m'avoir montré ce que sont vraiment les mathématiques et à quel point elles pouvaient être belles.

Enfin je souhaite faire des non-remerciements à Reed Hastings et Marc Randolph (Netflix) ainsi qu'à François Theurel (le Fossoyeur de Films) et Renaud Jesionek (le Cap'tain du Nexus VI) qui m'ont pris beaucoup de temps mais ont quand même grandement contribué à enrichir ma culture cinématographique.

Résumé

Cette thèse a été motivée par la volonté de créer un algorithme interprétable en analyse de la régression. Dans un premier temps, nous nous sommes concentrés sur les algorithmes interprétables les plus courants : les algorithmes à bases de règles de décisions. Malheureusement, les conditions théoriques sur ces algorithmes engendrent une perte d'interprétabilité lorsque la dimension augmente. Partant du principe que moins il y a de règles, meilleure est l'interprétabilité, nous avons introduit une nouvelle famille d'algorithmes à base d'un petit nombre de règles dites significatives. Ce principe a été traduit en une mesure d'interprétabilité permettant la comparaison entre algorithmes générant des règles.

Nous avons ensuite introduit une nouvelle méthode pour générer des estimateurs interprétables de la fonction de régression. L'idée repose sur la notion de recouvrements des données. L'objectif est de construire à partir des données un recouvrement de l'espace des variables explicatives au lieu d'imposer une partition comme pour les algorithmes à bases de règles usuels. Chaque élément du recouvrement est sélectionné selon un critère de significativité ou d'insignifiance. Les éléments significatifs servent à décrire le modèle et les éléments insignifiants permettent d'obtenir un recouvrement. Une partition est construite à partir du recouvrement pour définir une prédiction. La méthode prédit la variable d'intérêt comme l'espérance conditionnelle empirique sur les cellules de la partition activées par les variables explicatives correspondantes. Ainsi, ces prédictions sont identiques à celles issues d'algorithmes de partitionnement dépendant des données et s'interprètent comme un minimiseur du risque empirique. Nous prouvons ainsi que de telles méthodes fournissent des estimateurs consistants de la fonction de régression sans utiliser la condition de rétrécissement des cellules qui apparaît dans la littérature. Ce faisant, nous réduisons le nombre d'éléments du recouvrement et nous améliorons l'interprétabilité du modèle obtenu.

À partir de cette théorie, nous avons développé deux algorithmes. Le premier, Covering Algorithm (CA), est un algorithme rendant interprétable Random Forests (RF), un algorithme vu comme une boîte noire non-interprétable. L'algorithme extrait des règles obtenues par RF un recouvrement de règles significatives et insignifiantes. Le second, Rule Induction Covering Estimator (RICE), ne conçoit que des règles significatives et insignifiantes contrairement à (CA). RICE en sélectionne un petit ensemble pour former un recouvrement. Les règles significatives sont utilisées pour interpréter le modèle et le recouvrement permet de définir un estimateur de la fonction de régression qui, sous certaines conditions, est consistant. Enfin, une version open-source du code est disponible sur GitHub.

Abstract

This thesis was motivated by the desire to make an interpretable algorithm for regression analysis. First, we focused on the most common interpretable algorithms, i.e., rule-based algorithms. Unfortunately, the theoretical conditions on these algorithms generate a loss of interpretability when the dimension increases. Starting from the principle that the fewer the rules, the better the interpretability, we have introduced a new family of algorithms based on a small number of so-called significant rules. This principle has been translated into a measure of interpretability allowing the comparison between algorithms generating rules.

Then, we have introduced a new method to generate interpretable estimator of the regression function, based on data-dependent coverings. The goal is to extract from the data a covering of the explanatory variables space instead of a partition. Each element of the covering is labeled as significant or insignificant. Significant elements are used to describe the model and insignificant elements are used to obtain a covering. Then, a partition is made from the covering to define an estimator. This estimator predicts the empirical conditional expectation on the cells of the partition. Thus, these estimators have the same writing as those resulting from data-dependent partitioning algorithms. We have proven the consistency of such estimators without the cell shrinking condition that appears in the literature, thus reducing the number of elements in the covering.

From this theory, we have developed two algorithms. The first, Covering Algorithm (CA), is an algorithm that makes Random Forests (RF) interpretable, an algorithm seen as a black box that cannot be interpreted. The algorithm extracts from the rules obtained by RF a covering of significant and insignificant rules. The second, Rule Induction Covering Estimator (RICE), designs only significant and insignificant rules unlike (CA). RICE selects a sparse set to form a covering. The significant rules are used to interpret the model and the covering makes it possible to define an estimator of the regression function which, under certain conditions, is consistent. Finally, an open-source version of the code is available on GitHub.

Contents

Plan de la thèse

Le chapitre 1, introductif, présente le cadre général de l'analyse de la régression et de l'interprétabilité des modèles prédictifs. Parmi les algorithmes interprétables, nous nous concentrons sur ceux à base de règles, dont l'état de l'art montre les limites d'interprétabilité. Ces limites ont motivé le travail de cette thèse. La fin du chapitre 1 est consacré à une revue de l'état de l'art et un résumé de l'ensemble des contributions.

Pour palier le manque d'interprétabilité, nous avons développé une nouvelle famille d'algorithmes de règles fondés sur des recouvrements de l'espace des variables explicatives au lieu des partitions usuellement étudiées. Les principes de ces algorithmes sont introduits dans le chapitre 2. Nous y présentons également un théorème de convergence faible de l'estimateur de la fonction de régression généré par de tels algorithmes. Ce chapitre se conclut par une illustration de cette théorie via un algorithme de recouvrement, Covering Algorithm, utilisant un Random Forest comme générateur de règles. Cet algorithme extrait un sous-ensemble parcimonieux de règles permettant de décrire le modèle de façon simple tout en gardant une bonne capacité prédictive. Ce chapitre reprend un article soumis à *Electronic Journal of Statistics*.

Cette nouvelle famille d'algorithme nous a permis de développer *RICE*, Rule Induction Covering Estimator, un algorithme interprétable adapté au problème de régression présenté dans le chapitre 3. Cet algorithme génère et sélectionne un petit nombre de règles en adaptant les principes théoriques des algorithmes de recouvrement présentés dans le chapitre précédent. Il fournit sous certaines conditions un estimateur consistant de la fonction de régression. Le code *Python* est disponible sur GitHub. La présentation de cet algorithme fait l'objet d'un article soumis à *Journal of Machine Learning Research*.

Enfin, en Annexe nous regroupons deux articles de conférences sur des algorithmes interprétables développés au cours de cette thèse. Le premier, *AdLearn*, est un algorithme de recouvrement dépendant des données développé chez *Advestis* dans le domaine de l'investissement socialement responsable. Les résultats ont été présentés lors de *The Seventh Public Investors Conference BIS / World Bank / Bank of Canada / Bank of Italy* [11]. Cet algorithme a la particularité de considérer chaque prédiction issue d'une règle comme un expert et d'agréger les prédictions en utilisant la théorie de l'agrégation d'experts développé dans [9]. Le second algorithme *RIPE*, Rule Induction Partitioning Estimator, a été présenté lors de *The 14th International Conference on Machine Learning and Data Mining* [46]. Il a été le premier algorithme fondé sur des recouvrements et utilisant le *partitioning trick* pour générer un estimateur de la fonction de régression. Mais les propriétés des règles que RIPE génère ne permettent pas, contrairement à RICE, d'obtenir de résultats théoriques de consistance. Le code *Python* est disponible sur GitHub.

"Sois le changement que tu veux voir dans le Monde."
Mahatma Gandhi.

Chapter 1

Introduction et contributions

En 1959 Arthur Samuel a défini le machine learning (ML) ou apprentissage automatique comme

> *"The field of study that gives computers the ability to learn without being explicitly programmed."*

Le ML est une discipline charnière entre l'informatique pour la partie application et l'apprentissage statistique pour la partie théorique. Le but de cette discipline est de développer des algorithmes informatiques capables à partir d'un ensemble de données de prédire/décrire un phénomène. La construction et l'évaluation de ces algorithmes se fait en suivant des principes développés en théorie de l'apprentissage statistique. Les lecteurs curieux pourront consulter [50] pour une introduction et une revue des différentes techniques de ML et les ouvrages [72] et [30] pour des introductions détaillées de la théorie de l'apprentissage statistique.

Dans cette thèse nous nous intéressons à l'analyse de la régression : le statisticien dispose de données composées d'une variable d'intérêt Y à valeurs dans $\mathbb{R}$ et d'un ensemble de variables $\mathbf{X} \in \mathbb{R}^d$, appelées variables explicatives, pouvant être utilisées pour expliquer la variable d'intérêt. L'objectif de cette analyse est d'être capable de prédire la valeur de la variable d'intérêt à partir de nouvelles observations des variables explicatives.

1.1 Le cadre général

Nous analysons la régression suivante : Soit $(\mathbf{X}, Y)$ un couple de variables aléatoires de loi $\mathbb{Q}$ inconnue à valeurs dans $\mathbb{R}^d \times \mathbb{R}$ tel que

$$Y = g^*(\mathbf{X}) + Z, \tag{1.1}$$

avec $\mathbb{E}[Z] = 0$, $\mathbb{V}(Z) = \sigma^2 > 0$ et g^* une fonction mesurable de $\mathbb{R}^d$ dans $\mathbb{R}$.

Nous faisons les hypothèses suivantes :

— Z est indépendante de $\mathbf{X}$; (H1)

— Y est bornée : $\mathbb{Q}(\mathcal{S}) = 1$ avec $\mathcal{S} = \mathbb{R}^d \times [-L, L]$, pour $L > 0$ (inconnue). (H2)

L'objectif est d'être capable de prédire la variable Y sachant $\mathbf{X}$. On note $\mathcal{G}$ l'ensemble des fonctions mesurables de $\mathbb{R}^d$ dans $\mathbb{R}$.

On observe un échantillon $D_n = ((\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n))$, où les couples $(\mathbf{X}_i, Y_i)$ sont indépendants et identiquement distribués (i.i.d) de loi $\mathbb{Q}$. À partir de D_n, on construit une fonction, appelée prédicteur,

$$\begin{array}{ccccccc} g & : & (\mathbb{R}^d \times \mathbb{R})^n & \times & \mathbb{R}^d & \to & \mathbb{R} \\ & & D_n & ; & \mathbf{x} & \mapsto & g(D_n, \mathbf{x}). \end{array}$$

Sauf ambiguïté, nous noterons, pour $\mathbf{x} \in \mathbb{R}^d$, $g_n(\mathbf{x}) = g(D_n, \mathbf{x})$, g étant vue comme une fonction de $\mathcal{G}$, aléatoire car dépendant de D_n.

La qualité d'un prédicteur est évaluée par la *fonction de perte* $\mathcal{L} : \mathcal{G} \to \mathbb{R}$ définie par :

$$\mathcal{L}(g) = \mathbb{E}_{(\mathbf{X},Y)\sim\mathbb{Q}}\left[(g_n(\mathbf{X}) - Y)^2 \mid D_n\right] \tag{1.2}$$

Sauf ambiguïté, nous noterons $\mathbb{E}_{\mathbb{Q}} = \mathbb{E}_{(\mathbf{X},Y)\sim\mathbb{Q}}$.

L'objectif lors de la construction d'un prédicteur est qu'il ait la plus petite perte possible. Or on sait que le meilleur prédicteur, au sens de la perte (1.2) est la *fonction de régression*. En effet on a

$$g^*(\mathbf{X}) = \arg\min_{g\in\mathcal{G}} \mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - Y)^2 \middle| \mathbf{X}\right] = \mathbb{E}\left[Y \middle| \mathbf{X}\right] \ \mathbb{Q}\text{-p.s.} \tag{1.3}$$

Pour plus de détail nous renvoyons à Hastie et al. [30, p. 18-22] ou encore Arlot and Celisse [2, p. 44].

Sachant que la perte minimale est atteinte pour g^* il semble naturel d'introduire la notion d'*excès de perte* $\ell : \mathcal{G} \times \mathcal{G} \to \mathbb{R}_+$. Elle est définie par

$$\ell(g^*, g) = \mathcal{L}(g) - \mathcal{L}(g^*). \tag{1.4}$$

L'utilisation de l'excès de perte permet de mettre en lumière le fait que vouloir construire le meilleur prédicteur et vouloir estimer la fonction de régression sont les mêmes objectifs. En effet nous avons la relation

$$\ell(g^*, g) = \mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))^2\right]$$

d'après le développement de la perte quadratique :

$$\begin{aligned}\mathcal{L}(g) = \mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - Y)^2\right] &= \mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - g^*(\mathbf{X}) + g^*(\mathbf{X}) - Y)^2\right] \\ &= \mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))^2\right] + \mathcal{L}(g^*) \\ &\quad + 2\mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))(g^*(\mathbf{X}) - Y)\right],\end{aligned}$$

le dernier terme de la somme étant nul car

$$\begin{aligned}\mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))(g^*(\mathbf{X}) - Y)\right] &= \mathbb{E}_{\mathbb{Q}}\left[\mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))(g^*(\mathbf{X}) - Y) \mid \mathbf{X}\right]\right] \\ &= \mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))(g^*(\mathbf{X}) - \mathbb{E}_{\mathbb{Q}}\left[Y \mid \mathbf{X}\right])\right] \\ &= 0 \text{ par (1.3)}.\end{aligned}$$

1.1.1 L'estimation de la fonction de régression

Pour construire un estimateur g_n de g^* à partir d'un échantillon D_n, nous utilisons un algorithme statistique. L'échantillon et le choix de l'algorithme conditionnent la classe de fonctions $G_n \subset \mathcal{G}$ à laquelle appartient l'estimateur g_n. La qualité de l'algorithme est donc mesurée par la perte (1.2) de l'estimateur g_n qu'il génère, et qui dépend de l'échantillon, mais également par le *risque*, notée R et défini par :

$$R(g^*, g_n) = \mathbb{E}_{D_n\sim\mathbb{Q}^{\otimes n}}\left[\ell(g^*, g_n)\right]. \tag{1.5}$$

Sauf ambiguïté, nous noterons $\mathbb{E}_{\mathbb{Q}^{\otimes n}} = \mathbb{E}_{D_n\sim\mathbb{Q}^{\otimes n}}$.

L'estimateur peut être sélectionné dans G_n comme étant le minimiseur de la perte empirique $\mathcal{L}_n$ définie par

$$\mathcal{L}_n(g) = \frac{1}{n}\sum_{i=1}^{n}(g(\mathbf{X}_i) - Y_i)^2. \tag{1.6}$$

On a alors

$$g_n \in \arg\min_{g \in G_n} \mathcal{L}_n(g) \tag{1.7}$$

Ce principe est appelé Empirical Risk Minimization (ERM) est a été introduit dans [74].

Lorsque l'on étudie un estimateur on commence par s'assurer qu'il est *universellement consistant*, c'est-à-dire qu'il converge vers g^* pour $n \to \infty$ quelle que soit la loi $\mathbb{Q}$. Nous rappelons les définitions de consistance d'un estimateur, présentées dans Györfi et al. [28, Def 1.1 et Def 1.2].

Définition 1.1.1. *Un estimateur g_n est dit universellement faiblement consistant si et seulement si pour toute loi $\mathbb{Q}$ telle que $\mathbb{E}[Y^2] < \infty$, on a*

$$\lim_{n\to\infty} R(g^*, g_n) = 0\,.$$

Définition 1.1.2. *Un estimateur g_n est dit universellement fortement consistant si et seulement si pour toute loi $\mathbb{Q}$ telle que $\mathbb{E}[Y^2] < \infty$, on a*

$$\lim_{n\to\infty} \ell(g^*, g_n) = 0, \quad \mathbb{Q}^{\otimes n}\text{-}p.s.$$

Remarque 1. Ces deux définitions nous permettent de lier un des objectifs de l'estimation, la consistance, avec des objectifs de prédiction comme la minimisation du risque (dans la définition 1.1.1) ou la minimisation de l'excès de perte (dans la définition 1.1.2).

1.1.2 Algorithmes de régression

En statistique non-paramétrique une méthode usuelle pour construire un estimateur de la fonction de régression consiste à utiliser un algorithme par moyenne locale de la forme

$$g_n(\mathbf{x}) = \sum_{i=1}^{n} W_{n,i}(\mathbf{x}) Y_i,$$

avec $W_{n,i}(\mathbf{x}) = W_{n,i}(\mathbf{X}_1, \ldots, \mathbf{X}_n; \mathbf{x}) \in \mathbb{R}$ donne du poids aux indices i tels que $\mathbf{X}_i$ est proche de $\mathbf{x}$. On peut citer par exemple l'algorithme à noyau de Nadaraya-Watson [52] ou celui des k-plus proches voisins.

Dans cette thèse nous nous concentrons sur les algorithmes à partition. Ces algorithmes découpent, à chaque étape n, l'espace $\mathbb{R}^d$ en une partition $\mathcal{P}_n = \{A_{1,n}, A_{2,n}, \ldots\}$ et l'estimateur généré peut s'écrire sous la forme précédente avec

$$W_{n,i}(\mathbf{x}) = \frac{\mathbf{1}_{\mathbf{X}_i \in A_n(\mathbf{x})}}{\sum_{i=1}^{n} \mathbf{1}_{\mathbf{X}_i \in A_n(\mathbf{x})}}, \tag{1.8}$$

où $A_n(\mathbf{x})$ est la cellule de $\mathcal{P}_n$ qui contient $\mathbf{x}$.

Remarque 2. Cette idée peut être généralisée en relâchant l'hypothèse que la collection $\mathcal{P}_n$ soit une partition : on peut considérer des collections $\mathcal{P}_n$ de sous-ensembles de $\mathbb{R}^d$ avec de possibles recouvrements et de possibles zones non couvertes. Dans ce cas, $W_{n,i}(\mathbf{x})$ peut ne pas être défini si aucune cellule ne contient $\mathbf{x}$. On trouve souvent la convention $0/0 = 0$. Cela peut se traduire par le fait qu'en l'absence d'information l'estimateur prédit 0. Mais cette convention peut être discutable. On pourrait se demander pourquoi ne pas prédire la moyenne empirique de Y si aucune cellule ne contient $\mathbf{x}$? Le problème de prédiction en l'absence d'information a été l'une des problématiques de cette thèse. La gestion des cas où les ensembles de $\mathcal{P}_n$ se recouvrent mutuellement, en a été une autre.

Algorithme à partition indépendante des données

Un premier type d'algorithme à partition regroupe les algorithmes à partition ne tenant compte que de la taille n de l'échantillon D_n. Le plus connu est le régressogramme [69]. Le théorème suivant, reposant sur les résultats de [65], donne des conditions suffisantes pour qu'un estimateur construit à partir d'un algorithme à partition soit faiblement universellement consistant. Ce théorème repose sur la notion de diamètre, notée *diam* et définie par :

$$diam(A) = sup\{d(\mathbf{x}, y); \mathbf{x} \in A, \mathbf{x} \in A\},$$

où d est la distance euclidienne sur $\mathbb{R}^d$.

Théorème 1.1.1. *Théorème de consistance faible, voir [28, Théorème 4.2]*
Soit g_n un estimateur défini sur une partition $\mathcal{P}_n = \{A_{1,n}, A_{2,n}, \ldots\}$. Si pour chaque sphère S centrée en l'origine on a :

$$\bullet \max_{\{j:A_{j,n}\cap S\neq\emptyset\}} diam(A_{j,n}) \to 0 \ (n \to \infty), \tag{1.9}$$

$$\bullet \frac{|\{j : A_{j,n} \cap S \neq \emptyset\}|}{n} \to 0 \ (n \to \infty), \tag{1.10}$$

alors

$$R\,(g^*, g_n) \to 0 \ (n \to \infty).$$

La première condition assure que les cellules de la partition tendent vers 0 dans un certain sens. Cette condition correspond à l'idée d'une estimation locale. La seconde traduit le fait que le nombre de cellules dans un ensemble borné est petit par rapport à n, ce qui permet d'assurer que chaque cellule contient plusieurs points de l'échantillon D_n et ainsi d'avoir une meilleure estimation.

Exemple 1. *Le régressogramme*
Le régressogramme est un algorithme à partition où chaque cellule a un volume égal à $(h_n)^d$ une valeur dépendante de n mais indépendante de D_n. La Figure 1.1 représente différentes partitions générées pour un échantillon D_n de plus en plus grand. On remarque que cette partition a des cellules contenant un nombre de points très hétérogènes. Quid de la qualité de prédiction dans une cellule ne contenant qu'une seule observation ?

Néanmoins, d'après le théorème 1.1.1 la consistance d'un estimateur issu d'un régressogramme est vérifiée si h_n vérifie que

$$h_n \to 0 \ (n \to \infty) \text{ et } n(h_n)^d \to \infty \ (n \to \infty).$$

■

Algorithmes à partition dépendant des données

Il existe un second type d'algorithmes qui construit une partition dépendant de D_n. On parle alors d'estimation à partition dépendante des données [28, Chap 13]. Ces algorithmes permettent de construire un estimateur en utilisant deux fois l'échantillon D_n. D'abord, pour construire une partition $\mathcal{P}_n = \mathcal{P}_n(D_n)$ de l'espace des variables explicatives à partir des données. Puis, $\mathcal{P}_n$ et D_n sont utilisés pour construire l'estimateur g_n en moyennant les Y_i pour lesquels $\mathbf{X}_i$ est proche de $\mathbf{X}$, c'est-à-dire dans le même cellule de la partition (1.8).

Exemple 2. *L'algorithme à partition cubique*
L'algorithme à partition cubique est comparable au régressogramme dans le sens où il

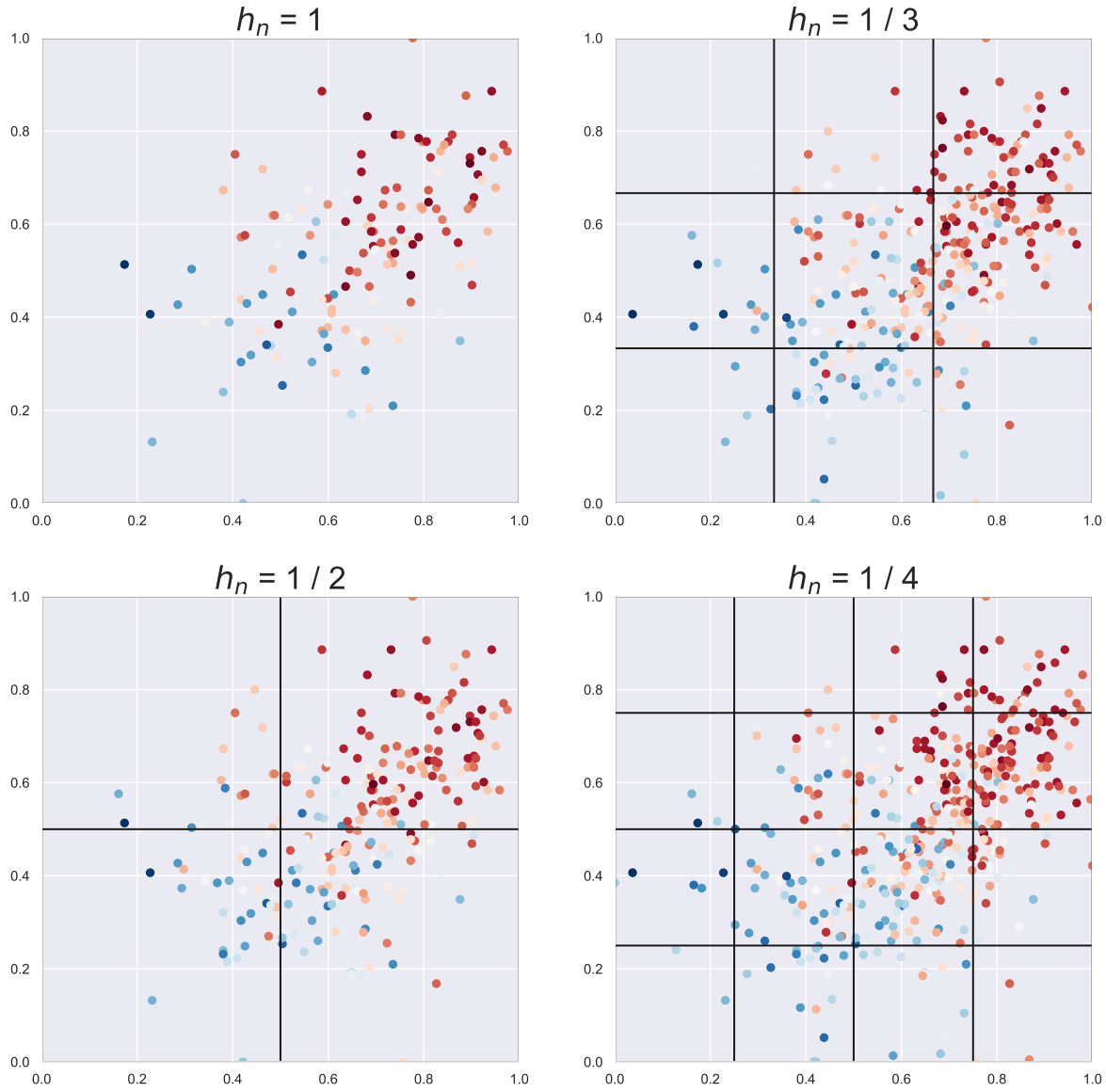


Figure 1.1 – Représentation de la partition d'un régressogramme pour différents D_n.

partitionne l'espace des variables explicatives en cellules géométriquement identiques [1]. La différence est dans le fait que la taille des cellules est déterminée à partir de D_n et d'un paramètre k.

Sans perdre de généralité nous supposons $d = 2$. L'algorithme partitionne un espace déterminé par un hyperrectangle $[L_n, R_n)^2$ dépendant de D_n. Soient $k_{min}(n)$ et $k_{max}(n) \in \mathbb{N}^*$ tels que $k_{min}(n) \leq k_{max}(n)$, dépendants uniquement de la taille de n. Le statisticien génère aléatoirement un $k \in \mathbb{N}$ tel que $k_{min}(n) \leq k \leq k_{max}(n)$. L'algorithme fournit alors la partition suivante :

$$\begin{aligned}\mathcal{P}_{n,k} = &\left\{\mathbb{R}^2 \setminus [L_n, R_n)^2\right\} \cup \\ &\{[L_n + i_1 h_{n,k}, L_n + (i_1+1)h_{n,k}[\times[L_n + i_2 h_{n,k}, L_n + (i_2+1)h_{n,k}[: \\ &i_1, i_2 \in \{0, \ldots, k-1\}\},\end{aligned}$$

avec $h_{n,k} = \dfrac{R_n - L_n}{k}$ la taille de la grille de la partition (voir la Figure 1.2).

■

1. Dans cet algorithme il y a toujours une cellule différente des autres. Il s'agit de celle recouvrant l'espace des valeurs non observées dans D_n

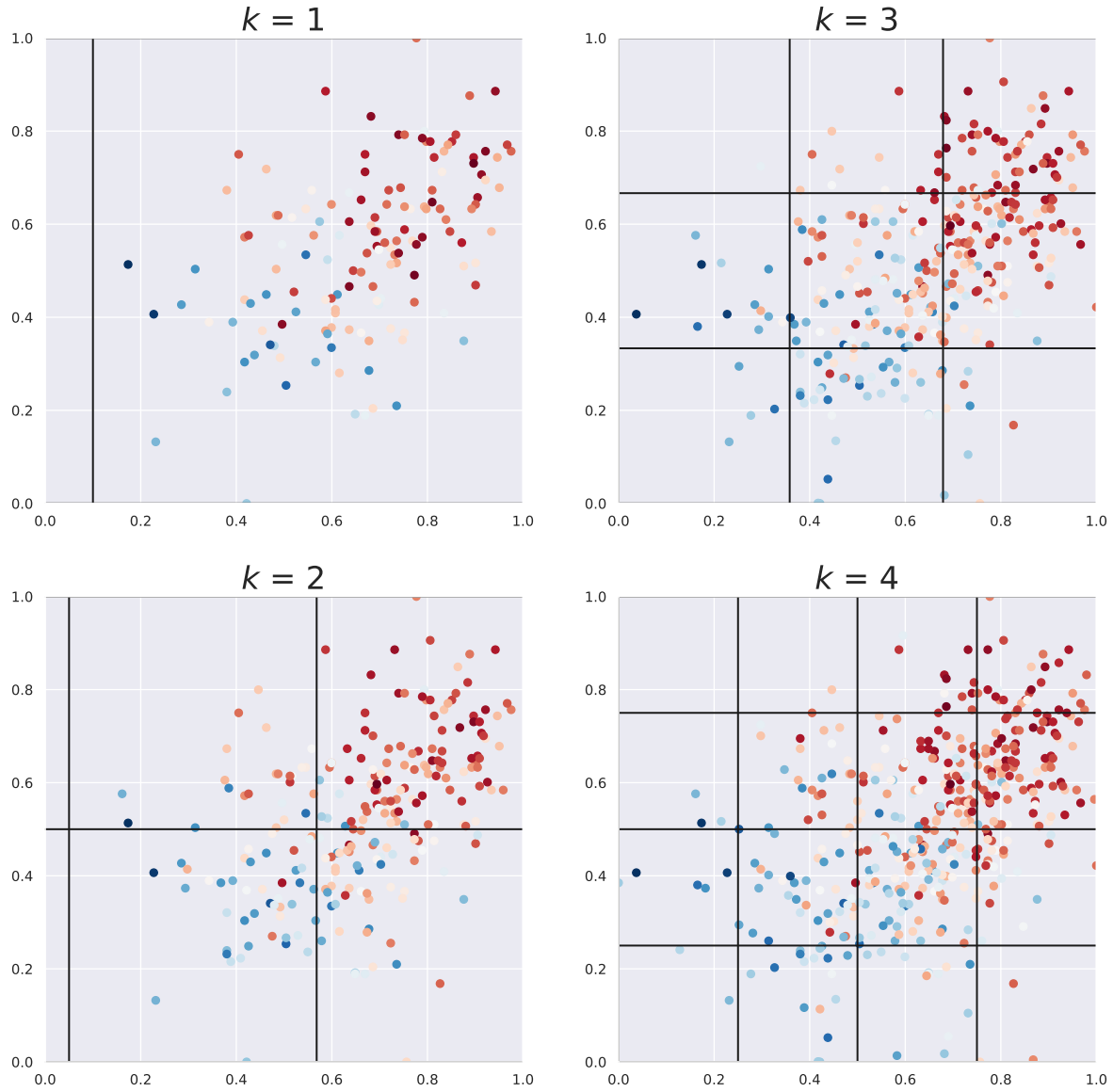


Figure 1.2 – Représentation de la partition cubique pour différent D_n.

Exemple 3. *L'algorithme à partition quantile*
L'algorithme à partition quantile est une variante de celui à partition cubique. Considérons le même jeux de données que pour l'exemple précédent. Tout comme l'algorithme à partition cubique, il partitionne un espace déterminé par un hyperrectangle $[L_n, R_n)^2$ dépendant uniquement de n. Soit $m_n \in \mathbb{N}$ choisi en fonction de n. L'algorithme construit une partition $\mathcal{P}_n$ de $[L_n, R_n)^2$ en utilisant les quantiles empiriques d'ordre m_n de D_n dans chaque dimension. On obtient alors une partition avec des hyperrectangles de taille variables (voir la Figure 1.3).

■

Exemple 4. *L'algorithme à partition par blocs statistiquement équivalents*
Le problème avec les partition précédentes c'est la possibilité d'avoir des cellules vides ce qui implique de devoir faire un choix pour la prédiction en l'absence d'information (cf la remarque 2). Pour palier à ce problème, on peut utiliser l'algorithme à partition par blocs statistiquement équivalents.

Considérons le même jeux de données que pour les exemples précédents. Soit $k_n \in \mathbb{N}$ choisi en fonction de n. La partition $\mathcal{P}_n$ construite par l'algorithme de partition par blocs statistiquement équivalents est telle que chaque cellule, sauf une, contient exactement k_n

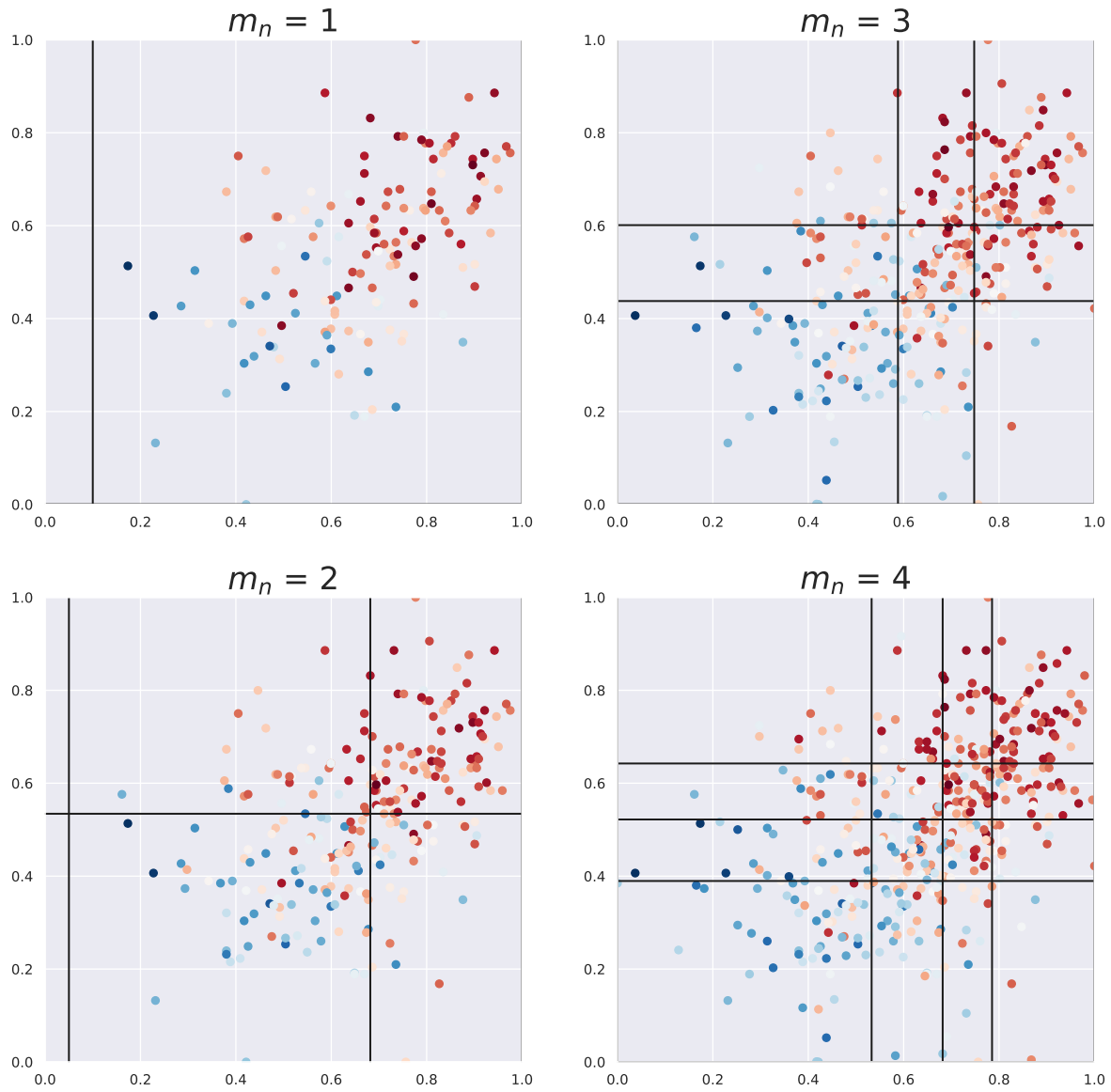


Figure 1.3 – Représentation de la partition quantile pour différent D_n.

points [2]. En pratique on utilise les quantiles empiriques pour identifier les intervalles (voir la Figure 1.4).

■

La complexité de l'algorithme peut être mesurée par une quantité combinatoire appelée *nombre de partitionnement*, introduit par [53] pour la régression et par [44] pour la classification. Cette quantité est comparable à la fonction de croissance (ou coefficient de pulvérisation) présentée en 1971 par Vapnik et Chervonenkis pour les classes de sous-ensembles [73]. Le nombre de partitionnement repose sur le nombre de partitions différentes qu'un algorithme peut générer indépendamment de D_n. Pour cela on introduit

$$\Pi_n = \left\{ \mathcal{P}_n \left(\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \right) : (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathbb{R} \right\}, \tag{1.11}$$

la famille des partitions issue d'un même algorithme. Et

$$M(\Pi_n) = \max_{\mathcal{P}_n \in \Pi_n} \{|\mathcal{P}_n|\},$$

le plus grand cardinal des éléments d'une famille de partitions Π_n.

2. En effet on a $|\mathcal{P}_n| = \lceil k_n/n \rceil$ donc il y a au plus une cellule qui ne vérifie pas cette condition.

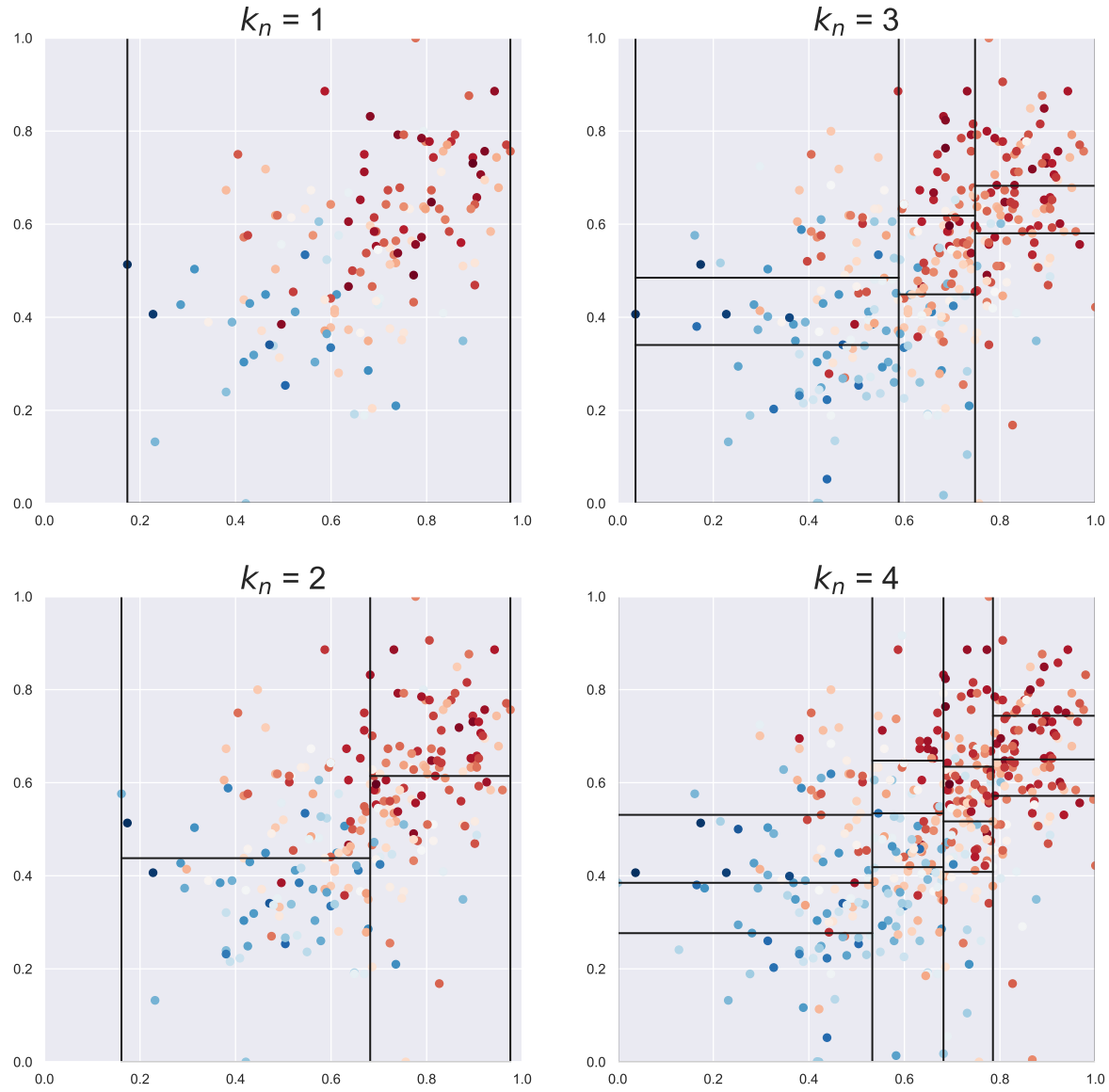


Figure 1.4 – Représentation de la partition par bloques statistiquement équivalents pour différent D_n.

Définition 1.1.3. *Soit Π_n une famille de partitions de $\mathbb{R}^d$. Pour $\mathbf{x}_1^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in (\mathbb{R}^d)^n$ on pose $\Delta(\mathbf{x}_1^n, \Pi_n)$ le nombre de partitions distinctes de $\mathbf{x}_1^n$ engendrées par les éléments de Π_n. Par distinctes on entend que l'ordre d'apparition n'importe pas. En d'autre terme $\Delta(\mathbf{x}_1^n, \Pi_n)$ est égale au nombre des différentes partitions $\{\mathbf{x}_1^n \cap A : A \in \mathcal{P}\}$ de $\mathbf{x}_1^n$ pour $\mathcal{P} \in \Pi_n$. On a*

$$\Delta(x_1^n, \Pi_n) = \#\left\{\{x_1^n \cap A : A \in \mathcal{P}\} : \mathcal{P} \in \Pi_n\right\}.$$

Le nombre de partitionnement $\Delta_n(\Pi_n)$ est défini par :

$$\Delta_n(\Pi_n) = \max_{\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in (\mathbb{R}^d)^n} \Delta(\mathbf{x}_1^n, \Pi_n).$$

Le nombre de partitionnement est le nombre maximum de partitions différentes de n'importe quel ensemble de n points, engendrées par les éléments de la partition issue de l'algorithme Π_n.

Exemple 5. *La partition quantile suite ...*

Reprenons l'algorithme à partition quantile de l'exemple 3 et calculons $M(\Pi_n)$ et $\Delta_n(\Pi_n)$. Chaque axe est découpée en m_n parties. Ainsi, les partitions générées comportent

au plus m_n^2 cellules. Nous avons alors

$$M(\Pi_n) = m_n^2.$$

Le calcul de $\Delta_n(\Pi_n)$ est un peu plus délicat. Supposons $d = 1$ et soit $x_1^n = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Sans perte de généralité nous supposons que $x_1 < x_2 < \cdots < x_n$. Pour avoir m_n intervalles il faut $m_n - 1$ bornes. On note $\{b_1, \ldots, b_{m_n-1}\}$ ces bornes, telles que $b_1 < b_2 < \cdots < b_{m_n-1}$. On pose $b_0 = -\infty$ et $b_{m_n} = +\infty$. Nous allons compter le nombre de partitionnements différents de x_1^n avec p cellules non vides pour $p \in \{1, \ldots, m_n\}$.

1. Pour $p = 1$ il faut qu'il existe $0 \leq j \leq m_n - 1$ tel que $b_j < x_1$ et $b_{j+1} > x_n$. Dans les deux cas le partitionnement de x_1^n sont les mêmes (tous les points dans une même cellule). Il y a donc une partition de x_1^n avec une cellule non vide.
2. Pour $p = 2$ nous obtenons le même partitionnement si pour un indice $i \in \{1, \ldots, n-1\}$ fixé il existe $j \in \{2, \ldots, m_n\}$ tel que $x_i < b_j < x_{i+1}$. Il y a donc $n - 1$ partitionnement de x_1^n avec 2 cellules non vides correspondant aux $n - 1$ choix possibles du plus grand indice des points contenus dans le premier élément de la partition.
3. Suivons le même raisonnement pour un partitionnement avec $p = 3$ cellules non vides. Dans ce cas, il faut fixer un couple (i, j) avec $1 \leq i < j \leq n - 1$ tels que i soit le plus grand indice des points contenus dans le premier élément de la partition et j le plus grand indice des points contenus dans le second élément de la partition. On a $\binom{n-1}{2}$ partitionnement différents.
4. Et ainsi de suite pour $p > 3$ parties.

En sommant les différents partitionnement de x_1^n avec 1 à m_n cellules non vides on a

$$\Delta_n(\Pi_n) \leq \sum_{l=1}^{m_n} \binom{n-1}{l-1},$$

où l représente le nombre de cellules non vides dans le partitionnement de x_1^n.

Pour $d > 1$ on applique le même raisonnement suivant chaque axe et on obtient

$$\Delta_n(\Pi_n) \leq \left(\sum_{l=1}^{m_n} \binom{n-1}{l-1} \right)^d.$$

On remarque que cette borne n'est valable que pour une partition complète. Si les points peuvent n'appartenir à aucun élément de la partition alors seule la borne $\Delta_n(\Pi_n) \leq \binom{n+m_n}{m_n}^d$ est valable. ■

Dans le théorème de consistance forte (voir Définition 1.1.2) pour les algorithmes à partition dépendante des données, présenté dans Nobel [53], on trouve deux conditions. La condition (1.12) requiert que la famille des partitions Π_n, à laquelle appartient la partition utilisée pour g_n, ne soit pas trop complexe. Cette complexité est contrôlée par le nombre maximal de cellules d'une partition de Π_n et par le nombre de partitionnement $\Delta_n(\Pi_n)$ (cf. (1.12)). Et la condition (1.13) porte sur la partition choisie pour générer l'estimateur g_n et impose que le diamètre des cellules rétrécisse et tende, d'une certaine façon, vers 0.

Théorème 1.1.2. *Théorème de consistance forte, voir Györfi et al. [28, Théorème 13.1] Si g_n et Π_n vérifient les conditions suivantes :*

$$\bullet \ \frac{M(\Pi_n)}{n} \vee \frac{\log(\Delta_n(\Pi_n))}{n} \to 0 \ (n \to \infty), \tag{1.12}$$

$$\bullet \ \textit{pour tout } \gamma > 0 \textit{ et } \delta \in (0,1),$$
$$\inf_{S : S \subset \mathbb{R}^d, \mu_{\mathbf{X}}(S) \geq 1-\delta} \mu_{\mathbf{X}} \left(\{ \mathbf{x} : diam(A_n(\mathbf{x}) \cap S) > \gamma \} \right) \to 0 \ (n \to \infty) \ p.s, \tag{1.13}$$

avec $\mu_{\mathbf{X}}$ la marginal de $\mathbf{X}$ et $A_n(\mathbf{x})$ la cellule de la partition utilisée pour g_n qui contient $\mathbf{x}$.

Alors

$$\ell(g^*, g_n) \to 0 \ (n \to \infty) \ p.s.$$

Exemple 6. *La partition quantile, suite et fin.*

Reprenons l'exemple 3, de l'algorithme à partition quantile. On note $L_n \in \mathbb{R}$ et $R_n \in \mathbb{R}$ les bornes telles que $[L_n, R_n]^2$ soit le plus petit hyperrectangle contenant les données. Si

- $R_n \to \infty, L_n \to -\infty \ (n \to \infty)$,
- $\dfrac{m_n^2}{n} \to 0 \ (n \to \infty)$, (1.14)
- $\dfrac{\log(n)}{m_n} \to 0 \ (n \to \infty)$, (1.15)
- $\dfrac{R_n - L_n}{m_n} \to 0 \ (n \to \infty)$, (1.16)

alors l'estimateur g_n défini pour $\mathcal{P}_n$ est universellement fortement consistant.

La preuve est une application directe du théorème précédent. En effet il faut vérifier que l'estimateur, g_n et $\mathcal{P}_n$ vérifient les conditions. On sait, par l'exemple 5, que:

$$M(\Pi_n) = m_n^2, \qquad \Delta(\Pi_n) \leq \binom{n+m_n}{m_n}^2.$$

A partir de ces résultats et en utilisant (1.14) et (1.15), on peut montrer (1.12). En effet on a

$$\Delta_n(\Pi_n) \leq \binom{n+m_n}{m_n}^2 \leq (n+m_n)^{2m_n} \leq (2n)^{2m_n}.$$

Il reste donc (1.13) à montrer. Soit $\gamma > 0$ et $\delta \in (0,1)$. Par (1.16) et pour n suffisamment grand on a

$$\begin{aligned}
\inf_{S: S\subset\mathbb{R}^2, \mu_{\mathbf{X}}(S)\geq 1-\delta} & \mu_{\mathbf{X}}\left(\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap S) > \gamma\}\right) \\
&\leq \mu_{\mathbf{X}}\left(\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap [L_n, R_n]^2) > \gamma\}\right) \\
&\leq \mu_{\mathbf{X}}\left(\left\{\mathbf{x} : \sqrt{2}\frac{R_n - L_n}{m_n} \geq \gamma\right\}\right) \\
&= \mu_{\mathbf{X}}(\emptyset) \\
&= 0.
\end{aligned}$$

■

Dans les deux théorèmes de consistance les conditions imposent un rétrécissement des cellules de la partition (la condition (1.9) pour le théorème 1.1.1 et la condition (1.13) pour le théorème 1.1.2). En d'autres termes, plus n augmente plus le nombre de cellules augmente. Sous ces conditions, les algorithmes à partition, conçus pour être simples à comprendre, génèrent de fait des estimateurs dont les cellules sont de moins en moins interprétables.

Remarques autour de la condition (1.13)

1. On peut remarquer dans un premier temps que cette condition est toujours vérifiée si $\mathbf{X}$ est une variable aléatoire discrète.

   Preuve. Soit $S \subset \mathbb{R}^d$ tel que $\mu_{\mathbf{X}}(S) \geq 1-\delta$. Étant donné que la condition est un infimum et que $\mathbf{X}$ est une variable aléatoire discrète on peut prendre S comme étant un singleton. Dans ce cas, quelque soit $x \in \mathbb{R}^d$ on a que $A_n(\mathbf{x}) \cap S$ est égale, soit à S, soit à l'ensemble vide. Dans ces deux cas le diamètre de l'intersection est nul. Ce qui implique que l'ensemble $\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap S) > \gamma\}$ est l'ensemble vide, qui par définition est de mesure nulle. □

2. La deuxième remarque consiste à faire le lien entre la condition (1.13) et la condition suivante:
$$diam(A_n(\mathbf{x})) \to 0 \ (n \to \infty) \ p.s. \tag{1.17}$$
   Évidement on a (1.17) qui implique (1.13).

   Preuve. Supposons (1.17) vraie et soit $\gamma > 0$. Alors:
$$\begin{aligned}
(1.17) &\implies \forall R \subset \mathbb{R}^d \text{ tel que } \mu_{\mathbf{X}}(R) > 0 \text{ p.s,} \\
&\quad diam(A_n(\mathbf{x}) \cap R) \to 0 \ (n \to \infty) \\
&\implies \forall \gamma > 0 \text{ et } \forall R \subset \mathbb{R}^d \text{ tel que } \mu_{\mathbf{X}}(R) > 0 \text{ p.s.,} \\
&\quad \{\mathbf{x} : diam(A_n(\mathbf{x}) \cap R) > \gamma\} \to \emptyset \ (n \to \infty) \\
&\implies \forall \gamma > 0 \text{ et } \forall R \subset \mathbb{R}^d \text{ tel que } \mu_{\mathbf{X}}(R) > 0 \text{ p.s.,} \\
&\quad \mu_{\mathbf{X}}\left(\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap R) > \gamma\}\right) \to 0 \ (n \to \infty).
\end{aligned}$$
   De plus on a que pour tout $\delta \in [0,1]$, $\gamma > 0$ et $R \subset \mathbb{R}^d$ tel que $\mu_{\mathbf{X}}(R) \geq 1-\delta$,
$$\begin{aligned}
&\inf_{S : S \in \mathbb{R}^d, \mu_{\mathbf{X}}(S) \geq 1-\delta} \mu_{\mathbf{X}}\left(\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap S) > \gamma\}\right) \\
&\qquad\qquad \leq \mu_{\mathbf{X}}\left(\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap R) > \gamma\}\right).
\end{aligned}$$
   Donc si (1.17) est vraie, le terme de droite tend vers 0 donc (1.13) est vérifiée. □

   Ce qui est intéressant c'est de savoir dans quel cas (1.17) et (1.13) sont équivalentes. À la lumière de la preuve précédente on remarque que (1.17) et (1.13) sont équivalentes si et seulement si pour tout $\gamma > 0$ et $\delta \in (0,1)$ on a :
$$\begin{aligned}
&\inf_{S : S \in \mathbb{R}^d, \mu_{\mathbf{X}}(S) \geq 1-\delta} \mu_{\mathbf{X}}\left(\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap S) > \gamma\}\right) \\
&\quad = \inf_{S : S \in \mathbb{R}^d, \mu_{\mathbf{X}}(S) = 1-\delta} \mu_{\mathbf{X}}\left(\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap S) > \gamma\}\right) \text{ p.s} \\
&\quad = \mu_{\mathbf{X}}\left(\{\mathbf{x} : diam(A_n(\mathbf{x}) \cap R) > \gamma\}\right) \text{ p.s,}
\end{aligned}$$
   quel que soit $R \subset \mathbb{R}^d$ tel que $\mu_{\mathbf{X}}(R) = 1-\delta$ p.s.

1.2 Machine learning et interprétabilité

Il existe une légende urbaine dans le domaine du ML relaté dans [79]:

> *"Once upon a time, the US Army wanted to use neural networks to automatically detect camouflaged enemy tanks. The researchers trained a neural net on 50 photos of camouflaged tanks in trees, and 50 photos of trees without tanks. Using standard techniques for supervised learning, the researchers trained the neural network to a weighting that correctly loaded the training set - output "yes" for the 50 photos of camouflaged tanks, and output "no" for the 50 photos of a forest. This did not ensure, or even imply, that new examples would be classified correctly. The neural network might have "learned" 100 special cases that would not generalize to any new problem. Wisely, the researchers had originally taken 200 photos, 100 photos of tanks and 100 photos of trees. They had used only 50 of each for the training set. The researchers ran the neural network on the remaining 100 photos, and without further training, the neural network classified all remaining photos correctly. Success confirmed! The researchers handed the finished work to the Pentagon, which soon handed it back, complaining that in their own tests the neural network did no better than chance at discriminating photos.*
>
> *It turned out that in the researchers' dataset, photos of camouflaged tanks had been taken on cloudy days, while photos of plain forest had been taken on sunny days. The neural network had learned to distinguish cloudy days from sunny days, instead of distinguishing camouflaged tanks from empty forest."*

Le but de cette histoire (citée également dans [19, 26]) est de mettre en garde contre l'utilisation d'algorithmes générant des modèles non interprétables tels que les algorithmes de type *Support Vector Machines* [72], *Neural networks* ou encore *Random Forests* [7]. Ces algorithmes sont souvent appelés *black box* dans la mesure ou il est très difficile, voir impossible, pour un statisticien de comprendre le modèle généré par l'algorithme. En dépit de leur manque d'interprétabilité, ces algorithmes sont le plus souvent très précis avec une perte (1.2) très faible. Ce qui conduit au compromis *Interprétabilité-Précision*.

Dans certains domaines comme la biologie, la médecine, la justice, la banque ou même la finance, être capable d'interpréter les modèles prédictifs est devenue un enjeu majeur. L'article [43] expose tous les enjeux et tout ce que l'on est susceptible d'attendre de l'interprétabilité d'un algorithme. La Figure 1.5 met en lumière l'intérêt croissant du phénomène dans le milieu scientifique. En effet de plus de plus d'articles statistiques traitant de l'interprétabilité sont publiés chaque année.

Malheureusement aujourd'hui encore, l'interprétabilité souffre d'un manque de rigueur dans sa définition. Dans [15], les auteurs la définissent de la façon suivante

> *"Interpret means to explain or to present in understandable terms. In the context of ML systems, interpretability is the ability to explain or to present in understandable terms to a human."*

Cette définition ne fournit pas un critère objectif pour évaluer l'interprétabilité d'un algorithme. Dans [51] les auteurs donnent la définition suivante

> *"We define interpretable machine learning as the use of machine-learning models for the extraction of relevant knowledge about domain relationships contained in data."*

Leas auteurs proposent d'évaluer l'interprétabilité en utilisant le cadre *PDR* pour Predictive, Description and Revelant. Il s'agit des trois désidératas pour évaluer et concevoir un algorithme interprétable : la précision des prédictions, la qualité de la description du

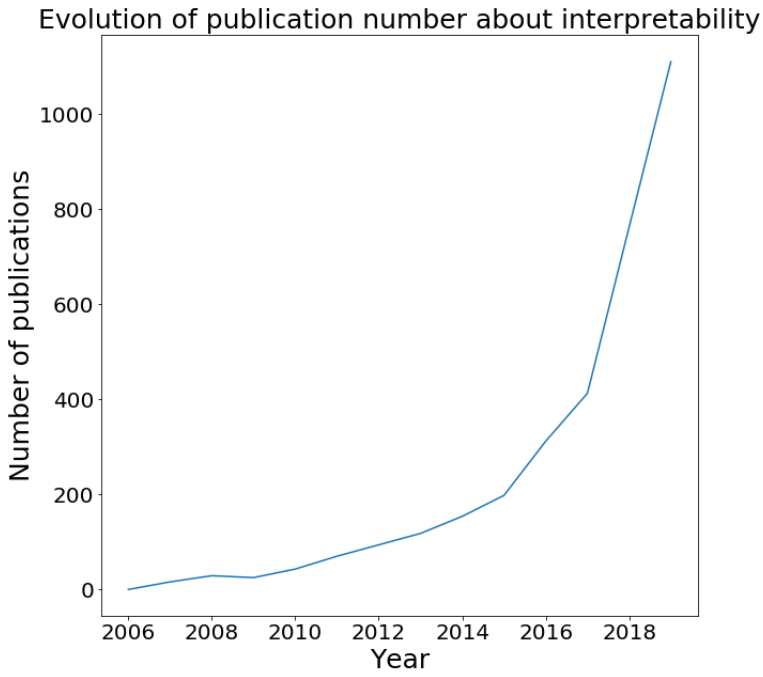


Figure 1.5 – Évolution du nombre de publications sur ArXiv avec le mot *interprétabilité* dans le résumé.

modèle et sa pertinence. Mais, là encore, les deux derniers désidératas sont évalués de façon subjective.

Finalement, même si l'on est capable de comprendre ces définitions il n'existe, à notre connaissance, aucune mesure ou définition formelle des capacités d'interprétabilité d'un algorithme statistique. Ce qui fait que nous ne sommes pas capable de comparer deux algorithmes en termes d'interprétabilité de façon objective.

Il existe deux grandes façons de construire un modèle considéré comme interprétable : la première est d'utiliser un modèle black-box puis de le résumer pour créer un algorithme interprétable dit *post-hoc*. De récentes recherches proposent d'utiliser des algorithmes d'explication tels que *LIME* [58], *DeepLIFT* [62] ou encore *SHAP* [45] pour interpréter les modèles black-box. Ces algorithmes essayent de mesurer l'importance des variables explicatives sur le processus de prédiction (voir [26] ou [24] pour une présentation des méthodes existantes). Mais ces algorithmes ne fournissent que des informations partielles qui peuvent ne pas être suffisantes pour des décisions sensibles d'un point de vue moral ou à fort enjeu [60]. La seconde méthode consiste à utiliser des algorithmes intrinsèquement interprétables, c'est-à-dire des algorithmes générant des modèles interprétables. Il existe deux grandes familles d'algorithmes intrinsèquement interprétables : les *arbres de décision* et les *ensembles de règles*.

1.2.1 Les arbres de décision

Un arbre de décision est un modèle structuré composé de *nœuds* et d'*arcs*. Chaque arc mène à un nœud représentant un test sur une variable explicative. Chaque suite de succès et d'échecs mène à des nœuds terminaux, appelés *feuilles*, représentant les valeurs prédites.

Les algorithmes d'arbres les plus connus sont *CART* [8], *ID3* [55], *C4.5* [56] et *RIPPER* [10].

La Figure 1.6 représente un arbre de régression décrivant les données *Boston housing*

[29]. Dans un souci d'illustration la profondeur de l'arbre a été fixée à 3 et le nombre de feuilles générées à 6. Dans chaque feuille on calcule la perte empirique (1.6), aussi appelée MSE, le nombres de points dans la cellule, *samples* et l'espérance conditionnelle empirique de Y sachant qu'on est dans la cellule, *value*. On renvoie au tableau IV p96-97 de [29] pour une explication des variables Y, RM, $LSTAT$, etc.

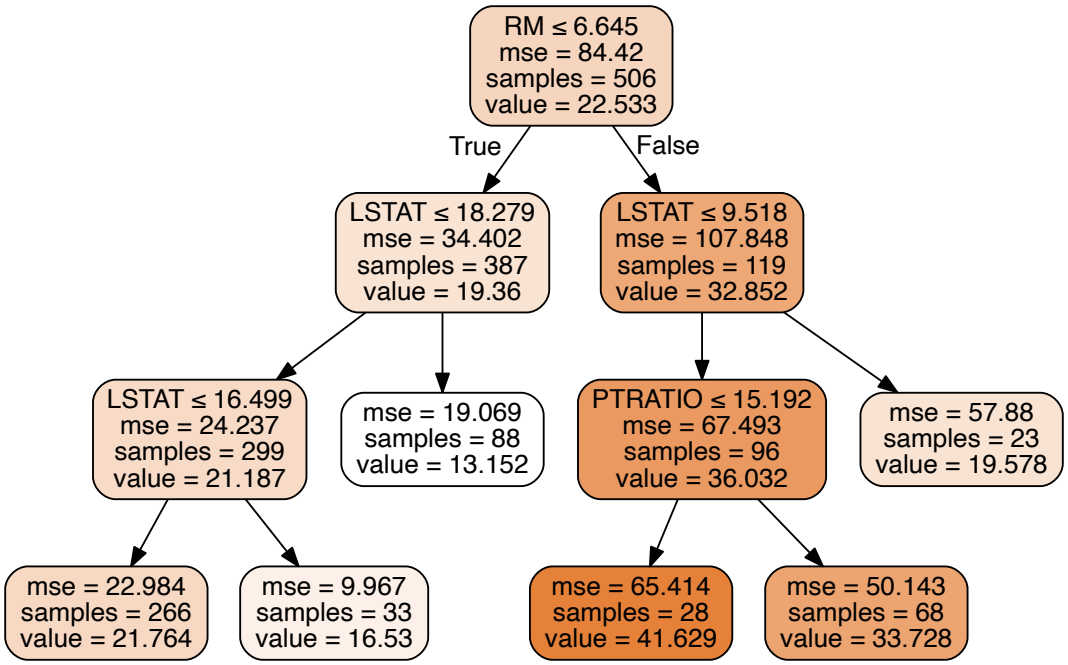


Figure 1.6 – Exemple d'arbre de décision.

Le processus de construction d'un arbre se fait par partition successive des données. Plus l'arbre est profond plus le nombre d'observations diminue. Il devient alors difficile d'identifier la meilleure coupure pour séparer les données en deux. Ce problème engendre souvent des arbres trop complexes expliquant très bien les données mais qui se généralisent mal dans le sens où les coupures dépendent des observations plus que de leur distribution.

En d'autres termes cela signifie que l'algorithme a également appris le bruit. On appelle ce phénomène le *sur-apprentissage*. Pour palier ce défaut, les statisticiens ajoutent une étape de post-processing appelé *élagage*. L'idée consiste à transformer certains nœuds en feuilles et donc à supprimer les sous-arbres rattachés à ces nœuds.

La construction d'un arbre binaire optimal est un problème NP-complet [34]. Néanmoins, plusieurs méthodes d'optimisation ont été développées ces dernières années, notamment la méthode *MIO* [5] dont les résultats pratique sont les plus performants.

1.2.2 Les ensembles de règles

Une règle est une assertion du type *If-Then* de la forme

$$\begin{aligned} \text{IF} \quad & (\mathbf{X}[1] \in c_1) \text{ And } (\mathbf{X}[2] \in c_2) \text{ And } \dots \text{ And } (\mathbf{X}[d] \in c_d) \\ \text{THEN} \quad & g_n(\mathbf{X}) = p \end{aligned} \tag{1.18}$$

avec $\mathbf{X}[j]$ la $j^{\text{ème}}$ coordonnée de $\mathbf{X}$ et $c_j \subseteq \mathbb{R}$ un intervalle.

Le nombre de $c_i \neq \mathbb{R}$ est appelé la *longueur* d'une règle. La partie *If* de la règle, appelée *condition de la règle*, est composée de la conjonction de d tests vérifiant si une

variable explicative (une coordonnée de $\mathbf{X}$) vérifie une certaine condition. La partie *Then* de la règle, appelée *conclusion*, est la prédiction de la valeur de la variable d'intérêt Y quand la règle est *active*, c'est-à-dire lorsque sa condition est satisfaite.

Remarque 3. Étant donné que chaque c_i est un intervalle de $\mathbb{R}$, l'ensemble $\mathbf{r} = \prod_{i=1}^{d} c_i$ forme un hyperrectangle de $\mathbb{R}^d$. De plus, une règle étant entièrement déterminée par sa condition, l'ensemble $\mathbf{r}$ est aussi appelé règle par abus de langage. On définit alors la fonction qui donne la longueur d'une règle $\mathbf{r}$, notée ℓ par : $\ell(\mathbf{r}) = d - \#\{1 \leq j \leq d; c_j = \mathbb{R}\}$.

Les règles sont des objets simples à comprendre et permettent de prendre une décision simple à interpréter tant que la longueur $\ell(\mathbf{r})$ des règles reste petite. Chaque décision issue de la prédiction de Y peut ainsi être expliquée par un ensemble de propriétés vérifiées par les données $\mathbf{X}$. Le tableau 1.1 représente un ensemble de règles de longueur maximale 3 décrivant également les données *Boston housing*. On retrouve le *If* dans la colonne *Conditions* et le *Then* dans la colonne *Value*. La colonne *Samples* est le nombre de points dans la règle. Il est important de noter qu'il y a des règles dont les conditions s'intersectent et peuvent donc s'activer en même temps (ex: $R2$ et $R3$ pour $LSTAT \in [7.76, 7.87]$). La gestion de ces conflits possibles est un enjeux important.

Rule	Conditions	Samples	Value
$R1$	$LSTAT \in [9.62, 37.97]$	293	17.74
$R2$	$LSTAT \in [1.73, 7.87]$	157	32.31
$R3$	$RM \in [3.86, 6.10]$ $LSTAT \in [7.76, 37.97]$ $DIS \in [1.20, 12.13]$	187	17.62
$R4$	$RM \in [6.42, 8.78]$ $LSTAT \in [1.73, 14.40]$ $DIS \in [1.20, 12.13]$	167	31.88
$R5$	$RM \in [6.09, 8.78]$ $LSTAT \in [1.73, 9.98]$ $CRIM \in [0.05, 88.98]$	136	32.12
$R6$	$LSTAT \in [5.06, 37.97]$ $PTRATIO \in [13.90, 22.0]$ $INDUS \in [15.01, 27.74]$	182	16.34

Table 1.1 – Exemple d'ensemble de règle.

1.2.3 Arbres vs Ensemble de règles

Un arbre peut formellement être écrit sous la forme d'un ensemble de règles, mais la réciproque est fausse. En effet un ensemble de règles n'a pas forcément de noeud racine et ne forme par non plus une partition. Il a été montré, dans [56], que les modèles fondés sur des arbres, même élagués, sont souvent plus complexes et plus difficiles à interpréter que des modèles à base d'ensembles de règles.

De plus les arbres reposent sur une partition récursive de l'espace des variables explicatives. De ce fait il n'y a aucune intersection entre les règles. Cette contrainte entraîne souvent que des sous-arbres identiques sont appris à différents endroits de l'arbre. Ce problème appelé problème de la *réplication des sous-arbres* a été traité dans [3]. Un autre inconvénient est qu'une telle procédure est contraignante dans le processus de recherche. En effet de tels algorithmes ne peuvent pas être parallélisés puisque chaque étape dépend de la précédente.

A l'inverse les ensembles de règles ne font pas nécessairement de partition ce qui permet d'éviter ce problème de réplication. Malheureusement cela engendre une autre difficulté

qui est de faire une prédiction lorsque plusieurs règles sont actives. De plus, en l'absence d'heuristique de parcours des données les algorithmes de génération de règles peuvent être très consommateurs de ressources. En effet il n'est absolument pas envisageable de générer toutes les règles possibles à cause du volume que cela représente.

1.2.4 Les algorithmes à ensemble

Pour éviter les problèmes liés à la contrainte de partition, évoqué ci-dessus, on peut également considérer des algorithmes à ensemble. Nous présentons ici quatre algorithmes à ensemble parmi les plus populaires.

Développée récemment, une idée est de considérer tous les nœuds (intérieurs et terminaux) d'un ensemble d'arbres comme des règles. Chaque prédiction sera alors une somme pondérée de plusieurs règles. L'algorithme *RuleFit* [20] génère un grand nombre de règles à partir d'arbres. Les activations de ces règles sont ensuite considérées comme des variables binaires valant 1 si la règle est active, 0 sinon. Ces nouvelles variables sont ajoutées à l'ensemble de variables explicatives pour une régression *LASSO* [66]. Les règles avec un coefficient non nul forment un ensemble de règles sélectionnées auquel s'ajoute les relations linéaires. Le Tableau 1.2 représente les six règles les plus importantes décrivant les données *Boston housing*[3] issues de *RuleFit*. On retrouve les même colonnes que le tableau 1.1 sauf que la colonne Value correspond aux coefficients issues de la régression LASSO et non pas au *Then* de la règle.

Toutefois, le nombre de règles dans le modèle final issu de *RuleFit* peut être très élevé. De plus, le signe du coefficient associé par la régression à la règle peut être différent de celui de la moyenne des Y dans cette règle, ce qui peut biaiser l'interprétabilité.

Rule	Conditions	Samples	Value
Linear	$LSTAT$	/	-0.40
Linear	AGE	/	-0.036
$R1$	$DIS < 1.40$ $PTRATIO > 17.9$ $LSTAT < 10.5$	5	10.1
$R2$	$RM > 6.62$ $NOX < 0.67$	116	2.26
$R3$	$RM < 7.45$ $DIS > 1.37$ $TAX > 219.0$	445	-2.27
$R4$	$DIS > 1.30$ $PTRATIO > 19.4$	207	-1.40

Table 1.2 – Exemple de règles les plus importantes de RuleFit

L'algorithme *Ender* [12] génère un nombre de règles fixé à l'avance. Elles sont générées de façon récursives telles que chaque nouvelle règle ajoutée soit celle dont l'ajout à l'ensemble de règles minimise la perte empirique (1.6). Cet algorithme est un algorithme d'ensemble où chaque règle est considérée comme un modèle très simple car générée de petite longueur. L'ensemble de ces modèles (règles) est ensuite agrégé en une moyenne pondérée. Le problème de cet algorithme est qu'il génère également un grand nombre de règles (plus de 1000 pour décrire les données *Boston housing*)

L'algorithme *Node harvest* de [49], développé dans le cadre de la classification, considère également tous les nœuds (intérieurs et terminaux) d'un ensemble d'arbres comme

3. On reprend ici le tableau [20, Table 2].

des règles. Chaque règle se voit associer un poids positif ou nul ajusté pour minimiser la perte empirique (1.6). La résolution de ce problème d'optimisation convexe permet d'obtenir un petit ensemble de règles avec des poids non nuls.

Plus récemment, [4] propose l'algorithme *SIRUS* qui génère un grand nombre d'arbres de faibles profondeurs. Il extrait alors les chemins (arcs plus nœuds) dont la probabilité empirique d'apparition est supérieure à un certain seuil. Afin d'assurer la stabilité de l'algorithme, les variables explicatives sont discrétisées. Cet algorithme fournit un petit ensemble de règles très stable par rapport à l'échantillon et garde un bon pouvoir prédictif. Il ne s'applique pour le moment que dans le cadre de la classification.

La section suivante présente les méthodes usuelles pour apprendre une règle et construire un ensemble de règles. Tout au long de cette section nous dressons un état de l'art de ces algorithmes et concluons en présentant les problèmes liés à l'utilisation de tels algorithmes pour l'analyse de la régression.

1.3 L'apprentissage des ensembles de règles

L'apprentissage de règles est une composante basique du machine learning et du data mining, il est étudié depuis les années 1960. Il a été le premier type d'algorithme de machine learning utilisé dans un processus industriel. On peut citer *GASOIL* [63], *BMT* [31] ou encore [40] utilisé dans le processus de contrôle. Pour une introduction détaillée sur l'apprentissage de règles nous renvoyons les lecteurs à l'ouvrage [22].

L'apprentissage de règles est le domaine de ML qui se concentre sur l'extraction de règles à partir d'un jeu de données. Les deux domaines principaux sont l'apprentissage de règles d'associations et l'apprentissage de règles prédictives. Le premier cherche à trouver des informations ou des modèles en fonction du jeu de données donné. L'algorithme le plus emblématique de recherche de règles d'associations est *APriori* [1]. Pour une revue des algorithmes d'apprentissage de règles les plus connus pour la description nous renvoyons les lecteurs à [80].

L'apprentissage de règles prédictives vise une description du modèle, afin de faire des prédictions à partir de nouvelles observations. C'est ce domaine que nous traitons dans cette thèse. Pour une revue des algorithmes d'apprentissage de règles pour la prédiction nous renvoyons le lecteur à l'article [23].

1.3.1 Apprendre une règle

Il existe deux méthodes de recherche pour apprendre une règle : la méthode *général à spécifique* ou *descendante* (top-down) et la méthode *spécifique à général* ou *ascendante* (bottom-up).

La méthode descendante commence par une règle générique et tente de la spécifier de façon récursive. L'algorithme *CART* [8] est établie sur cette méthode. Par exemple, l'algorithme 1 part d'une règle toujours vérifiée qui prédit la moyenne de la variable d'intérêt sur un échantillon (ligne 1) et rétrécie les bornes de l'intervalle pour ne garder que les valeurs de Y les plus grandes. Par souci de simplicité on suppose que $d = 1$ et $Y \in [-1, 1]$ dans l'algorithme 1.

Inversement, la méthode ascendante part d'une règle spécifique et tente de la généraliser de façon récursive. La méthode d'élagage [8] est une méthode ascendante. Par exemple, l'algorithme 2 présenté ci-dessous, part d'une règle toujours très spécifique (définie sur un point de l'échantillon ligne 3) qui prédit le maximum de la variable d'intérêt sur un échantillon et élargit les bornes de l'intervalle pour ne garder que des Y positifs. Par souci de simplicité on suppose que $d = 1$ et $Y \in [-1, 1]$.

Algorithm 1: Exemple d'algorithme top-down naïf

Input: $D_n = (x_i, y_i)_{i=1,\dots,n}$ un échantillon;
Output: $\mathbf{r}$ un intervale de $\mathbb{R}$ définissant une règle;

$\mathbf{r} = \mathbb{R}$;
$bmin = min(\{x_1, \dots, x_n\})$;
$bmax = \arg\max_{b \in \mathbf{r}} \sum_{i=1}^{n} y_i \mathbf{1}_{x_i \in [bmin, b]}$;
$bmin = \arg\min_{l \in \mathbf{r}} \sum_{i=1}^{n} y_i \mathbf{1}_{x_i \in [l, bmax]}$;
$\mathbf{r} = [bmin, bmax]$;
return $\mathbf{r}$;

Algorithm 2: Exemple d'algorithme bottom-up naïf

Input: $D_n = (x_i, y_i)_{i \leq 0}$ un échantillon;
Output: $\mathbf{r}$ un interval de $\mathbb{R}$ définissant une règle;

$i_{max} = \arg\max_{i \in \{1,\dots,n\}} y_i$;
$bmin = bmax = x_{i_{max}}$;
$\mathbf{r} = [bmin, bmax]$;
$bmax = \arg\max_{b \in \{x_1,\dots,x_n\}} \sum_{i=1}^{n} y_i \mathbf{1}_{x_i \in [bmin, b]}$;
$bmin = \arg\min_{l \in \{x_1,\dots,x_n\}} \sum_{i=1}^{n} y_i \mathbf{1}_{x_i \in [bmin, b]}$;
$\mathbf{r} = [bmin, bmax]$;
return $\mathbf{r}$;

On observe qu'en général la première méthode fournit des règles plus générales que la seconde, ce qui la rend plus adaptée à l'apprentissage sur des jeux de données bruités. Elle est également la plus utilisée dans les systèmes pratiques. En revanche la méthode ascendante est plus efficace lorsque le phénomène que l'on veut étudier s'est peu produit dans l'échantillon.

On peut remarquer que ces méthodes de raffinement d'une règle peuvent être très coûteuses en temps de calcul à mesure que n augmente si les variables explicatives sont continues. Comment trouve-t-on des bornes pour définir une condition dans (1.18) ? La méthode usuelle est de discrétiser les données. Soit par une méthode qualitative en choisissant des groupes usuels pour les professionnels du secteur (ex : $poids \leq 50$, $50 < poids \leq 60$, etc.), soit en utilisant des méthodes quantitatives comme dans [18], [16], [64] ou encore l'algorithme BRL (Bayesian Rule Lists) [41] et *SIRUS* [4] qui appliquent une discrétisation en utilisant les quantiles empiriques.

1.3.2 Apprendre un ensemble de règles

Pour définir un modèle à partir d'un apprentissage de règles, il existe deux familles de stratégies algorithmiques : *Divide-and-Conquer,* qui est une partition récursive de l'ensemble de données en optimisant un critère sur tous les nœuds successeurs [55] et *Separate-and-Conquer* qui cherche les meilleures règles qui expliquent une partie du jeu de données puis supprime les points couverts et conquiert récursivement le jeu de données restant en apprenant plus de règles [21] .

La première stratégie ne conçoit pas de règles qui se chevauchent, évitant ainsi les conflits de prédictions lorsque deux ou plusieurs conditions de règles sont vérifiées. Dans ce cas, les règles peuvent être ordonnées, formant ainsi une *liste de décision* introduit dans [59]. Les listes de décisions sont un ensemble de règles qui peuvent s'écrire sous la forme *If-ThenIf-. . . -Then.*

Dans la deuxième stratégie, on obtient un ensemble de règles et il faut gérer les

chevauchements. Soit en transformant l'ensemble des règles en une liste de décisions, soit en utilisant l'une des méthodes suivantes : *Frequency-Based*, *Naive Bayesian* et *Double Induction*, pour une description de ces méthodes nous renvoyons le lecteur à [42]. L'autre problème de cette stratégie est que la couverture n'est pas assurée. Cela signifie que pour les nouvelles données, il peut n'y avoir aucune règle active (c.f la remarque 2). Une solution courante consiste à prédire la classe majoritaire ou la moyenne, selon le scénario proposé par [71]. Une autre méthode proposée par [17] consiste à appliquer des règles qui garantissent un recouvrement.

1.3.3 Les problèmes liés à la régression

L'analyse de la régression à partir de règles n'a pas été particulièrement bien étudiée. Contrairement à la classification, il existe peu de solutions commerciales (on peut néanmoins citer Cubist 5 [4]). En fait, l'apprentissage des règles de régression pose de nombreux problèmes en plus de ceux liés à l'apprentissage des règles.

Premièrement, comme mentionné précédemment dans un scénario de régression, la variable d'intérêt prend ses valeurs dans $\mathbb{R}$ et l'objectif d'un algorithme prédictif est de prédire la valeur de la variable à partir de nouvelles données. Mais, on peut remarquer que la conclusion d'une règle, de la forme (1.18), est une valeur numérique constante, ce qui induit un biais lorsque la fonction de régression n'est pas constante.

Le deuxième problème concerne les critères d'arrêt. En effet, dans un problème de régression, le critère est la réduction de la variance autour de la prédiction de la règle. Les meilleures règles sont donc celles qui sont réduites à un point. Une solution courante consiste à imposer une limite minimale au taux de couverture des règles, c'est-à-dire au ratio du nombre de points dans la règle sur n. Par exemple l'algorithme *FORS* [37] utilise le principe de *minimal description length* (MDL) pour élaborer un élagage de règles. Ce qui peut s'interpréter comme une application du principe du rasoir d'Ockham *"Pluralitas non est ponenda sine necessitate"* [5].

La technique la plus populaire pour l'apprentissage d'une règle de régression consiste à utiliser un algorithme d'arbre de décisions où les arbres appris sont transformés en listes de décision [59]. Par exemple, l'algorithme *M5 Rules* [33], propose d'utiliser des arbres de façon récursive sur l'ensemble de données. Tout d'abord, il extrait une seule règle d'un arbre, puis supprime toutes les données couvertes par cette règle, l'algorithme réitère le processus sur les données restantes jusqu'à ce que touts les points soient couverts. On a donc à faire à une stratégie Separate-and-Conquer avec une méthode descendante.

Une autre méthode répandue consiste à convertir le problème de régression en problème de classification et ainsi utiliser un algorithme d'apprentissage des règles pour un problème de classification. La première idée est de discrétiser la variable d'intérêt comme dans [75] et [68]. Mais l'ensemble de règles d'apprentissage dépend trop du nombre d'éléments choisi pour la discrétisation. Un moyen d'éviter ce problème est présenté dans [35], [36]. Dans [39] le problème de régression se transforme en un problème de classification binaire dont l'objectif est de prédire si la valeur de sortie est supérieure ou inférieure à un seuil choisi.

Enfin, une idée originale, présentée dans [67], consiste à construire un arbre de régression où chaque nœud terminal prédit un modèle linéaire (ajusté sur les données de l'échantillon dans la feuille) à la place d'une valeur constante.

4. http://www.rulequest.com/cubist-info.html
5. "Les multiples ne doivent pas êtres utilisés sans nécessité."

1.3.4 Estimateur à base de règles

L'objectif des algorithmes d'apprentissage de règles prédictives est de générer un ensemble de règles $\mathcal{P}_n$. Cet ensemble est utilisé pour décrire le modèle appris et permet de faire des prédictions en fonction de nouvelles observations. D'après (1.7), si $\mathcal{P}_n$ forme une partition de $\mathbb{R}^d$, c'est-à-dire toutes les règles $\mathbf{r}$ sont disjointes et $\bigcup_{\mathbf{r}\in\mathcal{P}_n} \mathbf{r} = \mathbb{R}^d$ alors l'estimateur de la fonction de régression s'écrit sous la forme

$$g_n(\mathbf{x}) = \sum_{\mathbf{r}\in\mathcal{P}_n} \frac{\sum_{i=1}^n Y_i \mathbf{1}_{\mathbf{X}_i\in\mathbf{r}}}{\sum_{i=1}^n \mathbf{1}_{\mathbf{X}_i\in\mathbf{r}}} \mathbf{1}_{\mathbf{x}\in\mathbf{r}}\,, \qquad \mathbf{x}\in\mathbb{R}^d. \tag{1.19}$$

En effet, dans ce cas, la classe G_n dans (1.7) est l'ensemble des fonctions constantes par morceaux sur les partitions P_n :

$$G_n = \mathcal{G}_c \circ \Pi_n := \left\{ g : \mathbb{R}^d \to \mathbb{R} : g = \sum_{\mathbf{r}\in\mathcal{P}_n} f_{\mathbf{r}} \mathbf{1}_{\mathbf{r}}, \mathcal{P}_n \in \Pi_n, f_{\mathbf{r}} \in \mathcal{G}_c \right\},$$

et la fonction constante qui minimise la perte empirique sur un ensemble $\mathbf{r}$ est l'espérance conditionnelle empirique de Y sachant $\mathbf{X}\in\mathbf{r}$.

1.4 Contributions

L'objectif de cette thèse est de présenter un algorithme interprétable, *RICE* pour *Rule Induction Covering Estimator*, qui génère et sélectionne un ensemble de règles pour construire un estimateur consistant de la fonction de régression.

Dans un premier temps, nous avons introduit un *indice d'interprétabilité* pour être en mesure de comparer les algorithmes dits interprétables (définition 1.4.1). Puis, partant du principe que moins il y a de règles, meilleure sera l'interprétabilité, nous avons mis en place un moyen efficace en terme de complexité de définir un estimateur à partir d'un recouvrement (proposition 1.4.1). Nous avons ainsi introduit une famille d'algorithmes dit à *recouvrement dépendant des données*. Enfin, nous avons prouvé qu'en générant un recouvrement composé de règles définies comme *signifiantes* et *insignifiantes* (définition 1.4.3), alors un algorithme à recouvrement dépendant des données génère un estimateur consistant de la fonction de régression (théorème 1.4.1).

Ce résultat, résumé dans la Proposition 1.4.3, repose sur les propriétés des classes $\mathbb{Q}$-Donsker et sur la notion de probabilité extérieure. La preuve utilise les taux de convergence des estimateurs de l'espérance et de la variance conditionnelles. C'est le principal résultat technique de cette thèse.

1.4.1 Mesure de l'interprétabilité

RICE a été développé en se fondant sur le triptyque *simplicité, stabilité et prédictivité* présenté dans [78]. La notion de prédictivité peut être facilement mesurée en calculant la perte (1.2) sur simulation et celle de stabilité a été traitée dans [77] et peut être évaluée en mesurant le taux de rotations des règles en fonction de l'échantillon comme dans [4]. Dans [78] la simplicité fait référence à la complexité de l'algorithme utilisé. Mais nous proposons plutôt de mesurer la simplicité du modèle généré. A notre connaissance, il n'existe actuellement pas de moyen quantitatif d'évaluer la simplicité d'un modèle à base de règles. Pour être en mesure de comparer notre algorithme à ceux existants, nous introduisons l'indice d'interprétabilité qui permet de comparer n'importe quel type de modèle à base de règles.

L'interprétabilité pour un estimateur à base de règles est atteint lorsque la longueur k de chaque règle est petite. Considérant qu'un estimateur avec une règle de longueur k est aussi interprétable que celui avec les k règles correspondantes de longueur 1, la possibilité d'interprétation obtient naturellement la propriété d'additivité. Nous sommes alors en mesure de définir l'interprétabilité d'un modèle fondé sur un ensemble de règles $\mathcal{P}$ comme suit :

Définition 1.4.1. *L'indice d'interprétabilité d'un modèle fondé sur un ensemble de règles $\mathcal{P}$ est défini par*

$$Int(g) := \sum_{\mathbf{r}\in\mathcal{P}} \ell(\mathbf{r}). \tag{1.20}$$

1.4.2 Introduction d'un recouvrement

L'idée usuelle d'imposer que l'union des conditions des règles sélectionnées forme une partition est une condition trop contraignante. En effet, les règles deviennent inter-dépendantes et plus le nombre de règles sélectionnées augmente plus l'espace de recherche diminue. Nous avons donc mis en place une méthode permettant de travailler avec des règles qui se chevauchent en imposant que l'union des conditions des règles sélectionnées forme un recouvrement. En l'absence de cette contrainte le modèle généré n'explique qu'une partie du vrai modèle. Cette façon de procéder rend la recherche de chaque règle indépendante et permet de paralléliser l'algorithme et donc d'explorer plus de possibilité dans le même laps de temps. Afin de pouvoir garantir une prédiction unique le recouvrement est transformé en partition en utilisant le partitioning trick. Ainsi nous obtenons un estimateur de la forme (1.19) à partir d'un ensemble de règles.

Considérer un recouvrement permet d'avoir un ensemble de règles beaucoup plus petit pour décrire le modèle. Et le fait de définir l'estimateur sur la partition engendrée par ce recouvrement permet de garder de bonnes propriétés comme la consistance.

Transformation d'un recouvrement en partition

Formellement, nous définissons une partition générée pour une collection de sous-ensembles $\mathcal{C}$ en utilisant l'ensemble des parties de $\mathcal{C}$, noté $2^{\mathcal{C}}$:

Définition 1.4.2. *Soit $\mathcal{C}$ une collection finie de sous-ensembles de $\mathbb{R}^d$ et soit $\mathbf{c} = \bigcup_{\mathbf{r}\in\mathcal{C}} \mathbf{r}$. On définit la fonction d'activation comme*

$$\varphi_{\mathcal{C}} : \mathbb{R}^d \mapsto 2^{\mathcal{C}}; \qquad \varphi_{\mathcal{C}}(\mathbf{x}) = \{\mathbf{r} \in \mathcal{C} : \mathbf{x} \in \mathbf{r}\}.$$

On définit $\mathcal{P}(\mathcal{C})$, la partition de $\mathbf{c}$ générée à partir de $\mathcal{C}$, comme

$$\mathcal{P}(\mathcal{C}) := \varphi_{\mathcal{C}}^{-1}(Im(\varphi_{\mathcal{C}})),$$

où $Im(\varphi_{\mathcal{C}})$ est l'image de $\varphi_{\mathcal{C}}$.

Nous illustrons la transformation de $\mathcal{P}$ sur un exemple à quatre éléments dans les Figures 1.7 et 1.8.

Remarque 4. Si $\mathcal{C}$ est un recouvrement de $\mathbb{R}^d$, alors $\mathcal{P}(\mathcal{C})$ est une partition de $\mathbb{R}^d$. De plus l'égalité $\mathcal{C} = \mathcal{P}(\mathcal{C})$ est vérifiée si et seulement si $\mathcal{C}$ est composée de sous-ensemble disjoints.

On définit également $\mathcal{P}_{\mathcal{C}}(\mathbf{r})$ pour tout élément $\mathbf{r}$ de $\mathcal{C}$ comme étant l'ensemble des cellules de $\mathcal{P}(\mathcal{C})$ contenues dans $\mathbf{r}$:

$$\mathcal{P}_{\mathcal{C}}(\mathbf{r}) := \{A \in \mathcal{P}(\mathcal{C}) : A \subseteq \mathbf{r}\}.$$

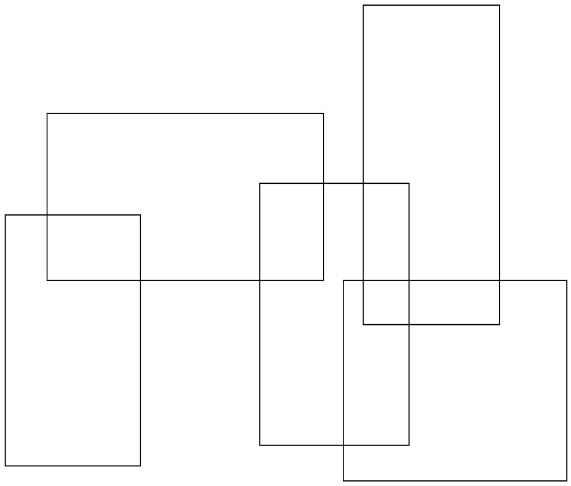


Figure 1.7 – Les cinqs éléments de $\mathcal{C}$.

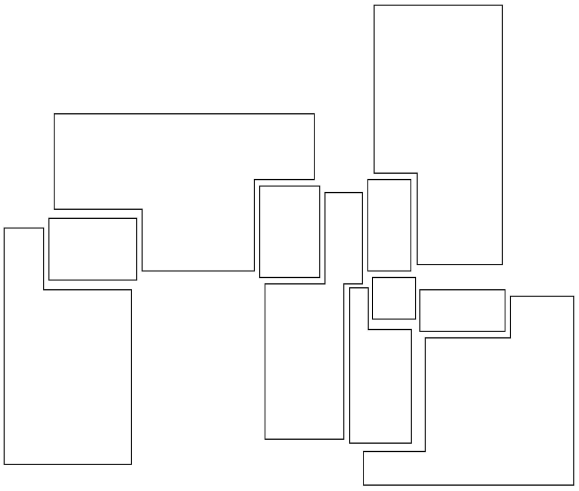


Figure 1.8 – Les onze cellules de $\mathcal{P}(\mathcal{C})$.

On introduit également la redondance maximale (resp. minimale) de $\mathcal{C}$ sur un sous-ensemble $\mathbf{r} \in \mathcal{C}$:

$$M(\mathcal{C}, \mathbf{r}) := \max_{x \in \mathbf{r}} \#\varphi_{\mathcal{C}}(x)$$
$$m(\mathcal{C}, \mathbf{r}) := \min_{x \in \mathbf{r}} \#\varphi_{\mathcal{C}}(x).$$

On utilisera les notations $M(\mathcal{C})$ au lieu de $M(\mathcal{C}, \mathbf{c})$ et au lieu $m(\mathcal{C})$ de $m(\mathcal{C}, \mathbf{c})$ in .

Remarque 5. Si $\mathcal{C}$ est une partition alors pour tout $\mathbf{r} \in \mathcal{C}$, on a $\mathcal{P}_{\mathcal{C}}(\mathbf{r}) = \{\mathbf{r}\}$ et $M(\mathcal{C}, \mathbf{r}) = m(\mathcal{C}, \mathbf{r}) = 1$.

À partir d'un recouvrement $\mathcal{C}$ de $\mathbb{R}^d$ nous pouvons construire un estimateur g_n défini par :

$$g_n(\mathbf{x}) = \sum_{A \in \mathcal{P}(\mathcal{C})} \frac{\sum_{i=1}^n Y_i \mathbf{1}_{\mathbf{X}_i \in A}}{\sum_{i=1}^n \mathbf{1}_{\mathbf{X}_i \in A}} \mathbf{1}_{\mathbf{x} \in A}\,, \qquad \mathbf{x} \in \mathbb{R}^d. \tag{1.21}$$

Par construction, il y a, au plus, un élement de $\mathcal{P}(\mathcal{C})$ contenant $\mathbf{x}$.

Le partitioning trick

Malheureusement, transformer un recouvrement $\mathcal{C}$ en partition $\mathcal{P}(\mathcal{C})$ est une opération complexe. En effet nous avons

$$\#\mathcal{C} \leq \#\mathcal{P}(\mathcal{C}) \leq \sum_{i=1}^{\#\mathcal{C}} \binom{\#\mathcal{C}}{i}.$$

Pour le voir il suffit de remarquer que pour $\mathcal{C} = \{\mathbf{r}_1, \ldots, \mathbf{r}_n\}$ on a pour tout $i \in \{1, \ldots, n\}$

$$\begin{aligned} P(\mathcal{C}) = &\{\mathbf{r}_i \setminus \cup_{j \neq i} \mathbf{r}_j : i \in \{1, \ldots, n\}\} \\ &\bigcup \{\mathbf{r}_i \cap \mathbf{r}_j \setminus \cup_{l \neq i,j} \mathbf{r}_l : i \in \{1, \ldots, n\}, j \in \{i+1, \ldots, n\}\} \\ &\bigcup \{\mathbf{r}_i \cap \mathbf{r}_j \cap \mathbf{r}_l \setminus \cup_{h \neq i, h \neq j, h \neq l} \mathbf{r}_h : i \in \{1, \ldots, n\}, j \in \{i+1, \ldots, n\}, l \in \{j+1, \ldots, n\}\} \\ &\bigcup \{\ldots\} \\ &\bigcup \{\cap_{i=1}^n \mathbf{r}_i\}\,. \end{aligned}$$

L'astuce est de remarquer qu'il n'est pas nécessaire de connaître $\mathcal{P}(\mathcal{C})$ pour calculer $g_n(\mathbf{x})$, mais uniquement la cellule de $\mathcal{P}(\mathcal{C})$ qui contient $\mathbf{x}$.

Pour ce faire, il faut d'abord identifier les règles activées par $\mathbf{x}$, c'est-à-dire telles que $\mathbf{x}$ appartient à l'hyperrectangle défini par leurs conditions (Fig 1.9, en haut à gauche). Et nous calculons l'intersection des règles actives (Fig 1.9, en bas à gauche). Puis nous calculons l'union des règles inactives (Fig 1.9, en haut à droite). Enfin, nous calculons la différence entre l'intersection des règles actives et l'union des règles inactives (Fig 1.9, en bas à droite). Cet sous-ensemble est la cellule de $\mathcal{P}(\mathcal{C})$ qui contient $\mathbf{x}$. Tout le processus est illustré dans le Figure 1.9.

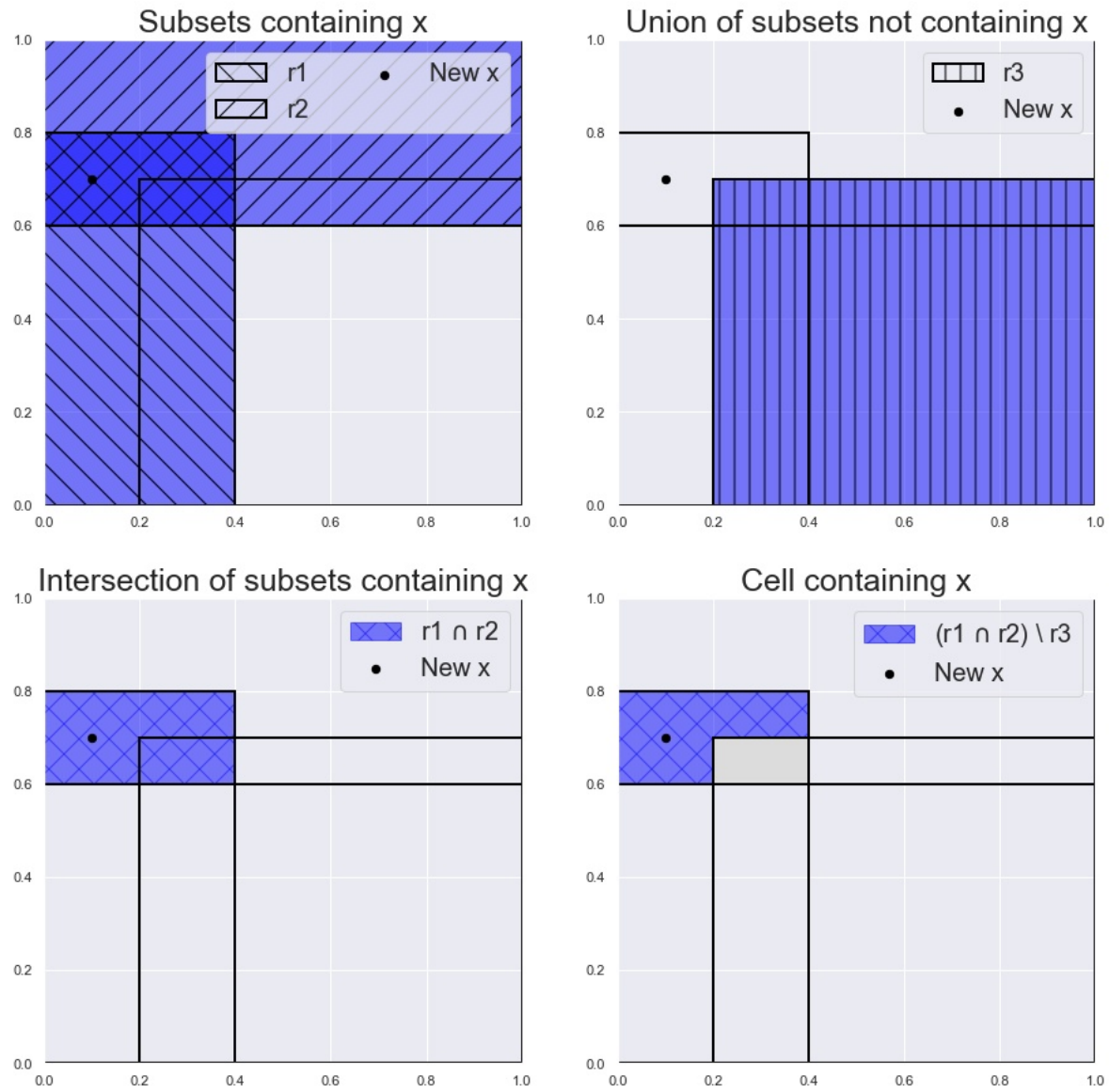


Figure 1.9 – Les différentes étapes du partitioning trick pour un ensemble de trois hyperrectangles $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$ de $[0,1]^2$ et une nouvelle observation $\mathbf{x} = (0.1, 0.7)$. Il est important de noter que la cellule qui contient $\mathbf{x}$, $(\mathbf{r}_1 \cap \mathbf{r}_2) \setminus \mathbf{r}_3$, n'est pas un hyperrectangle et ne peut pas être vue comme une règle au sens de la définition (1.18).

Grâce au partitioning trick, la complexité pour calculer une prédiction à partir d'un recouvrement est linéaire en le nombre de points n et le nombre d'éléments dans le recouvrement. Ce résultat est présenté dans la proposition suivante.

Proposition 1.4.1. *Soit $\mathcal{C}$ un recouvrement de $\mathbb{R}^d$ composé de R sous-ensembles. Alors, la complexité pour calculer $g_n(\mathbf{x})$ pour une nouvelle observation $\mathbf{x} \in \mathbb{R}^d$ est $O(nR)$.*

Preuve. Il suffit de remarquer que $g_n(\mathbf{x})$ peut s'écrire sous la forme:

$$g_n(\mathbf{x}) = \frac{\sum_{j=1}^{n} y_j k(\mathbf{x}, \mathbf{x}_j, \mathcal{C})}{\sum_{j=1}^{n} k(\mathbf{x}, \mathbf{x}_j, \mathcal{C})},$$

avec $k(\mathbf{x}, \mathbf{x}_j, \mathcal{C}) = \prod_{i=1}^{R} \left(\mathbf{1}_{\mathbf{x}\in\mathbf{r}_i}\mathbf{1}_{\mathbf{x}_j\in\mathbf{r}_i} + \mathbf{1}_{\mathbf{x}\notin\mathbf{r}_i}\mathbf{1}_{\mathbf{x}_j\notin\mathbf{r}_i}\right)$. La complexité $O(nR)$ apparait immédiatement, car il suffit de considérer les $n+1$ vecteurs $(\mathbf{1}_{\tilde{\mathbf{x}}\in\mathbf{r}})_{\mathbf{r}\in\mathcal{C}}$ avec $\tilde{\mathbf{x}} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}\}$. □

1.4.3 Contrôle de l'erreur d'approximation sans contrainte sur le diamètre des cellules

L'excès de perte (1.4) peut être décomposé en deux termes grâce au Lemme 10.1 [28] suivant :

Lemme 1.4.1. *Soit G_n une classe de fonctions $g : \mathbb{R}^d \to [-L, L]$ dépendant d'un échantillon $D_n = ((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n))$. Si g_n vérifie (1.7) alors*

$$\ell(g^*, g_n) \leq 2 \sup_{g\in G_n} |\mathcal{L}_n(g) - \mathcal{L}(g)| + \inf_{g\in G_n} \mathbb{E}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))^2\right].$$

Le premier terme est appelé *erreur d'estimation*. Il correspond à la distance entre la perte de l'estimation (1.6) et la perte (1.2) de la meilleure fonction de G_n. Le second terme est appelé *erreur d'approximation*. Il mesure à quel point la fonction de régression g^* peut être bien estimée par une fonction de G_n.

Il est facile de remarquer que ces deux termes jouent des rôles opposés par rapport à la taille de G_n. En effet, plus G_n sera complexe, c'est-à-dire que l'algorithme construira des fonctions variées (linéaires, polynomiales, sinusoïdales, etc.) plus le terme d'erreur d'approximation sera faible. Mais le terme d'erreur d'estimation sera élevé. Il faut donc réaliser un compromis estimation/approximation.

Dans le cas de partitions indépendantes des données, l'erreur d'estimation est contrôlée automatiquement grâce à la mesurabilité de la fonction de régression et la loi forte des grands nombres. Les conditions du théorème 1.1.1 assurent quant à elles un contrôle de l'erreur d'approximation en imposant aux cellules de la partition un diamètre qui tend vers 0 pour $n \to \infty$.

Dans le théorème 1.1.2, pour les partitions dépendantes des données, la condition (1.12) permet de contrôler l'erreur d'estimation en contrôlant la complexité de l'algorithme et la condition (1.13) permet, quant à elle, de contrôler l'erreur d'approximation. En effet, la complexité est mesurée avec le nombre de partitions, $\Delta_n(\Pi)$, que l'algorithme peut générer et donc la complexité de G_n et en assurant un rétrécissement des cellules de la partition on assure une diminution de l'erreur d'approximation dans chacune d'elles.

Pour garantir la consistance et l'interprétabilité au sens de la définition 1.4.1 il faut pouvoir contrôler l'erreur d'approximation sans imposer un rétrécissement des cellules. Pour contrôler cette erreur en considérant des recouvrements nous introduisons la notion de suite acceptable de recouvrements dépendant des données.

On introduit les notations empiriques suivantes :

$$\mathbb{Q}_n(\mathbf{r} \times I) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{\mathbf{X}_i\in\mathbf{r}}\mathbf{1}_{Y_i\in I},$$

$$\mathbb{E}_n[h(Y) \mid \mathbf{X} \in \mathbf{r}] := \frac{\sum_{i=1}^{n} h(Y_i)\mathbf{1}_{\mathbf{X}_i\in\mathbf{r}}}{\sum_{i=1}^{n} \mathbf{1}_{\mathbf{X}_i\in\mathbf{r}}}$$

et

$$\mathbb{V}_n(Y \mid \mathbf{X} \in \mathbf{r}) := \mathbb{E}_n\left[Y^2 \mid \mathbf{X} \in \mathbf{r}\right] - \mathbb{E}_n\left[Y \mid \mathbf{X} \in \mathbf{r}\right]^2.$$

Par souci de simplification d'écriture $\mathbb{Q}_n(\mathbf{r})$ au lieu de $\mathbb{Q}_n(\mathbf{r} \times \mathbb{R})$ de même pour $\mathbb{Q}(\mathbf{r})$ au lieu de $\mathbb{Q}(\mathbf{r} \times \mathbb{R})$ pour tout $\mathbf{r} \subseteq \mathbb{R}^d$.

En considérant la notation classique $x_+ = \max\{x, 0\}$ pour tout $x \in \mathbb{R}$ nous introduisons la définition suivante.

Définition 1.4.3. *Une suite $(\mathcal{C}_n)_{n\geq 1}$ de recouvrements de $\mathbb{R}^d$ dépendants des données est* ***acceptable*** *si et seulement si elle vérifie les deux conditions suivantes :*

1. *la* ***condition de couverture :*** *(H3)*

$$\exists\ \alpha \in [0, 1/2),\ \forall \mathbf{r} \in \mathcal{C}_n,\ \mathbb{Q}_n(\mathbf{r}) > n^{-\alpha} \quad p.s.,$$

pour n suffisamment grand;

2. *la* ***condition de significativité :*** *(H4)*
il existe deux suites $\beta_n \to 0$ et $\varepsilon_n \to 0$ telles que:

$$\mathcal{C}_n = \mathcal{C}_n^s \cup \mathcal{C}_n^i \quad p.s.,$$

pour n suffisamment grand, avec $\mathcal{C}_n^s$ le sous-ensemble significatif défini par

$$\mathcal{C}_n^s := \left\{\mathbf{r} \in \mathcal{C}_n\ :\ \beta_n \left|\mathbb{E}_n[Y|\mathbf{X} \in \mathbf{r}] - \mathbb{E}_n[Y]\right| \geq \sqrt{(\mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) - \sigma^2)_+}\right\}, \quad (1.22)$$

le $\mathcal{C}_n^i$ sous-ensemble insignifiant défini par

$$\mathcal{C}_n^i := \left\{\mathbf{r} \in \mathcal{C}_n \setminus \mathcal{C}_n^s : \varepsilon_n \geq \sqrt{(\mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) - \sigma^2)_+}\right\}, \quad (1.23)$$

et leurs redondances vérifient

$$\frac{M(\mathcal{C}_n^s)}{m(\mathcal{C}_n^s)} = o_{\mathbb{P}}(\beta_n^{-2} \wedge n^{1/2-\alpha})$$

et

$$\frac{M(\mathcal{C}_n^i)}{m(\mathcal{C}_n^i)} = o_{\mathbb{P}}(\varepsilon_n^{-2} \wedge n^{1/2-\alpha}) \quad ,$$

avec $M(\mathcal{C}) := \max_{\mathbf{x}\in\mathbf{c}} \#\{\mathbf{r} \in \mathcal{C} : \mathbf{x} \in \mathbf{r}\}$, $m(\mathcal{C}) := \min_{\mathbf{x}\in\mathbf{c}} \#\{\mathbf{r} \in \mathcal{C} : \mathbf{x} \in \mathbf{r}\}$ et $\mathbf{c} = \bigcup_{\mathbf{r}\in\mathcal{C}} \mathbf{r}$.

Les éléments de $\mathcal{C}_n^s$ sont appelés éléments significatifs *et ceux de $\mathcal{C}_n^i$,* éléments insignifiants.

La condition de couverture (H3) garantit la qualité des estimateurs de l'espérance et de la variance, conditionnellement à $\mathbf{X} \in \mathbf{r}$. En effet, il faut un nombre minimum de points pour être en mesure de garantir un bon taux de convergence. La condition de significativité (1.22) assure, d'une certaine façon, que la *variance intra-groupe* des éléments signifiants d'un recouvrement est contrôlée par la *variance inter-groupe*. La condition d'insignifiance (1.23) garantit que la variance conditionnelle d'un élément insignifiant tend vers la variance du bruit.

Remarque 6. Il est intéressant de noter que la vérification des conditions (H3), (1.22) et (1.23) pour un élément du recouvrement ne nécessite pas de connaitre les autres éléments de ce recouvrement. Par conséquence, la recherche de tels éléments dans une base de données D_n, peut être aisément parallélisée.

1.4.4 Taux de convergence des estimateurs conditionnels

La définition d'une suite acceptable de recouvrement est établie à partir de l'utilisation d'estimateurs *plug-in* des espérances et des variances conditionnelles. Mais, à notre connaissance, les seuls résultats théoriques sur le taux de convergence pour de tels estimateurs sont ceux issus des travaux de [25]. En s'inspirant de ces résultats, nous en proposons ici une version plus adaptée à notre cas qui repose sur les notions de processus empiriques, de classes $\mathbb{Q}$-Donsker et de probabilité extérieure, $\mathbb{P}^*$, rappelées dans la Section 1.5.1.

Ce résultat repose sur la notion d'application *tendue* et *asymptotiquement tendue* dont nous rappelons les définitions [70, Chapitre 18].

Définition 1.4.4. *Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace de probabilité*

1. *Une application aléatoire M définie sur Ω et à valeur dans un espace métrique $(\mathbb{D}, d)$ est dite tendue si et seulement si $\forall \epsilon > 0, \exists K \subset \mathbb{D}$ compact tel que*

$$\mathbb{P}(M \notin K) < \varepsilon. \tag{1.24}$$

2. *Une suite d'applications aléatoires $(M_n)_{n\in\mathbb{N}}$ définie sur Ω et à valeur dans un espace métrique $(\mathbb{D}, d)$ est dite asymptotiquement tendue si et seulement si $\forall \varepsilon > 0, \exists K \subset \mathbb{D}$ compact tel que $\forall \delta > 0$*

$$\limsup_{n\to\infty} \mathbb{P}^*(M_n \notin K^\delta) < \varepsilon, \tag{1.25}$$

avec $K^\delta = \{y \in \mathbb{D} : d(y, K) < \delta\}$ un δ-voisisnage de K pour tout $K \subset \mathbb{D}$ et $\delta > 0$.

Remarque 7. Si $\mathbb{D} = \mathbb{R}$, alors (M_n) est asymptotiquement tendu si et seulement si

$$\forall \varepsilon > 0, \exists M > 0 \ \limsup_{n\to\infty} \mathbb{P}^*(|M_n| > M) < \varepsilon.$$

On note alors $M_n = O_{\mathbb{P}^*}(1)$.

Nous rappelons le résultat suivant.

Proposition 1.4.2. *Soit $\mathcal{F}$ une classe $\mathbb{Q}$-Donsker de fonctions alors $\sqrt{n}\,\|\mathbb{Q}_n - \mathbb{Q}\|_{\mathcal{F}}$ est asymptotiquement tendu, ce qu'on note :*

$$\|\mathbb{Q}_n - \mathbb{Q}\|_{\mathcal{F}} \in O_{\mathbb{P}^*}(n^{-1/2}),$$

en considérant la norme sup, $\|v_n\|_{\mathcal{F}} = \sup_{f\in\mathcal{F}} |v_n(f)|$.

Remarque 8. Si $\mathcal{C}$ est une classe $\mathbb{Q}$-Donsker alors :

$$\|\mathbb{Q}_n - \mathbb{Q}\|_{\mathcal{C}} \in O_{\mathbb{P}^*}(n^{-1/2}),$$

en considérant la norme sup, $\|v_n\|_{\mathcal{C}} = \sup_{A\in\mathcal{C}} |v_n(A)|$.

Taux de convergence de l'estimateur de l'espérance conditionnelle

On rappelle qu'on observe un échantillon $D_n = ((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n))$ où les couples $(\mathbf{X}_i, Y_i)$ sont indépendants et identiquement distribués de loi $\mathbb{Q}$ et à valeur dans $\mathcal{S} \subseteq \mathbb{R}^{d+1}$. On munit $\mathcal{S}$ d'une algèbre borélienne $\mathcal{B}_{\mathcal{S}}$. Le résultat suivant est inspiré de [25, Proposition 3.2].

Proposition 1.4.3. *Soient $\mathcal{B} \subseteq \mathcal{B}_{\mathcal{S}}$ et $\mathcal{F}_{\mathcal{B}} := \{f\mathbf{1}_A : f \in \mathcal{F}, A \in \mathcal{B}\}$ où $\mathcal{F}$ est une classe de fonctions de $\mathcal{L}^1(\mathbb{Q})$ uniformément bornées. Si $\mathcal{B}$ et $\mathcal{F}_{\mathcal{B}}$ sont des classes $\mathbb{Q}$-Donsker alors pour tout $\alpha \in [0, 1/2)$ et avec $\mathcal{B}_n := \{A \in \mathcal{B}, \mathbb{Q}_n(A) \geq n^{-\alpha}\}$,*

$$\sup_{f\in\mathcal{F}} \sup_{A\in\mathcal{B}_n} \left|\mathbb{E}_n[f \mid A] - \mathbb{E}[f \mid A]\right| = O_{\mathbb{P}^*}(n^{\alpha-1/2}).$$

Preuve. Soit $\varepsilon > 0$. Avant tout, on sait que pour tout $f \in \mathcal{F}$ et $A \in \mathcal{B}_n$, étant donné que $\mathbb{Q}_n(A) > 0$ on a $\mathbb{Q}(A) > 0$ et,

$$\begin{aligned}
&|\mathbb{E}_n[f \mid A] - \mathbb{E}[f \mid A]| \\
&= \left|\frac{\int_A f d\mathbb{Q}_n}{\mathbb{Q}_n(A)} - \frac{\int_A f d\mathbb{Q}}{\mathbb{Q}(A)}\right| \\
&= \left|\frac{\mathbb{Q}(A)\left(\int_A f d\mathbb{Q}_n - \int_A f d\mathbb{Q}\right) + (\mathbb{Q}(A) - \mathbb{Q}_n(A))\int_A f d\mathbb{Q}}{\mathbb{Q}(A)\mathbb{Q}_n(A)}\right| \\
&\leq \left|\frac{\int_A f d\mathbb{Q}_n - \int_A f d\mathbb{Q}}{\mathbb{Q}_n(A)}\right| + \left|(\mathbb{Q}(A) - \mathbb{Q}_n(A))\frac{\int_A f d\mathbb{Q}}{\mathbb{Q}(A)\mathbb{Q}_n(A)}\right|. \qquad (1.26)
\end{aligned}$$

Or, d'après la proposition 1.4.2,

$$\sup_{\tilde{f}\in\mathcal{F},\tilde{A}\in B}\left|\int_{\tilde{A}} \tilde{f} d\mathbb{Q}_n - \int_{\tilde{A}} \tilde{f} d\mathbb{Q}\right| = O_{\mathbb{P}^*}(n^{-1/2})$$

et

$$\sup_{\tilde{A}\in\mathcal{B}}\left|\mathbb{Q}_n(\tilde{A}) - \mathbb{Q}(\tilde{A})\right| = O_{\mathbb{P}^*}(n^{-1/2}).$$

Ainsi, d'après la remarque 7, il existe $M > 0$ tel que pour tout n suffisamment grand on a

$$\mathbb{P}^*\left\{\sup_{\tilde{f}\in\mathcal{F},\tilde{A}\in B}\left|\int_{\tilde{A}} \tilde{f} d\mathbb{Q}_n - \int_{\tilde{A}} \tilde{f} d\mathbb{Q}\right| > Mn^{-1/2}\right\} < \frac{\varepsilon}{2}$$

et

$$\mathbb{P}^*\left\{\sup_{\tilde{A}\in\mathcal{B}}\left|\mathbb{Q}_n(\tilde{A}) - \mathbb{Q}(\tilde{A})\right| > Mn^{-1/2}\right\} < \frac{\varepsilon}{2}$$

c'est-à-dire $\mathbb{P}^*(\Omega_n) \geq 1 - \varepsilon$ avec

$$\Omega_n := \left\{\sup_{\tilde{f}\in\mathcal{F},\tilde{A}\in B}\left|\int_{\tilde{A}} \tilde{f} d\mathbb{Q}_n - \int_{\tilde{A}} \tilde{f} d\mathbb{Q}\right| \leq Mn^{-1/2}\right\} \bigcap \left\{\sup_{\tilde{A}\in B}\left|\mathbb{Q}_n(\tilde{A}) - \mathbb{Q}(\tilde{A})\right| \leq Mn^{-1/2}\right\}.$$

Or sachant (1.26) et avec $c := \sup_{f\in\mathcal{F},x\in\mathcal{S}} |f(x)| < \infty$, comme $\mathbb{Q}_n(A) \geq n^{-\alpha}$, pour n suffisamment grand, dans l'événement Ω_n, on a

$$\sup_{f\in\mathcal{F},A\in\mathcal{B}_n}\left|\mathbb{E}_n[f \mid A] - \mathbb{E}[f \mid A]\right| \leq Mn^{\alpha-1/2}(1 + c),$$

puisque $\frac{\int_A f d\mathbb{Q}}{\mathbb{Q}(A)} \leq c$.

Finalement, on a prouvé que $\forall \varepsilon > 0, \exists M > 0, \exists N \in \mathbb{N}^*/\forall n \geq N$,

$$\mathbb{P}^*\left\{\sup_{f\in\mathcal{F},A\in\mathcal{B}_n} |\mathbb{E}_n[f \mid A] - \mathbb{E}[f \mid A]| > Mn^{\alpha-1/2}\right\} < \varepsilon$$

et donc $\forall \varepsilon > 0, \exists M > 0$ tels que

$$\limsup_{n\to\infty} \mathbb{P}^*\left\{\sup_{f\in\mathcal{F},A\in\mathcal{B}_n} |\mathbb{E}_n[f \mid A] - \mathbb{E}[f \mid A]| > Mn^{\alpha-1/2}\right\} \leq \varepsilon$$

qui, avec la remarque 7, prouvent la proposition. □

Taux de convergence de l'estimateur de la variance conditionnelle

A partir de la proposition 1.4.3 il est facile de déduire le corollaire suivant :

Corollaire 1.4.1. *Soit $\mathcal{B} \subseteq \mathcal{B}_{\mathcal{S}}$ une classe $\mathbb{Q}$-Donsker. Si Y est bornée alors pour tout $i \in \mathbb{N}^*$ et pour tout $\alpha \in [0, 1/2)$, avec $\mathcal{B}_n := \{A \in \mathcal{B}, \mathbb{Q}_n(A) \geq n^{-\alpha}\}$ on a*

$$\sup_{A \in \mathcal{B}_n} \left| \mathbb{E}_n \left[Y^i \mid (\mathbf{X}, Y) \in A \right] - \mathbb{E} \left[Y^i \mid (\mathbf{X}, Y) \in A \right] \right| = O_{\mathbb{P}^*}(n^{\alpha - 1/2}), \tag{1.27}$$

et

$$\sup_{A \in \mathcal{B}_n} \left| \mathbb{V}_n \left[Y \mid (\mathbf{X}, Y) \in A \right] - \mathbb{V} \left[Y \mid (\mathbf{X}, Y) \in A \right] \right| = O_{\mathbb{P}^*}(n^{\alpha - 1/2}). \tag{1.28}$$

Preuve de (1.27). On note $L = \operatorname{ess\,sup} Y$ où ess sup désigne la borne supérieure essentielle et soient $i \in \mathbb{N}$, et $f_i \in \mathcal{L}^1(\mathbb{Q})$ définie par

$$\begin{aligned} f_i : \mathbb{R}^d \times [-L, L] &\to [-L^i, L^i] \\ (x, y) &\mapsto y^i. \end{aligned}$$

f_i est bornée et $\{f_i\}$ est une classe de Donsker par application directe du théorème centrale limite. Le résultat est alors une application directe de la Proposition 1.4.3. □

Preuve de (1.28). Cette partie découle directement de (1.27) étant donné que Y est bornée et que

$$\mathbb{V}_n \left[Y \mid (\mathbf{X}, Y) \in A \right] := \mathbb{E}_n \left[Y^2 \mid (\mathbf{X}, Y) \in A \right] - \mathbb{E}_n \left[Y \mid (\mathbf{X}, Y) \in A \right]^2.$$

□

1.4.5 Théorème de consistance à partir d'un recouvrement

Le principal résultat théorique de cette thèse est un théorème de consistance pour un estimateur de la fonction de régression généré par un algorithme à recouvrement dépendant des données. Ce théorème s'appuie sur le partitioning trick (Section 1.4.2), la définition d'une suite acceptable de recouvrements (Section 1.4.3) et le résultat sur le taux de convergence des estimateurs de moments conditionnels (Section 1.4.4).

Théorème 1.4.1. *Supposons (H1) et (H2). Soit $(\mathcal{C}_n)$ une suite acceptable de recouvrements (i.e., vérifiant (H3) et (H4)) telle que :*

$$\mathcal{M}(\Pi_n) \vee \log(\Delta_n(\Pi_n)) = o(n), \tag{H5}$$

avec $\Pi_n := \left\{ \mathcal{P}(\mathcal{C}_n((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))) : (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathbb{R} \right\}$ *pour tout $n \in \mathbb{N}^*$;*

$$\forall n \in \mathbb{N}^*, \{\mathbf{c} \times \mathbb{R}, \mathbf{c} \in \mathcal{C}_n\} \subseteq \mathcal{B}, \tag{H6}$$

avec $\mathcal{B}$ une classe $\mathbb{Q}$-Donsker.

Alors l'estimateur g_n défini par (1.21) *est faiblement consistant:*

$$\ell(g^*, g_n) = o_{\mathbb{P}}(1). \tag{1.29}$$

La condition (H5) est la même que la condition (1.12) du théorème 1.1.2 pour la consistance des estimateurs issus d'un algorithme à partition dépendante des données. Comme expliqué précédemment, cette condition permet de contrôler l'erreur d'estimation

en garantissant que la classe de fonction G_n à laquelle appartient g_n n'est pas trop complexe. L'originalité de ce théorème est dans le contrôle de l'erreur d'approximation par les conditions (H3), (H4) et (H6) au lieu d'une condition sur le diamètre des cellules ((1.9) et (1.13)). On peut également remarquer que dans ce théorème il n'y a aucune condition sur les cellules de la partition induite par le recouvrement sur lesquelles est défini l'estimateur g_n. Cela permet d'avoir une plus grande richesse de partitions comme illustré dans la Figure 1.9.

1.4.6 Algorithmes à recouvrement interprétables

Nous avons présenté dans [46] (chapitre 5) le premier algorithme générant un estimateur de la forme (1.21) à partir d'un recouvrement de règles. L'algorithme *RIPE* a été développé en se fondant sur le partitioning trick (voir Section 1.4.2). Il permet d'obtenir un ensemble de règles de petite taille pour décrire des modèles complexes tout en préservant la stabilité, grâce à une discrétisation des variables explicatives ainsi qu'un bon pouvoir prédictif.

Mais, bien que très interprétable, les propriétés statistiques du recouvrement de règles ne permettaient pas d'obtenir de résultat de consistance sur l'estimateur. L'algorithme *Covering Algorithm* a été développé pour illustrer la notion de règles signifiantes et insignifiante introduite dans le chapitre 2. Cet algorithme utilise un Random Forest comme générateur de règles. Il extrait un petit ensemble de règles signifiantes et insignifiantes en considérant toutes les feuilles et tous les nœuds de tous les arbres. Le problème est que pour assurer un recouvrement l'algorithme doit ajouter une "non-règle" qui s'active si aucune règle n'est active. L'absence de contrôle sur cette "non-règle" et sur la complexité du générateur de règle (au sens de la condition (H5)) ne permet pas d'assurer la consistance sans l'ajout d'hypothèses.

L'algorithme *RICE*, introduit dans le chapitre 3, est, en ce sens, une amélioration de RIPE et de Covering Algorithm. Il reprend l'algorithme RIPE pour la génération de règles et utilise l'algorithme de sélection de Covering Algorithm pour sélectionner une suite acceptable de recouvrements. La fusion de ces deux algorithmes nous permet d'avoir un algorithme interprétable et de nous placer dans les conditions du théorème de consistance 1.4.1, ce qui nous permet de prouver la consistance faible de l'estimateur de la fonction de régression généré par RICE.

De plus, pour chaque algorithme, un package Python a été développé et mis en ligne sur GitHub.

1.5 Rappels et heuristique de preuve

Dans cette section nous rappelons quelques propriétés asymptotiques des processus empiriques. Nous présentons également le Théorème 1.4.1 dans un contexte *idéal* afin de mettre en lumière les grands principes de la preuve.

1.5.1 Rappels sur les processus empiriques

Soit $S_1, S_2, \ldots$ une suite de variables aléatoires i.i.d de loi inconnue $\mathbb{Q}$ et prenant leurs valeurs dans un espace $\mathcal{S} \subseteq \mathbb{R}^d$. On munit $\mathcal{S}$ d'une tribu borélienne $\mathbb{B}_{\mathcal{S}}$. Avec la notation $\mathbb{Q}f := \int f d\mathbb{Q}$, on considère le processus empirique $v_n(f) := n^{1/2}(\mathbb{Q}_n f - \mathbb{Q}f)$, où $\mathbb{Q}_n$ désigne la mesure empirique.

Le résultat théorique sur le taux de convergence des estimateurs d'espérances conditionnelles (Proposition 1.4.2) repose sur les propriétés des classes de fonctions de $\mathcal{L}^2(\mathbb{Q})$ de $\mathcal{S}$ dans $\mathbb{R}$ dites $\mathbb{Q}$-Donsker.

Définition 1.5.1. *On dit que $\mathcal{F}$ est une classe $\mathbb{Q}$-Donsker si et seulement si la suite $\{v_n(f) : f \in \mathcal{F}\}$ converge en loi vers $G_{\mathbb{Q}}$ un $\mathbb{Q}$-pont brownien.*

Remarque 9. Par abus de langage on dit qu'une classe d'ensemble $\mathcal{C}$ est $\mathbb{Q}$-Donsker si la classe des fonctions caractéristiques de ses ensembles, $\mathcal{I}_{\mathcal{C}} := \{\mathbf{1}_A : A \in \mathcal{C}\}$, est une classe $\mathbb{Q}$-Donsker.

À ce jour il n'existe pas de condition nécessaire et suffisante pour qu'une classe $\mathcal{F}$ soit $\mathbb{Q}$-Donsker. Mais une condition suffisante est donnée par [70, Theorem 19.4]. Ce théorème montre que pour qu'une classe de fonctions mesurables $\mathcal{F}$ soit $\mathbb{Q}$-Donsker il suffit que son nombre de crochets $N_{[\,]}(\epsilon, \mathcal{F}, \mathcal{L}^2(\mathbb{Q}))$ ne grossisse pas trop vite vers l'infini lorsque ϵ tend vers 0. Cette vitesse se mesure par l'entropie à crochet définie par

$$J_{[\,]}(\delta, \mathcal{F}, \mathcal{L}^2(\mathbb{Q})) = \int_0^\delta \sqrt{\log N_{[\,]}(\epsilon, \mathcal{F}, \mathcal{L}^2(\mathbb{Q}))} d\epsilon.$$

Théorème 1.5.1. *Toute classe $\mathcal{F}$ de fonctions mesurables vérifiant $J_{[\,]}(1, \mathcal{F}, \mathcal{L}^2(\mathbb{Q})) < \infty$ est une classe $\mathbb{Q}$-Donsker.*

La notion de limite en probabilité pour une suite de variable aléatoire doit être généralisée pour une suite d'applications à valeurs dans des espaces métriques qui ne sont pas des espaces euclidiens (dans ce cas les ensembles bornés et fermés ne sont pas nécessairement compacts) et la mesurabilité n'est pas garantie. Pour contourner la difficulté de mesurabilité, on introduit la notion de probabilité extérieure, notée $\mathbb{P}^*$, et on considère le taux de convergence en $O_{\mathbb{P}^*}$ au lieu de l'usuel $O_{\mathbb{P}}$.

Définition 1.5.2. *Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace de probabilité, alors pour tout $B \subseteq \Omega$, la probabilité extérieure $\mathbb{P}^*(B)$ est définie par*

$$\mathbb{P}^*(B) := \inf \{\mathbb{P}(A) : B \subset A, A \in \mathcal{A}\}.$$

Définition 1.5.3. *On dit qu'une suite $S_1, S_1, \ldots$ est dans $O_{\mathbb{P}^*}(a_n)$ pour une suite de nombre positifs $a_1, a_2, \ldots$ si et seulement si pour tout $\epsilon > 0$ il existe des constantes $M > 0$ et $N \in \mathbb{N}$ telles que pour tout $n > N$ on a*

$$\mathbb{P}^* \left\{ \left| \frac{S_n}{a_n} \right| > M \right\} \leq \epsilon.$$

1.5.2 Heuristique de preuve du théorème 1.4.1

Nous présentons une preuve du théorème 1.4.1 dans le cas particulier d'une partition et dans un cadre où les espérances et variances conditionnelles sont connues.

Théorème 1.5.2. *Soit g_n défini par (1.19) sur un ensemble de règles $\mathcal{P}_n$. Si*

- $\dfrac{\mathcal{M}(\Pi_n)}{n} \vee \dfrac{\log(\Delta_n(\Pi_n))}{n} \to 0 \ (n \to \infty)$,
- *$\mathcal{P}_n$ forme une partition de $\mathbb{R}^d$,*
- *Il existe deux suites $\beta_n \to 0$ et $\varepsilon_n \to 0$ telles que :*

$$\mathcal{P}_n = \mathcal{P}_n^s \cup \mathcal{P}_n^i \quad p.s,$$

avec $\mathcal{P}_n^s$ le sous-ensemble signifiant défini par

$$\mathcal{P}_n^s := \left\{ \mathbf{r} \in \mathcal{P}_n : \beta_n \left| \mathbb{E}[Y | \mathbf{X} \in \mathbf{r}] - \mathbb{E}[Y] \right| \geq \sqrt{(\mathbb{V}(Y | \mathbf{X} \in \mathbf{r}) - \sigma^2)_+} \right\},$$

et $\mathcal{P}_n^i$ le sous-ensemble insignifiant défini par

$$\mathcal{P}_n^i := \left\{ \mathbf{r} \in \mathcal{P}_n \setminus \mathcal{P}_n^s : \varepsilon_n \geq \sqrt{(\mathbb{V}(Y | \mathbf{X} \in \mathbf{r}) - \sigma^2)_+} \right\},$$

alors

$$\ell\left(g^*, g_n\right) \to 0 \ (n \to \infty) \ p.s.$$

Remarque 10. On remarque que dans l'énoncé de ce théorème la condition de couverture (H3) n'est plus nécessaire puisque les conditions de signifiance et d'insignifiance ne portent pas sur des estimateurs des moments. De plus nous obtenons un résultat de consistance forte contrairement au Théorème 1.4.1.

Preuve. D'après le Lemme 1.4.1 un estimateur est consistant si les erreurs d'approximation et d'estimation tendent vers 0. Commençons par prouver que l'erreur d'approximation tend vers 0 en utilisant la seconde condition du Théorème 1.5.2.

Erreur d'approximation : Soit G_n la classe des fonctions mesurables définies sur la partition P_n

$$G_n = \left\{ g : \mathbb{R}^d \to \mathbb{R} : g = \sum_{A \in \mathcal{P}_n} f_A \mathbf{1}_A, f_A \in \mathcal{G}_c \right\},$$

où $\mathcal{G}_c$ est l'ensemble des fonctions constante de $\mathbb{R}^d$ dans $[-L, L]$. Pour tout $g_n \in G_n$ on a

$$\inf_{g \in G_n} \mathbb{E}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))^2\right] \leq \mathbb{E}\left[(g_n(\mathbf{X}) - g^*(\mathbf{X}))^2\right],$$

et on pose

$$\mathbb{W}_n := \mathbb{E}\left[(g_n(\mathbf{X}) - g^*(\mathbf{X}))^2\right].$$

L'idée de la preuve repose sur l'identification de l'erreur d'approximation comme étant une variance intra-groupe. En effet, soit $\tilde{g}_n \in G_n$ définie par

$$\tilde{g}_n(\mathbf{x}) = \mathbb{E}\left[Y | \mathbf{X} \in A(\mathbf{x})\right],$$

pour tout $\mathbf{x} \in \mathbb{R}^d$ où $A(\mathbf{x}) \in \mathcal{P}_n$ désigne la cellule qui contient $\mathbf{x}$. On a alors

$$\mathbb{W}_n := \sum_{A \in \mathcal{P}_n} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X} \in A\right] - g^*(\mathbf{X})\right)^2 \middle| \mathbf{X} \in A\right] \mathbb{P}\left(\mathbf{X} \in A\right),$$

la variance intra-groupe des règles $A \in \mathcal{P}_n$.
On pose

$$\mathbb{W}_n^s := \sum_{A \in \mathcal{P}_n^s} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X} \in A\right] - g^*(\mathbf{X})\right)^2 \middle| \mathbf{X} \in A\right] \mathbb{P}\left(\mathbf{X} \in A\right)$$

$$\text{et } \mathbb{W}_n^i := \sum_{A \in \mathcal{P}_n^i} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X} \in A\right] - g^*(\mathbf{X})\right)^2 \middle| \mathbf{X} \in A\right] \mathbb{P}\left(\mathbf{X} \in A\right).$$

Étant donné que $\mathcal{P}_n = \mathcal{P}_n^s \cup \mathcal{P}_n^i$ et que $\mathcal{P}_n^s \cap \mathcal{P}_n^i = \emptyset$, on a

$$\mathbb{W}_n = \mathbb{W}_n^s + \mathbb{W}_n^i.$$

L'idée est d'utiliser les propriétés des sous-ensembles (in)signifiants pour montrer que $\mathbb{W}_n^s$ et $\mathbb{W}_n^i$ tendent vers 0.

Variance intra-groupe des insignifiantes : Il est facile de montrer que par hypothèse on a

$$\begin{aligned} \mathbb{W}_n^i &\leq \varepsilon_n^2 \\ &\xrightarrow[n \to \infty]{p.s} 0. \end{aligned}$$

Variance intra-groupe des signifiantes : En utilisant la décomposition de variance on a

$$\mathbb{W}_n^s \leq \mathbb{V}(g^*(\mathbf{X})) - \mathbb{B}_n^s, \tag{1.30}$$

avec

$$\mathbb{B}_n^s := \sum_{A \in \mathcal{P}_n^s} \left(\mathbb{E}\left[Y \mid \mathbf{X} \in A\right] - \mathbb{E}\left[Y\right]\right)^2 \mathbb{P}\left(\mathbf{X} \in A\right),$$

la variance inter-groupe des signifiantes. Or, d'après la propriété de signifiance on a que

$$\mathbb{B}_n^s \geq \sum_{A \in \mathcal{P}_n^s} \beta_n^{-2} (\mathbb{V}(Y|\mathbf{X} \in A) - \sigma^2) \mathbb{P}\left(\mathbf{X} \in A\right)$$

De plus, par indépendance de Z et $\mathbf{X}$, on a

$$\begin{aligned}
\mathbb{V}(Y|\mathbf{X} \in A) - \sigma^2 &= \mathbb{E}\left[\left(Y - \mathbb{E}\left[Y \mid \mathbf{X} \in A\right]\right)^2 \mid \mathbf{X} \in A\right] - \sigma^2 \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X} \in A\right] - g^*(\mathbf{X})\right)^2 \mid \mathbf{X} \in A\right].
\end{aligned}$$

On a donc

$$\mathbb{B}_n^s \geq \beta_n^{-2} \sum_{A \in \mathcal{P}_n^s} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X} \in A\right] - g^*(\mathbf{X})\right)^2 \mid \mathbf{X} \in A\right] \mathbb{P}\left(\mathbf{X} \in A\right).$$

On reconnaît $\mathbb{W}_n^s$ dans le terme de droite. Donc, par hypothèse et d'après (1.30), nous pouvons conclure que

$$\begin{aligned}
\mathbb{W}_n^s &\leq \frac{\mathbb{V}(g^*(\mathbf{X}))}{1 + \beta_n^{-2}} \\
&\xrightarrow[n \to \infty]{p.s} 0.
\end{aligned}$$

Erreur d'estimation: Il reste à montrer que l'erreur d'estimation tend vers 0 en utilisant la première condition du Théorème 1.4.1. On rappelle que G_n est l'ensemble des fonctions constantes par morceaux à valeur dans $[-L, L]$ sur les éléments de la partition $\mathcal{P}(D_n)$. En introduisant la notion de famille de partitions Π_n définie par (1.11) on a

$$\sup_{g \in G_n} \left|\mathcal{L}_n\left(g\right) - \mathcal{L}\left(g\right)\right| \leq \sup_{g \in \mathcal{G}_c \circ \Pi_n} \left|\mathcal{L}_n\left(g\right) - \mathcal{L}\left(g\right)\right|,$$

avec

$$\mathcal{G}_c \circ \Pi_n := \left\{ g : \mathbb{R}^d \to \mathbb{R} : g = \sum_{A \in \mathcal{P}_n} f_A \mathbf{1}_A, \mathcal{P}_n \in \Pi_n, f_A \in \mathcal{G}_c \right\}.$$

La suite de la preuve repose sur les mêmes arguments que celle du [28, Théorème 13.1] mais sera détaillée dans le chapitre 2. □

"We think too much and feel too little"
Charlie Chaplin.

Chapter 2

Consistent Regression using Data-Dependent Coverings

2.1 Introduction

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and we consider the following regression setting: $(\mathbf{X}, Y)$ is a couple of random variables in $\mathbb{R}^d \times \mathbb{R}$ of unknown distribution $\mathbb{Q}$ such that

$$Y = g^*(\mathbf{X}) + Z,$$

where $\mathbb{E}[Z] = 0$, $\mathbb{V}(Z) = \sigma^2$ and g^* is a measurable function from $\mathbb{R}^d$ to $\mathbb{R}$.
We make the following common assumptions:

— Z is independent of $\mathbf{X}$ and $\sigma^2 \geq 0$ is known; (H7)

— Y is bounded: $\mathbb{Q}(\mathcal{S}) = 1$ with $\mathcal{S} = \mathbb{R}^d \times [-L, L]$, for some $L > 0$ (unknown). (H8)

Given a sample $\mathbf{D}_n = ((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n))$, we aim at predicting Y conditionally on $\mathbf{X}$. The observations $(\mathbf{X}_i, Y_i)$ are independent and identically distributed (i.i.d.) from the distribution $\mathbb{Q}$. The accuracy of a regression function $g : \mathbb{R}^d \to \mathbb{R}$ is measured by its quadratic risk, defined as

$$\mathcal{L}(g) = \mathbb{E}_{\mathbb{Q}}\left[(g(\mathbf{X}) - Y)^2\right].$$

Thanks to Hypothesis (H7), we have

$$g^*(\mathbf{X}) = \mathbb{E}[Y|\mathbf{X}] = \arg\min_g \mathcal{L}(g) \text{ a.s,} \tag{2.1}$$

where the arg min is taken over the class of all measurable regression functions.

The regression functions generated from the data $\mathbf{D}_n$ by a learning algorithm are called estimators of g^*. We consider a set of regression functions G_n that contains all such estimators. Let $\mathbb{Q}_n$ be the empirical distribution of the sample $\mathbf{D}_n$. We define the empirical risk, the empirical risk minimizer and the minimizer of the risk over G_n as, respectively,

$$\mathcal{L}_n(g) = \frac{1}{n}\sum_{i=1}^{n}(g(\mathbf{X}_i) - Y_i)^2, \; g_n = \arg\min_{g \in G_n} \mathcal{L}_n(g) \text{ and } \tilde{g}_n = \arg\min_{g \in G_n} \mathcal{L}(g). \tag{2.2}$$

The aim of this chapter is to provide interpretable learning algorithms that generate G_n so that the associated empirical risk minimizer g_n is consistent, i.e., g_n converges to g^* as $n \to \infty$. More precisely, we show the weak consistency of the estimator g_n, i.e., its excess of risk

$$\ell(g^*, g_n) = \mathcal{L}(g_n) - \mathcal{L}(g^*) = \mathbb{E}[(g_n(\mathbf{X}) - g^*(\mathbf{X}))^2] = o_{\mathbb{P}}(1).$$

As explained in [43], there are several meanings of *interpretability* depending on the users desiderata and the expected properties of algorithms. In this chapter, we use the definition of model interpretability from [6]: **Interpretability is the degree to which an observer can understand the cause of a decision.**

2.1.1 Rule-based algorithms using partitions and coverings

In this chapter we consider algorithms generating interpretable models that are rule-based, such as *CART* [8], *ID3* [55], *C4.5* [56], *FORS* [37], *M5 Rules* [33]. In these models, the regression function is explained by the realization of a simple condition, an *If-Then* statement of the form:

$$\begin{array}{rl} \text{IF} & (\mathbf{X}[i_1] \in c_1) \text{ And } (\mathbf{X}[i_2] \in c_2) \text{ And } \ldots \text{ And } (\mathbf{X}[i_k] \in c_k) \\ \text{THEN} & g_n(\mathbf{X}) = p \end{array} \tag{2.3}$$

where $\mathbf{X}[i]$ is the i^{th} coordinate of $\mathbf{X}$ and $c_i \subseteq \mathbb{R}$ is an interval.

The *If* part, called the *condition* of the rule, or simply the rule, is composed of the conjunction of $k \leq d$ tests, each of which checking whether a feature (a coordinate of $\mathbf{X}$) satisfies a specified property or not and k is called the *length* of the rule. The *Then* part, called the *conclusion* of the rule, is the estimated value when the rule is *activated*, i.e., when the condition in the *If* part is satisfied. The rules are easy to understand and allow an interpretable decision process when k is small. For a review of the best-known algorithms for descriptive and predictive rule learning, see [80] and [23].

Formally, the models generated by such algorithms are defined by a corresponding *data-dependent partition* $\mathcal{P}_n$ of $\mathbb{R}^d$. Each element of the partition is named a *cell* and the empirical risk minimizer associated to $\mathcal{P}_n$ satisfies

$$g_n(\mathbf{x}) = \sum_{A \in \mathcal{P}_n} \frac{\sum_{i=1}^n Y_i \mathbf{1}_{X_i \in A}}{\sum_{i=1}^n \mathbf{1}_{X_i \in A}} \mathbf{1}_{\mathbf{x} \in A} \,, \qquad \mathbf{x} \in \mathbb{R}^d \,. \tag{2.4}$$

Those algorithms use the dataset $\mathbf{D}_n$ twice; first, the partition $\mathcal{P}_n = \mathcal{P}_n(\mathbf{D}_n)$ is chosen according to the dataset, second, this partition and the data are used to compute $g_n(\mathbf{x})$ as in (2.4). Note that g_n is the empirical risk minimizer among the class G_n of all piecewise constant functions over $\mathcal{P}_n$ denoted $\mathcal{G}_c \circ \mathcal{P}_n$. The major issue for these algorithms is the model interpretability, which requires a small value for the length k of the rule, whereas the consistency of the estimator is usually proved for conditions implying that $k = d$, *i.e.* a high model complexity.

In order to reduce the complexity of the model, we present a method of generating a partition. The idea is to generate a *data-dependent covering* $\mathcal{C}_n = \mathcal{C}_n(\mathbf{D}_n)$ of $\mathbb{R}^d$ rather than a partition. To do so, the dataset $\mathbf{D}_n$ is used to identify subsets of $\mathbb{R}^d$ that fulfill some specific conditions (we detail them in the next section). As elements of coverings can overlap, the construction of the subsets fulfilling these conditions can be done separately, which is not doable for the cells of partitions. Using a covering instead of a partition we ensure consistency without a condition on shrinkage of the cells. Moreover, each subset of the covering defines a rule with a small length k. Thus, we obtain a regression function described by a covering formed by simple rules rather than a partition formed by complex rules:

$$\begin{array}{rl} \text{IF} & (\mathbf{X} \in \mathbf{r}_1) \text{ And } (\mathbf{X} \in \mathbf{r}_2) \text{ And } \ldots \text{ And } (\mathbf{X} \in \mathbf{r}_l) \\ \text{THEN} & g_n(\mathbf{X}) = p \end{array}$$

where, for $j = 1, \ldots, l$

$$\mathbf{r}_j := \Big\{ \mathbf{x} : (\mathbf{x}[i_{j,1}] \in c_{j,1}) \text{ And } (\mathbf{x}[i_{j,2}] \in c_{j,2}) \text{ And } \ldots \text{ And } (\mathbf{x}[i_{j,k_j}] \in c_{j,k_j}) \Big\},$$

with $k_j \ll d$.

To estimate the value p, a partition $\mathcal{P}(\mathcal{C}_n)$ is generated from the covering $\mathcal{C}_n$ as an intermediate calculation. Formally, we define the partition generated from any collection of subsets $\mathcal{C}$ using the power set $2^{\mathcal{C}}$ gathering all subsets of $\mathcal{C}$:

Definition 2.1.1. *Let $\mathcal{C}$ be a finite collection of subsets of $\mathbb{R}^d$ and let $\mathbf{c} = \bigcup_{\mathbf{r}\in\mathcal{C}} \mathbf{r}$. We define the activation function as*

$$\varphi_{\mathcal{C}} : \mathbb{R}^d \mapsto 2^{\mathcal{C}}; \qquad \varphi_{\mathcal{C}}(\mathbf{x}) = \{\mathbf{r} \in \mathcal{C} : \mathbf{x} \in \mathbf{r}\}.$$

Then $\mathcal{P}(\mathcal{C})$, the partition of $\mathbf{c}$ generated from $\mathcal{C}$, is defined as

$$\mathcal{P}(\mathcal{C}) := \varphi_{\mathcal{C}}^{-1}(Im(\varphi_{\mathcal{C}})).$$

We provide a more explicit characterisation of $\mathcal{P}(\mathcal{C})$ in the following Proposition :

Proposition 2.1.1. *Let $\mathcal{C}$ be a finite collection of sets of $\mathbb{R}^d$. Then*

$$\mathcal{P}(\mathcal{C}) = \Big\{ \bigcap_{\mathbf{r}\in\tilde{\mathcal{C}}} \mathbf{r} \setminus \bigcup_{\mathbf{r}\in\mathcal{C}\setminus\tilde{\mathcal{C}}} \mathbf{r} : \tilde{\mathcal{C}} \subseteq \mathcal{C} \Big\} \setminus \{\emptyset\}.$$

Proof. By the definition of $\varphi_{\mathcal{C}}$, for any $\mathbf{x} \in \mathbb{R}^d$, $\varphi_{\mathcal{C}}^{-1}(\varphi_C(\mathbf{x})) = \bigcap_{\mathbf{r}\in\mathcal{C}:\mathbf{x}\in\mathbf{r}} \mathbf{r} \setminus \bigcup_{\mathbf{r}\in\mathcal{C}:\mathbf{x}\notin\mathbf{r}} \mathbf{r}$. Thus

$$\begin{aligned} A \in \mathcal{P}(\mathcal{C}) = \varphi_{\mathcal{C}}^{-1}(Im(\varphi_{\mathcal{C}})) &\iff \exists \mathbf{x} \in \mathbb{R}^d / A = \varphi_{\mathcal{C}}^{-1}(\varphi_C(\mathbf{x})) = \bigcap_{\substack{\mathbf{r}\in\mathcal{C}\\ \mathbf{x}\in\mathbf{r}}} \mathbf{r} \setminus \bigcup_{\substack{\mathbf{r}\in\mathcal{C}\\ \mathbf{x}\notin\mathbf{r}}} \mathbf{r} \\ &\iff \exists \tilde{\mathcal{C}} \subseteq \mathcal{C} / A = \bigcap_{\mathbf{r}\in\tilde{\mathcal{C}}} \mathbf{r} \setminus \bigcup_{\mathbf{r}\in\mathcal{C}\setminus\tilde{\mathcal{C}}} \mathbf{r} \text{ and } A \neq \emptyset. \end{aligned}$$

□

We have illustrated this transformation $\mathcal{P}$ on an example of five elements in Figures 1.7 and 1.8.

Remark 1. If $\mathcal{C}$ is a covering of $\mathbb{R}^d$, then $\mathcal{P}(\mathcal{C})$ is a partition of $\mathbb{R}^d$. The relation $\mathcal{C} = \mathcal{P}(\mathcal{C})$ holds if and only if $\mathcal{C}$ is a partition of $\cup_{\mathbf{r}\in\mathcal{C}}\mathbf{r}$.

Definition 2.1.2. *Let $\mathcal{C}$ be a covering of $\mathbb{R}^d$ and $\mathcal{C}' \subseteq \mathcal{C}$. The cells of $\mathcal{P}(\mathcal{C})$ wich are included in an element of $\mathcal{C}'$ are gathered in $\mathcal{P}_{\mathcal{C}}(\mathcal{C}')$:*

$$\mathcal{P}_{\mathcal{C}}(\mathcal{C}') := \{A \in \mathcal{P}(\mathcal{C}) : \exists \mathbf{r} \in \mathcal{C}',\ A \subseteq \mathbf{r}\}.$$

Remark 2. By definition, we have the identity $\mathcal{P}_{\mathcal{C}}(\mathcal{C}) = \mathcal{P}(\mathcal{C})$.

Definition 2.1.3. *Let $\mathcal{C}$ be a finite collection of subsets of $\mathbb{R}^d$. We introduce the maximal (resp. minimal) redundancy of $\mathcal{C}$ on a subset $\mathbf{r} \subseteq \mathbb{R}^d$:*

$$M(\mathcal{C}, \mathbf{r}) := \max_{x\in\mathbf{r}} \#\varphi_{\mathcal{C}}(x)$$

$$m(\mathcal{C}, \mathbf{r}) := \min_{x\in\mathbf{r}} \#\varphi_{\mathcal{C}}(x).$$

We shorten $M(\mathcal{C}, \mathbf{c})$ in $M(\mathcal{C})$ and $m(\mathcal{C}, \mathbf{c})$ in $m(\mathcal{C})$.

Remark 3. If $\mathcal{C}$ is a partition then for any $\mathbf{r} \in \mathcal{C}$, we have $\mathcal{P}_{\mathcal{C}}(\mathbf{r}) = \{\mathbf{r}\}$ and $M(\mathcal{C}, \mathbf{r}) = m(\mathcal{C}, \mathbf{r}) = 1$.

By using this transformation on a data-dependent covering, $\mathcal{C}_n$, we get the partition $\mathcal{P}(\mathcal{C}_n)$ and the associated estimator (2.4). The major difference compared to an estimator defined on a data-dependent partition is its interpretability. Moreover, using a partition from a data-dependent covering in place of a data-dependent partition generates a more complex partition where cells are not necessarily conjunctions of tests as in (2.3).

As the construction of a partition from a covering is time consuming, it is important to note that the partition $\mathcal{P}(\mathcal{C}_n)$ does not need to be constructed. The trick is to identify the unique cell of $\mathcal{P}(\mathcal{C}_n)$ which contains some $\mathbf{x} \in \mathbb{R}^d$ used for calculating the prediction at $\mathbf{x}$. By creating binary vectors of size $\#\mathcal{C}_n$, whose value is 1 if $\mathbf{x}$ fulfilled the rule's condition and 0 otherwise, this cell identification becomes a simple sequence of vectorial operations. Figure 1.9 is an illustration of this process (see Chapter 5 for more details).

All the estimators generated by the data-dependent covering algorithm belong to the class

$$G_n := \mathcal{G}_c \circ \mathcal{P}(\mathcal{C}_n) \tag{2.5}$$

of piecewise constant functions on the partition $\mathcal{P}(\mathcal{C}_n)$ such that $\forall g \in G_n, \forall x \in \mathbb{R}^d, |g(x)| \leq L$.

Hence, from definitions (2.2) we have

$$g_n(\mathbf{x}) = \sum_{A \in \mathcal{P}(\mathcal{C}_n)} \frac{\sum_{i=1}^n Y_i \mathbf{1}_{X_i \in A}}{\sum_{i=1}^n \mathbf{1}_{X_i \in A}} \mathbf{1}_{\mathbf{x} \in A}\,, \qquad \mathbf{x} \in \mathbb{R}^d\,, \tag{2.6}$$

and the risk minimizer over G_n is

$$\tilde{g}_n(\mathbf{x}) = \sum_{A \in \mathcal{P}(\mathcal{C}_n)} \frac{\mathbb{E}[Y \mathbf{1}_{\mathbf{X} \in A}]}{\mathbb{P}(\mathbf{X} \in A)} \mathbf{1}_{\mathbf{x} \in A}\,, \qquad \mathbf{x} \in \mathbb{R}^d\,. \tag{2.7}$$

The functions g_n and $\tilde{g}_n$ are indeed both in $\mathcal{G}_n$, although the latter is not computable from the data only.

Remark 4. The definition (2.6) of g_n guarantees that $\forall \mathbf{x} \in \mathbb{R}^d, |g_n(\mathbf{x})| \leq L$ so that L does not need to be known.

In the following subsection we discuss the important notion of interpretability.

2.1.2 Interpretability

In many fields, such as healthcare, marketing or asset management, decision makers prefer interpretable models rather to models with better accuracy but uninterpretable. As mentioned in [43], there is no rigorous mathematical foundation of the concept. In this chapter, interpretability corresponds to parsimonious characterization of the estimators of g^* generated by a given algorithm, *i.e.* the facility to describe the generated model in human words. Nowadays, the most popular and efficient algorithms for regression, such as Support Vector Machines, Neural networks, Random Forests,... are uninterpretable. The lack of interpretability comes from the complexity of the models they generate. We refer to them as black-box models. Usually, these black-box models have an optimal accuracy. We assert that the family of covering algorithms described here, can achieve a better Interpretability-Accuracy trade-off by reducing the complexity of the generated models keeping Accuracy guarantees, i.e., weak consistency.

There exist two ways of constructing interpretable models. The first one is to create black-box models and then to summarize them to create a so-called *post-hoc* interpretable algorithm. For example, recent researches propose to use explanation models,

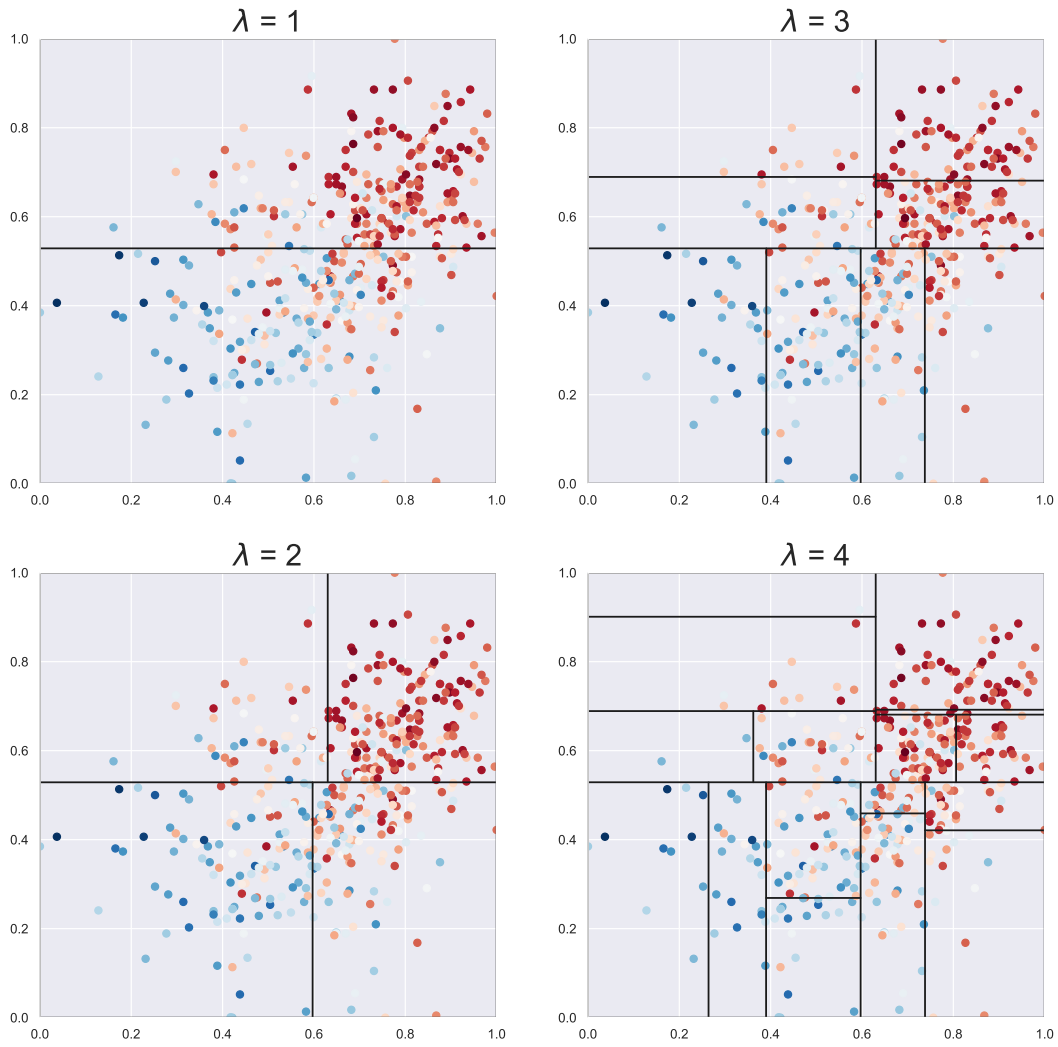


Figure 2.1 – Partitions generated by fully deployed decision tree algorithm, it means without pruning, for a maximal depth $\lambda \in \{1, 2, 3, 4\}$.

such as *LIME* [58], *DeepLIFT* [62] or *SHAP* [45], to interpret black-box models. These explanation models try to measure the importance of a feature (a coordinate of $\mathbf{X}$) on the prediction process (see [27] for a survey of existing methods). The second way to interpretability is to use *intrinsic* interpretable algorithms, it means algorithms that only generate interpretable models, such as rule-based algorithms.

The interpretability of the rule-based estimator is achieved when the length k of each rule is small and there is not too many rules. Considering that an estimate with one rule of length k is as interpretable as the one with the corresponding k rules of length 1, the interpretability gets naturally the additivity property. With this in mind, we are able to quantify the interpretability of an estimator g_n generated by a set of rules $\mathcal{C}_n$.

Definition 2.1.4. *The interpretability index of an estimator g_n generated by a set of rules $\mathcal{C}_n$ is defined by*

$$Int(g_n) := \sum_{\mathbf{r} \in \mathcal{C}_n} length(\mathbf{r}). \tag{2.8}$$

However in order to prove the consistency of the estimator g_n, one usually applies results such as Theorem 13.1 in [28] under the condition of shrinkage of the cells (Condition 13.10 in [28]). Each rule (2.3) must have a length $k = d$ in order to fulfill this sufficient condition without extra condition on the feature space. Then, for large d, the condition becomes uninterpretable. Moreover, as illustrated in Figure 2.1, the number of cells

necessary to have an accurate model is very large as the more precise the partition, the more complex the model.

For an estimator defined on a data-dependent covering, each prediction is explained by a small set of fulfilled rules, which are easy to understand, see Table 2.3 in Section 2.4 for an example. Even if the partition generated may be finer and more complex than a classical data-dependent partition, the explanation of the prediction is given by the covering and not the partition, and it remains understandable by humans, as illustrated in Figure 1.9.

Despite the fact that the parsimony of the selected set of rules is not theoretically guaranteed, the redundancy conditions (2.11) and (2.12) described below are heading in the right direction.

We obtain a consistent estimator g_n by carefully constructing the covering elements. We can apply none of the classical approaches based on Stone's theorem [65] because the covering is data-dependent nor based on Theorem 13.1 in [28] as Condition 13.10 in [28] forces rules to be complex ($k = d$). The key notion of this chapter is the notion of suitable data-dependent covering introduced in Section 2.2. Proposition 1.4.3 provides the main tool to prove the weak consistency of suitable data-dependent covering estimators stated in Theorem 2.2.1. Proposition 1.4.3 of independent interest is given in Section 2.3. Finally we apply our approach on covering elements using Random Forest as rule generator in Section 2.4.

2.2 Main result

We denote for any $\mathbf{r} \subseteq \mathbb{R}^d$ such that $\mathbb{Q}_n(\mathbf{r}) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{\mathbf{X}_j \in \mathbf{r}} > 0$

$$\mathbb{E}_n[Y \mid \mathbf{X} \in \mathbf{r}] := \frac{\frac{1}{n}\sum_{i=1}^n Y_i \mathbf{1}_{\mathbf{X}_i \in \mathbf{r}}}{\mathbb{Q}_n(\mathbf{r})}$$

and

$$\mathbb{V}_n(Y \mid \mathbf{X} \in \mathbf{r}) := \mathbb{E}_n\left[Y^2 \mid \mathbf{X} \in \mathbf{r}\right] - \mathbb{E}_n\left[Y \mid \mathbf{X} \in \mathbf{r}\right]^2.$$

In the same way, we define $\mathbb{E}[Y|\mathbf{X} \in \mathbf{r}] := \frac{\mathbb{E}[Y\mathbf{1}_{\mathbf{X}\in\mathbf{r}}]}{\mathbb{P}(\mathbf{X}\in\mathbf{r})}$ and $\mathbb{V}[Y|\mathbf{X} \in \mathbf{r}] := \mathbb{E}[Y^2|\mathbf{X} \in \mathbf{r}] - (\mathbb{E}[Y|\mathbf{X} \in \mathbf{r}])^2$.

2.2.1 Significance and coverage conditions

We introduce some conditions on each element of the covering. We use the classical notation $x_+ = \max\{x, 0\}$ for any $x \in \mathbb{R}$.

Definition 2.2.1. *We call a sequence $(\mathcal{C}_n)_{n\geq 1}$ of data-dependent coverings of $\mathbb{R}^d$* ***suitable*** *if it satisfies the two following conditions:*

1. *the* ***coverage condition****:* *(H9)*

$$\exists\ \alpha \in [0, 1/2),\ \forall \mathbf{r} \in \mathcal{C}_n,\ \mathbb{Q}_n(\mathbf{r}) > n^{-\alpha} \quad a.s., \tag{2.9}$$

for n sufficiently large;

2. *the* ***significance condition****:* *(H10)*

there exists two sequences $\beta_n \to 0$ and $\varepsilon_n \to 0$ such that:

$$\mathcal{C}_n = \mathcal{C}_n^s \cup \mathcal{C}_n^i \quad a.s., \tag{2.10}$$

for n sufficiently large, where the significant subsets $\mathcal{C}_n^s$ are defined by

$$\mathcal{C}_n^s := \Big\{ \mathbf{r} \in \mathcal{C}_n \,:\, \beta_n \big| \mathbb{E}_n[Y|\mathbf{X} \in \mathbf{r}] - \mathbb{E}_n[Y] \big| \geq \sqrt{(\mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) - \sigma^2)_+} \Big\}, \quad (2.11)$$

the insignificant subsets $\mathcal{C}_n^i$ are defined by

$$\mathcal{C}_n^i := \Big\{ \mathbf{r} \in \mathcal{C}_n \setminus \mathcal{C}_n^s : \varepsilon_n \geq \sqrt{(\mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) - \sigma^2)_+} \Big\}, \quad (2.12)$$

and their redundancies satisfy

$$\frac{M(\mathcal{C}_n^s)}{m(\mathcal{C}_n^s)} = o_{\mathbb{P}}(\beta_n^{-2} \wedge n^{1/2-\alpha}) \quad (2.13)$$

and

$$\frac{M(\mathcal{C}_n^i)}{m(\mathcal{C}_n^i)} = o_{\mathbb{P}}(\varepsilon_n^{-2} \wedge n^{1/2-\alpha}) \quad . \quad (2.14)$$

The coverage condition (H9) guarantees that the empirical within group expectation is a good estimation of the within group expectation. Up to our knowledge, the definitions of significant and insignificant elements of a covering in (H10) are new. An element fulfills the significance condition (2.11) if its conditional expectation is sufficiently different from the unconditional expectation. It ensures, in some sense, that the *within-group variances* of coverings with significant elements are controlled by the *between-group variances*. The insignificant condition (2.12) guarantees that the conditional variance of the insignificant elements shrinks to the noise variance. Both conditions (H9) and (H10) can be checked for each element of the covering separately. Thus the construction of such subsets can be parallelized, which allow designing algorithms less complex in comparison of usual ones.

Remark 5. An easy way to ensure (2.13) and (2.14) is to avoid inclusion between elements of the covering. Let $(\mathcal{C}_n)$ be a sequence of coverings that fulfills (H9). We consider $1 \leq i \leq \#\mathcal{C}_n$ any ordering of the covering. If

$$\mathbb{Q}_n\Big(\mathbf{r}_i \bigcap \Big\{ \bigcup_{1 \leq j \leq i-1} \mathbf{r}_j \Big\}\Big) \leq \gamma\, \mathbb{Q}_n(\mathbf{r}_i)\,, \qquad 1 \leq i \leq \#\mathcal{C}_n\,.$$

then the cardinality of $\mathcal{C}_n$ is upper bounded by $\frac{n^\alpha}{1-\gamma}$ for every n sufficiently large. Indeed, by the inclusion-exclusion principle we get

$$1 = \mathbb{Q}_n(\mathcal{C}_n) = \sum_{i=1}^{\#\mathcal{C}_n} \mathbb{Q}_n(\mathbf{r}_i \setminus \cup_{1 \leq j \leq i-1} \mathbf{r}_j) \geq \#\mathcal{C}_n\,(1-\gamma)\,n^{-\alpha}\,,$$

Thus (2.13) and (2.14) can be checked for any $\alpha \in [0, 1/4)$, using the fact that $M(\mathcal{C}_n^s)$ and $M(\mathcal{C}_n^i)$ are smaller than $\frac{n^\alpha}{1-\gamma}$ and setting $\beta_n = o_{\mathbb{P}}(n^{-\alpha/2})$ and $\varepsilon_n = o_{\mathbb{P}}(n^{-\alpha/2})$.

Example 1. The significant condition (2.11) can hold for a subset $\mathbf{r}$ with arbitrary diameter that does not satisfy Condition 13.10 of [28]. For instance, consider the case $g^* = \mathbf{1}_A$ for some Borel set A such that $0 < \mathcal{P}(\mathbf{X} \in A) < 1$. Then $\mathbf{r} = A$ is a significant subset as it satisfies the condition (2.11) with high probability for any β_n such that $n^{-1/4} = o(\beta_n)$ and n sufficiently large. Indeed, from the Strong Law of Large Numbers $k_n := \#\{X_i \in A\} \sim n\mathcal{P}(\mathbf{X} \in A)$ a.s. as $n \to \infty$. On the one hand, we obtain thanks to several applications of the Central Limit Theorem

$$\begin{aligned}
\big| \mathbb{E}_n[Y|\mathbf{X} \in A] - \mathbb{E}_n[Y] \big| &\geq \mathbb{E}_n[Y|\mathbf{X} \in A] - \mathbb{E}_n[Y] \\
&= 1 - \frac{k_n}{n} + \frac{1}{k_n} \sum_{i=1}^n Z_i \mathbf{1}_{\mathbf{X}_i \in A} - \frac{1}{n} \sum_{i=1}^n Z_i \\
&= 1 - \mathbb{P}(\mathbf{X} \in A) + O_{\mathbb{P}}(n^{-1/2}).
\end{aligned}$$

On the other hand, we obtain

$$\begin{aligned}(\mathbb{V}_n(Y|\mathbf{X}\in A)-\sigma^2)_+ &\leq |\mathbb{V}_n(Y|\mathbf{X}\in A)-\sigma^2| \\ &= \left|\frac{1}{k_n}\sum_{i=1}^n Z_i^2\mathbf{1}_{\mathbf{X}_i\in A} - \Big(\frac{1}{k_n}\sum_{i=1}^n Z_i\mathbf{1}_{\mathbf{X}_i\in A}\Big)^2 - \sigma^2\right| \\ &= O_{\mathbb{P}}(n^{-1/2}).\end{aligned}$$

Then

$$\begin{aligned}\beta_n\big|\mathbb{E}_n[Y|\mathbf{X}\in A]-\mathbb{E}_n[Y]\big| - \sqrt{(\mathbb{V}_n(Y|\mathbf{X}\in A)-\sigma^2)_+} \\ \geq \beta_n\big(1-\mathbb{P}(\mathbf{X}\in A)+O_{\mathbb{P}}(n^{-1/2})+O_{\mathbb{P}}(\beta_n^{-1}n^{-1/4})\big)\end{aligned}$$

Thus (2.11) holds for $\mathbf{r}=A$ with high probability for n sufficiently large. Note that for similar reasons (2.11) also holds with high probability for $\mathbf{r}=A^c$, $n^{-1/4}=o(\beta_n)$ and n sufficiently large. Finally, conditions (2.9), (2.13) and (2.14) are easily checked on the partitions $\mathcal{C}_n=\mathcal{P}_n=\{A,A^c\}$ that constitute a suitable coverings sequence with high probability for n large enough.

Remark 6. The significant condition (2.11) does not follow from a condition on the diameter of the subset. On the opposite, the insignificant condition (2.12) can follow from a condition on the diameter of the subset, see Proposition 2.3.3.

2.2.2 Partitioning number

To control the complexity of families of partitions, some tools introduced in [53, Sec. 1.2] are recalled (see also [28, Def 13.1]).

Definition 2.2.2. *Let Π be a family of partitions of $\mathbb{R}^d$.*

1. *The maximal number of cells in a partition of Π is denoted by*

$$\mathcal{M}(\Pi):=\sup\{\#\mathcal{P}:\mathcal{P}\in\Pi\}.$$

2. *For a set $\mathbf{x}_1^n=\{\mathbf{x}_1,\dots,\mathbf{x}_n\}\in(\mathbb{R}^d)^n$, let*

$$\Delta(\mathbf{x}_1^n,\Pi):=\#\big\{\{\mathbf{x}_1^n\cap A:A\in\mathcal{P}\}:\mathcal{P}\in\Pi\big\}$$

   be the number of distinct partitions of $\mathbf{x}_1^n$ induced by elements of Π.

3. *The partitioning number $\Delta_n(\Pi)$ of Π is defined by:*

$$\Delta_n(\Pi):=\max_{\mathbf{x}_1^n\in(\mathbb{R}^d)^n}\Delta(\mathbf{x}_1^n,\Pi).$$

The partitioning number is the maximal number of different partitions of any n points set that can be induced by elements of Π.

2.2.3 Consistency of data-dependent covering algorithms

In the following, we use the classical notion of Donsker class that is discussed in details in Section 2.3.

Theorem 2.2.1. *Assume that $\mathbb{Q}$ satisfies (H7) and (H8). Let $(\mathcal{C}_n)$ be a suitable data-dependent covering sequence (i.e., it satisfies (H9) and (H10)) fulfilling the two following conditions:*

$$\mathcal{M}(\Pi_n) \vee \log(\Delta_n(\Pi_n)) = o(n), \tag{H11}$$

where $\Pi_n := \{\mathcal{P}(\mathcal{C}_n(d_n)) : d_n \in \mathcal{S}^n\}$ for any $n \in \mathbb{N}^*$;

$$\forall n \in \mathbb{N}^*, \{\mathbf{c} \times \mathbb{R}, \mathbf{c} \in \mathcal{C}_n\} \subseteq \mathcal{B}, \tag{H12}$$

where $\mathcal{B}$ is a $\mathbb{Q}$-Donsker class.

Then the estimator g_n defined by (2.6) is weakly consistent:

$$\ell(g^*, g_n) = o_{\mathbb{P}}(1). \tag{2.15}$$

The proof of this theorem is postponed to Section 2.3.

This theorem gives us conditions on data-dependent covering algorithms to ensure that the generated empirical risk minimizer g_n converges in probability to the regression function g^* defined in (2.1). The condition (H11) is a classical one (*e.g.* [28, Conditions (13.7) and (13.8)]) used to ensure that the family of partitions Π_n is not too "complex". It means that the maximal number of cells in a partition, and the logarithm of the partitioning number, are small compared to the sample size. This condition guarantees that the estimation error tends to 0. The conditions (H9), (H10) and (H12) guarantee that the approximation error tends to 0 without any condition on the diameter of the cells.

2.3 Proof of Theorem 2.2.1

In order to prove the main theorem, we need some preliminary results based on notions of $\mathbb{Q}$-Donsker class and outer probability.

The outer probability, defined for $A \subseteq \Omega$ by

$$\mathbb{P}^*(A) := \inf\left\{\mathbb{P}(\tilde{A}) : A \subset \tilde{A}, \tilde{A} \in \mathcal{A}\right\}$$

is introduced to handle functions which are not necessarily measurable. The notation $O_{\mathbb{P}^*}(1)$ stands for *asymptotically tight* instead of the usual $O_{\mathbb{P}}(1)$ (*bounded in probability*). See [70, Chapter 18].

Let us define for any $f : \mathcal{S} \to \mathbb{R}$ in $\mathcal{L}^1(\mathbb{Q})$, $v_n f := \sqrt{n}(\mathbb{Q}_n f - \mathbb{Q} f)$ and consider the empirical process indexed by a set $\mathcal{F}$ of such functions: $\{v_n f : f \in \mathcal{F}\}$.

Definition 2.3.1. *[70, Section 19.2] $\mathcal{F}$ is called $\mathbb{Q}$-Donsker if the sequence of processes $\{v_n f : f \in \mathcal{F}\}$ converges in distribution to a tight limit process in the space $\ell^\infty(\mathcal{F})$.*

The limit process is then a $\mathbb{Q}$-Brownian bridge.

Definition 2.3.2. *A class of sets $\mathcal{B} \subseteq \mathcal{B}_{\mathcal{S}}$ is called $\mathbb{Q}$-Donsker if $\mathcal{I}_{\mathcal{B}} := \{\mathbf{1}_A : A \in \mathcal{B}\}$ is a $\mathbb{Q}$-Donsker class of functions.*

We remind that, given two functions l and u, the bracket $[l, u]$ is the set of all functions f with $l \leq f \leq u$. An ϵ-bracket in $L_2(\mathbb{Q})$ is a bracket $[l, u]$ with $\|u - l\|_{L_2(\mathbb{Q})} < \epsilon$. The bracketing number $N_{[]}(\varepsilon, \mathcal{F}, \mathcal{L}^2(\mathbb{Q}))$ is the minimum number of ϵ-brackets needed to cover $\mathcal{F}$. (The bracketing functions l and u must have finite $L_2(\mathbb{Q})$-norms but need not belong to $\mathcal{F}$.) A simple condition for a class to be $\mathbb{Q}$-Donsker is that $N_{[]}(\varepsilon, \mathcal{F}, \mathcal{L}^2(\mathbb{Q}))$ do not grow too fast to infinity as ε tends to 0. The speed is measured in terms of the bracketing integral defined by

$$J_{[]}(\delta, \mathcal{F}, \mathcal{L}^2(\mathbb{Q})) = \int_0^\delta \sqrt{\log N_{[]}(\varepsilon, \mathcal{F}, \mathcal{L}^2(\mathbb{Q}))} d\varepsilon.$$

Theorem 2.3.1. *[70, Theorem 19.4] Every class $\mathcal{F}$ of measurable functions with $J_{[]}(1, \mathcal{F}, \mathcal{L}^2(\mathbb{Q})) < \infty$ is $\mathbb{Q}$-Donsker.*

A straightforward application of this theorem is the following result.

Lemma 2.3.1. *Let $\mathcal{C}$ be the set of all hyperrectangles of $\mathbb{R}^d$:*

$$\mathcal{C} := \left\{ [\mathbf{a}, \mathbf{a} + \mathbf{h}] \subseteq \mathbb{R}^d : \mathbf{a} \in \mathbb{R}^d, \mathbf{h} \in \mathbb{R}^d_+ \right\} \cup \{\emptyset\},$$

then $\mathcal{B} := \{\mathbf{c} \times \mathbb{R} : \mathbf{c} \in \mathcal{C}\}$ is a $\mathbb{Q}$-Donsker class.

Proof. Let $\varepsilon > 0$ and consider the sequence $\{\mathbf{t}_i\}_{i \in \{0, \dots, \lceil 2d/\varepsilon \rceil\}}$ defined as follows: for any $k \in \{1, \dots, d\}$,

$$\begin{cases} t_{0,k} = -\infty \\ \forall i \in \{1, \dots, \lfloor 2d/\varepsilon \rfloor\}, t_{i,k} = \sup\{t \in \mathbb{R} : \mathbb{Q}(\mathbb{R}^{k-1} \times]t_{i-1,k}, t[\times \mathbb{R}^{d-k} \times [-L, L]) < \varepsilon/2d\} \\ t_{\lceil 2d/\varepsilon \rceil, k} = +\infty \end{cases}$$

From the definition of $\mathbf{t}_i$'s we have $\forall k \in \{1, \dots, d\}$, $\forall i \in \{1, \dots, \lceil 2d/\varepsilon \rceil\}$,

$$\begin{aligned} &\mathbb{Q}\left(\mathbb{R}^{k-1} \times]t_{i-1,k}, t_{i,k}[\times \mathbb{R}^{d-k} \times [-L, L]\right) \leq \varepsilon/2d \\ &\mathbb{Q}\left(\mathbb{R}^{k-1} \times]t_{i-1,k}, t_{i,k}] \times \mathbb{R}^{d-k} \times [-L, L]\right) \geq \varepsilon/2d \\ &\mathbb{Q}\left(\mathbb{R}^{k-1} \times]t_{0,k}, t_{i,k}] \times \mathbb{R}^{d-k} \times [-L, L]\right) \geq i\varepsilon/2d \\ &\mathbb{Q}\left(\mathbb{R}^{k-1} \times]t_{0,k}, t_{\lfloor 2d/\varepsilon \rfloor, k}] \times \mathbb{R}^{d-k} \times [-L, L]\right) \geq 1 - \varepsilon/2d \text{ (since } \lfloor 2d/\varepsilon \rfloor \geq 2d/\varepsilon - 1) \\ &\mathbb{Q}\left(\mathbb{R}^{k-1} \times]t_{\lfloor 2d/\varepsilon \rfloor, k}, +\infty[\times \mathbb{R}^{d-k} \times [-L, L]\right) \leq \varepsilon/2d \end{aligned}$$

Hence, $\forall i \in \{1, \dots, \lceil 2d/\varepsilon \rceil\}$ and $\forall k \in \{1, \dots, d\}$ we have that

$$\mathbb{Q}\left(\mathbb{R}^{k-1} \times]t_{i-1,k}, t_{i,k}[\times \mathbb{R}^{d-k} \times [-L, L]\right) \leq \varepsilon/2d.$$

Consider now the set of brackets defined by (see Fig 2.2)

$$A := \left\{ [\mathbf{1}_{\prod_{k=1}^d [t_{i_k+1,k}, t_{j_k-1,k}]}, \mathbf{1}_{\prod_{k=1}^d]t_{i_k,k}, t_{j_k,k}[}], \forall k \in \{1, \dots, d\}, 0 \leq i_k < j_k \leq \lceil 2d/\varepsilon \rceil \right\}.$$

We have $\#A = (\lceil 2d/\varepsilon \rceil (\lceil 2d/\varepsilon \rceil + 1)/2)^d$ and

$$\begin{aligned} &d(\mathbf{1}_{\prod_{k=1}^d [t_{i_k+1,k}, t_{j_k-1,k}] \times [-L,L]}, \mathbf{1}_{\prod_{k=1}^d]t_{i_k,k}, t_{j_k,k}[\times [-L,L]}) \\ &= \int_{\mathbb{R}^d} \left| \mathbf{1}_{\prod_{k=1}^d]t_{i_k,k}, t_{j_k,k}[\times [-L,L]} - \mathbf{1}_{\prod_{k=1}^d [t_{i_k+1,k}, t_{j_k-1,k}] \times [-L,L]} \right| d\mathbb{Q} \\ &\leq \int_{\mathbb{R}^d} \sum_{k=1}^d \mathbf{1}_{\mathbb{R}^{k-1} \times \left(]t_{i_k,k}, t_{i_k+1,k}[\cup]t_{j_k-1,k}, t_{j_k,k}[\right) \times \mathbb{R}^{d-k} \times [-L,L]} d\mathbb{Q} \\ &\leq \sum_{k=1}^d \mathbb{Q}\left(\mathbb{R}^{k-1} \times (]t_{i_k,k}, t_{i_k+1,k}[\cup]t_{j_k-1,k}, t_{j_k,k}[) \times \mathbb{R}^{d-k} \times \mathbb{R}\right) \\ &\leq d\left(\varepsilon/2d + \varepsilon/2d\right) \\ &\leq \varepsilon. \end{aligned}$$

The term after the first equality corresponds to the integration of the hatched area in Figure 2.2 and the term after the next inequality corresponds to the integration of the area delimited by the dotted lines.

Thus, the $\mathcal{L}_1(\mathbb{Q})$-size of the brackets is not larger than ε. Since $\mathbb{Q}f^2 = \mathbb{Q}f$ for every $f \in \mathcal{I}_{\mathcal{C}} = \{\mathbf{1}_A : A \in \mathcal{C}\}$, the $L_2(\mathbb{Q})$-size of the brackets is not larger than $\sqrt{\varepsilon}$.

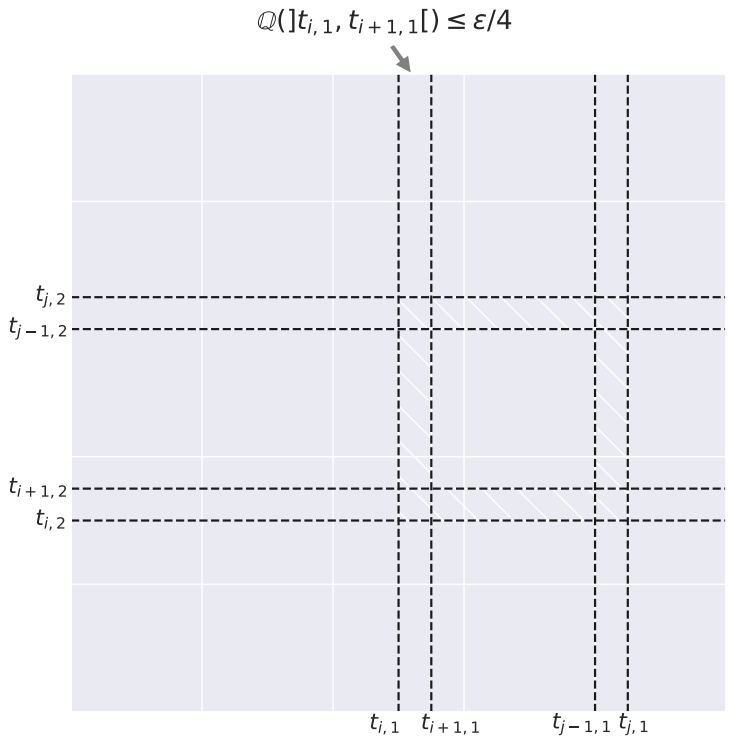


Figure 2.2 – Example of bracket for $d = 2$. With $l = \mathbf{1}_{[t_{i+1,1};t_{j-1,1}]\times[t_{i+1,2};t_{j-1,2}]}$ and $u = \mathbf{1}_{[t_{i,1};t_{j,1}]\times[t_{i,2};t_{j,2}]}$, for any rectangle A, $\mathbf{1}_A \in [l, u]$ if and only if its boundary $\bar{A} \setminus \mathring{A}$ is included in the hatched area.

Let $f \in \mathcal{I}_{\mathcal{C}}$. Then $\exists \mathbf{a} \in \mathbb{R}^d, \mathbf{h} \in \mathbb{R}_+^d$ s.t. $f = \mathbf{1}_{[\mathbf{a},\mathbf{a}+\mathbf{h}]\times[-L,L]}$. We set, for any $k \in \{1, \dots, d\}$,

$$i_k^* := \max\{\iota \in \{0, \dots, \lceil 2d/\varepsilon \rceil\} : t_{\iota,k} < a_k\}$$
$$j_k^* := \min\{\iota \in \{0, \dots, \lceil 2d/\varepsilon \rceil\} : t_{\iota,k} > a_k + h_k\}.$$

There always exist j_k^* and i_k^* since the sets $\{\iota \in \{0, \dots, \lceil 2d/\varepsilon \rceil\} : t_{\iota,k} < a_k\}$ and $\{\iota \in \{0, \dots, \lceil 2d/\varepsilon \rceil\} : t_{\iota,k} > a_k + h_k\}$ are not empty (they contain respectively 0 and $\lceil 2d/\varepsilon \rceil$) and, by construction, $j_k^* > i_k^*$. Moreover,

$$\mathbf{1}_{\prod_{k=1}^d [t_{i_k^*+1}, t_{j_k^*-1}]\times[-L,L]} \leq \mathbf{1}_{[\mathbf{a},\mathbf{a}+\mathbf{h}]\times[-L,L]} \leq \mathbf{1}_{\prod_{k=1}^d]t_{i_k^*}, t_{j_k^*}[\times[-L,L]}$$

and

$$[\mathbf{1}_{\prod_{k=1}^d [t_{i_k^*+1}, t_{j_k^*-1}]\times[-L,L]}, \mathbf{1}_{\prod_{k=1}^d]t_{i_k^*}, t_{j_k^*}[\times[-L,L]}] \in A.$$

Thus, $\forall f \in \mathcal{I}_{\mathcal{C}}, \exists [l, u] \in A$ such that $l \leq f \leq u$.

It follows that $N_{[]}(\sqrt{\varepsilon}, \mathcal{I}_{\mathcal{C}}, \mathcal{L}^2(\mathbb{Q})) \leq \left(\frac{\lceil 2d/\varepsilon \rceil(\lceil 2d/\varepsilon \rceil+1)}{2}\right)^d$. Hence,

$$J_{[]}(1, \mathcal{I}_{\mathcal{C}}, \mathcal{L}^2(\mathbb{Q})) < \infty.$$

According to Theorem 2.3.1, this guarantees that $\mathcal{I}_{\mathcal{C}}$ is a $\mathbb{Q}$-Donsker class.

□

Now, with $\mathbb{Q}f := \int f d\mathbb{Q}$ and $\mathbb{Q}_n f := \int f d\mathbb{Q}_n$, for any $f \in \mathcal{L}^1(\mathbb{Q})$, if $\mathcal{F}$ is a $\mathbb{Q}$-Donsker class of functions, then the empirical process $((\sqrt{n}(\mathbb{Q}_n f - \mathbb{Q}f))_{f\in\mathcal{F}})_{n\in\mathbb{N}}$ is asymptotically tight as a sequence of maps with values in $\ell^\infty(\mathcal{F})$ (this is a consequence of Prohorov's Theorem adapted to this framework – see Theorem 18.12 in [70]). Keeping in mind that a compact set in $\ell^\infty(\mathcal{F})$ is bounded, we have:

Proposition 2.3.1. *[70, Theorem 18.12] Let $\mathcal{F}$ be a $\mathbb{Q}$-Donsker class of functions. Then*

$$\|\mathbb{Q}_n - \mathbb{Q}\|_{\mathcal{F}} = O_{\mathbb{P}^*}(n^{-1/2}),$$

where for any $v : \mathcal{F} \to \mathbb{R}$, $\|v\|_{\mathcal{F}} = \sup_{f\in\mathcal{F}} |v(f)|$.

Remark 7. If $\mathcal{B} \subseteq \mathcal{B}_{\mathcal{S}}$ is a $\mathbb{Q}$-Donsker class of sets, where $\mathcal{B}_{\mathcal{S}}$ is the Borel set on $\mathcal{S}$, then

$$\|\mathbb{Q}_n - \mathbb{Q}\|_{\mathcal{B}} = O_{\mathbb{P}^*}(n^{-1/2}),$$

where for any $v : \mathcal{B} \to \mathbb{R}$, $\|v\|_{\mathcal{B}} = \sup_{A\in\mathcal{B}} |v(A)|$.

Remark 8. It can be checked that if $(Z_n)_{n\in\mathbb{N}}$ is a sequence of non-negative random variables, $(a_n)_{n\in\mathbb{N}} \in (\mathbb{R}^+)^{\mathbb{N}}$ such that $a_n = o_{\mathbb{P}}(1)$ and $(M_n)_{n\in\mathbb{N}}$ is a sequence of maps such that $M_n = O_{\mathbb{P}^*}(1)$ and $Z_n \leq a_n M_n$ for any n, then $Z_n \xrightarrow[n\to+\infty]{\mathbb{P}} 0$.

The usual notion of boundedness in probability for sequences of random variables need be generalized because sequences of maps are to be considered, with values in metric spaces which are not Euclidean spaces (thus bounded and closed sets need not be compact) and which are not guaranteed to be measurable. We need involve the outer probability $\mathbb{P}^*$.

Definition 2.3.3. *[70, Chapter 18] A sequence $(M_n)_{n\in\mathbb{N}}$ of maps defined on Ω and with values in a metric space $(\mathbb{D}, d)$ is said to be asymptotically tight if*

$$\forall \varepsilon > 0, \exists K \subset \mathbb{D} \text{ compact } \forall \delta > 0, \limsup_{n\to\infty} \mathbb{P}^*(M_n \notin K^\delta) < \varepsilon,$$

with $K^\delta = \{y \in \mathbb{D} : d(y, K) < \delta\}$ for any $K \subset \mathbb{D}$ and $\delta > 0$.

Remark 9. If $\mathbb{D} = \mathbb{R}$, (M_n) is asymptotically tight if and only if

$$\forall \varepsilon > 0, \exists M > 0 / \limsup_{n\to\infty} \mathbb{P}^*(|M_n| > M) < \varepsilon.$$

2.3.1 Empirical estimation of conditional expectations

We shall also use the following proposition, which is inspired by the work of [25] (Proposition 3.2).

Proposition 2.3.2. *Let $\mathcal{B} \subseteq \mathcal{B}_{\mathcal{S}}$ and let $\mathcal{F}_{\mathcal{B}} := \{f\mathbf{1}_A : f \in \mathcal{F}, A \in \mathcal{B}\}$ where $\mathcal{F}$ is a set of functions in $\mathcal{L}^1(\mathbb{Q})$ uniformly bounded. If $\mathcal{B}$ and $\mathcal{F}_{\mathcal{B}}$ are $\mathbb{Q}$-Donsker classes then for any $\alpha \in [0, 1/2)$ and with $\mathcal{B}_n := \{A \in \mathcal{B}, \mathbb{Q}_n(A) \geq n^{-\alpha}\}$,*

$$\sup_{f\in\mathcal{F}} \sup_{A\in\mathcal{B}_n} \left|\mathbb{E}_n[f \mid A] - \mathbb{E}[f \mid A]\right| = O_{\mathbb{P}^*}(n^{\alpha-1/2}).$$

Proof. Let $\varepsilon > 0$. First, for any $f \in \mathcal{F}$ and $A \in \mathcal{B}_n$, since $\mathbb{Q}_n(A) > 0$ and then $\mathbb{Q}(A) > 0$,

$$\begin{aligned}
&|\mathbb{E}_n[f \mid A] - \mathbb{E}[f \mid A]| \\
&= \left|\frac{\int_A f d\mathbb{Q}_n}{\mathbb{Q}_n(A)} - \frac{\int_A f d\mathbb{Q}}{\mathbb{Q}(A)}\right| \\
&= \left|\frac{\mathbb{Q}(A)\left(\int_A f d\mathbb{Q}_n - \int_A f d\mathbb{Q}\right) + (\mathbb{Q}(A) - \mathbb{Q}_n(A))\int_A f d\mathbb{Q}}{\mathbb{Q}(A)\mathbb{Q}_n(A)}\right| \\
&\leq \left|\frac{\int_A f d\mathbb{Q}_n - \int_A f d\mathbb{Q}}{\mathbb{Q}_n(A)}\right| + \left|(\mathbb{Q}(A) - \mathbb{Q}_n(A))\frac{\int_A f d\mathbb{Q}}{\mathbb{Q}(A)\mathbb{Q}_n(A)}\right|. \qquad (2.16)
\end{aligned}$$

Now, according to Proposition 2.3.1,

$$\sup_{\tilde{f}\in\mathcal{F},\tilde{A}\in B}\left|\int_{\tilde{A}}\tilde{f}d\mathbb{Q}_n-\int_{\tilde{A}}\tilde{f}d\mathbb{Q}\right|=O_{\mathbb{P}^*}(n^{-1/2})$$

and

$$\sup_{\tilde{A}\in\mathcal{B}}\left|\mathbb{Q}_n(\tilde{A})-\mathbb{Q}(\tilde{A})\right|=O_{\mathbb{P}^*}(n^{-1/2}).$$

Thus, According to Remark 9, there exists $M>0$ such that for any n large enough,

$$\mathbb{P}^*\left\{\sup_{\tilde{f}\in\mathcal{F},\tilde{A}\in B}\left|\int_{\tilde{A}}\tilde{f}d\mathbb{Q}_n-\int_{\tilde{A}}\tilde{f}d\mathbb{Q}\right|>Mn^{-1/2}\right\}<\frac{\varepsilon}{2}$$

and

$$\mathbb{P}^*\left\{\sup_{\tilde{A}\in\mathcal{B}}\left|\mathbb{Q}_n(\tilde{A})-\mathbb{Q}(\tilde{A})\right|>Mn^{-1/2}\right\}<\frac{\varepsilon}{2}$$

so that $\mathbb{P}^*(\Omega_n)\geq 1-\varepsilon$ with

$$\Omega_n:=\left\{\sup_{\tilde{f}\in\mathcal{F},\tilde{A}\in B}\left|\int_{\tilde{A}}\tilde{f}d\mathbb{Q}_n-\int_{\tilde{A}}\tilde{f}d\mathbb{Q}\right|\leq Mn^{-1/2}\right\}\bigcap \left\{\sup_{\tilde{A}\in B}\left|\mathbb{Q}_n(\tilde{A})-\mathbb{Q}(\tilde{A})\right|\leq Mn^{-1/2}\right\}.$$

Then (2.16) yields, with $c:=\sup_{f\in\mathcal{F},x\in\mathcal{S}}|f(x)|<\infty$ and since $\mathbb{Q}_n(A)\geq n^{-\alpha}$, for n large enough, in the event Ω_n,

$$\sup_{f\in\mathcal{F},A\in\mathcal{B}_n}\left|\mathbb{E}_n\left[f\mid A\right]-\mathbb{E}\left[f\mid A\right]\right|\leq Mn^{\alpha-1/2}(1+c),$$

since $\frac{\int_A f d\mathbb{Q}}{\mathbb{Q}(A)}\leq c$.

Finally, it has been proven that $\forall\varepsilon>0,\exists M>0,\exists N\in\mathbb{N}^*/\forall n\geq N$,

$$\mathbb{P}^*\left\{\sup_{f\in\mathcal{F},A\in\mathcal{B}_n}\left|\mathbb{E}_n\left[f\mid A\right]-\mathbb{E}\left[f\mid A\right]\right|>Mn^{\alpha-1/2}\right\}<\varepsilon$$

and then $\forall\varepsilon>0,\exists M>0$ such that

$$\limsup_{n\to\infty}\mathbb{P}^*\left\{\sup_{f\in\mathcal{F},A\in\mathcal{B}_n}\left|\mathbb{E}_n\left[f\mid A\right]-\mathbb{E}\left[f\mid A\right]\right|>Mn^{\alpha-1/2}\right\}\leq\varepsilon$$

which, together with Remark 9 again, proves the proposition. □

Corollary 2.3.1. *Let $\mathcal{B}\subseteq\mathcal{B}_{\mathcal{S}}$ be a $\mathbb{Q}$-Donsker class. If Y is bounded then for any $i\in\mathbb{N}$ and any $\alpha\in[0,1/2)$, with $\mathcal{B}_n:=\{A\in\mathcal{B},\mathbb{Q}_n(A)\geq n^{-\alpha}\}$ we have*

$$\sup_{A\in\mathcal{B}_n}\left|\mathbb{E}_n\left[Y^i\mid(\mathbf{X},Y)\in A\right]-\mathbb{E}\left[Y^i\mid(\mathbf{X},Y)\in A\right]\right|=O_{\mathbb{P}^*}(n^{\alpha-1/2}),\tag{2.17}$$

and

$$\sup_{A\in\mathcal{B}_n}\left|\mathbb{V}_n\left[Y\mid(\mathbf{X},Y)\in A\right]-\mathbb{V}\left[Y\mid(\mathbf{X},Y)\in A\right]\right|=O_{\mathbb{P}^*}(n^{\alpha-1/2}).\tag{2.18}$$

Proof of (2.17). Let $L=\operatorname{ess\,sup}Y$, $i\in\mathbb{N}$, and $f_i\in\mathcal{L}^1(\mathbb{Q})$ be defined by

$$\begin{aligned}f_i:\mathbb{R}^d\times[-L,L]&\to[-L^i,L^i]\\(\mathbf{x},y)&\mapsto y^i.\end{aligned}$$

f_i is bounded and $\{f_i\}$ is finite thus Donsker. The result is then a straightforward application of Proposition 2.3.2. □

Proof of (2.18). This part follows from (2.18) since Y is bounded and

$$\mathbb{V}_n\left[Y \mid (\mathbf{X}, Y) \in A\right] := \mathbb{E}_n\left[Y^2 \mid (\mathbf{X}, Y) \in A\right] - \mathbb{E}_n\left[Y \mid (\mathbf{X}, Y) \in A\right]^2 .$$

□

It seems that the result of Corollary 2.3.1, which is of independent interest, does not appear as such in the existing literature. As a first application of Corollary 2.3.1, we show that any partition with shrinking cell diameters is a suitable covering. We define the diameter of a cell $\mathbf{r}$ as $\mathrm{Diam}(\mathbf{r}) = \sup_{x\in\mathbf{r},\, x'\in\mathbf{r}} \|x - x'\|$, where $\|\cdot\|$ is any norm of $\mathbb{R}^d$.

Proposition 2.3.3. *Consider a sequence $(\mathcal{P}_n)_{n\in\mathbb{N}}$ of data-dependent partitions, that satisfies the coverage condition* (2.9) *and such that*

$$\bigcup_{n\in\mathbb{N}^*} \bigcup_{\mathbf{r}\in\mathcal{P}_n} (\mathbf{r}\times\mathbb{R})$$

is a.s. a $\mathbb{Q}$-Donsker class. If g^* is uniformly continuous and if

$$\max_{\mathbf{r}\in\mathcal{P}_n} \mathit{Diam}(\mathbf{r}) = o_{\mathbb{P}}(1) \tag{2.19}$$

then the sequence $(\mathcal{P}_n)$ is suitable.

Proof. Let us show that each cell is significant or insignificant. Thanks to Condition (2.9), Corollary 2.3.1 Eq. (2.18) and Remark 8,

$$\max_{\mathbf{r}\in\mathcal{P}_n} |\mathbb{V}_n(Y|\mathbf{X}\in\mathbf{r}) - \mathbb{V}(Y \mid \mathbf{X}\in\mathbf{r})| = O_{\mathbb{P}}(n^{\alpha-1/2}). \tag{2.20}$$

Moreover $\mathbb{V}(Y \mid \mathbf{X}\in\mathbf{r}) = \mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X}\in\mathbf{r}) + \sigma^2$. Thus, as the redundancy condition (2.14) is automatically satisfied for cells of a partition, the desired result will follow if we check that

$$\varepsilon_n := \max_{\mathbf{r}\in\mathcal{P}_n} \sqrt{(\mathbb{V}_n(Y|\mathbf{X}\in\mathbf{r}) - \sigma^2)_+}$$

converges to 0.

From (2.20) we remark that

$$\varepsilon_n \le \max_{\mathbf{r}\in\mathcal{P}_n} \sqrt{\mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X}\in\mathbf{r})} + O_{\mathbb{P}}(n^{\alpha/2-1/4}) .$$

For all n, if $\mathbf{r}\in\mathcal{P}_n$, then $\mathbf{r}\times\mathbb{R}\in\mathcal{B}_{\mathcal{S}}$. We denote $\mathbf{X}_\mathbf{r}$ and $\mathbf{X}'_\mathbf{r}$ two independent variables distributed as $\mathbf{X}$ given that $\mathbf{X}\in\mathbf{r}$. We obtain

$$\begin{aligned}\mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X}\in\mathbf{r}) &= \mathbb{V}(g^*(\mathbf{X}_\mathbf{r})) \\ &= \tfrac{1}{2}\mathbb{V}\big(g^*(\mathbf{X}_\mathbf{r}) - g^*(\mathbf{X}'_\mathbf{r})\big) \\ &\le \tfrac{1}{2}\mathbb{E}\left[(g^*(\mathbf{X}_\mathbf{r}) - g^*(\mathbf{X}'_\mathbf{r}))^2\right] .\end{aligned}$$

Thus, if we denote w the modulus of continuity of g^*, we get

$$\sqrt{\mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X}\in\mathbf{r})} \le 2^{-1/2} w(\mathrm{Diam}(\mathbf{r})).$$

By uniform continuity, the condition (2.19) implies that

$$\varepsilon_n \le 2^{-1/2} \max_{\mathbf{r}\in\mathcal{P}_n} \Big(w(\mathrm{Diam}(\mathbf{r}))\Big) + O_{\mathbb{P}}(n^{\alpha/2-1/4}) = o_{\mathbb{P}}(1) .$$

Thus, from (2.12), each cell which is not significant is insignificant and the corresponding covering sequence is suitable. □

Remark 10. The condition of uniform continuity of g^* in Proposition 2.3.3 may be simply raised. Indeed, from [28, Corollary A.1], g^* can be approximated arbitrarily closely in $\mathcal{L}^2(\mathbb{Q}_\mathbf{X})$ by functions of $C_0^\infty(\mathbb{R}^d)$ where $\mathbb{Q}_\mathbf{X}$ is the marginal distribution of $\mathbf{X}$.

2.3.2 Estimation-approximation decomposition

The excess loss (2.1) can be decomposed into two terms using the following lemma:

Lemma 2.3.2 (Lemma 10.1 of [28]). *Let G_n be a class of functions $g : \mathbb{R}^d \to [-L, L]$ depending on the data $D_n = ((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n))$. If g_n satisfies (2.2) then*

$$\ell(g^*, g_n) \leq 2 \sup_{g \in G_n} |\mathcal{L}_n(g) - \mathcal{L}(g)| + \inf_{g \in G_n} \mathbb{E}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))^2\right].$$

Hence, to prove (2.15) it is sufficient to prove that:

$$\sup_{g \in G_n} |\mathcal{L}_n(g) - \mathcal{L}(g)| = o_{\mathbb{P}}(1). \tag{2.21}$$

and

$$\inf_{g \in G_n} \mathbb{E}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))^2\right] = o_{\mathbb{P}}(1). \tag{2.22}$$

The *estimation error* (2.21) controls the distance between the best function in G_n and g_n. The *approximation error* (2.22) is the smallest error for a function of G_n.

The two terms have opposite behaviors. Indeed, if G_n is not too complex the empirical loss will be close to the loss uniformly over G_n. Thus, the error due to the minimization of the empirical loss instead of the loss will be small. On the other hand, the loss cannot be better than for the best function of G_n. So, G_n must be complex enough. It is the classical *Approximation/Estimation* trade-off.

2.3.3 Approximation Error

In this subsection, we prove (2.22) using hypotheses (H7), (H8), (H9), (H10) and (H12).

First, let us define $\mathcal{C}_n^{i\backslash s} = \{\mathbf{r} \setminus \mathbf{c}_n^s : \mathbf{r} \in \mathcal{C}_n^i\}$ where $\mathbf{c}_n^s = \cup_{\mathbf{r} \in \mathcal{C}_n^s} \mathbf{r}$. Notice that $\mathcal{P}(\mathcal{C}_n^{i\backslash s}) = \left\{A \setminus \mathbf{c}_n^s : A \in \mathcal{P}(\mathcal{C}_n^i)\right\}$. Indeed, by Proposition 2.1.1:

$$\begin{aligned}
\mathcal{P}(\mathcal{C}_n^{i\backslash s}) &= \left\{\bigcap_{\mathbf{r} \in \tilde{\mathcal{C}}} \mathbf{r} \setminus \bigcup_{\mathbf{r} \in \mathcal{C}_n^{i\backslash s} \setminus \tilde{\mathcal{C}}} \mathbf{r} : \tilde{\mathcal{C}} \subseteq \mathcal{C}_n^{i\backslash s}\right\} \\
&= \left\{\bigcap_{\mathbf{r} \in \tilde{\mathcal{C}}^i} (\mathbf{r} \setminus \mathbf{c}_n^s) \setminus \bigcup_{\mathbf{r} \in \mathcal{C}_n^i \setminus \tilde{\mathcal{C}}^i} (\mathbf{r} \setminus \mathbf{c}_n^s) : \tilde{\mathcal{C}}^i \subseteq \mathcal{C}_n^i\right\} \\
&= \left\{\left(\bigcap_{\mathbf{r} \in \tilde{\mathcal{C}}^i} \mathbf{r} \setminus \bigcup_{\mathbf{r} \in \mathcal{C}_n^i \setminus \tilde{\mathcal{C}}^i} \mathbf{r}\right) \setminus \mathbf{c}_n^s : \tilde{\mathcal{C}}^i \subseteq \mathcal{C}_n^i\right\} \\
&= \left\{A \setminus \mathbf{c}_n^s : A \in \mathcal{P}(\mathcal{C}_n^i)\right\}.
\end{aligned} \tag{2.23}$$

So that, by Proposition 2.1.1 again, $\mathcal{P}(\mathcal{C}_n)$ is a partition finer than $\mathcal{P}(\mathcal{C}_n^s) \cup \mathcal{P}(\mathcal{C}_n^{i\backslash s})$. Hence, with

$$\tilde{g}_n = \sum_{A \in \mathcal{P}(\mathcal{C}_n^s) \cup \mathcal{P}(\mathcal{C}_n^{i\backslash s})} \mathbb{E}[Y | \mathbf{X} \in A] \mathbf{1}_A,$$

we have $\tilde{g}_n \in \mathcal{G}_n$ and

$$\inf_{g \in \mathcal{G}_n} \sqrt{\mathbb{E}\left[(g(\mathbf{X}) - g^*(\mathbf{X}))^2\right]} \leq \sqrt{\mathbb{E}\left[(\tilde{g}_n(\mathbf{X}) - g^*(\mathbf{X}))^2\right]}$$

Thus, to prove (2.22), it suffices to show that $\mathbb{W}_n = o_{\mathbb{P}}(1)$ where

$$\mathbb{W}_n := \mathbb{E}\left[(\tilde{g}_n(\mathbf{X}) - g^*(\mathbf{X}))^2\right].$$

We will repeatedly make use of the following equalities:

$$\mathbb{E}[g^*(\mathbf{X}) \mid \mathbf{X} \in \mathbf{r}] = \mathbb{E}[Y \mid \mathbf{X} \in \mathbf{r}], \qquad \forall \mathbf{r} \subseteq \mathbb{R}^d. \tag{2.24}$$

and

$$\mathbb{V}(Y|\mathbf{X} \in \mathbf{r}) = \mathbb{V}(g^*(\mathbf{X})|\mathbf{X} \in \mathbf{r}) + \sigma^2, \qquad \forall \mathbf{r} \subseteq \mathbb{R}^d. \tag{2.25}$$

Let us first remark that

$$\begin{aligned}
\mathbb{W}_n &= \mathbb{E}\left[\left(\sum_{A \in \mathcal{P}(\mathcal{C}_n^s) \cup \mathcal{P}(\mathcal{C}_n^{i\setminus s})} \mathbb{E}\left[Y \mid \mathbf{X} \in A\right] \mathbf{1}_A(\mathbf{X}) - g^*(\mathbf{X})\right)^2\right] \\
&= \sum_{A' \in \mathcal{P}(\mathcal{C}_n^s) \cup \mathcal{P}(\mathcal{C}_n^{i\setminus s})} \mathbb{E}\left[\left(\sum_{A \in \mathcal{P}(\mathcal{C}_n^s) \cup \mathcal{P}(\mathcal{C}_n^{i\setminus s})} \mathbb{E}\left[g^*(\mathbf{X}) \mid \mathbf{X} \in A\right] \mathbf{1}_A(\mathbf{X}) - g^*(\mathbf{X})\right)^2 \mathbf{1}_{A'}(\mathbf{X})\right] \\
&= \sum_{A' \in \mathcal{P}(\mathcal{C}_n^s) \cup \mathcal{P}(\mathcal{C}_n^{i\setminus s})} \mathbb{E}\left[\left(\mathbb{E}\left[g^*(\mathbf{X}) \mid \mathbf{X} \in A'\right] - g^*(\mathbf{X})\right)^2 \mathbf{1}_{A'}(\mathbf{X})\right] \\
&= \sum_{A' \in \mathcal{P}(\mathcal{C}_n^s) \cup \mathcal{P}(\mathcal{C}_n^{i\setminus s})} \mathbb{E}\left[\left(\mathbb{E}\left[g^*(\mathbf{X}) \mid \mathbf{X} \in A'\right] - g^*(\mathbf{X})\right)^2 \middle| \mathbf{X} \in A'\right] \times \mathbb{P}\left(\mathbf{X} \in A'\right)
\end{aligned} \tag{2.26}$$

which shows that $\mathbb{W}_n$ is a within-group variance for the variable $g^*(\mathbf{X})$ and the groups $\mathcal{P}(\mathcal{C}_n^s) \cup \mathcal{P}(\mathcal{C}_n^{i\setminus s})$.

According to the decomposition

$$\begin{aligned}
\mathbb{W}_n = &\underbrace{\sum_{A \in \mathcal{P}(\mathcal{C}_n^s)} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X} \in A\right] - g^*(\mathbf{X})\right)^2 \middle| \mathbf{X} \in A\right] \mathbb{P}\left(\mathbf{X} \in A\right)}_{\mathbb{W}_n^s} \\
&+ \underbrace{\sum_{A \in \mathcal{P}(\mathcal{C}_n^{i\setminus s})} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X} \in A\right] - g^*(\mathbf{X})\right)^2 \middle| \mathbf{X} \in A\right] \mathbb{P}\left(\mathbf{X} \in A\right)}_{\mathbb{W}_n^{i\setminus s}},
\end{aligned}$$

it is sufficient to prove that $\mathbb{W}_n^s \xrightarrow[n\to\infty]{\mathbb{P}} 0$ and $\mathbb{W}_n^{i\setminus s} \xrightarrow[n\to\infty]{\mathbb{P}} 0$.

To deal with $\mathbb{W}_n^s$, we start from the decomposition of the total variance into the within-group and the between-group variances:

$$\begin{aligned}
\mathbb{W}_n^s &= \mathbb{E}[(g^*(\mathbf{X}) - \mathbb{E}[g^*(\mathbf{X})])^2 \mathbf{1}_{\mathbf{c}_n^s}(\mathbf{X})] - \mathbb{B}_n^s \\
&\leq \mathbb{V}(g^*(\mathbf{X}) - \mathbb{B}_n^s
\end{aligned} \tag{2.27}$$

where

$$\mathbb{B}_n^s := \sum_{A \in \mathcal{P}(\mathcal{C}_n^s)} \left(\mathbb{E}\left[g^*(\mathbf{X}) \mid \mathbf{X} \in A\right] - \mathbb{E}\left[g^*(\mathbf{X})\right]\right)^2 \mathbb{P}\left(\mathbf{X} \in A\right).$$

The between group variance $\mathbb{B}_n^s$ will be lower estimated by a function of $\mathbb{W}_n^s$ from which a lower bound of $\mathbb{W}_n^s$ will follow. The key point of this lower bound is to use the definition of $\mathcal{C}_n^s$ in (H4). To make the terms controlled by this significant condition appear, we lower bound $\mathbb{B}_n^s$ by a sum over the elements of $\mathcal{C}_n^s$ instead of elements of $\mathcal{P}(\mathcal{C}_n^s)$ and the expectations is replaced by empirical expectations. After applying the inequality provided by the significant condition we'll have to do the reverse operation to make $\mathbb{W}_n^s$ appear again.

Let us first lower bound $\mathbb{B}_n^s$ by a sum over the elements of $\mathcal{C}_n^s$ instead of elements of $\mathcal{P}(\mathcal{C}_n^s)$ and replace the expectations by empirical expectations:

$$\begin{aligned}
\mathbb{B}_n^s &= \sum_{\mathbf{r}\in\mathcal{C}_n^s} \sum_{A\in\mathcal{P}_{\mathcal{C}_n^s}(\mathbf{r})} \frac{1}{\#\varphi_{\mathcal{C}_n^s}(A)} \left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right] - \mathbb{E}\left[g^*(\mathbf{X})\right]\right)^2 \mathbb{P}\left(\mathbf{X}\in A\right) \\
&\geq \sum_{\mathbf{r}\in\mathcal{C}_n^s} \frac{1}{M(\mathcal{C}_n^s,\mathbf{r})} \sum_{A\in\mathcal{P}_{\mathcal{C}_n^s}(\mathbf{r})} \left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right] - \mathbb{E}\left[g^*(\mathbf{X})\right]\right)^2 \mathbb{P}\left(\mathbf{X}\in A\right) \\
&\geq \frac{1}{M(\mathcal{C}_n^s)} \sum_{\mathbf{r}\in\mathcal{C}_n^s} \sum_{A\in\mathcal{P}_{\mathcal{C}_n^s}(\mathbf{r})} \left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right] - \mathbb{E}\left[g^*(\mathbf{X})\right]\right)^2 \mathbb{P}\left(\mathbf{X}\in A \mid \mathbf{X}\in\mathbf{r}\right)\mathbb{P}\left(\mathbf{X}\in\mathbf{r}\right) \\
&\geq \frac{1}{M(\mathcal{C}_n^s)} \sum_{\mathbf{r}\in\mathcal{C}_n^s} \left(\sum_{A\in\mathcal{P}_{\mathcal{C}_n^s}(\mathbf{r})} \mathbb{E}\left[Y \mid \mathbf{X}\in A\right]\mathbb{P}\left(\mathbf{X}\in A \mid \mathbf{X}\in\mathbf{r}\right) - \mathbb{E}\left[g^*(\mathbf{X})\right] \right)^2 \mathbb{P}\left(\mathbf{X}\in\mathbf{r}\right) \\
&= \frac{1}{M(\mathcal{C}_n^s)} \sum_{\mathbf{r}\in\mathcal{C}_n^s} \Big(\mathbb{E}\left[Y \mid \mathbf{X}\in\mathbf{r}\right] - \mathbb{E}\left[g^*(\mathbf{X})\right]\Big)^2 \mathbb{P}\left(\mathbf{X}\in\mathbf{r}\right) \\
&\geq \frac{1}{M(\mathcal{C}_n^s)} \sum_{\mathbf{r}\in\mathcal{C}_n^s} \left(V_{n,\mathbf{r}\times\mathbb{R}}^2 \ - \Delta_n\right) \mathbb{P}(\mathbf{X}\in\mathbf{r}) \qquad (2.28)
\end{aligned}$$

where we applied Jensen's inequality for the third to last inequality and where

$$\Delta_n := \sup_{A\in\mathcal{B}_n} \{V_{n,A}^2 - U_A^2\}$$

with for any $A\in\mathcal{B}$,

$$\begin{aligned}
U_A &:= \mathbb{E}\left[Y \mid (\mathbf{X},Y)\in A\right] - \mathbb{E}\left[g^*(\mathbf{X})\right] \\
V_{n,A} &:= \mathbb{E}_n\left[Y \mid (\mathbf{X},Y)\in A\right] - \mathbb{E}_n\left[Y\right]
\end{aligned}$$

and $\mathcal{B}_n = \{A\in\mathcal{B} \text{ s.t. } \mathbb{Q}_n(A) > n^{-\alpha}\}$.

Continuing (2.28) with the definition of $\mathcal{C}_n^s$ in (H4) in mind,

$$\begin{aligned}
\mathbb{B}_n^s &\geq \frac{1}{M(\mathcal{C}_n^s)} \sum_{\mathbf{r}\in\mathcal{C}_n^s} \left(\beta_n^{-2}(\mathbb{V}_n(Y|\mathbf{X}\in\mathbf{r}) - \sigma_n^2) - \Delta_n\right) \mathbb{P}(\mathbf{X}\in\mathbf{r}) \\
&\geq \frac{1}{M(\mathcal{C}_n^s)} \sum_{\mathbf{r}\in\mathcal{C}_n^s} \left(\beta_n^{-2}(\mathbb{V}_n(Y|\mathbf{X}\in\mathbf{r}) - \sigma^2) - \beta_n^{-2}|\sigma_n^2 - \sigma^2| - \Delta_n\right) \mathbb{P}(\mathbf{X}\in\mathbf{r}).
\end{aligned}$$

Since $\mathbb{W}_n^s = \sum_{A\in\mathcal{P}_{\mathcal{C}_n}(\mathcal{C}_n^s)} \mathbb{V}(g^*(\mathbf{X})|\mathbf{X}\in A)\mathbb{P}\left(\mathbf{X}\in A\right)$, this last term can be lower bounded by $\mathbb{W}_n^s$ if the empirical variances are replaced by variances and the sum over $\mathcal{C}_n^s$ by a sum over $\mathcal{P}(\mathcal{C}_n^s)$. Let us then define

$$\Delta_n' := \sup_{A\in\mathcal{B}_n} \left\{\left|\mathbb{V}(Y|(\mathbf{X},Y)\in A) - \mathbb{V}_n(Y|(\mathbf{X},Y)\in A)\right|\right\}$$

and write

$$\begin{aligned}
\mathbb{B}_n^s &\geq \frac{1}{M(\mathcal{C}_n^s)} \sum_{\mathbf{r}\in\mathcal{C}_n^s} \left(\beta_n^{-2}(\mathbb{V}(Y \mid \mathbf{X}\in\mathbf{r}) - \sigma^2 - \Delta_n') - \beta_n^{-2}|\sigma_n^2 - \sigma^2| - \Delta_n\right) \mathbb{P}(\mathbf{X}\in\mathbf{r}) \\
&= \frac{1}{M(\mathcal{C}_n^s)} \sum_{\mathbf{r}\in\mathcal{C}_n^s} \left(\beta_n^{-2}(\mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X}\in\mathbf{r}) - \Delta_n') - \beta_n^{-2}|\sigma_n^2 - \sigma^2| - \Delta_n\right) \mathbb{P}(\mathbf{X}\in\mathbf{r}) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{by (2.25)} \\
&= \frac{\beta_n^{-2}}{M(\mathcal{C}_n^s)} \times \sum_{\mathbf{r}\in\mathcal{C}_n^s} \Big(\mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in\mathbf{r}\right] - g^*(\mathbf{X})\right)^2 \mid \mathbf{X}\in\mathbf{r}\right] \\
&\quad - (\Delta_n' + |\sigma_n^2 - \sigma^2| + \beta_n^2\Delta_n)\Big)\mathbb{P}(\mathbf{X}\in\mathbf{r}) \\
&\geq \frac{\beta_n^{-2}}{M(\mathcal{C}_n^s)} \times \sum_{\mathbf{r}\in\mathcal{C}_n^s} \sum_{A\in\mathcal{P}_{\mathcal{C}_n^s}(\mathbf{r})} \left(\mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in\mathbf{r}\right] - g^*(\mathbf{X})\right)^2 \mid \mathbf{X}\in A\right]\right) \mathbb{P}(\mathbf{X}\in A) \\
&\quad - (\beta_n^{-2}\Delta_n' + \beta_n^{-2}|\sigma_n^2 - \sigma^2| + \Delta_n) \qquad\qquad \text{since } \textstyle\sum_{\mathbf{r}\in\mathcal{C}_n^s} \mathbb{P}(\mathbf{X}\in\mathbf{r}) \leq M(\mathcal{C}_n^s) \\
&\geq \frac{\beta_n^{-2}}{M(\mathcal{C}_n^s)} \times \sum_{\mathbf{r}\in\mathcal{C}_n^s} \sum_{A\in\mathcal{P}_{\mathcal{C}_n^s}(\mathbf{r})} \left(\mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right] - g^*(\mathbf{X})\right)^2 \mid \mathbf{X}\in A\right]\right) \mathbb{P}(\mathbf{X}\in A) \\
&\quad - (\beta_n^{-2}\Delta_n' + \beta_n^{-2}|\sigma_n^2 - \sigma^2| + \Delta_n) \\
&\geq \beta_n^{-2}\frac{m(\mathcal{C}_n^s)}{M(\mathcal{C}_n^s)} \times \sum_{A\in\mathcal{P}(\mathcal{C}_n^s)} \left(\mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right] - g^*(\mathbf{X})\right)^2 \mid \mathbf{X}\in A\right]\right) \mathbb{P}(\mathbf{X}\in A) \\
&\quad - (\beta_n^{-2}\Delta_n' + \beta_n^{-2}|\sigma_n^2 - \sigma^2| + \Delta_n) \\
&= \beta_n^{-2}\frac{m(\mathcal{C}_n^s)}{M(\mathcal{C}_n^s)}\mathbb{W}_n^s - (\beta_n^{-2}\Delta_n' + \beta_n^{-2}|\sigma_n^2 - \sigma^2| + \Delta_n).
\end{aligned}$$

Together with (2.27), this yields

$$\mathbb{W}_n^s \leq \frac{\mathbb{V}(g^*(\mathbf{X})) + \beta_n^{-2}\Delta_n' + \beta_n^{-2}|\sigma_n^2 - \sigma^2| + \Delta_n}{1 + \beta_n^{-2}\frac{m(\mathcal{C}_n^s)}{M(\mathcal{C}_n^s)}}.$$

Under (H6), Corollary 1.4.1 applies and we obtain

$$\begin{aligned}
\Delta_n &= \sup_{A\in\mathcal{B}_n} \{(V_{n,A} - U_A)(V_{n,A} + U_A)\} \\
&\leq \sup_{A\in\mathcal{B}_n} \Big\{\left|\mathbb{E}_n[Y \mid (\mathbf{X},Y)\in A] - \mathbb{E}[Y \mid (\mathbf{X},Y)\in A]\right| \\
&\quad + \left|\mathbb{E}[g^*(\mathbf{X})] - \mathbb{E}_n[Y]\right|\Big\} \times 4L \\
&\leq 4L \times \Big(\sup_{A\in\mathcal{B}_n} \left|\mathbb{E}_n[Y|(\mathbf{X},Y)\in A] - \mathbb{E}[Y|(\mathbf{X},Y)\in A]\right| \\
&\quad + \left|\mathbb{E}[g^*(\mathbf{X})] - \mathbb{E}[g^*(\mathbf{X})]\right| + \Big|\underbrace{\mathbb{E}[g^*(\mathbf{X})]}_{\mathbb{E}[Y]} - \mathbb{E}_n[Y]\Big|\Big) \\
&\leq 4L \times \left(O_{\mathbb{P}^*}(n^{\alpha-1/2}) + O_{\mathbb{P}}(n^{-1/2})\right) \\
&= O_{\mathbb{P}^*}(n^{\alpha-1/2})
\end{aligned}$$

Corollary 1.4.1 also yields

$$\Delta_n' = O_{\mathbb{P}^*}(n^{\alpha-1/2}).$$

Since it is moreover assumed that $|\sigma_n^2 - \sigma^2| = O_{\mathbb{P}}(n^{\alpha-\frac{1}{2}})$, (2.13) of Condition (H10) and Remark 8 lead to

$$\mathbb{W}_n^s \xrightarrow[n\to\infty]{\mathbb{P}} 0.$$

To deal with $\mathbb{W}_n^{i\setminus s}$, remember (2.23) and write

$$\begin{aligned}
\mathbb{W}_n^{i\setminus s} &= \sum_{A\in\mathcal{P}(\mathcal{C}_n^{i\setminus s})} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right]-g^*(\mathbf{X})\right)^2 \mathbf{1}_A(\mathbf{X})\right] \\
&= \sum_{A\in\mathcal{P}(\mathcal{C}_n^{i})} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\setminus \mathbf{c}_n^s\right]-g^*(\mathbf{X})\right)^2 \mathbf{1}_{A\setminus\mathbf{c}_n^s}(\mathbf{X})\right] \\
&\leq \sum_{A\in\mathcal{P}(\mathcal{C}_n^{i})} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right]-g^*(\mathbf{X})\right)^2 \mathbf{1}_{A\setminus\mathbf{c}_n^s}(\mathbf{X})\right] \\
&\leq \sum_{A\in\mathcal{P}(\mathcal{C}_n^{i})} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right]-g^*(\mathbf{X})\right)^2 \mathbf{1}_{A}(\mathbf{X})\right] \\
&\leq \frac{1}{m(\mathcal{C}_n^i)}\sum_{\mathbf{r}\in\mathcal{C}_n^i}\sum_{A\in\mathcal{P}_{\mathcal{C}_n^i}(\mathbf{r})} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in A\right]-g^*(\mathbf{X})\right)^2 \mathbf{1}_{A}(\mathbf{X})\right] \\
&\leq \frac{1}{m(\mathcal{C}_n^i)}\sum_{\mathbf{r}\in\mathcal{C}_n^i}\sum_{A\in\mathcal{P}_{\mathcal{C}_n^i}(\mathbf{r})} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in \mathbf{r}\right]-g^*(\mathbf{X})\right)^2 \mathbf{1}_{A}(\mathbf{X})\right] \\
&\leq \frac{1}{m(\mathcal{C}_n^i)}\sum_{\mathbf{r}\in\mathcal{C}_n^i} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in \mathbf{r}\right]-g^*(\mathbf{X})\right)^2 \mathbf{1}_{\mathbf{r}}(\mathbf{X})\right] \\
&= \frac{1}{m(\mathcal{C}_n^i)}\sum_{\mathbf{r}\in\mathcal{C}_n^i} \mathbb{E}\left[\left(\mathbb{E}\left[Y \mid \mathbf{X}\in \mathbf{r}\right]-g^*(\mathbf{X})\right)^2 \middle| \mathbf{X}\in\mathbf{r}\right]\mathbb{P}\left(\mathbf{X}\in\mathbf{r}\right) \\
&= \frac{1}{m(\mathcal{C}_n^i)}\sum_{\mathbf{r}\in\mathcal{C}_n^i} \left(\mathbb{V}(Y|\mathbf{X}\in\mathbf{r})-\sigma^2\right)\mathbb{P}\left(\mathbf{X}\in\mathbf{r}\right) \qquad \text{by (2.24) and (2.25)} \\
&\leq \frac{1}{m(\mathcal{C}_n^i)}\sum_{\mathbf{r}\in\mathcal{C}_n^i} \left(\mathbb{V}_n(Y|\mathbf{X}\in\mathbf{r})-\sigma_n^2+|\sigma_n^2-\sigma^2|+\Delta_n'\right)\mathbb{P}\left(\mathbf{X}\in\mathbf{r}\right) \\
&\leq \frac{1}{m(\mathcal{C}_n^i)}\sum_{\mathbf{r}\in\mathcal{C}_n^i} \left(\varepsilon_n^2+|\sigma_n^2-\sigma^2|+\Delta_n'\right)\mathbb{P}\left(\mathbf{X}\in\mathbf{r}\right) \qquad \text{under (H4)} \\
&\leq \frac{M(\mathcal{C}_n^i)}{m(\mathcal{C}_n^i)}\left(\varepsilon_n^2+|\sigma_n^2-\sigma^2|+\Delta_n'\right) \xrightarrow[n\to\infty]{\mathbb{P}} 0 \\
&\qquad\qquad \text{under (2.14) of Condition (H10) and with Remark 8.}
\end{aligned}$$

2.3.4 Estimation Error

In this subsection we prove (2.21) using hypotheses (H7), (H8) and (H11).

Recall from (2.5) that G_n is the set of piecewise constant functions with values in $[-L, L]$ on the elements of the partition $\mathcal{P}(\mathcal{C}_n(D_n))$. Then, with the definition of Π_n in (H11) in mind,

$$\sup_{g\in G_n}\left|\mathcal{L}_n\left(g\right)-\mathcal{L}\left(g\right)\right| \leq \sup_{g\in\mathcal{G}_c\circ\Pi_n}\left|\mathcal{L}_n\left(g\right)-\mathcal{L}\left(g\right)\right|,$$

where $\mathcal{G}_c$ is the set of constant functions $\mathbb{R}^d \to [-L, L]$ and

$$\mathcal{G}_c\circ\Pi_n := \left\{g:\mathbb{R}^d\to\mathbb{R} : g=\sum_{A\in\mathcal{P}_n} f_A\mathbf{1}_A, \mathcal{P}_n\in\Pi_n, f_A\in\mathcal{G}_c\right\}.$$

The following is based on the same idea as [28, Theorem 13.1].

According to [28, Theorem 9.1 and Problem 10.4] we have,

$$\mathbb{P}\left\{\sup_{g\in\mathcal{G}_c\circ\Pi_n}|\mathcal{L}_n(g)-\mathcal{L}(g)|>\varepsilon\right\} \leq 8\mathbb{E}\left[\mathcal{N}_1\left(\frac{\varepsilon}{32L},\mathcal{G}_c\circ\Pi_n,\mathbf{X}_1^n\right)\right]\exp\left\{\frac{-n\varepsilon^2}{128.(4L^2)^2}\right\}, \quad (2.29)$$

where $\mathbf{X}_1^n=\{\mathbf{X}_1,\ldots,\mathbf{X}_n\}$.

Here $\mathcal{N}_1(\varepsilon,\mathcal{G}_c\circ\Pi_n,\mathbf{X}_1^n)$ is the random variable corresponding to the minimal number $N\in\mathbb{N}$ such that there exists functions $g_1,\ldots,g_N:\mathbb{R}^d\to[-L,L]$ with the property that for every $g\in\mathcal{G}_c\circ\Pi_n$ there is a $j\in\{1,...,N\}$ such that

$$\frac{1}{n}\sum_{i=1}^n|g(\mathbf{X}_i)-g_j(\mathbf{X}_i)|\leq\varepsilon.$$

This number is called the *ε-covering number* of $\mathcal{G}_c\circ\Pi_n$. It can be interpreted as the complexity of the class. Then using [28, Lemma 13.1] we have

$$\mathcal{N}_1\left(\frac{\varepsilon}{32L},\mathcal{G}_c\circ\Pi_n,\mathbf{X}_1^n\right) \leq \Delta(\Pi_n)\left\{\sup_{z_1,\ldots,z_m\in\{\mathbf{X}_1,\ldots,\mathbf{X}_n\},m\leq n}\mathcal{N}_1\left(\frac{\varepsilon}{32L},\mathcal{G}_c,z_1^m\right)\right\}^{\mathcal{M}(\Pi_n)},$$

According to [28, Lemma 9.2] for any set of function $\mathcal{G}$ and any sample z_1^m we have

$$\mathcal{N}_1\left(\frac{\varepsilon}{32L},\mathcal{G},z_1^m\right)\leq\mathcal{M}_1\left(\frac{\varepsilon}{32L},\mathcal{G},z_1^m\right),$$

where $\mathcal{M}_1(\varepsilon,\mathcal{G},z_1^m)$ is the maximal $N\in\mathbb{N}$ such that there exists functions $g_1,\ldots,g_N\in\mathcal{G}$ with

$$\frac{1}{n}\sum_{i=1}^m|g_j(z_i)-g_k(z_i)|\geq\varepsilon,$$

for all $1\leq j<k\leq N$. It is called L_1 *ε-packing* of $\mathcal{G}$ on z_1^m. See [28, Definition 9.4 (c)].

Now, from the definition of $\mathcal{G}_c$,

$$\sup_{z_1,\ldots,z_m\in\{\mathbf{X}_1,\ldots,\mathbf{X}_n\},m\leq n}\mathcal{M}_1(\varepsilon,\mathcal{G}_c,z_1^m)=\left\lceil\frac{2L}{\varepsilon}\right\rceil.$$

Finally,

$$\sup_{z_1,\ldots,z_m\in\{\mathbf{X}_1,\ldots,\mathbf{X}_n\},m\leq n}\mathcal{N}_1\left(\frac{\varepsilon}{32L},\mathcal{G}_c\circ\Pi_n,z_1^m\right)\leq\Delta(\Pi_n)\left\lceil\frac{64L^2}{\varepsilon}\right\rceil^{\mathcal{M}(\Pi_n)}. \quad (2.30)$$

According to (2.29) and (2.30) we have:

$$\mathbb{P}\left\{\sup_{g\in G_n}\left|\frac{1}{n}\sum_{i=1}^n|g(\mathbf{X}_i)-Y_i|^2-\mathbb{E}\left[|g(\mathbf{X})-Y|^2\right]\right|>\varepsilon\right\} \leq 8\Delta_n(\Pi_n)\left\lceil\frac{64L^2}{\varepsilon}\right\rceil^{\mathcal{M}(\Pi_n)}\exp\left(-\frac{n\varepsilon^2}{128.(4L^2)^2}\right)$$

and since

$$\begin{aligned}
&8\Delta_n(\Pi_n)\left\lceil\frac{64L^2}{\varepsilon}\right\rceil^{\mathcal{M}(\Pi_n)}\exp\left(-\frac{n\varepsilon^2}{2048L^4}\right)\\
&\leq 8\exp\left(\log\Delta_n(\Pi_n)+\mathcal{M}(\Pi_n)\log\left(\left\lceil\frac{64L^2}{\varepsilon}\right\rceil\right)-\frac{n\varepsilon^2}{2048.L^4}\right)\\
&\leq 8\exp\left(-\frac{n}{L^4}\left(\frac{\varepsilon^2}{2048}-\frac{\log\Delta_n(\Pi_n)L^4}{n}-\frac{\mathcal{M}(\Pi_n)L^4\log\left(\left\lceil\frac{64L^2}{\varepsilon}\right\rceil\right)}{n}\right)\right),
\end{aligned}$$

this concludes the proof of (2.21) and of Theorem 2.2.1.

2.4 Illustrations

In this section we propose a simple algorithm to generate a suitable sequence of data-dependent coverings using the Random Forests algorithm (RF) of [7] as rule generator. The interest is double; first, it shows that there exists a sequence of suitable data-dependent coverings in practice as in Definition 2.2.1. Second, it proves the consistency of a rule-based estimator generated from RF as soon as the condition (H11) on the complexity of RF is satisfied. Until now there are few results about the consistency of an estimator generated by RF, we may cite [14, 61].

Let $\mathcal{C}$ be the set of all hyperrectangles of $\mathbb{R}^d$ defined by

$$\mathcal{C} := \left\{ [\mathbf{a}, \mathbf{a}+\mathbf{h}] \subseteq \mathbb{R}^d : \mathbf{a} \in \mathbb{R}^d, \, \mathbf{h} \in \mathbb{R}_+^d \right\} \cup \{\emptyset\}.$$

Lemma 2.3.1 ensures that that any set of rules $\mathcal{C}_n$ such that $\mathcal{C}_n \subseteq \mathcal{C}$, which is the case of sets of rules generated by RF, fulfills (H12).

2.4.1 Covering Algorithm

The proposed algorithm generates an estimator using data-dependent coverings. It is decomposed into four steps.

1. Generation of RF with m_{tree} trees with a maximal depth controlled by a fixed maximal number of rules generated by RF [1].

2. Extraction of all significant and insignificant rules according to (H9) and (H10) from all nodes and leaves of all trees generated by RF that have a length lower than or equal to $k_{max} \in \{1, \dots, d\}$ for a chosen $\alpha \in (0, 1/2)$, $\beta_n = n^{\alpha/2-1/4}$, $\varepsilon_n = \beta_n \sigma_n(Y)$, where $\sigma_n(Y)$ is the empirical standard deviation of Y.

3. Selection of a minimal set of rules using Algorithm 3. A rule is added to the current set of rules if and only if it has at least a rate $\gamma \in (0, 1)$ of points not covered by the current set of rules. The smaller γ the smaller the number of rules $\#\mathcal{C}_n$ [2].

4. If the selected set of rules does not form a covering, generation of a unique *no-rule* that is one of the smallest hyperrectangles satisfying (H9) containing the remaining points.

Remark 11. We do not use step 4 in any of the examples treated below as we already obtain a covering at the third step. If the no-rule had to be added to ensure a covering, one should check that the no-rule is either a significant or an insignificant rule in order to obtain a consistent Covering Algorithm.

1. The redundancies conditions (2.13) and (2.14) are automatically controlled by the fixed maximal number of rules generated by RF.

2. This step was already described in Remark 5 in order to control the redundancy. Here it is only used for controlling the number of rules as the redundancy is already controlled by the maximal number of rules generated by RF.

Algorithm 3: Selection of minimal set of rules

Input:
— the rate $0 < \gamma < 1$;
— a set of significant rules S;
— a set of insignificant rules I;

Output:
— a minimal set of rules $\mathcal{C}_n$;

$\mathcal{C}_n \leftarrow \arg\max_{\mathbf{r}\in S} \mathbb{Q}_n(\mathbf{r})$;
$S \leftarrow S \setminus \mathcal{C}_n$;
while $\sum_{\mathbf{r}\in\mathcal{C}_n} \mathbb{Q}_n(\mathbf{r}) < 1$ **do**
  $\mathbf{r}^* \leftarrow \arg\max_{\mathbf{r}\in S} \mathbb{Q}_n(\mathbf{r})$;
  if $\mathbb{Q}_n(\mathbf{r}^* \cap \{\cup_{\mathbf{r}\in\mathcal{C}_n}\mathbf{r}\}) \leq \gamma\, \mathbb{Q}_n(\mathbf{r}^*)$ **then**
    $\mathcal{C}_n \leftarrow \mathcal{C}_n \cup \mathbf{r}^*$;
  $S \leftarrow S \setminus \mathbf{r}^*$;
  if $\#S = 0$ **then**
    Break ;
end
if $\sum_{\mathbf{r}\in\mathcal{C}_n} \mathbb{Q}_n(\mathbf{r}) < 1$ **then**
  while $\sum_{\mathbf{r}\in\mathcal{C}_n} \mathbb{Q}_n(\mathbf{r}) < 1$ **do**
    $\mathbf{r}^* \leftarrow \arg\min_{\mathbf{r}\in I} \mathbb{V}_n(Y|\mathbf{X}\in\mathbf{r})$;
    if $\mathbb{Q}_n(\mathbf{r}^* \cap \{\cup_{\mathbf{r}\in\mathcal{C}_n}\mathbf{r}\}) \leq \gamma\, \mathbb{Q}_n(\mathbf{r}^*)$ **then**
      $\mathcal{C}_n \leftarrow \mathcal{C}_n \cup \mathbf{r}^*$;
    $I \leftarrow I \setminus \mathbf{r}^*$;
    if $\#I = 0$ **then**
      Break ;
  end
return $\mathcal{C}_n$;

2.4.2 Artificial data

Here we consider the same data set as in [20]. We generate $n = 5,000$ data following the regression setting

$$Y = g^*(\mathbf{X}) + Z,$$

where $d = 100$ (the dimension of $\mathbf{X}$) and

$$g^*(\mathbf{X}) = 9\prod_{j=1}^{3}\exp\left(-3\,(1-X_j)^2\right) - 0.8\exp\left(-2\,(X_4 - X_5)\right) + 2\sin^2(\pi\cdot X_6) - 2.5\,(X_7 - X_8)\,, \quad (2.31)$$

and $Z \sim \mathcal{N}(0,\sigma^2)$. The value of $\sigma > 0$ was chosen to produce a two-to-one signal-to-noise ratio. The variables were generated from a uniform distribution on $\{0/10,\ldots,9/10\}$. It is important to notice that only the first eight variables are informative; the 92 others are just noise. Coefficients multiplying each of the terms in g^* were chosen to ensure that variables have approximately equal influence.

In order to evaluate the error without including the error of the noise, we consider the following criterion

$$MSE^* = \mathbb{E}_{\mathbb{Q}}\left[|g^*(\mathbf{X}) - g_n(\mathbf{X})|^2\right].$$

We approximate it using $50,000$ test observations sampled independently from $\mathbb{Q}$.

Execution

We run $M = 100$ simulations. For each simulation we set the maximal number of rules generated by the Random Forest (RF) at $20,000$ and the number of trees at $m_{tree} = 100$. The maximal length of a rule is fixed at $k_{max} = 3$. And we set $\alpha = 1/2 - 1/100$ and $\gamma = 0.95$. We compare results with the ones of RF and RuleFit [20].

Results

We calculate the MSE^* of each algorithm for each experience and sum up it in Figure 2.3. All results are described in Table 2.1.

Figures 2.4 and 2.5 display the frequency of occurrence of a variable in at least one rule of the selected set of rules and the average occurrence of a variable in the selected set of rules respectively. We emphasize that only the informative variables are involved in the selected rules (excepted X_{41} which appears one time in 100 simulations). It means that the support of g^* is well identified.

Nevertheless, the linear term in (2.31) cannot be well fitted by a finite set of rules. This setting is highly favorable to RuleFit that is the only tested algorithm including linear components. However, RuleFit involves at least 10 and up to 40 linear components (see Table 2.1). Its accuracy is high but many noise variables are included in its generated model. Hence the a posteriori analysis of the importances of the variables and rules of the generated model is crucial, see Section 9.1.2 of [20].

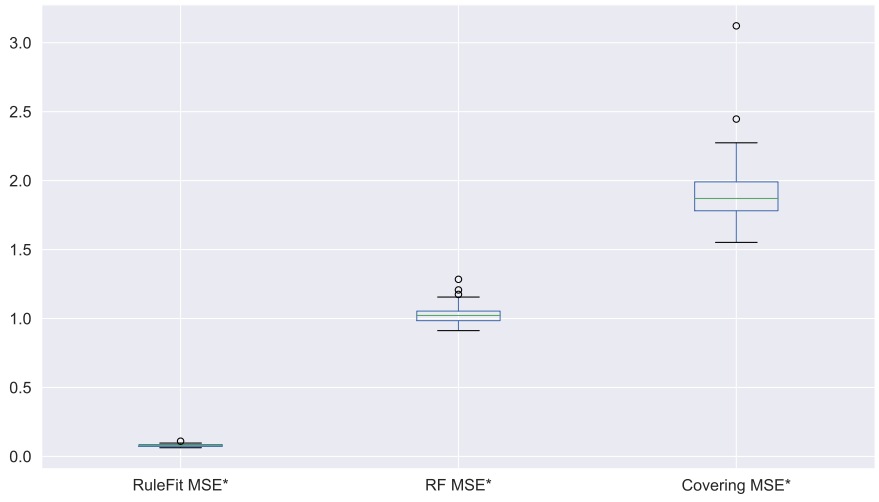


Figure 2.3 – Box-plot of the MSE^* of each algorithm.

Random Forest	Nb rules	Interpretability	MSE^*
mean	12595.96	60835.88	1.03
std	3.90	339.69	0.07
min	12582.00	59826.00	0.91
25%	12594.00	60652.50	0.98
50%	12598.00	60892.00	1.02
75%	12598.00	61100.00	1.05
max	12600.00	61380.00	1.28

Covering Algorithm	Nb rules	Interpretability	MSE^*
mean	33.56	100.11	1.90
std	3.47	10.52	0.20
min	25.00	74.00	1.55
25%	31.00	93.00	1.78
50%	34.00	101.50	1.87
75%	36.00	108.00	1.99
max	41.00	122.00	3.12

RuleFit	Nb Rules	Nb linear	Interpretability	MSE^*
mean	185.27	25.11	453.98	0.08
std	19.81	6.76	59.13	0.01
min	147.00	10.00	341.00	0.06
25%	171.00	20.75	407.75	0.07
50%	184.50	25.00	450.00	0.08
75%	199.00	29.00	496.00	0.09
max	271.00	40.00	719.00	0.11

Table 2.1 – Summary of the M experiences for each algorithm. Nb Rules is the number of rules, Nb Linear (for RuleFit) is the number of linear components in the generated model and Interpretability is the interpretability index.

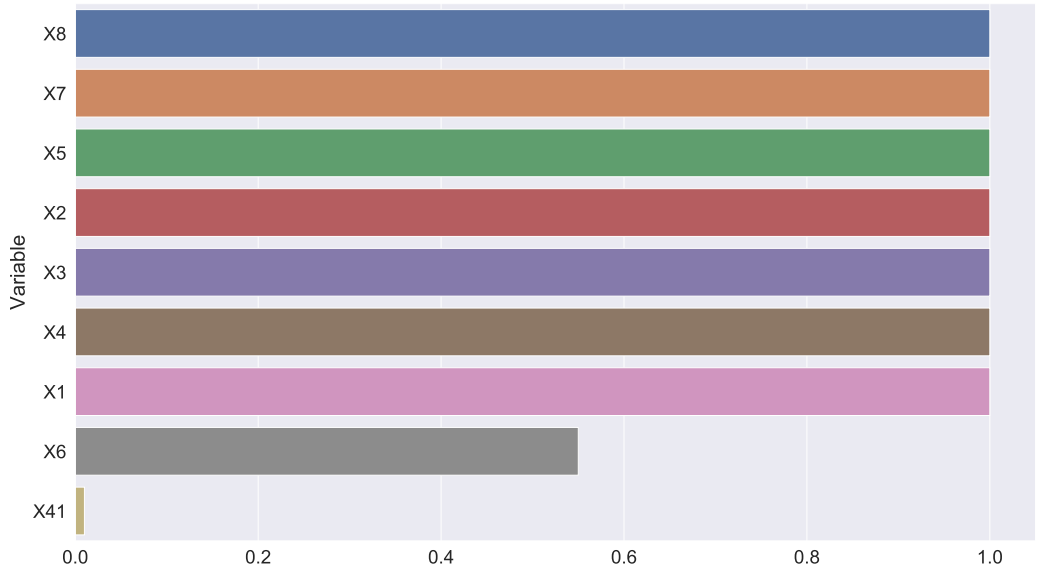


Figure 2.4 – Empirical probability of occurrence in at least one rule of the selected set of rules generated by the Covering Algorithm.

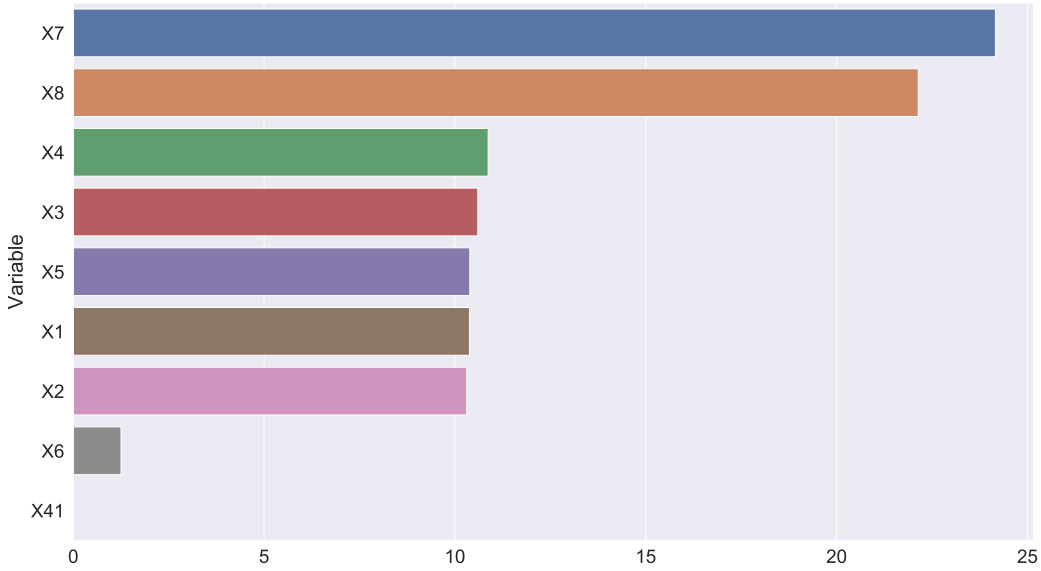


Figure 2.5 – Average occurrence in the selected set of rules generated by the Covering Algorithm.

2.4.3 Real data

In this example we use the well-known *Boston housing* dataset. It consists in $n = 506$ neighborhoods in the Boston area and 14 statistics for each neighborhood describes in Table 2.2 (see Table IV p96-97 [29] for more details about the variables). The regression model explains the median house price Y of each neighborhood by the $d = 13$ others statistics of this neighborhood. The data are randomly split into a training set and a test set, with a ratio of 70% / 30%, respectively.

	Y	CRIM	ZN	INDUS	CHAS	NOX	RM
mean	22.53	3.61	11.36	11.14	0.069	0.55	6.28
std	9.19	8.60	23.32	6.86	0.25	0.11	0.70
min	5.00	0.006	0.0	0.46	0.0	0.38	3.56
25%	17.02	0.08	0.0	5.19	0.0	0.44	5.88
50%	21.20	0.25	0.0	9.69	0.0	0.53	6.20
75%	25.00	3.67	12.50	18.10	0.0	0.62	6.62
max	50	88.97	100.0	27.74	1.0	0.87	8.78

	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
mean	68.57	3.79	9.54	408.23	18.45	356.67	12.65
std	28.14	2.10	8.70	168.53	2.16	91.29	7.14
min	2.90	1.12	1.0	187.0	12.6	0.32	1.73
25%	45.02	2.10	4.0	279.0	17.40	375.37	6.95
50%	77.50	3.20	5.0	330.0	19.0	391.44	11.36
75%	94.07	5.18	24.0	666.0	20.20	396.22	16.95
max	100.0	12.12	24.00	711.0	22.0	396.90	37.97

Table 2.2 – Description of variables.

Execution

We set the maximal number of rules generated by the Random Forest (RF) at $20,000$ and the number of trees at $m_{tree} = 100$. It generates $8,490$ rules and the RF estimator has an interpretability index of $Int(g_n) = 29,524$ according to (2.8). Among these rules $4,066$ have a length lower than or equal to $k_{max} = 3$. In this application the real value of σ^2 is unknown. We estimate it by σ_n^2 the minimal variance of the rules fulfilling the covering condition (2.9). We have $\sigma_n^2 = 0.815$. Considering $\sigma^2 = \sigma_n^2$ and $\alpha = 1/2 - 1/100$, the Covering Algorithm described in Section 2.4.1 extracts $1,306$ rules that are significant according to (2.11) and 2.083 that are insignificant according to (2.12). Then, the selection process (Algorithm 3), with $\gamma = 0.95$, extracts a set of 6 significant rules, which cover 99% of the data, and adds 2 insignificant rules $R4$ and $R7$ to cover all data. The interpretability index obtained by the covering estimator is $Int(g_n) = 20$. We compare results with the ones of RF and RuleFit.

Results

Regarding the accuracy on the test set, RF has a MSE of 10.47 and Covering Algorithm has a MSE of 25.35. This loss of accuracy is the price of turning a black box model into an interpretable one. According to (2.8) we have $Int(g_n) = 20$ for the Covering Algorithm and $Int(g_n) = 29524$ for the RF, which is a huge improvement of the interpretability index.

Rule	Conditions	Coverage	Prediction	Variance
$R1$	$LSTAT \in [9.62, 37.97]$	0.58	17.74	26.03
$R2$	$LSTAT \in [1.73, 7.87]$	0.31	32.31	78.57
$R3$	$RM \in [3.86, 6.10]$ $LSTAT \in [7.76, 37.97]$ $DIS \in [1.20, 12.13]$	0.37	17.62	22.52
$R4$	$RM \in [6.10, 6.54]$ $LSTAT \in [1.73, 9.98]$ $DIS \in [1.42, 12.13]$	0.12	23.39	4.62
$R5$	$RM \in [6.42, 8.78]$ $LSTAT \in [1.73, 14.40]$ $DIS \in [1.20, 12.13]$	0.33	31.88	72.88
$R6$	$RM \in [6.09, 8.78]$ $LSTAT \in [1.73, 9.98]$ $CRIM \in [0.05, 88.98]$	0.27	32.12	76.50
$R7$	$RM \in [3.86, 6.12]$ $LSTAT \in [8.13, 14.40]$ $TAX \in [187.0, 298.0]$	0.08	21.17	3.69
$R8$	$LSTAT \in [5.06, 37.97]$ $PTRATIO \in [13.90, 22.0]$ $INDUS \in [15.01, 27.74]$	0.36	16.34	35.70

Table 2.3 – Summary of the selected rules generated by Covering Algorithm.

Algorithm	Nb rules	Interpretability	MSE
Random Forest	8490	29524	10.47
Covering Algorithm	8	20	25.35
RuleFit	602 and 1 linear component	602	10.46

Table 2.4 – Comparison between Random Forest, Covering Algorithm and RuleFit on real data.

Comments

This application on real data emphasizes that data-dependent coverings are very efficient to generate an interpretable rule-based model. Using Table 2.2 we are able to translate in natural language the significant rules in Table 2.3, accordingly with the definition of interpretability of [6].

— Rules $R1$ and $R2$ indicate that the $LSTAT$ variable, which represents the percent of lower status population, is inversely related to the median house price. This relation is easily identified by the linear component of RuleFit, which corresponds to the unique variable $LSTAT$ (See Table 2.4).
— Rules $R3$, $R5$ and $R6$ show that for $LSTAT$ close to the median, the number of rooms RM is determining the median house price.
— Rule $R8$ can be interpreted by the fact that a very high proportion of non-retail business acres per town $INDUS$ has a negative influence on the median house price Y that cannot be offset by a percentage of the lower status $LSTAT$ and a pupil-teacher ratio $PTRATIO$ lower than the first quartile.

The variables DIS and $CRIM$ seem superfluous as the associated rules $R3$, $R5$ and $R6$ cover most of their range.

2.5 Conclusion

In this chapter, we provide a general setting for studying the consistency of interpretable rule-based estimators. The novelty is to introduce the notion of covering composed by two kinds of sets, the significant and the insignificant ones. The significant sets are thought as interpretable sets by construction. The insignificant ones are thought as small sets in which variances tend to zero. We provide Covering Algorithm that extracts from any rule generator a suitable data-dependent covering. We apply it to RF and we compare its result with the ones of RF and RuleFit [20]. The loss of accuracy in the prediction is the cost to pay to have an interpretable model according to our definition of interpretability. Monte Carlo experiments in section 2.4.2 show that Covering Algorithm, seeking interpretability, is identifying the support of the regression function as a byproduct. The loss of accuracy of Covering Algorithm may be due to the conflict between model identification and regression estimation identified by [76]. It is worthwhile to mention that despite its loss of accuracy, Covering Algorithm is still weakly consistent.

Our methodology based on coverings is very effective for generating interpretable models. The use of RF as the rule-generator of Covering Algorithm is questionable; in Section 2.4.2, we pointed out the possible negative effect of the randomization procedure in RF for identifying informative variables. In Section 2.4.3, we observed some rules generated by RF with length potentially artificially too high. In Chapter 3, we present an algorithm that generates significant, insignificant rules and a suitable sequence of data-dependent coverings satisfying (H11) on its own. The difficulty we encounter is to ensure that the generated rules cover all the data and the adaptation in cases of unknown variance estimated as in Section 2.4.3.

"Education is an admirable thing, but it is well to remember from time to time that nothing that is worth knowing can be taught."
Oscar Wilde.

Chapter 3

Rule Induction Covering Estimator: A New Data Dependent Covering Algorithm

3.1 Introduction

The massive use of machine learning algorithms in current life and sensitive fields such as medicine, criminal justice system, autonomous cars, asset management, has caused users to ask questions on the used algorithms functionalities and how to understand their decisions. Indeed, as related in [79], using algorithms with incomprehensible models output may be risky even if they are very efficient. This kind of algorithms are called black-box models because their internal workings are unknown. To avoid these risks, some suggest to use interpretable algorithms. But, as explained in [43], there are several meanings of interpretability depending on the users desiderata and the expected properties of algorithms. For example an algorithm used in criminal justice system must be *ethic* as explain in [15]. In this chapter, we use the definition of model interpretability from [6]: **Interpretability is the degree to which an observer can understand the cause of a decision.**

There are two common solutions to generate an interpretable model: *intrinsic* interpretable algorithms and *post-hoc* interpretable algorithms. The post-hoc interpretability consists in considering a black-box algorithm and trying to interpret it using algorithms which measure the importance of each explanatory variables (features) in the decision. The most popular ones are *LIME* [58], *DeepLIFT* [62] or *SHAP* [45]. See [27] for a survey of existing methods. The intrinsic interpretability, consisting in considering intrinsically interpretable algorithms. It means algorithms which output models easy to understand for humans. There are two common families of algorithms: tree-based algorithms, and rule-based algorithms. A tree may always be turned into a set of rules as in [Quinlan et al.] and [38]. Hence, without loss of generality we focus on set of rules generator algorithms. On a feature space $\mathbb{R}^d$, a *rule* is an *If-Then* statement of the form:

$$\begin{aligned} \text{IF} \quad & x \in \mathbf{r} \\ \text{THEN} \quad & \text{Prediction} = p \end{aligned} \tag{3.1}$$

where $\mathbf{r} \subset \mathbb{R}^d$. In this chapter all $\mathbf{r}$ considered are hyperrectangles: $\mathbf{r} = c_1 \times \cdots \times c_d$, where for all j, $c_j \subseteq \mathbb{R}$ is an interval.

The *If* part, called the *condition* of the rule $\mathbf{r}$, is then composed of $l \leq d$ tests, each of which checking whether a feature (a coordinate of $\mathbf{x}$) satisfies a specified property or not and k (the number of c_j's which are not $\mathbb{R}$) is called the *length* of the rule and is denoted

by $\ell(\mathbf{r})$:

$$\ell(\mathbf{r}) = d - \#\{1 \leq j \leq d; c_j = \mathbb{R}\}. \tag{3.2}$$

The *Then* part, called the *conclusion* of the rule, is the predicted value when the rule is *activated*, i.e., when the condition in the *If* part is satisfied. The rules are easy to understand and allow an interpretable decision process when l is small. For a review of the best-known algorithms for descriptive and predictive rule learning, see [80] and [23].

By an abuse of notation we do not distinguish between a rule and the hyperrectangle which defines its condition in the following.

The main issue is that a rule is interpretable only if the condition is easy to understand. For hyperrectangle cells, this means that each generated rule has few features implied in its condition. For instance, tree-based algorithms such as *CART* [8], *ID3* [55], *C4.5* [56], are interpretable if their maximal depth is small. But theoretical results such as Stone's Theorem (65 or 28, Theorem 13.1) requires a shrinkage, in probability, of the cells of the generated partition. Then the depths of the trees have to equal the dimension of the feature space for large enough sample sizes. Thus, each rule has a condition implying all features and it is uninterpretable if the dimension is high. In order to make this argument rigorous, we define the interpretability index of a rule based algorithm as follows.

Definition 3.1.1. *The interpretability index Int of a rule-based algorithm is the sum of the lengths of the rules involved in the predictor.*

That one can construct cells implying all features by intersecting d rules of unit length motivates the definition of Int. We say that an algorithm is interpretable when Int remains reasonable. Tree-based algorithms that need many trees with shrinking diameters (in probability) cannot be interpretable according to our definition.

Using rule-based algorithms for a regression is challenging and not been well investigated. There are few algorithm and none of them has had a significant impact. We may cite *FORS* [37] or CUBIST. Most of rule-based algorithms used for a regression setting, discretize the variable Y to bring back to a classification setting. Recently the focus is on the ensemble techniques such as *RuleFit* [20], *Ender* [13].

In this chapter we present a deterministic predictive rule-based algorithm for regression settings that generates highly interpretable models and predictors that are accurate with theoretical guarantee as consistency. This algorithm is an update of the *RIPE* algorithm presented in Chapter 5 based on the *data-dependent coverings* introduced in Chapter 2. In Chapter 2 we have presented an extraction of rules generated by Random Forests [7].

3.2 Preliminaries

3.2.1 Regression setting

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and let $(\mathbf{X}, Y)$ be random variables in $\mathbb{R}^d \times \mathbb{R}$ of unknown distribution $\mathbb{Q}$. We consider the following regression setting

$$Y = g^*(\mathbf{X}) + Z,$$

where $\mathbb{E}[Z] = 0$, $\mathbb{V}(Z) = \sigma^2$ and g^* is a measurable function from $\mathbb{R}^d$ to $\mathbb{R}$. We assume that Z is independent of $\mathbf{X}$ and Y is bounded in absolute value by some unknown $L > 0$. Contrary to the regression setting in Chapter 2 σ^2 is not assumed to be known.

Given a sample $D_n = ((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n))$ where the observations $(\mathbf{X}_i, Y_i)$ are independent and identically distributed (i.i.d.) from the distribution $\mathbb{Q}$, we aim at predicting Y conditionally on $\mathbf{X}$.

To do so, we use a learning algorithm which provides from D_n a random function g: $\mathbb{R}^d \to [-L, L]$ called a *predictor*. Its accuracy is measured by its quadratic risk, defined as

$$\mathcal{L}(g) = \mathbb{E}\left[(g(\mathbf{X}) - Y)^2\right]. \tag{3.3}$$

Its empirical risk is defined by

$$\mathcal{L}_n(g) = \frac{1}{n}\sum_{i=1}^{n}(g(\mathbf{X}_i) - Y_i)^2. \tag{3.4}$$

Learning algorithms use the *Empirical Risk Minimization* (ERM) principle [74] to generate the predictor. It means that a predictor g_n generated by a learning algorithm fulfills

$$g_n \in \arg\min_g \mathcal{L}_n(g) \tag{3.5}$$

where the arg min is taken among the set of all predictors that the algorithm can generate from D_n.

3.2.2 Regression based on coverings

Usually, interpretable predictive algorithms are based on splitting the feature space into a partition where each cell describes a local phenomenon. In other words, they generate a partition $\mathcal{P}_n$ from the dataset D_n and build an estimator using the empirical risk minimization. So, for a given partition $\mathcal{P}_n$ the predictor is defined as

$$g_n(\mathbf{x}) = \sum_{A\in\mathcal{P}_n} \frac{\sum_{i=1}^n Y_i \mathbf{1}_{\mathbf{X}_i\in A}}{\sum_{i=1}^n \mathbf{1}_{\mathbf{X}_i\in A}} \mathbf{1}_{\mathbf{x}\in A}\,, \qquad \mathbf{x}\in\mathbb{R}^d. \tag{3.6}$$

In Section 3.3, we discuss the case 0/0.

In this chapter, we present an algorithm based on a *data-dependent covering*. The idea is to generate, from the dataset D_n, a covering $\mathcal{C}_n = \mathcal{C}_n(D_n)$ of $\mathbb{R}^d$ instead of a partition. This covering is turned into a partition $\mathcal{P}(\mathcal{C}_n)$ using Definition 2.1.1. We have illustrated this transformation $\mathcal{P}$ on a covering $\mathcal{C}$ of five rectangles in Figures 1.7 and 1.8.

Hence we have the predictor

$$g_n(\mathbf{X}) = \sum_{A\in\mathcal{P}(\mathcal{C}_n)} \frac{\sum_{i=1}^n Y_i \mathbf{1}_{X_i\in A}}{\sum_{i=1}^n \mathbf{1}_{X_i\in A}} \mathbf{1}_{\mathbf{X}\in A}\,, \qquad \mathbf{X}\in\mathbb{R}^d, \tag{3.7}$$

This definition is based on the ERM principle among the piecewise constant predictors on $\mathcal{P}(\mathcal{C}_n)$. However the predictors from Random Forests (RF) and other random rule algorithms do not satisfy the ERM principle (3.5): these are based on the averaging of the activated rules at the point $\mathbf{x}$. This averaging is justified empirically and theoretically as it lowers the variance of the prediction thanks to the independence of the random rules as explained by [7]. The averaging does not decrease the approximation error of the prediction. On the opposite, our prediction following the relation (3.7) is the empirical conditional expectation of the intersection of the activated rules. Under some conditions, the approximation error of the prediction is then decreased but not the variance. To control the variance, we only consider coverings of rules that are *significant* (see Definition 1.4.3 below) as in the RIPE algorithm of [46]. But in this former work the significant test to select rules does not allow to control the approximation error.

3.2.3 Approximately suitable data-dependent coverings

In order to ensure the consistency of data-dependent covering algorithm, [47] have introduced the notion of *suitable* sequence of data-dependent coverings. This definition used the notations of empirical measures for any $\mathbf{r} \subseteq \mathbb{R}^d$ such that

$$\mathbb{Q}_n(\mathbf{r} \times \mathbb{R}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\mathbf{X}_j \in \mathbf{r}} > 0.$$

To simplify the notation we use $\mathbb{Q}_n(\mathbf{r})$ instead of $\mathbb{Q}_n(\mathbf{r} \times \mathbb{R})$ when no ambiguity is possible.

We also define

$$\mathbb{E}_n[Y \mid \mathbf{X} \in \mathbf{r}] := \frac{\sum_{i=1}^{n} Y_i \mathbf{1}_{\mathbf{X}_i \in \mathbf{r}}}{\sum_{i=1}^{n} \mathbf{1}_{\mathbf{X}_j \in \mathbf{r}}}$$

the empirical conditional expectation and

$$\mathbb{V}_n(Y \mid \mathbf{X} \in \mathbf{r}) := \mathbb{E}_n\left[Y^2 \mid \mathbf{X} \in \mathbf{r}\right] - \mathbb{E}_n\left[Y \mid \mathbf{X} \in \mathbf{r}\right]^2$$

the empirical conditional variance.

A suitable sequence of data-dependent coverings, Definition 2.2.1, satisfies the coverage and the significance conditions. The first one controls the minimal number of observations in a rule, ensuring that its prediction based on the empirical conditional expectation is accurate. The second one controls the variance of the rules in two different ways; the significant rules are predicting significantly differently than the average and may have a large variance, the insignificant rules are not significant and have a small variance.

The main drawback of Definition 2.2.1 relies on the use of the parameter σ^2, which is unknown in practice. A simple solution is to replace σ^2 by any consistent estimator of σ_n^2 in the definition of (in)significant rules (3.12) and (3.13). See Section 3.5.7 for such an estimator.

Definition 3.2.1. *We call a sequence $(\hat{\mathcal{C}}_n)_{n\geq 1}$ of data-dependent collections of sets of $\mathbb{R}^d$* ***approximately suitable*** *if it satisfies the two following conditions:*

1. *the* ***coverage condition****:*

$$\exists \alpha \in [0, 1/2),\ \forall \mathbf{r} \in \hat{\mathcal{C}}_n,\ \mathbb{Q}_n(\mathbf{r}) > n^{-\alpha} \quad \mathbb{P} - a.s, \tag{3.8}$$

   for n sufficiently large;

2. *the* ***approximately significance condition****: there exist two sequences $\beta_n \to 0$ and $\varepsilon_n \to 0$ such that:*

$$\hat{\mathcal{C}}_n = \hat{\mathcal{C}}_n^s \cup \hat{\mathcal{C}}_n^i \quad \mathbb{P} - a.s, \tag{3.9}$$

   for n sufficiently large, where the approximately significant subsets $\hat{\mathcal{C}}_n^s$ are defined by

$$\hat{\mathcal{C}}_n^s := \left\{\mathbf{r} \in \hat{\mathcal{C}}_n : \beta_n \left|\mathbb{E}_n[Y|\mathbf{X} \in \mathbf{r}] - \mathbb{E}_n[Y]\right| \geq \sqrt{(\mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) - \sigma_n^2)_+}\right\}, \tag{3.10}$$

   the approximately insignificant subsets $\hat{\mathcal{C}}_n^i$ are defined by

$$\hat{\mathcal{C}}_n^i := \left\{\mathbf{r} \in \hat{\mathcal{C}}_n \setminus \hat{\mathcal{C}}_n^s : \varepsilon_n \geq \sqrt{(\mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) - \sigma_n^2)_+}\right\}, \tag{3.11}$$

   where $x_+ = \max\{x, 0\}$.
   Assume moreover that $\hat{\mathcal{C}}_n^s$ and $\hat{\mathcal{C}}_n^i$ satisfy

$$\frac{M(\hat{\mathcal{C}}_n^s)}{m(\hat{\mathcal{C}}_n^s)} = o_{\mathbb{P}}(\beta_n^{-2} \wedge n^{1/2-\alpha}) \tag{3.12}$$

and

$$\frac{M(\hat{\mathcal{C}}_n^i)}{m(\hat{\mathcal{C}}_n^i)} = o_{\mathbb{P}}(\varepsilon_n^{-2} \wedge n^{1/2-\alpha}) \tag{3.13}$$

where $M(\mathcal{C}) := \max_{\mathbf{x}\in\mathbb{R}^d} \#\varphi_{\mathcal{C}}(\mathbf{x})$ *and* $m(\mathcal{C}) := \min_{\mathbf{x}\in\mathbb{R}^d} \#\varphi_{\mathcal{C}}(\mathbf{x})$ *are the maximal and minimal redundancies of* $\mathcal{C}$ *a collection of subsets of* $\mathbb{R}^d$.

Note that, in practice it is not guaranteed to obtain a complete covering of $\mathbb{R}^d$ with (in)significant hyperrectangles only. Thus we add an extra assumption (hypothesis (H13)) requiring that at the limit the (in)significant hyperrectangles cover the support of the observations.

3.3 Prediction based on quasi-coverings by hyperrectangles

One practical drawback using prediction (3.6) instead of the averaging is when $\mathbf{x} \in A$ with $A \in \mathcal{P}(\hat{\mathcal{C}}_n)$ such that $\sum_{i=1}^n \mathbf{1}_{\mathbf{X}_i\in A} = 0$. In Chapter 2, we tackle this issue by predicting arbitrarily 0 in this case as it is usually done by the rule $0/0 = 0$ advocated in [28, Chapter 4]. However, for many A such that $\sum_{i=1}^n \mathbf{1}_{\mathbf{X}_i\in A} = 0$ we do not know how to interpret this arbitrary choice or any other arbitrary ones (such as $\mathbb{E}_n[Y]$ which seems more accurate). In this sense, we do not give a prediction in such a case.

We denote $\hat{\mathcal{C}}_n^s$ and $\hat{\mathcal{C}}_n^i$ the set of significant rules such that $\hat{\mathcal{C}}_n = \hat{\mathcal{C}}_n^s \cup \hat{\mathcal{C}}_n^i$. In practice we predict as following with

$$A_n^s(\mathbf{x}) = \bigcap_{\mathbf{r}\in\hat{\mathcal{C}}_n^s:\mathbf{x}\in\mathbf{r}} \mathbf{r} \setminus \bigcup_{\mathbf{r}\notin\hat{\mathcal{C}}_n^s:\mathbf{x}\in\mathbf{r}} \mathbf{r}$$

and

$$A_n^i(\mathbf{x}) = \left(\bigcap_{\mathbf{r}\in\hat{\mathcal{C}}_n^i:\mathbf{x}\in\mathbf{r}} \mathbf{r} \setminus \bigcup_{\mathbf{r}\notin\hat{\mathcal{C}}_n^i:\mathbf{x}\in\mathbf{r}} \mathbf{r} \right) \setminus \mathbf{c}_n^s,$$

where $\mathbf{c}_n^s = \cup_{\mathbf{r}\in\hat{\mathcal{C}}_n^s} \mathbf{r}$.

1. If there exists $\mathbf{r} \in \hat{\mathcal{C}}_n^s$ such that $\mathbf{x} \in \mathbf{r}$, then we predict
   (a) If it exists i such that $\mathbf{X}_i \in A_n^s(\mathbf{x})$, then
   $$g_n(\mathbf{x}) = \frac{\sum_{i:\mathbf{X}_i\in A_n^s(x)} Y_i}{\#\{i : \mathbf{X}_i \in A_n^s(\mathbf{x})\}},$$
   (b) If not there is no prediction.
2. If for all $\mathbf{r} \in \hat{\mathcal{C}}_n^s$ we have $\mathbf{x} \notin \mathbf{r}$ and there exists $\mathbf{r} \in \hat{\mathcal{C}}_n^i$ such that $\mathbf{x} \in \mathbf{r}$, then we predict
   (a) If it exists i such that $\mathbf{X}_i \in A_n^i(\mathbf{x})$, then
   $$g_n(\mathbf{x}) = \frac{\sum_{i:\mathbf{X}_i\in A_n^i(\mathbf{x})} Y_i}{\#\{i : \mathbf{X}_i \in A_n^i(\mathbf{x})\}},$$
   (b) If not we predict the "no-rule" i.e.,
   $$g_n(\mathbf{x}) = \frac{\sum_{i:\mathbf{X}_i\notin\cup_{\mathbf{r}\in\hat{\mathcal{C}}_n^s}\mathbf{r}} Y_i}{\#\{i : \mathbf{X}_i \notin \cup_{\mathbf{r}\in\hat{\mathcal{C}}_n^s} \mathbf{r}\}},$$
3. If for all $\mathbf{r} \in \hat{\mathcal{C}}_n$ $\mathbf{x} \notin \mathbf{r}$ then there is no prediction.

3.4 Consistency of estimator based on quasi-coverings by hyperrectangles

Theorem 2.2.1 presents sufficient assumptions for an algorithm to generate a consistent estimator g_n of g^* based on a suitable sequence of coverings of hyperrectangles. We required extra conditions controlling the complexity of the family of finer partitions that permit to build by countable unions of cells all the partitions that the algorithm can generate. These conditions use are based on the concept of partitioning number introduced in [53, Sec. 1.2] (see also [28, Def 13.1]).

Definition 3.4.1. *Let Π be a family of partitions of $\mathbb{R}^d$.*

1. *The maximal number of cells in a partition of Π is denoted by*

$$\mathcal{M}(\Pi) := \sup\{\#\mathcal{P} : \mathcal{P} \in \Pi\}.$$

2. *For a set $\mathbf{x}_1^n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in (\mathbb{R}^d)^n$, let*

$$\Delta(\mathbf{x}_1^n, \Pi) := \#\big\{\{\mathbf{x}_1^n \cap A : A \in \mathcal{P}\} : \mathcal{P} \in \Pi\big\}$$

   be the number of distinct partitions of $\mathbf{x}_1^n$ induced by elements of Π.

3. *The partitioning number $\Delta_n(\Pi)$ of Π is defined by:*

$$\Delta_n(\Pi) := \max_{\mathbf{x}_1^n \in (\mathbb{R}^d)^n} \Delta(\mathbf{x}_1^n, \Pi).$$

The partitioning number is the maximal number of different partitions of any n points set that can be induced by elements of Π.

Let $\Pi_n = \{\mathcal{P}(\mathcal{C}_n(D'_n)) : D'_n \in (\mathbb{R}^d \times \mathbb{R})^n\}$ be the family of partitions induced by the coverings $\mathcal{C}_n(D'_n)$ that can be generated by the algorithm. And let $\widetilde{\Pi}_n$ the family of finer partitions $\widetilde{\mathcal{P}}$ in the following sense: for any $\mathcal{P} \in \Pi_n$ there exists a partition $\widetilde{\mathcal{P}} \in \widetilde{\Pi}_n$ such that $\mathcal{P}$ can be expressed as a set of countable unions of cells of $\widetilde{\mathcal{P}}$.

We work under the assumption that there exists an intermediate sequence $m_n = o(n^{1/d})$ such that

$$\mathcal{M}(\widetilde{\Pi}_n) \le (m_n)^d \quad \text{and} \quad \Delta_n(\widetilde{\Pi}_n) \le \left(\sum_{l=1}^{m_n} \binom{n-1}{l-1}\right)^d. \tag{3.14}$$

We are now ready to state a slightly extended version of Theorem 2.2.1 that is applicable to RICE algorithm.

Theorem 3.4.1. *Consider an algorithm generating an approximately suitable sequence $(\hat{\mathcal{C}}_n)_{n\ge1}$ of data-dependent collections of sets of $\mathbb{R}^d$ fulfilling the two following conditions:*

$$\mathbb{P}\Big(\bigcup_{\mathbf{r}\in\hat{\mathcal{C}}_n(D_n)} \{\mathbf{r}\times\mathbb{R}\}\Big) \to 1\,, \qquad n\to\infty\,. \tag{H13}$$

Assume that $\hat{\mathcal{C}}_n(D_n)$ can be completed with insignificant rules in order to constitute a covering of $\mathbb{R}^d$ and that σ_n^2 satisfies $|\sigma_n^2 - \sigma^2| = O_{\mathbb{P}}(n^{\alpha-1/2})$, then the estimator g_n defined as (3.7) is universally weakly consistent,

$$\mathbb{E}_{\mathbb{Q}}[(g_n(\mathbf{X}) - g^*(\mathbf{X}))^2] = o_{\mathbb{P}}(1).$$

Proof. The proof is a straightforward extension of Theorem 2.2.1. This theorem asserts the weak consistency of predictors satisfying (3.7) for suitable coverings under two conditions

$$\mathcal{M}(\Pi_n) \vee \log(\Delta_n(\Pi_n)) = o(n), \tag{H14}$$

$$\forall n \in \mathbb{N}^*, \{\mathbf{c} \times \mathbb{R}, \mathbf{c} \in \mathcal{C}_n\} \subseteq \mathcal{B}, \tag{H15}$$

where $\mathcal{B}$ is a $\mathbb{Q}$-Donsker class.

Condition (H15) is fulfilled on hyperrectangles by an application of Lemma 2.3.1. Then, let us show that (3.14) is sufficient for (H5). We have that $\mathcal{M}(\widetilde{\Pi}_n) \leq (m_n)^d$ and $\Delta_n(\widetilde{\Pi}_n) \leq (4n)^{m_n^d}$ (see Example 5). And since $m_n = o(n^{1/d})$ we have $\mathcal{M}(\widetilde{\Pi}_n) \vee \log(\Delta_n(\widetilde{\Pi}_n)) = o(n)$. Finally, to control the estimation error we use the following inequality

$$\sup_{g \in G_n} |\mathcal{L}_n(g) - \mathcal{L}(g)| \leq \sup_{g \in \mathcal{G}_c \circ \Pi_n} |\mathcal{L}_n(g) - \mathcal{L}(g)| \leq \sup_{g \in \mathcal{G}_c \circ \widetilde{\Pi}_n} |\mathcal{L}_n(g) - \mathcal{L}(g)|,$$

where $\mathcal{G}_c$ is the set of constant functions $\mathbb{R}^d \to [-L, L]$ and

$$\mathcal{G}_c \circ \Pi_n := \left\{ g : \mathbb{R}^d \to \mathbb{R} : g = \sum_{A \in \mathcal{P}_n} f_A \mathbf{1}_A, \mathcal{P}_n \in \Pi_n, f_A \in \mathcal{G}_c \right\}.$$

The set of functions $\mathcal{G}_c \circ \widetilde{\Pi}_n$ is defined similarly.

We are now discussing the extension of Theorem 2.1 of Chapter 2 under the approximation of σ^2 by σ_n^2 in (3.12) and (3.13) and ((H13)) instead of a covering $\mathcal{C}_n$ of $\mathbb{R}^d$. Indeed an inspection of the proof of Theorem 2.1 of Chapter 2 shows that σ^2 there always appears with an empirical approximation of the order $\Delta_n' := \sup_{\mathbf{r} \in \mathcal{C}_n} \left\{ \left| \mathbb{V}(Y|\mathbf{X} \in \mathbf{r}) - \mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) \right| \right\} = O_{\mathbb{P}}(n^{\alpha - 1/2})$. Thus, the extra approximation of σ^2 with σ_n^2 is harmless and the weak consistency of g_n follows if $\hat{\mathcal{C}}_n$ was a covering.

As we have no guarantee than $\hat{\mathcal{C}}_n$ is a covering of $\mathbb{R}^d$, we consider $\hat{\mathbf{c}}_n = \bigcup_{\mathbf{r} \in \hat{\mathcal{C}}_n} \mathbf{r}$ and $\mathbb{R}^d \setminus \hat{\mathbf{c}}_n$, the subset of $\mathbb{R}^d$ which is not covered by $\hat{\mathcal{C}}_n$. Let us show that the consistency of the estimator of g^* can be extended to this case.

We recall the approximation error term $\mathbb{E}[(g_n(\mathbf{X}) - g^*(\mathbf{X}))^2]$ that was proved to be $o_{\mathbb{P}}(1)$ Theorem 2.1 of Chapter 2 for any rule estimators based on covering on (approximatively) significant and insignificant rules only. We consider g_n^c the complete covering based on $\hat{\mathcal{C}}_n$ completed with insignificant rules and $\mathcal{P}_n^c$ the induced partition of $\mathbb{R}^d$. Denoting $\hat{\mathcal{P}}_n$ the partition induced by $\hat{\mathcal{C}}_n$ over $\hat{\mathbf{c}}_n$ then we have

$$\begin{aligned}
\mathbb{E}_{\mathbb{Q}}[(g_n(\mathbf{X}) - g^*(\mathbf{X}))^2] &\leq 2(\mathbb{E}_{\mathbb{Q}}[(g_n(\mathbf{X}) - g_n^c(\mathbf{X}))^2] + \mathbb{E}_{\mathbb{Q}}[(g_n^c(\mathbf{X}) - g^*(\mathbf{X}))^2]) \\
&\leq 2\mathbb{E}_{\mathbb{Q}}[(g_n(\mathbf{X}) - g_n^c(\mathbf{X}))^2] + o_{\mathbb{P}}(1) \\
&\leq \sum_{A \in \mathcal{P}_n^c} 2\mathbb{E}_{\mathbb{Q}}[(g_n(\mathbf{X}) - g_n^c(\mathbf{X}))^2 | \mathbf{X} \in A] \mathbb{P}(\mathbf{X} \in A) + o_{\mathbb{P}}(1) \\
&\leq \sum_{A \in \mathcal{P}_n^c \setminus \hat{\mathcal{P}}_n} 2\mathbb{E}_{\mathbb{Q}}[(g_n(\mathbf{X}) - g_n^c(\mathbf{X}))^2 | \mathbf{X} \in A] \mathbb{P}(\mathbf{X} \in A) + o_{\mathbb{P}}(1) \\
&\leq 2L^2 \sum_{A \in \mathcal{P}_n^c \setminus \hat{\mathcal{P}}_n} \mathbb{P}(\mathbf{X} \in A) + o_{\mathbb{P}}(1) \\
&\leq 2L^2 \mathbb{P}(\mathbf{X} \in \mathbb{R}^d \setminus \hat{\mathbf{c}}_n) + o_{\mathbb{P}}(1).
\end{aligned}$$

Under ((H13)) and using the previous $\mathbb{P}$ negligibility result, we have

$$\mathbb{P}(\mathbf{X} \in \mathbb{R}^d \setminus \hat{\mathbf{c}}_n) = 1 - \mathbb{P}\Big(\mathbf{X} \in \bigcup_{\mathbf{r} \in \hat{\mathcal{C}}_n(\mathbb{D}_n)} \mathbf{r}\Big) = o(1).$$

Thus one can consider that $\mathcal{C}_n$ is a covering of $\mathbb{R}^d$ and the weak consistency still holds $\mathbb{E}_{\mathbb{Q}}[(g_n(\mathbf{X}) - g^*(\mathbf{X}))^2] = o_{\mathbb{P}}(1)$. □

Remark 12. The safe strategy of prediction presented in Section 3.3 does not alter the asymptotic properties of the algorithms for any $\mathbf{x}$ in the support of the $\mathbf{X}_i$ as such cases disappear $\mathbb{P}$-a.s when $n \to \infty$ under condition (H13). Indeed, under (H13) the case of no prediction 3. is asymptotically $\mathbb{P}$-negligible.

3.5 RICE: the algorithm

The Rule Induction Covering Algorithm (RICE) aims at designing quasi-coverings satisfying (H13) based on hyperrectangles which are all either approximately significant or insignificant (Definition 3.2.1). We supply an open source code of RICE in *Python* on GitHub.

The algorithm's steps (see Algorithm 4) consist in designing a set of hyperrectangle rules (see Algorithms 7 and 8) on a discretization of the feature space (see Algorithm 6), and then to select (see Algorithm 5) among them a subset of the *significant rules* and a subset of the *insignificant rules* such that (3.12) and (3.13) are fulfilled with σ_n^2 instead of σ^2. There is no guarantee that these sets of rules together constitute a covering of the feature space and even not a covering of the sample: whether condition (H13) is satisfied will be discussed.

Name	Range	Description	Default value
m_n	$\mathbb{N}^*$, such that $(m_n)^d = o(n)$	Sharpness of the discretization	$n^{1/2d}$
α	$(0, 1/2)$	Coverage rate	$1/2 - 1/100$
γ	$(0, 1)$	Intersection rate	0.95
k	$\mathbb{N}^*$	Maximal number of candidate rules	150
l_{max}	$\{1, \ldots, d\}$	Maximal length of rules	3

Table 3.1 – Main parameters of the algorithm RICE with their default values.

Algorithm 4: Main

Input:

— m_n the number of bins
— α the coverage rate parameter
— γ the intersection rate parameter
— k the maximal number of candidate rules
— l_{max} the maximal length of the rules
— D_n the dataset

Output:

— $\mathcal{C}_n$ the set of selected rules

$\beta_n \leftarrow n^{\alpha/2-1/4}$ the significant coefficient;
$\varepsilon_n \leftarrow \beta_n \mathbb{V}_n(Y)$ the insignificant coefficient;
$\tilde{\mathbf{X}} \leftarrow Discretize(\mathbf{X}, m_n)$ discretization of each feature into modalities $1, \ldots, m_n$;
$D_n = (\tilde{\mathbf{X}}, Y)$;
$\mathcal{R}_n \leftarrow Calc\_length\_1(\tilde{\mathbf{X}}, \alpha, m_n)$;
if $l_{max} \geq 2$ **then**
    $\mathcal{R}_1 \leftarrow$ the k candidates rules of length 1 in $\mathcal{R}_n$;
    $\mathcal{R}' \leftarrow Calc\_length(\tilde{\mathbf{X}}, \alpha, \mathcal{R}_1, \mathcal{R}_1)$;
    $\mathcal{R}_n \leftarrow append(\mathcal{R}_n, \mathcal{R}')$;
    if $l_{max} \geq 3$ **then**
        for $l = 3, \ldots, l_{max}$ **do**
            $\mathcal{R}_{l-1} \leftarrow$ the k candidates rules of length $l-1$ in $\mathcal{R}_n$;
            $\mathcal{R}' \leftarrow Calc\_length(\tilde{\mathbf{X}}, \alpha, \mathcal{R}_1, \mathcal{R}_{l-1})$;
            $\mathcal{R}_n \leftarrow append(\mathcal{R}_n, \mathcal{R}')$;
$S \leftarrow Get\_significant(\mathcal{R}_n, D_n, \beta_n)$;
$I \leftarrow Get\_insignificant(\mathcal{R}_n \setminus S, D_n, \varepsilon_n)$;
$\hat{\mathcal{C}}_n \leftarrow Select(S, I, \gamma)$;
Return $\hat{\mathcal{C}}_n$;

3.5.1 Discretization

Designing all rules when features are continuous variables, is too complex and too time-consuming to be feasible in practice. So, to increase speed by limiting the searching area, each variable of the feature space is discretized into m_n classes with $m_n = o(n^{1/d})$. As in Chapter 5 the empirical quantiles of each variable are considered (when it has more than m_n distinct values). The algorithm is described in Algorithm 6. The algorithmic complexity of the discretization is $O(ndm_n)$. After the discretization, each class for each variable covers approximately n/m_n points of D_n.

The discretization is interesting since it guarantees that a covering of the sample is a covering of the feature space. Indeed, if we have a covering of the sample then for any new observation we have at least one activated rule. Without the discretization process, the new observation may be a new extremmum for example.

Moreover, the discretization process ensures that any partition generated by RICE, Π_n, can be expressed as a set of countable unions of cells of the quantile partition with the parameter m_n, $\widetilde{\Pi}_n$. Hence $\widetilde{\Pi}_n$ fulfills (3.14) (see for instance Example 6).

Using discretization is a common method when $\mathbf{X} \in \mathbb{R}^d$, we may cite [18, 16, 64] which use quantitative methods for discretization and the algorithms *BRL* (Bayesian Rule Lists) [41] and *SIRUS* [4] which use the empirical quantiles.

3.5.2 Rules designing

The rule designing process is similar to the RIPE in Chapter 5 The rule designing is processed recursively on the rule length l defined in (3.2).

A rule, $\mathbf{r}$, is completely defined by a binary vector of length n equal to 1 if $\mathbf{X}_i \in \mathbf{r}$ and 0 otherwise for $i \in \{1, \dots, n\}$. Hence, the complexity to design one rule is $O(n)$.

Rules of length 1

RICE first designs all rules of length 1 which fulfill (3.8). Thanks to the discretization we have at most $\frac{(m_n+1)m_n}{2}$ rules by variables, so the number of rules of length 1, $\mathcal{R}_1$, is bounded by

$$\#\mathcal{R}_1 \leq d\left(\frac{(m_n+1)m_n}{2}\right).$$

From this bound it is easy to see that the complexity of evaluating all rules of length 1 is $O(ndm_n^2)$. This part is described as Algorithm 7.

Rules of length l

The design of rules of length l, corresponding to Algorithm 8, is made by intersections of rules of length 1 and of rules of length $l-1$. Some of these intersections can be of length smaller than l and are discarded. The intersections which do not fulfill the coverage condition (3.8) are discarded, too.

Formally, the design of rules of length l is made by a *suitable intersection* of rules of length 1 and rule of length $l-1$.

Definition 3.5.1 (Def 2.3 Chapter 5). *Let $d_n = (\mathbf{x}_i)_{1 \leq i \leq n}$ be a sequence of observation of $\mathbf{X}$. Two rules $\mathbf{r}$ and $\mathbf{s}$ define a suitable intersection if and only if they satisfy the two following conditions:*

1. ***Intersection condition:***

$$\begin{aligned} &\{\mathbf{x} \in \mathbf{r} \cap \mathbf{s}; \mathbf{x} \in d_n\} \neq \emptyset, \\ &\{\mathbf{x} \in \mathbf{r}; \mathbf{x} \in d_n\} \not\subseteq \{\mathbf{x} \in \mathbf{s}; \mathbf{x} \in d_n\} \text{ and } \{\mathbf{x} \in \mathbf{s}; \mathbf{x} \in d_n\} \not\subseteq \{\mathbf{x} \in \mathbf{r}; \mathbf{x} \in d_n\}. \end{aligned} \tag{3.15}$$

2. ***Length condition:***

$$\ell(\mathbf{r} \cap \mathbf{s}) = \ell(\mathbf{r}) + \ell(\mathbf{s}). \tag{3.16}$$

The intersection condition (3.15) ensures that the algorithm does not increase the length of a rules without adding information. It means that $\mathbf{r}$ and $\mathbf{s}$ must not be satisfied for the same observations of d_n. The length condition (3.16) means that $\mathbf{r}$ and $\mathbf{s}$ have no marginal index in common. In other words $\mathbf{r}$ and $\mathbf{s}$ must satisfied that

$$\{1 \leq j \leq d; c_j^{\mathbf{r}} \neq \mathbb{R}\} \cap \{1 \leq j \leq d; c_j^{\mathbf{s}} \neq \mathbb{R}\} = \emptyset.$$

Despite the discretization, the number of rules increases too fast with l to be all designed. So, to design rules of length $l+1$ for $l \geq 2$, RICE selects a given number $k \in \mathbb{N}^*$ of rules of each length 1 and l. They are selected according to their empirical risks (3.4). These rules are called *candidate rules*. Thus

$$\#\mathcal{R}_{l+1} \leq k^2, \qquad \forall\, l \in \{1, \ldots, d-1\}. \tag{3.17}$$

These formula permits to have a control on the maximal number of generated rules. Let $l_{max} \in \mathbb{N}^*$ a given maximal length for the generated rules. For a given number of candidate $k \in \mathbb{N}^*$ the maximal number R of generated rules is bounded by

$$R \leq \#\mathcal{R}_1 + l_{max} k^2. \tag{3.18}$$

From this bound, it is easy to see that the complexity of evaluating all rules of length l obtained from their intersections is $O(nk^2)$.

3.5.3 Get (in)significant rules

To consider significant and insignificant rules in practice a (consistent) estimator σ_n^2 of σ^2 is needed. It can be defined as follows, where $\mathcal{R}_n$ is the set of rules designed as described in Section 3.5.2.

Definition 3.5.2. *The estimator σ_n^2 is the smallest empirical conditional variance over all the rules in $\mathcal{R}_n$:*

$$\sigma_n^2 = \min_{\mathbf{r}_n \in \mathcal{R}_n} \mathbb{V}_n(Y \mid \mathbf{X} \in \mathbf{r}_n).$$

To select significant and insignificant rules, RICE applies a filtering of the rules. RICE selects $S \subseteq \mathcal{R}_n$ the set of the significant rules by considering rules $\mathbf{r} \in \mathcal{R}_n$ which satisfied the following inequality

$$\beta_n \left| \mathbb{E}_n[Y|\mathbf{X} \in \mathbf{r}] - \mathbb{E}_n[Y] \right| \geq \sqrt{(\mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) - \sigma_n^2)_+},$$

where $\beta_n = n^{\alpha/2 - 1/4}$.

The set $I \subseteq \mathcal{R}_n \setminus S$ of the insignificant rules is selected by considering rules $\mathbf{r} \in \mathcal{R}_n$ not in S which satisfied the following inequality

$$\varepsilon_n \geq \sqrt{(\mathbb{V}_n(Y|\mathbf{X} \in \mathbf{r}) - \sigma_n^2)_+},$$

where $\epsilon_n = \beta_n \mathbb{V}_n(Y)$.

3.5.4 Selection of rules

RICE then selects a subset $\hat{\mathcal{C}}_n$ of the set of rules $\mathcal{R}_n$ it has designed. First, it tries to create an empirical covering with significant rules of $\mathcal{R}_n$. To increase the chance to get a covering, the algorithm considers the rules by decreasing order of covering rates $\mathbb{Q}_n(\mathbf{r})$ and adds them to the currently selected set of rules $\mathcal{S}$, subject to the condition that

$$\mathbb{Q}_n\Big(\mathbf{r} \bigcap \Big\{ \bigcup_{\mathbf{r}' \in \mathcal{S}} \mathbf{r}' \Big\}\Big) \leq \gamma \mathbb{Q}_n(\mathbf{r}) \,, \qquad \mathbf{r} \in \mathcal{R}_n \setminus \mathcal{S} \,, \qquad 0 < \gamma < 1 \,. \tag{3.19}$$

This condition guarantees that (3.12) and (3.13) are fulfilled (see Remark 5). The selection ends if the current selected set of rules $\mathcal{S}$ forms a covering or if $\mathcal{S}$ has browsed all significant rules.

In the latter case, the algorithm tries to form a covering by adding insignificant rules to the currently selected set of rules $\mathcal{S}$, avoiding inclusion using the recursive condition that $\mathcal{P}_n(\mathbf{r} \setminus \{\cup_{\mathbf{r}' \in \mathcal{S}} \mathbf{r}'\}) > 0$. The insignificant rules are browsed by increasing order of empirical variance. This step is described in Algorithm 5.

3.5.5 Complexity and parallelization

As mentioned before the algorithm can be easily parallelized. Throughout the process, except for the selection, rules are designed in an independent way. Thus, each loop on rules may be done on different cores. Assume that we have an unlimited number of cores, then the complexity of RICE can be hugely decreased.

The discretization can be done on each dimension, each bin in $\{1, \ldots, m_n\}$ and each $i \in \{1, \ldots, n\}$ at the same time. So, the complexity of Algorithm 6 is the same as the complexity of finding a quantile (line 7) which is $O(n)$ on average [32].

The complexity of the design of rules of length 1, Algorithm 7, may be reduced to $O(n)$ instead of $O(ndm_n^2)$. Indeed, we can do the designing of the $\frac{d}{2}\Big((m_n+1)m_n\Big)$ rules at the same time. Hence, the complexity of Algorithm 7 parallelized is the same as the design of a rule (line 5) which is $O(n)$.

Finally, for Algorithm 8, loops on $\mathcal{R}_1$ and $\mathcal{R}_{l-1}$ can be done at the same time. Thus, we have k^2 intersection operations that can be made in the same time. The complexity of Algorithm 8 parallelized becomes the same as the evaluation of a suitable intersection (lines 4 and 5) which is $O(2n)$.

Unfortunately, Algorithm 5 cannot be parallelized. It complexity depends on n, the maximal number of designed rules (3.18) and the minimal coverage rate to be able to estimate an average number of rules to have a covering.

In conclusion, the dimension of the feature space is not an issue. The number of candidates k and the maximal length l_{max} must be reasonably small (smaller than 500 and 5 respectively) to avoid an uncontrollable increase in the complexity of the select part.

3.5.6 Discussion

Adding insignificant rules to significant rules allows to ensure a covering without considering a *no-rule* as in Chapter 2 Section 2.4. The theoretical difficulty with the no-rule (the complementary of significant rules) is that it is not necessarily a hyperrectangle and it may not fulfill the coverage condition (3.8). Thus it is not clear whether it belongs to fixed Donsker class for any $n \in \mathbb{N}$. To circumvent this issue one considers insignificant rules despite they do not improve the interpretation of the predictor as most of them predict around the mean $\mathbb{E}_n[Y]$. Moreover, as we control drastically the number of rules

Algorithm 5: Select

Input:
— S the set of significant rules
— I the set of insignificant rules
— γ the intersection rate

Output:
— $\hat{\mathcal{C}}_n$ the minimal set of rules

$\hat{\mathcal{C}}_n \leftarrow \arg\max_{\mathbf{r}\in S} \mathbb{Q}_n(\mathbf{r})$;
if $|\hat{\mathcal{C}}_n| > 1$ **then**
  $\hat{\mathcal{C}}_n \leftarrow \arg\min_{\mathbf{r}\in\hat{\mathcal{C}}_n} \mathcal{L}_n(\mathbf{r})$ where $\mathcal{L}_n(\mathbf{r}) := \frac{1}{n}\sum_{i=1}^n \left(\mathbb{E}_n[Y|\mathbf{X}\in\mathbf{r}]\mathbf{1}_{\mathbf{X}_i\in\mathbf{r}} - Y_i\right)^2$;
$S \leftarrow S \setminus \hat{\mathcal{C}}_n$;
while $\mathbb{Q}_n(\cup_{\mathbf{r}\in\hat{\mathcal{C}}_n}\mathbf{r}) < 1$ **do**
  $\mathbf{r}^* \leftarrow \arg\max_{\mathbf{r}\in S} \mathbb{Q}_n(\mathbf{r})$;
  if $|\mathbf{r}^*| > 1$ **then**
    $\mathbf{r}^* \leftarrow \arg\min_{\mathbf{r}\in\mathbf{r}^*} \mathcal{L}_n(\mathbf{r})$;
  if $\mathbb{Q}_n\big(\mathbf{r}^* \cap (\cup_{\mathbf{r}\in\hat{\mathcal{C}}_n}\mathbf{r})\big) \le \gamma\, \mathbb{Q}_n(\mathbf{r}^*)$ **then**
    $\hat{\mathcal{C}}_n \leftarrow \hat{\mathcal{C}}_n \cup \{\mathbf{r}^*\}$;
  $S \leftarrow S \setminus \{\mathbf{r}^*\}$;
  if $\#S = 0$ **then**
    Break ;
end
while $\mathbb{Q}_n(\cup_{\mathbf{r}\in\hat{\mathcal{C}}_n}\mathbf{r}) < 1$ **do**
  $\mathbf{r}^* \leftarrow \arg\min_{\mathbf{r}\in I} \mathbb{V}_n(Y|\mathbf{X}\in\mathbf{r})$;
  if $\mathbb{Q}_n\big(\mathbf{r}^* \cap (\cup_{\mathbf{r}\in\hat{\mathcal{C}}_n}\mathbf{r})\big) < \mathbb{Q}_n(\mathbf{r}^*)$ **then**
    $\hat{\mathcal{C}}_n \leftarrow \hat{\mathcal{C}}_n \cup \{\mathbf{r}^*\}$;
  $I \leftarrow I \setminus \{\mathbf{r}^*\}$;
  if $\#I = 0$ *And* $\mathbb{Q}_n(\cup_{\mathbf{r}\in\hat{\mathcal{C}}_n}\mathbf{r}) < 1$ **then**
    Print *"Warning: Covering is not completed!"*;
    Print *"The covering ratio is:"*;
    Print $\mathbb{Q}_n(\cup_{\mathbf{r}\in\hat{\mathcal{C}}_n}\mathbf{r})$;
    Break ;
end
return $\hat{\mathcal{C}}_n$;

thanks to the number of candidates k there is no guarantee that the union of the selected rules covers the sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$. Clearly the condition (H13) may not be satisfied and the weak consistency of the procedure is not guaranteed. In this case RICE algorithm provides a warning message with the total coverage ratio of the selected rule. In all our experiments with $k = 150$ RICE has always generated a covering of the feature space.

3.5.7 Estimation of the noise variance

We provide some natural conditions under which the estimator σ_n^2 of σ^2 is consistent.

Lemma 3.5.1. *Assume that a rule $\mathbf{r}_n$ can be chosen in $\mathcal{R}_n$ for each $n \in \mathbb{N}^*$ such that*

$$\mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X} \in \mathbf{r}_n) \xrightarrow[n\to\infty]{\mathbb{P}} 0. \tag{3.20}$$

Then σ_n^2 is a consistent estimator of σ^2. Moreover if $\mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X} \in \mathbf{r}_n) = O_{\mathbb{P}}(n^{\alpha-1/2})$ then $|\sigma^2 - \sigma_n^2| = O_{\mathbb{P}}(n^{\alpha-1/2})$ and σ_n^2 satisfies the condition in Theorem 3.4.1.

Proof. First, for any rule $\mathbf{r}$ fulfilling (3.8) and with

$$\Delta'_n = \sup_{\mathbf{r} \text{ rule fulfilling (3.8)}} |\mathbb{V}_n(Y \mid \mathbf{X} \in \mathbf{r}) - \mathbb{V}(Y \mid \mathbf{X} \in \mathbf{r})|, \tag{3.21}$$

we have

$$\begin{aligned}\mathbb{V}_n(Y \mid \mathbf{X} \in \mathbf{r}) &\geq \mathbb{V}(Y \mid \mathbf{X} \in \mathbf{r}) - \Delta'_n \\ &= \mathbb{V}(g^*(X) \mid \mathbf{X} \in \mathbf{r}) + \underbrace{\mathbb{V}(Z \mid \mathbf{X} \in \mathbf{r})}_{=\mathbb{V}(Z)=\sigma^2} - \Delta'_n.\end{aligned}$$

Thus we obtain $\sigma_n^2 \geq \sigma^2 - \Delta'_n$.

Now, since for any rule $\mathbf{r}$ fulfilling (3.8),

$$\begin{aligned}\mathbb{V}_n(Y \mid \mathbf{X} \in \mathbf{r}) &\leq \mathbb{V}(Y \mid \mathbf{X} \in \mathbf{r}) + \Delta'_n \\ &= \mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X} \in \mathbf{r}) + \sigma^2 + \Delta'_n,\end{aligned}$$

we have $\sigma_n^2 \leq \mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X} \in \mathbf{r}) + \sigma^2 + \Delta'_n$.

Finally, if $\mathbf{r}$ fulfills (3.20) and since, by Corollary 2.3.1, $\Delta'_n = O_{\mathbb{P}}(n^{\alpha-1/2}) = o_{\mathbb{P}}(1)$ (rules are hyperrectangles of $\mathbb{R}^d$), the set of which is a Donsker class according to Lemma 2.3.1, we conclude that

$$\sigma_n^2 \xrightarrow[n\to\infty]{\mathbb{P}} \sigma^2.$$

□

A rule $\mathbf{r}_n$ obviously fulfills (3.20) as soon as $g^*_{|\mathbf{r}_n}$ is constant for any n large enough. Such rules can reasonably be expected to be generated by RICE when g^* is a constant piecewise function (e.g. if g^* is a rule or a set of rules).

Another case where (3.20) holds is provided by the following lemma. Notice that, although the assumption of shrinking diameter for all rules is not desirable, the assumption that there should exist at least one sequence of rules with shrinking diameter is much milder.

Let us denote for any $\mathbf{r} \subset \mathbb{R}^d$ and with $\mathbf{X}'$ distributed as (and independent of) $\mathbf{X}$,

$$\begin{aligned}\mathrm{Diam}_{\mathbb{Q}}(\mathbf{r}) &= \inf\{a \geq 0 : \mathbb{Q}(\|\mathbf{X} - \mathbf{X}'\| \leq a | \mathbf{X} \in \mathbf{r} \text{ and } \mathbf{X}' \in \mathbf{r}) = 1\} \\ &\leq \underbrace{\sup\{\|\mathbf{x} - \mathbf{x}'\| : \mathbf{x}, \mathbf{x}' \in \mathbf{r}\}}_{\mathrm{Diam}(\mathbf{r})}.\end{aligned}$$

By convention $\mathrm{Diam}_{\mathbb{Q}}(\mathbf{r}) = 0$ if $\mathbb{Q}(\mathbf{X} \in \mathbf{r}) = 0$.

Lemma 3.5.2. *Assume that g^* is uniformly continuous and that a rule $\mathbf{r}_n$ can be chosen in $\mathcal{R}_n$ for each n such that*

$$\mathit{Diam}_{\mathbb{Q}}(\mathbf{r}_n) = o_{\mathbb{P}}(1).$$

Then $(\mathbf{r}_n)_{n\in\mathbb{N}^*}$ fulfills (3.20). Moreover, if g^* is Lipschitz continuous and

$$\mathit{Diam}_{\mathbb{Q}}(\mathbf{r}_n) = O_{\mathbb{P}}(n^{\alpha/2-1/4})$$

then $|\sigma^2 - \mathbb{V}(Y \mid \mathbf{X} \in \mathbf{r}_n)| = O_{\mathbb{P}}(n^{\alpha-1/2})$.

Proof. We denote $\mathbf{r} = \mathbf{r}_n$ for simplicity and $\mathbf{X}_{\mathbf{r}}$ and $\mathbf{X}'_{\mathbf{r}}$ two independent variables distributed as $\mathbf{X}$ given that $\mathbf{X} \in \mathbf{r}$. We obtain

$$\begin{aligned}\mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X} \in \mathbf{r}) &= \mathbb{V}(g^*(\mathbf{X}_{\mathbf{r}})) \\ &= \tfrac{1}{2}\mathbb{V}\big(g^*(\mathbf{X}_{\mathbf{r}}) - g^*(\mathbf{X}'_{\mathbf{r}})\big) \\ &\leq \tfrac{1}{2}\mathbb{E}\left[(g^*(\mathbf{X}_{\mathbf{r}}) - g^*(\mathbf{X}'_{\mathbf{r}}))^2\right].\end{aligned}$$

Thus, if we denote w the modulus of continuity of g^*, we get

$$\mathbb{V}(g^*(\mathbf{X}) \mid \mathbf{X} \in \mathbf{r}) \leq \tfrac{1}{2} w(\mathrm{Diam}_{\mathbb{P}}(\mathbf{r}))^2.$$

The result follows by uniform continuity of g^*.

□

3.6 Applications

We present several applications to present advantages and limits of using RICE algorithm. In the following, we denote by $\mathcal{S}^*$ the support of Y, $\mathcal{S}_n$ the support of the estimator g_n. To measure the difference between the two supports we introduce the distance $\mathcal{H}$ defined by:

$$\mathcal{H}(\mathcal{S}_n, \mathcal{S}^*) = |\mathcal{S}^* \setminus \mathcal{S}_n| + |\mathcal{S}_n \setminus \mathcal{S}^*|.$$

For each application RICE has been used with its default parameters (see Table 3.1) and it has generated a covering of the feature space. Nevertheless, as mentioned before a new observation may be in a cell of the partition generated by RICE without any observation used during the rules-designing process. In this case, either RICE is not able to make prediction (case 1. (b)) or RICE predicts the "no-rule" (case 2. (b)). Observations for which RICE is not able to make a prediction have been removed in the calculation of MSE^*. They represent a proportion between 1% and 3% of the test observations.

3.6.1 Artificial data

Here we consider the same data set as in [20] already used in Section 2.4.2. We generate $n = 5,000$ data points following the regression setting

$$Y = g^*(\mathbf{X}) + Z,$$

where $d = 100$ (the dimension of $\mathbf{X}$) and

$$g^*(\mathbf{X}) = 9 \prod_{j=1}^{3} \exp\left(-3\,(1 - X_j)^2\right) - 0.8 \exp\left(-2\,(X_4 - X_5)\right) + 2\sin^2(\pi \cdot X_6) - 2.5\,(X_7 - X_8)\,, \quad (3.22)$$

and $Z \sim \mathcal{N}(0, \sigma^2)$. The value of $\sigma > 0$ was chosen to produce a two-to-one signal-to-noise ratio. The variables were generated from a uniform distribution on $\{0/10, \ldots, 9/10\}$. It is important to notice that only the first eight variables are informative; the 92 others are just noise. Coefficients multiplying each of the terms in g^* were chosen to ensure that variables have approximately equal influence.

In order to evaluate the error without including the error of the noise, we consider the following error

$$MSE^* = \mathbb{E}_{\mathbb{Q}}\left[|g^*(\mathbf{X}) - g_n(\mathbf{X})|^2\right].$$

We approximate the criteria MSE^* using $50,000$ test observations sampled independently from $\mathbb{Q}$.

Execution

We run $M = 100$ simulations. For each simulation we set the maximal number of rules generated by the Random Forest (RF) at $20,000$ and the number of trees at $m_{tree} = 100$.

Results

According to Figure 3.1 RICE seems less accurate than RF and RuleFit. It was expected because it is the cost of interpretability. Indeed, from Table 3.2, the interpretability index of RICE is largely smaller than those of others. Moreover, RICE generates rules (significant and insignificant) implying only variables of the support of Y. This is confirmed by the following result

$$\forall s \in \{1, \dots, M\} \text{ we have } \mathcal{H}(\mathcal{S}_n^s, \mathcal{S}^*) = 0,$$

where $\mathcal{S}_n^s$ is the support of the estimator g_n generated by RICE for the simulation s.

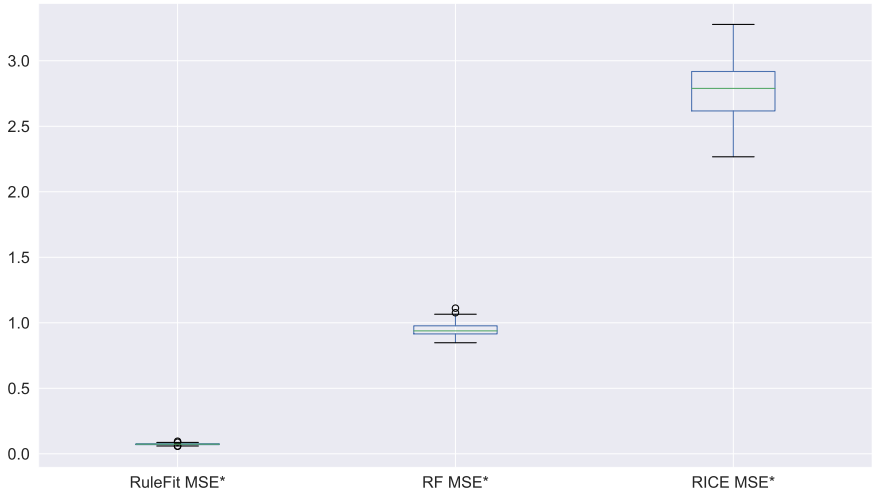


Figure 3.1 – Box-plot of the MSE^* of each algorithm.

The average occurrence of a variable in a selected set of rules is depicted in Figure 3.2. Variables X_7 and X_8, corresponding to the linear relation, are the most representative variables. On another side, X_6 which corresponds to the sinusoidal relation is the less represented variable. This result makes sense because a sinusoidal signal is harder to explain with rules than a linear one. Indeed, a linear signal may be described with two rules: one greater than the mean and one less than the mean.

One can notice that the Covering Algorithm introduced in Section 2.4.1 seems better than RICE for this application. One reason might be the fact that Covering Algorithm uses a "no-rule" to be able to generate a covering. It permits to limit the number of selected rules but this algorithm cannot generate a consistent estimator of the regression function contrary to RICE. Moreover, RICE discretizes features which decreases the accuracy of the prediction but makes the algorithm less complex (see (3.14)) while Covering Algorithm uses a Random Forest as rule generator.

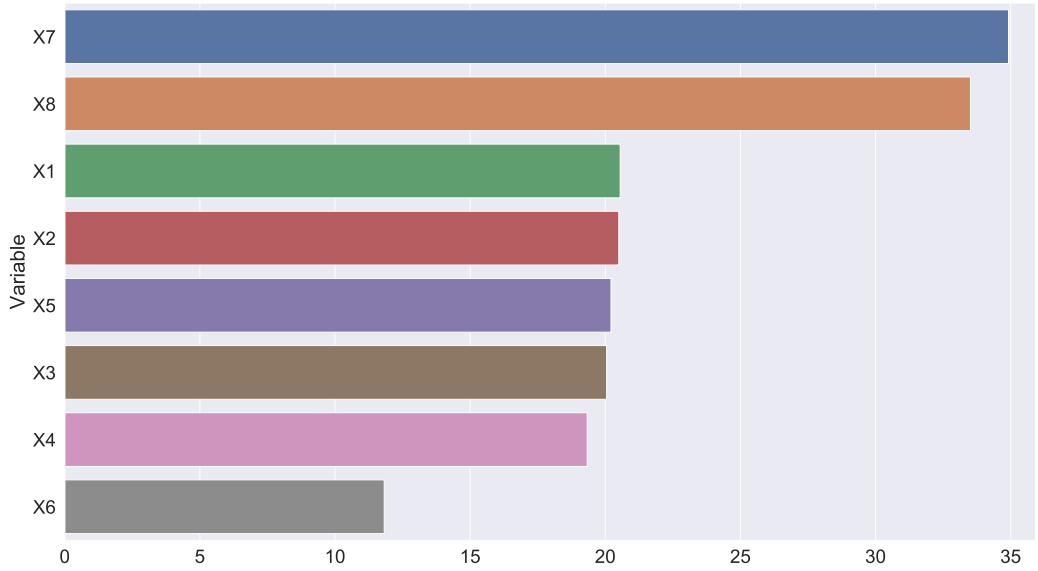


Figure 3.2 – Average occurrence in the selected set of rules generated by RICE.

Random Forest	Nb rules	Interpretability	MSE^*
mean	12596.36	60870.42	0.95
std	3.62	349.37	0.05
min	12584.00	60060.00	0.86
25%	12596.00	60640.50	0.92
50%	12598.00	60851.00	0.94
75%	12598.00	61126.50	0.98
max	12600.00	61530.00	1.12

RICE	Nb rules	Interpretability	MSE^*
mean	63.58	180.89	2.81
std	8.24	21.85	0.23
min	51.00	148.00	2.29
25%	58.00	166.00	2.66
50%	62.00	174.50	2.82
75%	67.25	191.50	2.95
max	99.00	264.00	3.33

RuleFit	Nb rules	Nb linear	Interpretability	MSE^*
mean	184.43	24.89	451.70	0.07
std	17.91	6.61	51.24	0.01
min	145.00	11.00	343.00	0.06
25%	170.75	20.00	411.75	0.07
50%	184.00	24.50	456.50	0.07
75%	199.00	29.00	492.25	0.08
max	234.00	42.00	568.00	0.10

Table 3.2 – Summary of the $M = 100$ experiences for each algorithm. Nb Rules is the number of rules, Nb Linear (for RuleFit) is the number of linear components in the generated model and Interpretability is the interpretability index.

3.6.2 Random hyperrectangles

In this application we consider the following regression setting

$$Y = g^*(\mathbf{X}) + Z,$$

where g^* is defined by a set of random rules generated by the Aglorithm 9 given in Appendix.

Execution 1

Let $i, j \in \{1, \dots, d\}$ such that $i < j$, we set a rule

$$\mathbf{r}^* = \prod_{k=1}^{i-1} \mathbb{R} \times [a_i, b_i] \times \prod_{k=i+1}^{j-1} \mathbb{R} \times [a_j, b_j] \times \prod_{k=j+1}^{d} \mathbb{R}$$

such that $0.02 \leq \mathbb{Q}(\mathbf{r}^* \times \mathbb{R}) \leq 0.05$. Then we set

$$g^*(\mathbf{X}) = \mathbf{1}_{\mathbf{X} \in \mathbf{r}^*}.$$

We generate $n = 2,000$ points to train the algorithms and test them on $20,000$. The aim of the first simulation is to show the behavior of the algorithms for $d \in \{10, 20, \ldots, 200\}$. For each value of d we run 100 simulations.

Results

In this application, we test the stability of the algorithms when we add noninformative variables. The support of g^* is composed of two variables. Figure 3.3 highlights the fact that RICE is invariant in dimension contrary to the others. This property comes from the sparse set of variables occurring in rules selected by RICE. Indeed, as shown in Table 3.3c, the support of the estimator generated by RICE is always composed of around 5 variables independently of the dimension of the feature space.

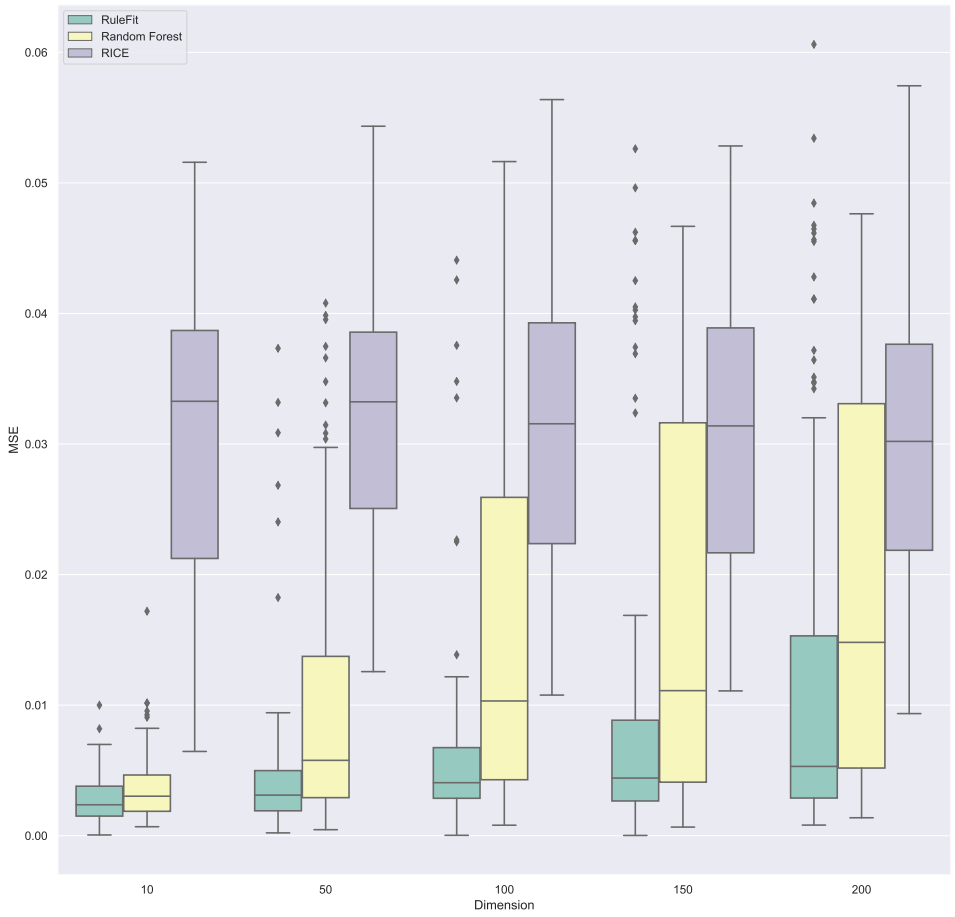


Figure 3.3 – Box-plot of the MSE^* of each algorithm.

d	Nb rules	Interpretability
10	9243.44	33902.60
20	8942.48	36273.56
30	8474.60	35731.46
40	7973.84	33994.34
50	7726.54	33403.88
60	7786.20	34010.02
70	7691.88	33772.74
80	7705.20	33992.32
90	7627.32	33875.16
100	6493.20	8456.26
110	6769.82	29948.56
120	6705.60	29617.42
130	7071.20	31504.18
140	6371.26	28132.10
150	6133.60	27020.92
160	6501.72	28763.94
170	6569.26	29157.72
180	6354.10	28229.52
190	6250.34	27728.86
200	6067.28	26859.14

(a) Random Forest.

d	Nb rules	Nb Linear	Interpretability
10	62.50	0.66	183.37
20	54.76	2.04	172.01
30	59.35	5.01	186.70
40	51.04	6.34	164.24
50	56.20	10.01	186.15
60	51.56	13.30	172.21
70	43.82	12.42	148.64
80	42.60	14.40	146.89
90	51.43	20.31	173.37
100	39.12	17.84	137.92
110	42.92	18.13	153.48
120	38.37	18.38	139.08
130	37.19	20.10	137.26
140	48.77	25.43	180.20
150	50.61	29.40	181.98
160	44.72	28.92	163.92
170	39.94	25.49	154.16
180	42.05	27.70	160.96
190	52.41	37.13	189.22
200	51.84	36.84	203.00

(b) RuleFit.

d	Nb rules	Interpretability	$\mathbb{P}_M(\mathcal{S}^* \subseteq \mathcal{S}_n)$	$\mathbb{P}_M(\mathcal{S}_n \subseteq \mathcal{S}^*)$	$\frac{1}{M}\sum_{s=1}^{M} \mathcal{H}(\mathcal{S}_n^s, \mathcal{S}^*)$
10	4.95	7.17	0.75	0.02	3.08
20	5.21	7.58	0.76	0.04	3.47
30	5.14	7.38	0.69	0.00	3.75
40	5.10	7.13	0.63	0.01	3.89
50	4.96	6.89	0.58	0.02	3.98
60	5.11	7.32	0.67	0.01	3.85
70	4.87	6.75	0.59	0.03	3.98
80	4.94	7.01	0.64	0.01	3.66
90	4.99	7.05	0.57	0.03	3.96
100	5.09	7.30	0.67	0.02	3.79
110	4.99	7.16	0.67	0.02	3.74
120	5.01	7.16	0.66	0.03	3.98
130	5.18	7.37	0.72	0.00	3.82
140	5.07	7.04	0.55	0.01	4.31
150	5.07	7.04	0.61	0.03	4.07
160	4.92	6.89	0.62	0.02	3.79
170	5.01	6.98	0.63	0.02	4.07
180	5.11	7.34	0.75	0.03	3.79
190	5.04	7.00	0.62	0.02	4.12
200	5.10	7.10	0.60	0.00	3.94

(c) RICE.

Table 3.3 – Average over the $M = 100$ simulations for each algorithm as a function of d.

Execution 2

We set $d = 100$ and we simulate $\mathbf{X} \sim \mathcal{U}([0,1], d)$. We generate $n = 2000$ points to train the algorithms. The aim of the second simulation is to show the behavior of the algorithms when the number of rules $n_{rules} \in \{1, 2, \ldots, 8\}$ increases. And we set

$$g^*(\mathbf{X}) = \sum_{i=1}^{n_{rules}} \mathbf{1}_{\mathbf{X} \in \mathbf{r}_i^*},$$

where each $\mathbf{r}_i^*$ is generated as in 3.6.2. For each value of n_{rules} we realize 100 simulations.

Results

In this application, we test the ability to identify a model more and more complex. As shown in Figure 3.4 the accuracy of all algorithm decrease with the complexity of the model. Moreover, one can notice that the gap between algorithms increases too.

Table 3.4 sums up results of each algorithm. One can notice that the more complex the model the less interpretable the output of algorithms. Nevertheless, RICE keeps an interpretability index largely lower than other algorithms.

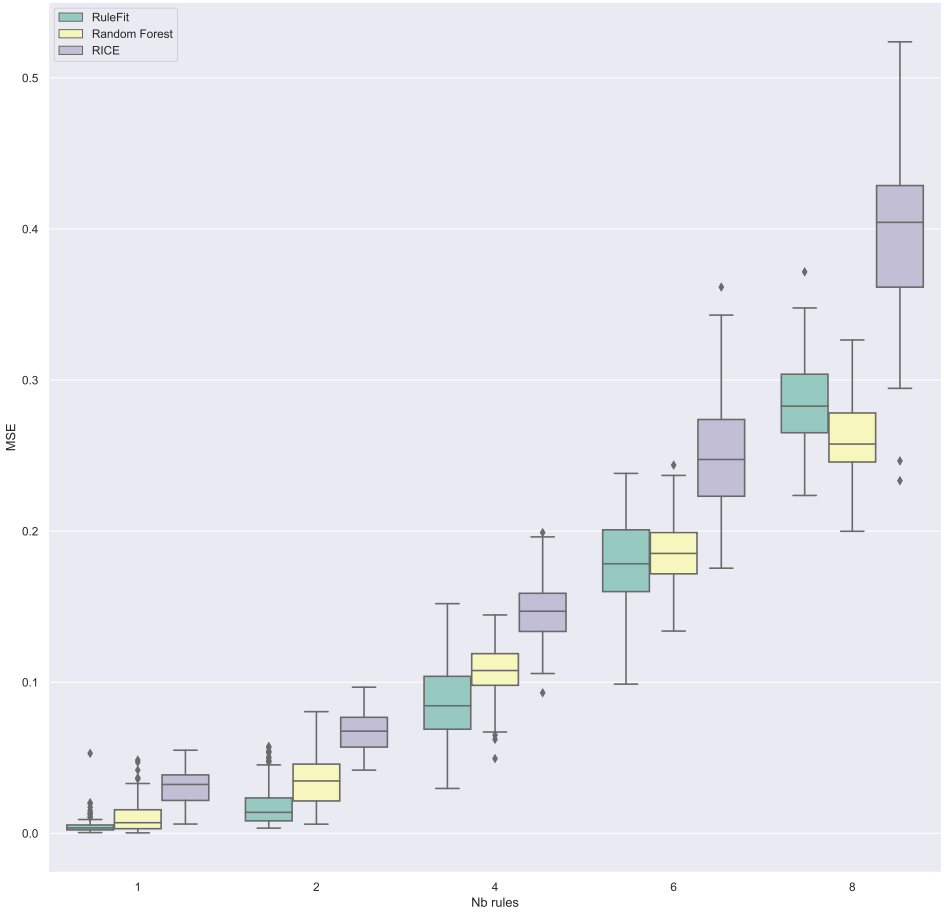


Figure 3.4 – Box-plot of the MSE^* of each algorithm.

n_{rules}	Nb rules	Interpretability
1	7453.92	33069.08
2	6951.94	29393.94
3	6548.28	27312.80
4	6810.78	29110.78
5	7052.20	30901.66
6	7395.16	32909.44
7	7517.36	34048.92
8	7821.02	35803.04

(a) Random Forest.

n_{rules}	Nb rules	Nb linear	Interpretability
1	41.02	18.87	142.35
2	88.04	17.70	282.91
3	157.33	15.16	475.82
4	208.65	13.61	609.67
5	248.38	11.68	715.55
6	262.12	8.32	740.33
7	295.27	9.37	824.32
8	317.22	9.69	880.78

(b) RuleFit.

n_{rules}	Nb rules	Int	$\mathbb{P}_M(\mathcal{S}^* \subseteq \mathcal{S}_n)$	$\mathbb{P}_M(\mathcal{S}_n \subseteq \mathcal{S}^*)$	$\frac{1}{M}\sum_{s=1}^{M} \mathcal{H}(\mathcal{S}_n^s, \mathcal{S}^*)$
1	4.99	7.13	0.58	0.01	4.04
2	5.32	7.52	0.26	0.06	4.27
3	6.55	9.71	0.14	0.07	5.53
4	7.91	12.02	0.10	0.05	5.88
5	8.94	13.81	0.02	0.03	6.91
6	10.40	16.49	0.03	0.04	7.47
7	12.87	20.75	0.01	0.01	8.95
8	15.30	25.36	0.00	0.01	10.40

(c) RICE.

Table 3.4 – Average over the $M = 100$ simulations for each algorithm as a function of n_{rules}, the number of rules used to create g^*.

3.7 Conclusion

In this chapter, we have introduced an interpretable algorithm, RICE, based on the notion of approximately suitable sequence of data-dependent covering. It is an empirical extension of that defined in Chapter 2. RICE is a rules-based algorithm thought to be deterministic to ensure stability of the output. It designs rules satisfying a coverage condition (3.8) and selects a sparse set of them, forming a covering (3.9) fulfilling approximately (in)significant conditions (3.10) and (3.11). This covering permits to generate a estimator of the regression function which is consistent under some conditions (see Theorem 3.4.1).

RICE has been compared to the algorithms Random Forest [7] and RuleFit [20]. These applications spotlight a link between interpretable behavior and support detection. Indeed, according to our interpretability index (Definition 3.1.1) the fewer the number of variables involved in the estimator the better the interpretability. Hence, to be interpretable and accurate algorithms have to select rules involving only variables of the support. Unfortunately, as explained in [76], an algorithm cannot generate a good estimator and identify the support at the same time.

The applications have shown that RICE is very efficient to identify the support of a simple model, as in Section 3.6.2, regardless of the dimension of the feature space. Its invariance to noise, its ability to identify informative variables and its theoretical results make it a promising algorithm.

Appendix: Algorithms of RICE

Algorithm 6: Discretize

Input:
— $\mathbf{X}$ the features
— m_n the number of bins

Output:
— $\tilde{\mathbf{X}}$ the discretized features

for $i = 1, \ldots, d$ **do**
  $\mathbf{x} \leftarrow \mathbf{X}[i]$;
  if $length(set(\mathbf{x})) > m_n$ **then**
    $bins \leftarrow [\,]$;
    for $j = 1, \ldots, m_n - 1$ **do**
      $p \leftarrow j/m_n$;
      $b \leftarrow get\_quantile(\mathbf{x}, p)$;
      $bins \leftarrow append(bins, b)$;
    $low \leftarrow min(\mathbf{x})$;
    $up \leftarrow max(\mathbf{x})$;
    for $j = 1, \ldots, n$ **do**
      $val \leftarrow 1$;
      for b *in* $bins$ **do**
        if $low \leq \mathbf{x}[j] < b$ **then**
          $\mathbf{x}[j] \leftarrow val$;
        $low \leftarrow b$;
        $val \leftarrow val + 1$;
      if $low \leq \mathbf{x}[j] \leq up$ **then**
        $\mathbf{x}[j] \leftarrow val$;
  $\tilde{\mathbf{X}}[i] \leftarrow \mathbf{x}$;
return $\tilde{\mathbf{X}}$;

Algorithm 7: Calc_length_1

Input:

— $\tilde{\mathbf{X}}$ the discretized features

— α the coverage rate parameter

— m_n the number of bins

Output:

— $\mathcal{R}$ the set of all rules of length 1 satisfying (3.8)

$\mathcal{R} \leftarrow \emptyset$;
for $i = 1, \ldots, d$ **do**
  for $b_{min} = 1, \ldots, m_n$ **do**
    for $b_{max} = b_{min}, \ldots, m_n$ **do**
      Set $\mathbf{r} \leftarrow \prod_{j=1}^{d} I_j$ with $\begin{cases} I_j = [1, m_n],\ j \neq i \\ I_i = [b_{min}, b_{max}] \end{cases}$;
      if $n(\mathbf{r}, \tilde{\mathbf{X}}) > n^{1-\alpha}$ **then**
        $\mathcal{R} \leftarrow append(\mathcal{R}, \mathbf{r})$;
return $\mathcal{R}$;

Algorithm 8: Calc_length

Input:

— $\tilde{\mathbf{X}}$ the discretized features

— α the coverage rate parameter

— $\mathcal{R}_1$ the k candidate rules of length 1

— $\mathcal{R}_{l-1}$ the k candidate rules of length $l-1$

Output:

— $\mathcal{R}_l$ a set of rules of length l satisfying (3.8)

$\mathcal{R}_l \leftarrow \emptyset$;
for $\mathbf{r}_1$ *in* $\mathcal{R}_1$ **do**
  for $\mathbf{r}_2$ *in* $\mathcal{R}_{l-1}$ **do**
    if $\mathbf{r}_2 \not\subseteq \mathbf{r}_1$ *AND* $\mathbf{r}_1 \not\subseteq \mathbf{r}_2$ **then**
      $\mathbf{r} \leftarrow \mathbf{r}_1 \cap \mathbf{r}_2$;
      if $n(\mathbf{r}, \tilde{\mathbf{X}}) > n^{1-\alpha}$ *and* $\ell(\mathbf{r}) = l$ **then**
        $\mathcal{R}_l \leftarrow append(\mathcal{R}_l, \mathbf{r})$;
  end
end
return $\mathcal{R}_l$

Appendix: Algorithm for the generation of random hyperrectangles

Algorithm 9: Random hyperrectangles generation

Input:

— d the dimension of features space;
— n_{rules} the number of rules;
— l_{max} the maximal length of a rule;
— c_{min} the minimal coverage rate of a rule;
— c_{max} the maximal coverage rate of a rule;

Output:

— $\mathcal{R}$ a set of hyperrectangles;

$\mathcal{R} \leftarrow \emptyset$;
for j *in* $[0, \ldots, n_{rules}]$ **do**
  $\mathbf{r} \leftarrow 1$;
  index $\leftarrow \emptyset$;
  l $\leftarrow \mathcal{U}([1, \ldots, l_{max}])$;
  for k *in* $[1, \ldots, l]$ **do**
    i $= \mathcal{U}([1, \ldots, d])$;
    while i *in* *index* **do**
      i $\leftarrow \mathcal{U}([1, \ldots, d])$;
    end
    bmin $\leftarrow \mathcal{U}([0, 1])$;
    while $bmin > 1 - c_{min}^{1/d}$ **do**
      bmin $\leftarrow \mathcal{U}([0, 1])$;
    end
    bmax$\leftarrow \mathcal{U}([0, 1])$;
    while $bmin + c_{max}^{1/d} \leq bmax \leq bmin + c_{min}^{1/d}$ **do**
      bmax $\leftarrow \mathcal{U}([0, 1])$;
    end
    index.append(i);
    $c_k \leftarrow \mathbf{1}_{X_i \in [bmin, bmax]}$;
  end
  $\mathbf{r} \leftarrow \prod_{k=1}^{l} c_k$;
  $\mathcal{R}$.append($\mathbf{r}$);
end
return $\mathcal{R}$

Conclusions and perspectives

In this thesis, we have introduced a family of rule-based algorithms generating a covering instead of an usual partition. The use of a covering makes it possible to decrease the number of selected rules needed to describe a model, thus creating a more interpretable output. We have introduced the notion of *interpretability index* to be able to compare the interpretable nature of algorithms.

Moreover, we have demonstrated that under some conditions on the selected rules forming a covering, we are able to get the weak consistency for the estimator based on the partition generated from this covering without any shrinkage condition on the diameter of the cells of the underlying partition. These conditions avoid an increasing number of selected rules when n tends to infinity. They are based on the notion of *significant* and *insignificant* rules, the significant rules being interpretable sets by construction and the insignificant rules being small sets with variances tending to zero.

During the last part of this thesis, we have developed an algorithm, RICE, with open-source code, based on the theory of covering algorithms. This algorithm designs rules and selects a sparse set of significant rules and completes the covering with insignificant rules. The significant rules are used to interpret the model and to identify the support. RICE makes an estimator defined on the partition generated from this covering which is, under some conditions, a consistent estimator of the regression function.

Nevertheless, the theoretical setting could be refined: unbounded Y may be considered by introducing a truncation operator as in [28], and strong consistency and rates of convergence of the data-dependent covering estimators may be established under slightly stronger assumptions. Furthermore, it could be interesting to have some conditions that would ensure a covering of the explanatory variables space. Finally, the scope could be broaden from the regression setting to the classification setting by adapting the significant condition accordingly.

The algorithm RICE could also be improved. The selection of k candidates necessary to increase the length of designing rules is based on the empirical risk, with the assumption that the intersection of two "good" rules should define a good rule-which is debatable. Indeed, two features could be noninformative individually taken but strongly explainable together. Conditions to guarantee a covering of the explanatory variables space would be an improvement too. Finally, the source code could be sharpen.

Finally, this thesis spotlights a link between interpretability and support detection. This link makes sense because the parsimony of variables involved in the prediction process is an important factor to be interpretable. Applications emphasize the interpretability property of RICE and thus its ability to detect the support.

Bibliography

[1] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM.

[2] Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79. Published in Statistics Surveys.

[3] Bagallo, G. and Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine learning*, 5(1):71–99.

[4] Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. (2019). Sirus: making random forests interpretable. *arXiv preprint arXiv:1908.06852*.

[5] Bertsimas, D. and Dunn, J. (2017). Optimal classification trees. *Machine Learning*, 106(7):1039–1082.

[6] Biran, O. and Cotton, C. (2017). Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8, page 1.

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

[8] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. CRC press.

[9] Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning and Games*. Cambridge university press.

[10] Cohen, W. (1995). Fast effective rule induction. In *Machine Learning Proceedings 1995*, pages 115–123. Elsevier.

[11] de Franco, C., Geissler, C., Margot, V., and Monnier, B. (2018). Esg investments: Filtering versus machine learning approaches.

[12] Dembczyński, K., Kotłowski, W., and Słowiński, R. (2008a). Solving regression by learning an ensemble of decision rules. In *International Conference on Artificial Intelligence and Soft Computing*, pages 533–544. Springer.

[13] Dembczyński, K., Kotłowski, W., and Słowiński, R. (2008b). Solving regression by learning an ensemble of decision rules. In *International Conference on Artificial Intelligence and Soft Computing*, pages 533–544. Springer.

[14] Denil, M., Matheson, D., and Freitas, N. (2013). Consistency of online random forests. In *International Conference on Machine Learning*, pages 1256–1264.

[15] Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

[16] Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *Machine Learning Proceedings 1995*, pages 194–202. Elsevier.

[17] Eineborg, M. and Boström, H. (2001). Classifying uncovered examples by rule stretching. In *International Conference on Inductive Logic Programming*, pages 41–50. Springer.

[18] Fayyad, U. M. and Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. 13th Int. Joint Conf. on Artificial Intelligence*, pages 1022–1027.

[19] Freitas, A. A. (2014). Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10.

[20] Friedman, J. and Popescu, B. (2008). Predective learning via rule ensembles. *The Annals of Applied Statistics*, pages 916–954.

[21] Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54.

[22] Fürnkranz, J., Gamberger, D., and Lavrač, N. (2012). *Foundations of rule learning*. Springer Science & Business Media.

[23] Fürnkranz, J. and Kliegr, T. (2015). A brief overview of rule learning. In *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, pages 54–69. Springer.

[24] Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE.

[25] Grunewalder, S. (2018). Plug-in estimators for conditional expectations and probabilities. In *International Conference on Artificial Intelligence and Statistics*, pages 1513–1521.

[26] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2019). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):93.

[27] Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., and Giannotti, F. (2018). A survey of methods for explaining black box models. *arXiv preprint arXiv:1802.01933*.

[28] Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. (2006). *A Distribution-Free Theory of Nonparametric Regression*. Springer Science & Business Media.

[29] Harrison Jr, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102.

[30] Hastie, T., Friedman, J., and Tibshirani, R. (2001). *The Elements of Statistical Learning*, volume 1. Springer series in statistics Springer, Berlin.

[31] Hayes-Michie, J. (1990). Pragmatica: Bulletin of the inductive programming special interest group. 1.

[32] Hoare, C. A. (1961). Algorithm 65: find. *Communications of the ACM*, 4(7):321–322.

[33] Holmes, G., Hall, M., and Prank, E. (1999). Generating rule sets from model trees. In *Australasian Joint Conference on Artificial Intelligence*, pages 1–12. Springer.

[34] Hyafil, L. and Rivest, R. (1976). Constructing optimal binary decision trees is np-complete. *Information processing letters*, 5(1):15–17.

[35] Indurkhya, N. and Weiss, S. M. (2001). Solving regression problems with rule-based ensemble classifiers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 287–292. ACM.

[36] Janssen, F. and Fürnkranz, J. (2011). Heuristic rule-based regression via dynamic reduction to classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1330.

[37] Karalič, A. and Bratko, I. (1997). First order regression. *Machine Learning*, 26(2-3):147–176.

[38] Kramer, S. (1996). Structural regression trees. In *AAAI/IAAI, Vol. 1*, pages 812–819. Citeseer.

[39] Langford, J., Oliveira, R., and Zadrozny, B. (2012). Predicting conditional quantiles via reduction to classification. *arXiv preprint arXiv:1206.6860*.

[40] Leech, W. (1986). A rule-based process control method with feedback. *Advanced in Intrumentation*, 41:169–175.

[41] Letham, B., Rudin, C., McCormick, T., and Madigan, D. (2015). Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371.

[42] Lindgren, T. and Boström, H. (2004). Resolving rule conflicts with double induction. *Intelligent Data Analysis*, 8(5):457–468.

[43] Lipton, Z. C. (2017). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

[44] Lugosi, G. and Nobel, A. (1996). Consistency of data-driven histogram methods for density estimation and classification. *The Annals of Statistics*, 24(2):687–706.

[45] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774.

[46] Margot, V., Baudry, J.-P., Guilloux, F., and Wintenberger, O. (2018). Rule induction partitioning estimator. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 288–301. Springer.

[47] Margot, V., Baudry, J.-P., Guilloux, F., and Wintenberger, O. (2019). Consistent regression using data-dependent coverings.

[48] Margot, V., Baudry, J.-P., Guilloux, F., and Wintenberger, O. (2020). Rule induction covering estimator : A new data dependent covering algorithm.

[49] Meinshausen, N. et al. (2010). Node harvest. *The Annals of Applied Statistics*, 4(4):2049–2072.

[50] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of machine learning*. MIT press.

[51] Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*.

[52] Nadaraya, E. (1964). On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142.

[53] Nobel, A. (1996). Histogram regression estimation using data-dependent partitions. *The Annals of Statistics*, 24(3):1084–1105.

[54] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[55] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

[56] Quinlan, J. R. (1993). C4.5: Programs for machine learning.

[Quinlan et al.] Quinlan, J. R. et al. Learning with continuous classes. World Scientific.

[58] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

[59] Rivest, R. L. (1987). Learning decision lists. *Machine learning*, 2(3):229–246.

[60] Rudin, C. (2018). Please stop explaining black box models for high stakes decisions. *arXiv preprint arXiv:1811.10154*.

[61] Scornet, E., Biau, G., Vert, J.-P., et al. (2015). Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741.

[62] Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*.

[63] Slocombe, S., Moore, K., and Zelouf, M. (1986). Engineering expert system applications. *Annual Conference of the BCS Special Group on Expert Systems*.

[64] Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In *Acm Sigmod Record*, volume 25, pages 1–12. ACM.

[65] Stone, C. (1977). Consistent nonparametric regression. *The Annals of Statistics*, pages 595–620.

[66] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

[67] Torgo, L. (1995). Data fitting with rule-based regression. In *Proceedings of the 2nd international workshop on Artificial Intelligence Techniques (AIT'95)*. Brno, Czech Republic.

[68] Torgo, L. and Gama, J. (1997). Regression using classification algorithms. *Intelligent Data Analysis*, 1(4):275–292.

[69] Tukey, J. (1961). Curves as parameters, and touch estimation. In *Proceedings of the 4th Berkeley Symposium*, volume 31, pages 681–694.

[70] Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press.

[71] Van Laer, W., De Raedt, L., and Dzeroski, S. (1997). On multi-class problems and discretization in inductive logic programming. In *International Symposium on Methodologies for Intelligent Systems*, pages 277–286. Springer.

[72] Vapnik, V. (1995). *The nature of statistical learning theory*. Springer Statistics for Engineering and Information Science.

[73] Vapnik, V. and Chervonenkis, A. (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pages 11–30. Springer.

[74] Vapnik, V. and Kotz, S. (1982). *Estimation of Dependences Based on Empirical Data*, volume 40. Springer-Verlag New York.

[75] Weiss, S. M. and Indurkhya, N. (1995). Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, 3:383–403.

[76] Yang, Y. (2005). Can the strengths of aic and bic be shared? a conflict between model identification and regression estimation. *Biometrika*, 92(4):937–950.

[77] Yu, B. et al. (2013). Stability. *Bernoulli*, 19(4):1484–1500.

[78] Yu, B. and Kumbier, K. (2019). Three principles of data science: predictability, computability, and stability (pcs). *arXiv preprint arXiv:1901.08152*.

[79] Yudkowsky, E. (2008). Artificial intelligence as a positive and negative factor in global risk. *Global catastrophic risks*, 1(303):184.

[80] Zhao, Q. and Bhowmick, S. S. (2003). Association rule mining: A survey. *Nanyang Technological University, Singapore*.

Appendix

Chapter 4

ESG Investments: Filtering Versus Machine Learning Approaches

Carmine de Franco [1], Christophe Geissler [2], Vincent Margot [3], Bruno Monnier [4].

We designed a machine learning algorithm that identifies patterns between ESG profiles and financial performances for companies in a large investment universe. The algorithm consists of regularly updated sets of rules that map regions into the high-dimensional space of ESG features to excess return predictions. The final aggregated predictions are transformed into scores which allow us to design simple strategies that screen the investment universe for stocks with positive scores. By linking the ESG features with financial performances in a non-linear way, our strategy based upon our machine learning algorithm turns out to be an efficient stock picking tool, which outperforms classic strategies that screen stocks according to their ESG ratings, as the popular best-in-class approach. Our paper brings new ideas in the growing field of financial literature that investigates the links between ESG behavior and the economy. We show indeed that there is clearly some form of alpha in the ESG profile of a company, but that this alpha can be accessed only with powerful, non-linear techniques such as machine learning.

Key words: Best-in-class approach, ESG, Machine Learning, Portfolio Construction, Sustainable Investments.

4.1 Introduction

The relationship between corporate social (CSP) and financial performances (CFP) is fairly an old theme in the economic research. In its earlier stages, it has met quite deep skepticism and critics: Nobel prize-winning economist Milton Friedman wrote in the New York Times Magazine, back in the 1970', that *"... there is one and only one social responsibility of business - to use its resources and engage in activities designed to increase its profits so long as it stays within the rules of the game, which is to say, engages in open and free competition without deception or fraud...."* [11].

1. PhD, Portfolio Manager at Ossiam, carmine.de-franco@ossiam.com.
2. CEO at Advestis, cgeissler@advestis.com.
3. PhD Candidate, vincent.margot@upmc.com.
4. CFA, Portfolio Manager at Ossiam, bruno.monnier@ossiam.com.

But the number of studies that highlights positive, or at least non-negative, relationship between social and financial performances has grown significantly since then, probably beginning with the initial work by [4] on the link between environmental virtue and financial performance. Fifty years latter the context has completely changed. The number of proponents of social and, more broadly, ESG integration in both corporate management and investors' choices has grown exponentially. And so has the number of financial products, funds and ETFs, that offer ESG versions of a large panel of investment strategies (mainly on equity and bonds).

The current approach now seems completely opposed to Friedman's one, and the most recent empirical literature highlights the link between ESG performance and alpha [7, 12, 28]. Nonetheless, the question regarding the relationship between CSP and CFP remains unanswered. Reviews of published paper (meta-analysis) highlight that the majority of empirical studies published on this theme reports non-negative or small positive relationship between CSP and CFP (see for example [20], [1], [27], [25], [17], [10]). Other researchers take a more optimistic view and report significant relationship between CSP and CFP [18, 9, 15] or at least that CSP is not detrimental to CFP as long as one manages to build the portfolio with care, even if there is no clear value added in ESG integration [16].

Although we do not share the very optimistic, and mostly overstated, enthusiasm about the direct relationship between ESG and financial performance, we do believe that there is a strong relationship between ESG and sustainability of corporate business. Therefore, ESG has an impact on financial performances and risks, but this does not come linearly. We welcome the efforts that investors are undertaking to include ESG criteria into their portfolio choices, and we clearly hope that this will trigger economic and cultural changes in corporate management. At the same time, we remain skeptic in front of the far too flaunted capability of basic ESG ratings to act as an alpha generator in a portfolio.

It remains true however that the very large set of ESG data, reports and analysis can contain useful information related to the strengths and weaknesses of corporations. Unfortunately, ESG ratings are, by construction, a composite measure that dramatically reduces this rich set of information.

Our contribution to the growing literature on this topic is to show that, empirically, there is no value added in portfolios based on simple ESG screenings. Although it usually comes with no harm to the performance, we do not find any alpha in such approaches. However, by recognizing the intrinsic value of the large panel of ESG indicators that are aggregated to form the ESG ratings, we show that it is possible to extract value from them, which, in turns, translates into real alpha. By exploring large data sets of specific ESG indicators, we are able to identify those who really have an impact on corporate financial performances. In a simplified example, we can agree on the fact that for a company in the utility sector, most likely, the environmental performance can be a discriminating criterion for financial performance; at the same time, governance can play an important role if we compare a utility company in Europe with one in an emerging country. Similarly, direct carbon emissions for banks are probably not as relevant as the exposures of these banks, through loans, to highly polluting companies would be. In short, aggregate measures as ESG ratings lose valuable information contained in the ESG indicators, which therefore lower their predictive power.

Searching for interesting patterns between specific ESG indicators and financial performance for a large set of companies remains out of reach for the standard tools available to econometricians. This search takes place in a very high-dimensional space and is not oriented by some a priori on these ESG features. To deal with this complexity, we developed a machine learning algorithm that allows us to identify features and patterns that are relevant to explain the link between CSP and CFP. The algorithm maps the regions in

our high-dimensional space of ESG features that have been consistently associated with outperformance or underperformance. In the econometric parlance, we look at those regions for which the conditional expectation of each stock's forward return is statistically positive (or negative), given that its relevant ESG features fall in these regions. We say that these relevant ESG features "activate" the region. By observing the ESG features we then obtain a significant signal regarding the future financial performance of the stock.

This identification is done with a set of rules that take the form of *If-Then* statements. The *If* statement identifies the region in the ESG space: in other words, the values that some ESG features have to take in order to activate the rule. The *Then* statement produces a prediction of the excess return, over the benchmark, that we can expect from a stock whose ESG features fall in that region. The final prediction is the aggregation of the predictions made by these rules and is transformed into a score $\{-1, 0, +1\}$. We therefore focus on the sign of the prediction of excess return rather than on its value. This usually makes the estimation more robust. The aggregation method mimics a panel of experts, each of which is expert on a very specific ESG feature (ex. Environment, Independence of the Board, ESG Reporting Verification, Employee Incidents, etc.) and makes a prediction given the ESG behavior of the company. When the aggregated prediction is close to zero, i.e the panel of experts is split between optimistic and pessimist forecasters, the final prediction is set at 0. The algorithm is regularly trained over time so that it can react and readjust to the new observed data.

The algorithm is used to design a very simple strategy that screens the investment universe and selects all stocks with a positive score. The resulting portfolio is compared with a classic ESG best-in-class portfolio, which consists of all stocks in the investment universe whose ESG ratings are above a given threshold within their peer groups. Our empirical results show that the simple machine learning screened portfolio significantly outperforms the ESG best-in-class approach and the benchmark.

This is in line with the economic belief that ESG data is valuable information to assess financial performance, but also confirms that aggregated ESG ratings are not suited to distinguish between outperformers and underperformers over the long run. Even if perfect distinction is out of reach, our results clearly point out the fact that there is alpha in the granular ESG data, but the relation between ESG and financial performance is definitely not linear. Furthermore, the predictive power of the scores vanishes with time. We proof indeed that regularly training the algorithm over time, and producing up-to-date sets of rules, are key components of the superior performance of the machine learning when it comes to stocks screening.

4.2 Data

The analyses in this paper are carried out on portfolios based on the investment universe defined by the capitalization-weighted MSCI World Index USD, that consists of the largest capitalization listed in the US, Canada, Western Europe, Japan, Australia, New Zealand, Hong Kong and Singapore. Portfolios are calculated in USD and net dividends are reinvested in the portfolio itself. Stock prices and dividends are taken from Thomson Reuters/Datastream. We reconstruct a proxy of the MSCI World Index by using end-of-month compositions as well as proxies for the US, Europe [5] and Asia Developed [6] benchmarks. We also consider sector portfolios derived from the MSCI World Index and the regional benchmarks by filtering on stocks that belong to the same sector: Consumer

5. Stocks in the MSCI World Index domiciled in Austria, Belgium, Denmark, Finland, France, Germany, Ireland, Italy, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom

6. Stocks in the MSCI World Index domiciled in Australia, Hong Kong, Japan, New Zealand and Singapore

Staples (CS), Consumer Discretionary (CD), Energy (EN), Financials (FI), HealthCare (HC), Industrials (IN), Information Technology (IT), Materials (MA), Telecommunication Services (TL), Utilities (UT).
For each company in the investment universe, we collect ESG ratings from Sustainalytics[7]. An ESG rating is a comprehensive measure based on three pillars, Environment, Social and Governance, that assesses the strengths and weaknesses of a company along these three directions. The pillars are themselves based on a large set of specific indicators. For the purposes of this study, the composite ESG rating is the arithmetic average of the three ratings Environment (E), Social (S) and Governance (G), each of which is itself the combination of roughly 50 narrower indicators. Finally, for each company, we consider its relative Peer Group, which consists of all companies with a similar business, hence comparable from a sustainability point of view.

4.3 The best-in-class approach

One of the most popular approaches to embed ESG criteria in the portfolio construction process is the so-called best-in-class approach. Given a threshold x, one excludes the stocks whose ESG ratings belong to the lowest x-quantile. The exclusion is usually carried across peer groups, i.e., groups of stocks with very similar characteristics. The reason behind this is twofold:

- Removing stocks with low ESG ratings within peer groups insures that the final economic mesh of the filtered universe remains similar to the initial investment universe.
- ESG ratings have a structural, sector-driven bias which usually favor specific sectors (ex. IT or HealthCare sectors) while penalizing others (ex. Energy or Utilities). Given this bias, the filtering over peer groups makes comparisons of ESG ratings independent of the sectors.

For the purpose of this study, an ESG best-in-class portfolio derived from a capitalization-weighted portfolio removes, within each peer group, the stocks whose ratings belong to the lowest x-quantile. The portfolio is finally scaled to sum up to one.
This approach, quite popular among investors, should not be thought of as a way to enhance performance. As Tables 4.1–4.4 show, ESG best-in-class filters applied to standard capitalization-weighted indexes do not bring outperformance.
Except for Europe and relatively low threshold levels, we find small but negative excess returns and negative information ratios for the ESG best-in-class portfolios over their benchmarks with almost unchanged risks. Although the approach does not create outperformance per se, it does not carry structural under-performance either. Optimistically, one could accept the fact that embedding ESG objectives in a portfolio does not significantly modify its risk/return profile.

Our findings are not in contradiction with the large literature that finds positive links between ESG and financial performance. But the consistency and durability over time of the ESG factor has been questioned since the very beginning. [3] find no significant relationship between social responsibility and corporate profitability, and similar results were obtained in [5] and [14]. [13] report that correlation between financial performance and social performance depends on the measure used to distinguish between high and low

7. One of the largest provider of ESG ratings.

Table 4.1 – World Developed

	ESG best-in-class			
	Bench.	10%	30%	50%
Ann. Performance	10.07%	10.01%	9.93%	9.51%
Ann. Volatility	13.34%	13.31%	13.44%	13.82%
Sharpe Ratio	0.73	0.73	0.72	0.67
Max. Drawdown	-21.91%	-21.79%	-22.02%	-22.57%
Information Ratio	0	-0.27	-0.25	-0.41

Table 4.2 – US

	ESG best-in-class			
	Bench.	10%	30%	50%
Ann. Performance	13.45%	13.25%	13.46%	13.2%
Ann. Volatility	14.61%	14.54%	14.49%	14.4%
Sharpe Ratio	0.9	0.89	0.91	0.9
Max. Drawdown	-18.99%	-18.87%	-18.71%	-18.04%
Information Ratio	0	-0.71	0.02	-0.18

Table 4.3 – Europe

	ESG best-in-class			
	Bench.	10%	30%	50%
Ann. Performance	6.37%	6.55%	6.47%	6.31%
Ann. Volatility	19.25%	19.19%	19.19%	19.29%
Sharpe Ratio	0.32	0.33	0.32	0.31
Max. Drawdown	-30.25%	-30.21%	-30.2%	-30.54%
Information Ratio	0	0.43	0.22	-0.11

Table 4.4 – Asia

	ESG best-in-class			
	Bench.	10%	30%	50%
Ann. Performance	6.83%	6.71%	6.41%	5.75%
Ann. Volatility	15.54%	15.71%	16%	16.2%
Sharpe Ratio	0.42	0.41	0.38	0.34
Max. Drawdown	-24.8%	-24.95%	-25.27%	-25.8%
Information Ratio	0	-0.21	-0.36	-0.46

Key performance indicators of the MSCI World Index and three capitalization-weighted regional benchmarks, together with ESG best-in-class filtered portfolios with different thresholds: 10%, 30% and 50%. Data is shown in USD from August 2009 to March 2018. Source MSCI, Datastream, Sustainalytics.

social performers.
Our results are more in line with [21] for which *"... the consideration of corporate social responsibility in stock market portfolios is neither a weakness nor a strength compared with conventional investments..."*. It should be noted that many fund managers and institutional investors surveys report that ESG is mostly looked as a risk mitigation tool in the first place [26], and eventually as a performance driver at longer horizon. We share the optimistic view of Nobel prize-winning economist Robert Shiller for which both society and the financial community would find the use of socially responsible practices mutually beneficial [22]. At the same time, we also believe that short to mid term financial performance is at best lowly correlated to ESG ratings, at least for such broad investment universes as the MSCI World Index (which contains more than 1,600 companies). We can list several reasons for this:

(i) The investment universes are relatively large and the aggregated ESG ratings have too little a signal-to-noise ratio to allow for an efficient selection of outperforming stocks.

(ii) ESG ratings are global metrics that embrace environmental, social and governance criteria. As such, they may be too reductive and we may lose a significant amount of information from the single indicator to the aggregated scores.

(iii) Granularity is key: As an example, it is likely that companies in specific sectors (ex. Energy) react differently to changes in the environment score (E) compared to the social score (S).

(iv) In the search for a rational economic theory behind ESG, some argue that by divesting low ESG rated companies, investors raise their cost of capital and, in turns, the return these companies have to offer to attract new investors. As such, in the short

run, they may show higher performances, but over time, the level of return they have to offer becomes unsustainable. Said otherwise, the action of divesting may take time to materialize in both investors' portfolios and low ESG rated companies (see for example [2]).

(v) The period considered in this study spans from the earlier stages of the recovery in 2009 to March 2018. Therefore, we are considering key performance indicators over a period of strong equity market, characterized by high returns and historically lower levels of volatility. This market regime can potentially affect the overall strength of ESG filtered portfolios.

To illustrate item (iii), we consider sector portfolios derived from the MSCI World Index and from the three regional benchmarks (US, Europe, Asia) and we apply both ESG and single pillar E, S and G, 30% best-in-class filtering. Tables 4.5–4.8 collect the results. For the sake of simplicity, we only show annualized excess returns over the relative benchmark sector portfolios and information ratios.

Overall, it is not straightforward to detect clear patterns between excess returns and ESG metrics conditionally to the regional benchmarks. But we can definitely detect specific triplets sector/region/metric that produce significant positive excess returns. Clearly, integrating ESG criteria in the Utilities sector enhances in-sample performances. But the right metric to use clearly depends on the geography: in the World Developed (Table 4.5) the best excess return for the Utilities sector is achieved when one uses the Governance (G) score only at 0.82%; in the US (Table 4.6) it is better to look at the composite ESG ratings which achieves 0.63%. In Europe (Table 4.7) it is the Environment score (E) that obtains the best result with 3.25% while in the Asia (Table 4.8) it is, once again, the composite ESG rating that achieves the highest excess return at 3.07%.

More generally, there is no sector nor metric for which the excess return of the best-in-class filtered sector achieves positive excess return in all the regions. Similarly, there is no region nor sector for which all metrics produce positive excess returns. Finally, no sector achieves positive excess returns across all regions and metrics. In other words, finding performance drivers when integrating ESG criteria in a best-in-class fashion is out of reach.

From Tables 4.5–4.8, only 12 out of 40 sector/metric portfolios in the World Developed region turn out to have positive excess return, and half of them are obtained when one considers the Governance (G) score. In the US we find positive excess returns in 14 out of 40, with no clear indication on the best metric to use. We notice though that all the metrics seem to work in the Utilities sector.

In Europe, we count 25 out of 40 sector/metric pairs with positive excess returns. For 4 sectors (Consumer Discretionary, Materials, Telecommunication Services and Utilities) all metrics work accurately. In Asia, we have 16 out of 40 portfolios with positive excess returns with no clear patterns between sectors and metrics, except for the Energy sector for which all metrics produce positive excess returns, even if their magnitudes are relatively small.

In conclusion, our empirical findings confirm that simple ESG filtering does not bring extra performance. Overall, it rather behaves as a small drag. Given the short period we consider, and the market regime that equity markets have experienced since 2009, we share the view that ESG best-in-class integration is, most likely, neutral to financial performance. Nevertheless, our results highlight the fact that geographies and sectors do not react to ESG criteria in the same way. But finding interesting and statistically significant patterns between ratings, pillars, their underlying narrow indicators (*features*) and financial performances, for more than 150 indicators on more than 1,600 companies

Table 4.5 – World Developed

	ESG	E	S	G
CS	-0.26% (-0.26)	-0.45% (-0.47)	-0.65% (-0.59)	**0.62%** **(0.53)**
CD	-0.33% (-0.3)	-0.46% (-0.46)	-0.65% (-0.4)	-1.22% (-0.65)
EN	-0.24% (-0.13)	-0.41% (-0.19)	-0.26% (-0.17)	**0.86%** **(0.31)**
FI	-0.51% (-0.43)	-0.46% (-0.42)	-0.87% (-0.63)	**0.19%** **(0.13)**
HC	-0.39% (-0.35)	-0.5% (-0.3)	-0.47% (-0.4)	-0.53% (-0.43)
IN	-0.31% (-0.28)	-0.32% (-0.27)	-0.32% (-0.31)	-0.33% (-0.19)
IT	-0.13% (-0.12)	-0.44% (-0.45)	-1.53% (-0.52)	**0.1%** **(0.05)**
MA	-0.09% (-0.05)	-0.09% (-0.03)	-0.17% (-0.08)	**0.32%** **(0.18)**
TL	**1.26%** **(0.64)**	**1.17%** **(0.74)**	**0.15%** **(0.06)**	**0.73%** **(0.21)**
UT	**0.16%** **(0.1)**	-0.94% (-0.39)	**0.72%** **(0.43)**	**0.82%** **(0.32)**

Table 4.6 – US

	ESG	E	S	G
CS	-0.94% (-0.84)	-0.46% (-0.43)	-0.46% (-0.32)	-0.26% (-0.12)
CD	-1.95% (-0.82)	-0.48% (-0.39)	-0.91% (-0.55)	-2.92% (-0.98)
EN	-0.09% (-0.06)	-0.66% (-0.44)	**0.01%** **(0.01)**	**0.55%** **(0.16)**
FI	-0.22% (-0.19)	-0.11% (-0.11)	-0.2% (-0.17)	**1.02%** **(0.67)**
HC	-0.23% (-0.22)	**0.19%** **(0.15)**	-0.55% (-0.5)	-0.58% (-0.37)
IN	**0.4%** **(0.54)**	**0.25%** **(0.32)**	**0.01%** **(0.01)**	-0.6% (-0.56)
IT	-1.36% (-0.96)	-0.7% (-0.75)	-1.76% (-0.51)	-0.37% (-0.11)
MA	**0.34%** **(0.18)**	**0.72%** **(0.32)**	-0.76% (-0.32)	-0.14% (-0.07)
TL	-0.38% (-0.37)	-0.32% (-0.38)	-0.4% (-0.2)	**1.32%** **(0.28)**
UT	**0.63%** **(0.56)**	**0.47%** **(0.38)**	**0.7% (0.7)**	**0.13%** **(0.12)**

Table 4.7 – Europe

	ESG	E	S	G
CS	**0.16%** **(0.14)**	**0.03%** **(0.03)**	-0.01% (-0.01)	**0.39%** **(0.32)**
CD	**0.51%** **(0.32)**	**0.45%** **(0.29)**	**0.18%** **(0.14)**	**1.07%** **(0.68)**
EN	-1.2% (-0.3)	-2.01% (-0.38)	-0.5% (-0.12)	-1.05% (-0.27)
FI	**0.31%** **(0.18)**	**0.04%** **(0.03)**	**0.33%** **(0.17)**	-0.28% (-0.21)
HC	-0.38% (-0.49)	-0.44% (-0.55)	-0.53% (-0.67)	-0.11% (-0.18)
IN	**0.03%** **(0.03)**	**0.03%** **(0.03)**	-0.39% (-0.35)	-0.67% (-0.44)
IT	-0.63% (-0.29)	-1.13% (-0.49)	**0.01%** **(0.08)**	-0.71% (-0.43)
MA	**0.12%** **(0.03)**	**0.23%** **(0.06)**	**0.51%** **(0.14)**	**0.6%** **(0.18)**
TL	**1.81%** **(0.67)**	**1.72%** **(0.63)**	**1.82%** **(0.62)**	**2.06%** **(0.71)**
UT	**1.79%** **(0.54)**	**3.25%** **(0.87)**	**0.09%** **(0.03)**	**0.41%** **(0.16)**

Table 4.8 – Asia

	ESG	E	S	G
CS	**0.6%** **(0.52)**	**0.37%** **(0.12)**	-0.3% (-0.28)	-0.18% (-0.09)
CD	-0.29% (-0.19)	-0.11% (-0.11)	**0.09%** **(0.04)**	**0.75%** **(0.37)**
EN	**0.12%** **(0.04)**	**0.34%** **(0.11)**	**0.03%** **(0.01)**	**0.25%** **(0.1)**
FI	-0.27% (-0.16)	-0.09% (-0.06)	-0.71% (-0.46)	-0.08% (-0.05)
HC	**0.29%** **(0.19)**	-0.67% (-0.27)	**0.14%** **(0.08)**	-0.03% (-0.02)
IN	-0.76% (-0.39)	-0.41% (-0.22)	-0.42% (-0.22)	-0.61% (-0.35)
IT	-1.32% (-0.59)	-0.92% (-0.4)	-1.01% (-0.43)	-0.96% (-0.28)
MA	-0.29% (-0.26)	**0.04%** **(0.03)**	-0.58% (-0.44)	-0.07% (-0.05)
TL	**0.17%** **(0.04)**	-0.18% (-0.11)	-0.61% (-0.09)	**1.83%** **(0.24)**
UT	**3.07%** **(0.65)**	-3.57% (-0.87)	**0.16%** **(0.04)**	**2.6%** **(0.55)**

Annualized excess returns (Information ratios) between capitalization-weighted sector portfolios and their ESG best-in-class filtered versions for the MSCI World Index and the derived regional benchmarks. In bold pairs sector/indicator for which the excess return is positive. Best-in-class filters are performed with the ESG rating together with the single pillars Environment (E), Social (S) and Governance (G) ratings. Data is shown in USD from August 2009 to March 2018. Source MSCI, Datastream, Sustainalytics.

in the MSCI World Index, over roughly 10 year, is out of reach for both human and linear statistic tools.

Next section introduces other techniques that can overcome this complexity and exploit this huge set of data.

4.4 Machine learning

In this section we introduce a deterministic, easily understandable machine-learning prediction algorithm, aimed at finding consistent and statistically significant patterns between ESG ratings and financial performances. The algorithm explores a high-dimensional data-set of ESG granular indicators for all the companies in our investment universe.

The goal of the algorithm, which falls in the category of supervised machine learning,

is to predict the (conditional) excess return of each company over the benchmark, given the specific values taken by some of its ESG indicators (the features). Stated differently, the algorithm identifies regions in the high-dimensional space of ESG features that are statistically related to financial outperformance or underperformance. Features include raw and derived ESG indicators [8], sector and country classifications, company's size and controversy indicators.

The regions are characterized by *rules* in the form *If-Then*, so that the algorithm finally consists of a set of such rules. The *If* statement is a list of conditions on the features $\mathbf{x}_t \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, where $\mathcal{X}_i$ is the set of possible outcomes of the feature i and $d = 447$ is the total number of features [9]. Therefore, a rule defines a hyper-rectangle of $\mathcal{X}$. The *Then* statement is the prediction of the 3-month forward excess return conditionally to the *If* statement. Since the rules correspond to hyper-rectangles in the feature space, we finally obtain relatively simple and understandable regions. Furthermore, to avoid over-fitting, the algorithm only selects a finite number of such rules. At each time t, the predictions of each rule are aggregated into one prediction, $\hat{y}_t$, through convex combination.
The algorithm is calibrated (*trained*) on the training set and the rules are used out-of-sample. The learning process works at two independent levels:

— At the end of year $N + 1$ we train the algorithm on an expanded data-set of features and stock total returns which contains the data-set used at the end of year N augmented of all the new observed data (features and stock total returns) from the end of year N to the end of year $N + 1$. To initialize the algorithm, we train it over 3 year of data (from 2009 to 2012). By expanding the data-set, the algorithm is able to access new data and explore new patterns, so that it can strengthen or nuance some rules that were previously discovered.

— On a daily basis, the algorithm can update the weights used to aggregate each rule's prediction, by over-weighting rules with a good prediction rate and under-weighting the others. Therefore, following day predictions will benefit from the *experience* the algorithm is gaining on the rules and their predictive power. The weight of each rule can be viewed as a confidence index. Of course, this is possible because the algorithm is able to assess the goodness of its predictions by looking at the realized 3-month return.

To avoid threshold effects, we transform the final prediction for each stock into a score $S \in \{-1, 0, +1\}$, where $+1$ stands for significantly positive excess return prediction, -1 for negative prediction and 0 for an uncertain prediction. The case where $S = 0$ is usually related to stocks for which some of their ESG indicators would eventually signal financial outperformance, while other ESG indicators rather signal potential undeperformance. The picture is then nuanced, and the algorithm cannot make a precise prediction. This is a very common situation in finance, where different indicators can yield different forecasts, so that, in aggregate, the forecast turns out to be uninformative.

The learning process is divided into two steps. Following [19] and [24], the training set D_N at the end of year N is divided into two sub-datasets: D_n the learning set and D_t the aggregation set, with $t >> n$ and $n + t = N$. The learning set D_n is used to design and

8. For each raw indicator, as for example the environment score (E), we also look at the derived indicator relative to the peer-group and the sector. All of these transformations can potentially contain useful information. On the other side, the use of both raw and derived indicators rapidly increases the dimension of the feature space d.

9. We use 164 ESG raw indicators, from which we derive peer group and sector relative indicators and 3 valuation indicators. In total 164*3 + 3 = 495. From these indicators we remove 48 indicators for which either the sector or the peer group derived indicators are too close, or for which historical data is missing.

select the set of rules used by the algorithm to make predictions. The aggregation set is used to fit the coefficients of the convex combination, in line with the *expert aggregation theory* of [6] and [23].

Independent Suitable Rules Let $D_N = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)) \in (\mathcal{X} \times \mathcal{R})^N$ be the training set. Here y_i denotes the 3-month return for some stock and $\mathbf{x}_i$ is the d-dimensional vector of its ESG features. The training set consists of a large but finite numbers of $(d+1)$-vectors spanning all stocks in the investment universe and all available dates. The training set $D_n \subseteq D_N$ includes the first n data points in D_N and $D_t = ((\mathbf{x}_{n+1}, y_{n+1}), \dots, (\mathbf{x}_N, y_N))$, the order being induced by the time.

Definition 4.4.1. *For any set $E \subset \mathcal{X}$, we define*

$$\mu(E, D_n) := \frac{\sum_{i=1}^{n} y_i \mathbf{1}_{\mathbf{x}_i \in E}}{\sum_{i=1}^{n} \mathbf{1}_{\mathbf{x}_i \in E}},$$

here, by convention, $\frac{0}{0} = 0$.

The set-valued map μ represents the conditional excess return of a stock over the benchmark, given that its ESG features $\mathbf{x}$ belong to E.

Definition 4.4.2. *Let $\mathbf{r}$ be a hyper-rectangle on $\mathcal{X}$: $\mathbf{r} = \prod_{k=1}^{d} I_k$ where each I_k is an interval of $\mathcal{X}_k$.*
A rule f is a function defined on $\mathbf{r} \times (\mathcal{X} \times \mathcal{R})^n$ as

$$f(\mathbf{x}, D_n) := \mu(\mathbf{r}, D_n) \, \forall \mathbf{x} \in \mathbf{r} \tag{4.4.1}$$

*The hyper-rectangle $\mathbf{r}$ is called the **condition** and $\mu(\mathbf{r}, D_n)$ is called the **prediction** of the rule f. The event $\{\mathbf{x} \in \mathbf{r}\}$ is called the **activation conditions** of the rule f.*

A rule f is completely defined by its condition $\mathbf{r}$. So, with an abuse of notation, we do not distinguish between a rule and its condition. We define two crucial numbers for a rule:

Definition 4.4.3. *Let f be a rule as in Definition 4.4.2 defined on $\mathbf{r} = \prod_{k=1}^{d} I_k$.*

1. *The* number of activations *of f in the sample D_n is*

$$n(\mathbf{r}, D_n) := \sum_{j=1}^{n} \mathbf{1}_{\mathbf{x}_j \in \mathbf{r}}.$$

2. *The* complexity *of f is*

$$cp(\mathbf{r}) = d - \#\{1 \leq k \leq d; I_k = \mathcal{X}_k\},$$

The algorithm does not consider all the possible rules, but only those with a given *coverage* and *significance*. We call these rules *suitable*, and their definition is given below.

Definition 4.4.4. *A rule f, defined on $\mathbf{r}$, is a* suitable rule *for the training set D_n if and only if it satisfies the two following conditions:*

1. ***Coverage condition.***

$$C_{min} \leq \frac{n(\mathbf{r}, D_n)}{n} \leq C_{max}, \tag{4.4.2}$$

with C_{min} and C_{max} are suitably chosen in the calibration step.

2. ***Significance condition.***

$$|\mu(\mathbf{r}, D_n) - \mu(\mathcal{X}, D_n)| \geq z(\mathbf{r}, D_n, \alpha), \tag{4.4.3}$$

for a chosen $\alpha \in [0, 1]$ *and function* z.

The coverage condition (4.4.2) excludes rules that are activated only on small sets (i.e with a low coverage rate, C_{min}) and rules that are too obvious (i.e with a high coverage rate, C_{max}). The threshold in the significance condition (4.4.3) is set such that the probability of falsely rejecting the null hypothesis $\mu(\mathbf{r}, D_n) = \mu(\mathcal{X}, D_n)$ is less than α. The parameter α permits to control the number of suitable rules. The higher α, the higher the number of suitable rules. In what follows, we generate rules of complexity $c \geq 2$ by a *suitable intersection* of rules of complexity 1 and rule of complexity $c - 1$.

Definition 4.4.5. *Two rules* f_i *and* f_j *defined on* $\mathbf{r}_i$ *and* $\mathbf{r}_j$ *respectively, form a* suitable intersection *if and only if they satisfy the two following conditions:*

1. ***Intersection condition:***

$$\begin{aligned} &\mathbf{r}_i \cap \mathbf{r}_j \neq \emptyset, \\ &n(\mathbf{r}_i \cap \mathbf{r}_j, D_n) \neq n(\mathbf{r}_i, D_n), \\ &n(\mathbf{r}_i \cap \mathbf{r}_j, D_n) \neq n(\mathbf{r}_j, D_n) \end{aligned} \tag{4.4.4}$$

2. ***Complexity condition:***

$$cp(\mathbf{r}_i \cap \mathbf{r}_j) = cp(\mathbf{r}_i) + cp(\mathbf{r}_j). \tag{4.4.5}$$

The intersection condition (4.4.4) avoids adding a useless condition for a rule. In other words, to define a suitable intersection, $\mathbf{r}_i$ and $\mathbf{r}_j$ must not be satisfied by the same points in D_n. The complexity condition (4.4.5) means that $\mathbf{r}_i$ and $\mathbf{r}_j$ have no marginal index in common.

Designing Suitable Rules The design of suitable rules is made recursively on their complexity. It stops at a complexity c if no rule is suitable or if the maximal complexity $c = cp_{max}$ is achieved.

Complexity 1: The first step is to find suitable rules of complexity 1. First notice that the complexity of evaluating all rules of complexity 1 is $O(ndm^2)$. Rules of complexity 1 are the base of the algorithm search heuristic. So all rules are considered and only suitable ones are kept, i.e rules that satisfied the coverage condition (4.4.2) and the significance condition (4.4.3). Since rules are considered independently, the search can be parallelized.

Complexity c: Among the suitable rules of complexity 1 and $c - 1$, we select M rules of each complexity (1 and $c - 1$) according to a chosen criterion. Then it generates rules of complexity c by pairwise *suitable intersection* according to the Definition 4.4.5. The complexity of evaluating all rules of complexity c, obtained from their intersections, is $O(nM^2)$. Here again, since rules are considered independently, the evaluation can be parallelized. The parameter M helps to control the computing time.

Selecting Suitable Rules We select a subset $\mathcal{S}$ from all suitable rules which maximizes the gains expected from rule in D_n and such as their conditions form a covering of $\mathcal{X}$.

Algorithm The calibration of the algorithm is structured in two parts: in the first one, it finds all suitable rules, and in the second one it retains only an optimal subset of it. To avoid threshold effects, overfitting and to manage the numerical complexity, we discretize each feature in $\mathcal{X}$ into m classes with empirical quantiles (modalities) [10]. Thus, each modality of each variable covers about $100/m$ percent of the sample. In practice, m must be inversely related to d: The higher the dimension of the problem, the smaller the number of modalities.

The parameters of the algorithm are:
- m, the sharpness of the discretization;
- $\alpha \in [0, 1]$, which specifies the false rejecting rate of the test;
- z, the significance function of the test;
- C_{max} and C_{min} the coverage bounds;
- cp_{max} the maximal complexity of a rule;
- and $M \in \mathbb{N}$, the number of rules of complexity 1 and $c-1$ used to define the rules of complexity c.

Aggregation Let $D_t = ((\mathbf{x}_{n+1}, y_{n+1}), \ldots, (\mathbf{x}_{n+t}, y_{n+t})) \in (\mathcal{X}, \mathcal{R})^t$, where $n + t = N$ be the aggregation set and let $\mathcal{S}$ be the set of R rules selected by the algorithm. At each time t, the predictions of each rule are aggregated into one prediction $\hat{y}_t$ as follows:

$$\hat{y}_t = \frac{\sum_{i=1}^{R} \pi_{i,t} f_i(\mathbf{x}_t, D_n)}{\sum_{i=1}^{R} \pi_{i,t} \mathbf{1}_{\mathbf{x}_t \in \mathbf{r}_i}} \tag{4.4.6}$$

with $\pi_{i,1} = 1/R$. When the realized value y_t is known, the weights $\pi_{.,t+1}$ are updated with the following formula

$$\pi_{i,t+1} = \pi_{i,t} \frac{e^{-\eta \ell(f_i(\mathbf{x}_t, D_n), y_t)}}{\sum_{i=1}^{R} e^{-\eta \ell(f_i(\mathbf{x}_t, D_n), y_t)}}, \tag{4.4.7}$$

with $\eta > 0$ and ℓ a convex loss function.

Remark 13. One can notice that $f_i(\mathbf{x}_t, D_n)$ is not defined if $\mathbf{x}_t \notin \mathbf{r}_i$. In (4.4.6), $\hat{y}_t$ is well defined for all t, since the set $\mathcal{S}$ is a covering of $\mathcal{X}$. In (4.4.7) we follow the methodology of the *sleeping expert aggregation* from [8].

Once trained, the machine learning algorithm produces predictions of the excess returns which are transformed into a scores $S \in \{-1, 0, +1\}$, given the out-of-sample ESG features $\mathbf{x}_t$ for each company. Table 4.9 shows some examples of rules taken from the learning process of the algorithm. The table lists three rules associated with positive predictions (opportunities) and five rules with negative predictions. Each rule consists of two features and two intervals. The "Relative To" properties indicate whether the feature must be calculated over all stocks in the universe (All), over a Sector, over a Peer Group, or whether we should look at the variations of the feature over time (Delta Score).

Whenever the values taken by the features for a given company fall in the given intervals (we say that the stock activates the rule) the algorithm makes a prediction on its excess return. It is important to remark that we aggregate all the predictions, and we transform the final aggregated prediction into a score $\{-1, 0, +1\}$, so that in the end we mainly look at the sign of the prediction rather than at its magnitude. We also remark that, while the set of rules remains unchanged for one year (until the next learning process), the output of the rules can change over time, because raw indicators can change and also because the aggregated weights of the rules change over time.

10. Of course such procedure is performed only on real-valued features with more than m different values. Categorical features are left unchanged.

Opportunity Rules: Positive excess return

Feature	Relative To	Activation Set	Rule Description
Business Ethics Incidents	Sector	[5, 9]	WHEN Business Ethics Incidents is high relative to sector AND Board Remuneration Disclosure is high relative to sector THEN Opportunity
Board Remuneration Disclosure	Sector	[5, 9]	
Board Independence	All	[9, 9]	WHEN Board Independence is at the maximum AND Board Remuneration Disclosure is high relative to sector THEN Opportunity
Board Remuneration Disclosure	Sector	[5, 9]	
Board Independence	All	[9, 9]	WHEN Board Independence is at the maximum AND Business Ethics Incidents is high relative to sector THEN Opportunity
Business Ethics Incidents	Sector	[5, 9]	

Risk Rules: Negative excess return

Feature	Relative To	Activation Set	Rule Description
Verification of ESG Reporting	Sector	[0, 7]	WHEN Verification of ESG Reporting is not high relative to sector AND Board Remuneration Disclosure is low relative to sector THEN Risk
Board Remuneration Disclosure	Sector	[0, 4]	
Quantitative Performance	All	[5, 9]	WHEN Quantitative Performance Score is high AND Board Remuneration Disclosure is low relative to sector THEN Risk
Board Remuneration Disclosure	Sector	[0, 4]	
Verification of ESG Reporting	All	[0, 6]	WHEN Verification of ESG Reporting is not high AND Quantitative Performance Score is high THEN Risk
Quantitative Performance	All	[6, 9]	
Gender Diversity of Board	Peer Group	[0, 8]	WHEN Gender Diversity of Board is not high relative to peer group AND Employee Incidents is very low relative to peer group THEN Risk
Employee Incidents	Peer Group	[0, 2]	
Green Logistics Programs	Delta Score	[0, 2]	WHEN Green Logistics Programs Delta Score is very low AND Qualitative Performance Delta Score is very low THEN Risk
Qualitative Performance	Delta Score	[0, 2]	

Table 4.9 – Some rules from the learning process of the algorithm at the end on 2012, 2013 and 2016. All features are discretized over 10 modalities (0 to 9) except for Qualitative Performance which is discretized over 6 modalities (0 to 5). High values for the features correspond to good ESG performance.

4.5 Machine learning application

We now test the predictive power of the machine learning algorithm developed in Section 4.4 compared with the classical best-in-class approach. More precisely, we try to assess whether filtering stocks over scores derived from the algorithm outperforms the standard filtering over ESG ratings (best-in-class). For the sake of simplicity, we only present the World Developed universe and, among the strategies presented in Section 4.3, we only consider the 30% best-in-class, as it is very close to what investors look at for their ESG portfolios. We recall that this strategy excludes, at each monthly review, the stocks whose ESG ratings are in the lower tercile within each peer group, and finally scale the weights so that their sum is one. To insure replicability of the portfolio, the ESG ratings are taken four days before the review date (which is end-of-month).
At the monthly review, we also build three portfolios based on the scores calculated with the machine learning algorithm, with the rules calculated at the end of the year that precedes the review:

Positive ML Screening: The portfolio selects all stocks in the investment universe whose scores are +1. The weights are finally scaled up to sum to one (maintaining

then the capitalization-weighting scheme of the benchmark)

Positive ML Screening Sector Matched: Same selection as for the Positive ML Screening portfolio, but the scaling of the weights is done in such a way that the final sector breakdown of the portfolio is matched to the benchmark's one.

Negative ML Screening: The portfolio selects all stocks in the investment universe whose scores are -1. The weights are finally scaled up to sum to one (maintaining then the capitalization-weighting scheme of the benchmark)

As before, the scores are taken four days before the review date. We consider the sector matched portfolio because the absolute screening usually introduces significant sector deviations with respect to the benchmark. Table 4.10 collects the main results for these portfolios since January 2013.

		ML Screening			ESG best-in-class
	Bench.	Positive	Positive Sect. Matched	Negative	30%
Ann. Performance	10.32%	13.07%	11.66%	8.31%	10.13%
Ann. Volatility	10.50%	11.14%	10.96%	10.95%	10.57%
Sharpe Ratio	0.94	1.14	1.03	0.72	0.92
Max. Drawdown	-18.07%	-14.99%	-16.46%	-22.47%	-17.91%
Information Ratio	-	1.01	0.58	-0.54	-0.32
Ann. Alpha	-	2.47%	1.15%	-1.81%	-0.24%

Table 4.10 – Key performance indicators of the MSCI World Index (Bench.), the capitalization-weighted selection filtered over positive scores from the ML algorithm, the one with the sector allocation matched to the benchmark, the one screened over negative scores and the 30% ESG best-in-class filtered portfolios. Data is shown in USD from January 2013 to March 2018. Source MSCI, Datastream, Sustainalytics.

Although we recognize that the period over which we can test the machine learning algorithm is relatively short (five years and three months), the results we obtain contain some interesting insights. First of all, the Positive ML Screening outperforms all the other portfolios: by 2.76% the benchmark on an annualized basis, by 2.94% the ESG best-in-class portfolio and by 4.77% the Negative ML Screening. And while the realized annual volatilities remain in the range 10.50% to 11.14%, there are significant differences in the realized maximum drawdowns: the Negative ML Screening shows a -22.47% loss from its peak, while the Positive ML Screening loss from its peak accounts for -14.99%.

These two combined results show that the machine learning algorithm is clearly able to distinguish between opportunity stocks (the ones with positive scores) from risky stocks (negative scores). Figure 4.1 shows the historical behavior of these two portfolios and the benchmark. We notice that the Positive ML Screening outperforms the Negative ML Screening over time, with the benchmark in between. Furthermore, in years when the benchmark shows very high performances with very low volatility, typically in bull market regimes, the differences between the two strategies are less pronounced. On the contrary, when the market is in bear regimes or it does not have clear trend, the Positive ML Screening clearly outperforms its negative counterpart, as shown in Table 4.11.
In years when the benchmark performance is very significant (2013 or 2017), the Positive ML Screening is still able to achieve some outperformance, but the spread with the Negative ML Screening is somehow lower than years when the market performance is negative or low (2014, 2015 and, most recently, 2018).

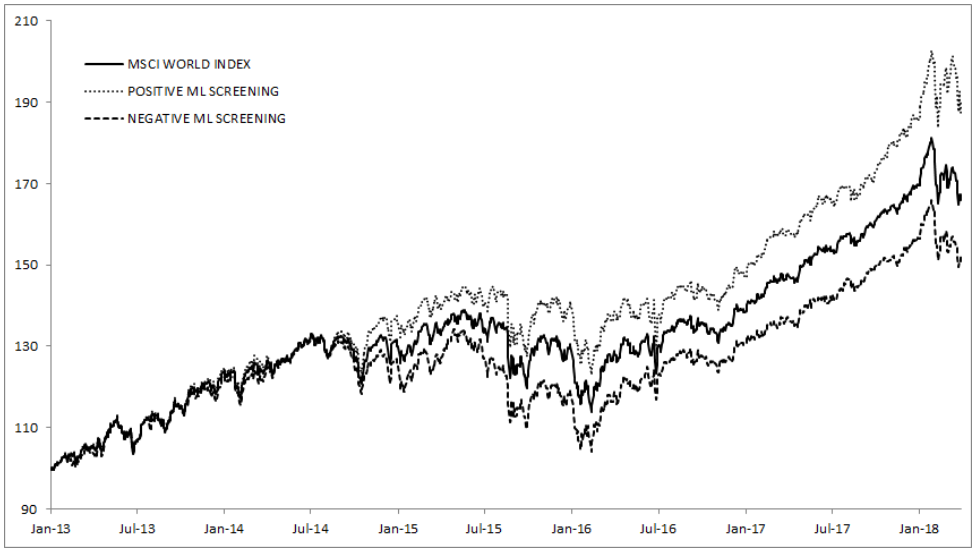


Figure 4.1 – Simulated strategy levels for the benchmark MSCI World Index, the capitalization-weighted selection filtered over positive scores from the ML algorithm (Positive ML Screening) and the one screened over negative scores (Negative ML Screening). Data in USD from January 2013 to March 2018. Base level = 100. Source MSCI, Datastream, Sustainalytics.

		Excess Return			
		ML Screening			ESG best-in-class
Year	Bench.	Positive	Positive Sect. Matched	Negative	30%
2013	23.95%	0.72%	0.12%	-1.39%	-0.14%
2014	4.97%	3.88%	3.65%	-2.75%	-0.17%
2015	-0.89%	3.79%	1.89%	-5.3%	0.07%
2016	7.4%	-2.14%	-1.91%	3.73%	-0.44%
2017	22.44%	3.82%	0.61%	-2.57%	-0.02%
2018	-1.37%	3.92%	2.24%	-1.63%	-0.24%

Table 4.11 – Calendar year performances for the MSCI World Index (Bench.) and the excess returns for the Positive, Positive Sect. Matched and Negative ML Screening as well as for the ESG 30% best-in-class portfolio. Data is shown in USD from January 2013 to March 2018. Source MSCI, Datastream, Sustainalytics.

Interestingly, the excess return of the sector matched version is also positive, even if lower in magnitude when compared to the Positive ML Screening. By neutralizing the sector component (because matched), the outperformance essentially comes from the stock picking.

For the Negative ML Screening, the excess return is always negative except for 2016. Finally, the best-in-class portfolio shows almost systematically small but negative excess returns, except in 2015 when it managed to outperform by 0.07%. Once again, our findings confirm the fact that for very large and diversified universes, the simple ESG filtering does not bring alpha, although it does not significantly reduce the performance with the best-in-class approach.

The effects of learning The machine learning algorithm is initially trained over three years of data and then yearly updated. During these regular updates, the algorithm learns from the new flow of data it can access: It can test its rules to confirm, nuance or remove some of them, and selects new rules linked to statistically significant patterns.

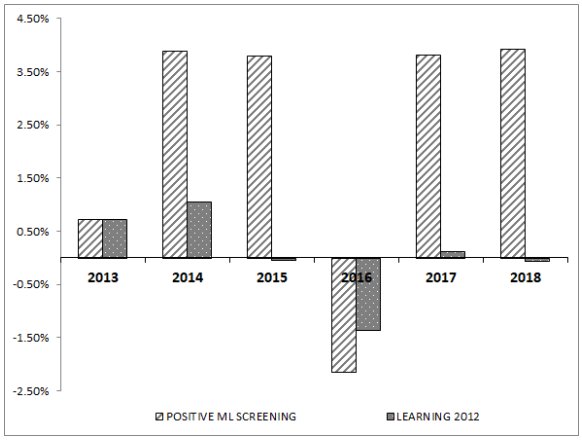
This learning process is key in the final performance of the model (and for the Positive ML Screening portfolio built upon it). To measure this effect, we form four portfolios named LEARNING Y, where Y = 2012, 2013, 2014, 2015 as follows:

— For each year Y, we consider the set of rules related to the learning at the end of the year Y.

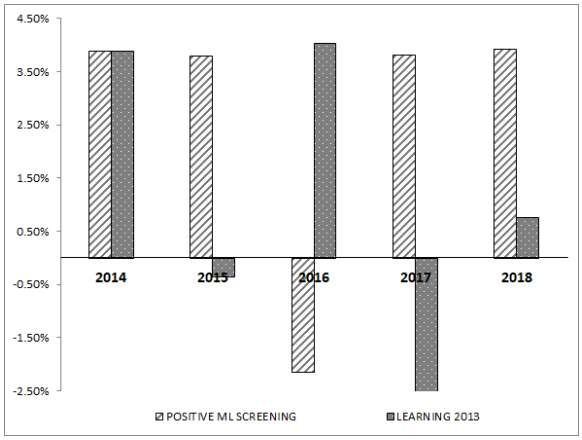
— We calculate the scores for all stocks in the universe from the end of year Y to March 2018 with this set of rules.

— LEARNING Y is built as Positive ML Screening, except that the underlying scores are calculated with the same, not updated set of rules calibrated at the end of year Y.

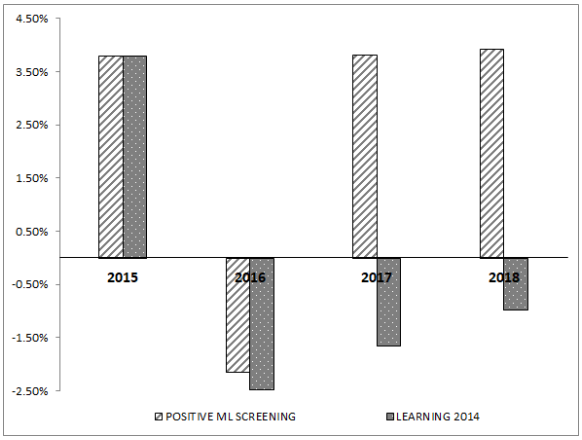
Said differently, LEARNING Y uses a unique, static set of rules that is never updated (no learning). By construction, the portfolios Positive ML Screening and LEARNING Y coincide over the period *January, 1st*, $Y+1$ to *December, 31st*, $Y+1$, because, over this period, they use the same set of rules (hence the same scores) to screen the investment universe. Figure 4.2 shows the calendar excess returns of these portfolios together with the Positive ML Screening portfolio over the benchmark MSCI World Index.



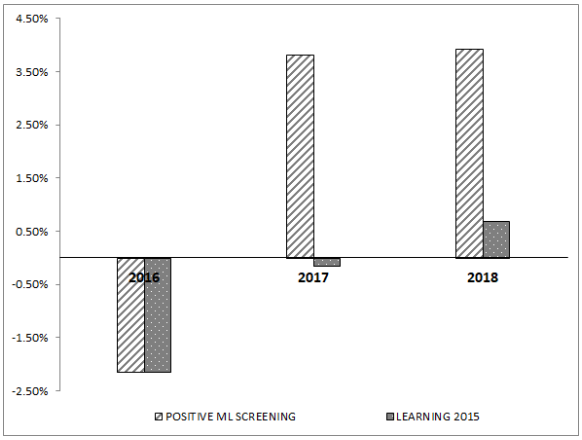
(a) Learning end of 2012



(b) Learning end of 2013



(c) Learning end of 2014



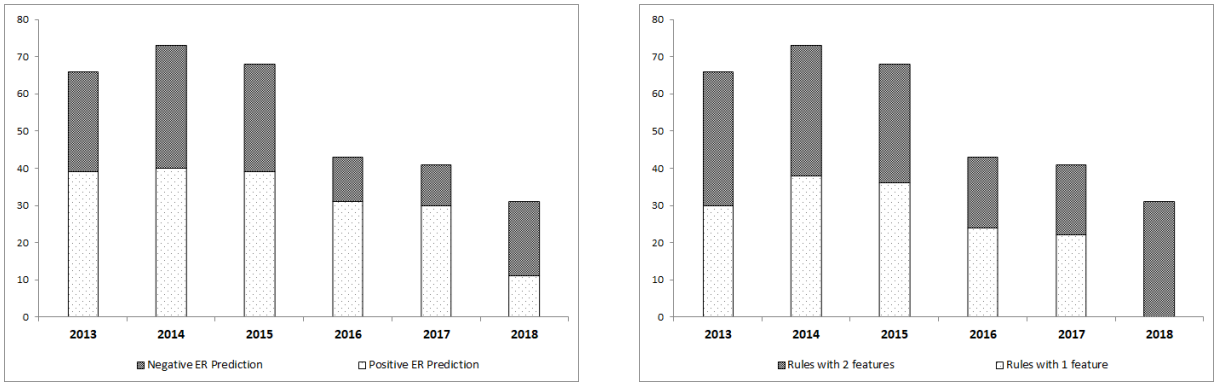
(d) Learning end of 2015

Figure 4.2 – Calendar excess returns of the Positive ML Screening and the four portfolios LEARNING 2012, LEARNING 2013, LEARNING 2014 and LEARNING 2015 over the MSCI World Index. Data is shown in USD from January 2013 to March 2018. Source MSCI, Datastream, Sustainalytics.

Since we only show out-of-sample results, the time frame of each LEARNING Y portfolio is different. In the majority of cases, we see that Positive ML Screening outperforms the LEARNING Y portfolios after the first year (since they are the same on the first year). Indeed, the excess return for the LEARNING Y portfolios usually shrinks to zero and

becomes even negative over time.

In other words, the predictive power of the scores vanishes over time, so that it is important to train the algorithm on the new observed data to update the set of rules.



(a) Positive vs. Negative

(b) Simple vs. Complex

Figure 4.3 – Number of rules at each update of the algorithm: (a) the split between rules that predict positive or negative excess returns (ER); (b) the split between rules that make use of one feature (Simple) or two features (Complex).

The number of rules used by the algorithm changes over time: as shown in Panel (a) of Figure 4.3, this number evolves in the range $[31, 73]$ with the split between positive rules (i.e., rules related to positive predictions of the excess return) and negative ones also changing over time. Interestingly, the number of rules related to negative excess return increased from 12 in 2016 to 20 in the latest 2018 learning. Panel (b) of Figure 4.3 shows the same number of rules split between simple rules (i.e., those that only make use of one feature) and complex rules (i.e., those that use two features, as the examples shown in Table 4.9). Both Figures 4.2-4.3 suggest that to extract alpha from the ESG features, one needs to regularly update the algorithm, and consider newly created set of rules to detect patterns between ESG profiles and financial performances.

4.6 Conclusion

The last few years have seen an increasing interest toward ESG investing and the integration of socially responsible principles at the portfolio construction level. Managers and investors are asked to complement pure financial objectives with extra-financial ones.

Our study brings some new ideas and insights onto the way investors could achieve ESG objectives in their investments. The literature on the theme is mixed: Initial studies were mostly skeptical on the benefit of ESG integration into the portfolio. Over time the mindset has evolved, and several studies have empirically proven that ESG integration in the portfolio does not lower performances. Most recently, the financial literature has gone one step further and claim that, indeed, ESG integration is a way to extract alpha or, at least, to reduce risks.

We do recognize the need for serious integration of ESG objectives alongside with classic financial ones, and that there exists an economic link between the ESG profile of a company and its financial performances over the long run. Nevertheless, we tend to agree with the pioneers of ESG research for which, at best, ESG integration does not significantly degrade financial performances, especially for large and diversified investment universes.

Because ESG profiles can impact financial performances in a non-linear way, and the impact can depend on the sector, the country or other specific characteristics of each company, we designed and implemented a sophisticated machine learning algorithm that

identifies patterns between ESG profiles and performances, statistically robust across the universe and over time.

The algorithm produces a set of rules, each rule identifying a region in the high-dimensional space of the ESG features, conditionally on which we can make a prediction on the stock's excess return. All the predictions are finally aggregated and transformed into a score taking values in $\{-1, 0, +1\}$, so that in the end we effectively look at the sign of the excess return rather than its magnitude.

With this algorithm, trained over time to keep it updated, we empirically proof that the link between ESG profiles and financial performances exists, but can only be accessed with non-linear techniques. Indeed a simple strategy that selects stocks whose scores are positive significantly outperforms the well known ESG best-in-class approach.

Bibliography

[1] Allouche, J. and Laroche, P. (2005). A meta-analytic investigation of the relationship between corporate social and financial performance. *Revue de gestion des ressources humaines, N. 57, pp. 18*

[2] Asness, C. (2017). Virtue is its Own Reward: Or, One Man's Ceiling is Another Man's Floor. *AQR Blog.* https://www.aqr.com/Insights/Perspectives/Virtue-is-its-Own-Reward-Or-One-Mans-Ceiling-is-Another-Mans-Floor

[3] Aupperle, K.E. and Carroll, A.B. and Hatfield, J.D. (1985). An empirical examination of the relationship between corporate social responsibility and profitability. *Academy of management Journal, Vol. 28, N. 2, pp. 446–463*

[4] Bragdon, J.H. and Marlin, J.A.T. (1972). Is pollution profitable? *Risk Management, Vol. 19, N. 4, pp. 9–18*

[5] Capelle-Blancard, G. and Monjon, S. (2012). Trends in the literature on socially responsible investment: Looking for the keys under the lamppost. *Business Ethics: A European Review, Vol. 21, N. 3, pp. 239–250*

[6] Cesa-Bianchi, N. and Lugosi, G. (2006) *Prediction, Learning and Games.* Cambridge university press

[7] Chong, J. and Phillips, G.M. (2016). ESG investing: A simple approach. *The Journal of Wealth Management Fall 2016, Vol. 19 N. 2, pp. 73–88*

[8] Devaine, M. and Gaillard, P. and Goude, Y. and Stoltz, G. (2013). Forecasting Electricity Consumption by Aggregating Specialized Experts. *Machine Learning, Vol. 90 N. 2, pp. 231–260*

[9] Filbeck, G. Holzhauer, H.M. and Zhao, X. (2014). Using social responsibility ratings to outperform the market: Evidence from long-only and active-extension investment strategies. *The Journal of Investing Spring 2014, Vol. 23 N. 1, pp. 79–96*

[10] Friede, G. and Bush, T. and Bassen, A. (2015). ESG and financial performance: aggregated evidence from more than 2000 empirical studies. *Journal of Sustainable Finance & Investment, Vol. 5, N. 4, pp. 210–233*

[11] Friedman, M. (1970). The social responsibility of business is to increase its profits. *The New York Times Magazine, September 13, 1970. Copyright 1970 by The New York Times Company.*

[12] Giese, G. and Ossen, A. and Bacon, S. (2016). ESG as a performance factor for smart beta indexes. *The Journal of Index Investing Winter 2016, Vol. 7, N. 3, pp. 7–20*

[13] Griffin, J.J. and Mahon, J.F. (1997). The corporate social performance and corporate financial performance debate: Twenty-five years of incomparable research. *Business & society, Vol. 36, N. 1, pp. 5–31*

[14] Humphrey, J.E. and Tan, D.T. (2014). Does it really hurt to be responsible? *Journal of business ethics, Vol. 122, N. 3, pp. 375–386*

[15] Indrani De, I. and Clayman, M.R. (2015). The benefits of socially responsible investing: An active manager's perspective. *The Journal of Investing Winter 2015, Vol. 24 N. 4, pp. 49–72*

[16] Kurtz, L. and Di Bartolomeo, D. (2011). The long-term performance of a social investment universe. *The Journal of Investing Fall 2011, Vol. 20 N. 3, pp. 95–102*

[17] Margolis, J.D. and Elfenbein, H.A. and Walsh, J.P. (2009). Does it pay to be good... and does it matter? A meta-analysis of the relationship between corporate social and financial performance. *Available at SSRN: https://ssrn.com/abstract=1866371orhttp://dx.doi.org/10.2139/ssrn.1866371*

[18] Peiris, D. and Evans, J. (2010). The relationship between environmental social governance factors and U.S. stock performance. *The Journal of Investing Fall 2010, Vol. 19, N. 3, pp. 104–112*

[19] Nemirovski, A. (2000). *Topics in Nonparametric.* Ecole d'Eté de Probabilités de Saint-Flour, Vol. 28, pp. 85

[20] Orlitzky, M. and Schmidt, F.L. and Rynes, S.L. (2003). Corporate social and financial performance: A meta-analysis. *Organization Studies, Vol. 24, N. 3, pp. 403–441*

[21] Revelli, C., and Viviani, J.L. (2015). Financial performance of socially responsible investing (SRI): What have we learned? A meta-analysis. *Business Ethics: A European Review, Vol. 24, N. 2, pp. 158–185*

[22] Shiller, R.J. (2013). Capitalism and financial innovation. *Financial Analysts Journal, Vol. 69, N. 1*

[23] Stoltz, G. (2010). Agrégation séquentielle de prédicteurs: méhodologie générale et applications á la prévision de la qualité de l'air et celle de la consommation électrique. *Journal de la Société Française de Statistique, Vol. 151, N. 2, pp.66–106*

[24] Tsybakov, A.B. (2003). *Optimal Rates of Aggregation.* Learning Theory and Kernel Machines, Springer, pp. 303-313

[25] Van Beurden, P. and Gössling, T. (2008). The worth of values – a literature review on the relation between corporate social and financial performance. *Journal of Business Ethics, Vol. 82, N. 2, pp. 407–424*

[26] Van Duuren, E. and Plantinga, A. and Scholtens, B. (2016). ESG Integration and the Investment Management Process: Fundamental Investing Reinvented. *Journal of Business Ethics, Vol. 138, N. 3, pp. 525–533*

[27] Wu, M.L. (2006). Corporate social performance, corporate financial performance, and firm size: A meta-analysis. *Journal of American Academy of Business, Vol. 8, N. 1, pp. 163–171*

[28] Zoltan, N. and Kassam, A. Lee, L.E. (2016). Can ESG add alpha? An analysis of ESG tilt and momentum strategies. *The Journal of Investing Summer 2016, Vol. 25, N. 2, pp. 113–124*

Chapter 5

Rule Induction Partitioning Estimator: Design of an interpretable prediction algorithm

Vincent Margot [1], Jean-Patrick Baudry [2], Frederic Guilloux [3], Olivier Wintenberger [4].

RIPE is a deterministic and easily *understandable* prediction algorithm developed for continuous and discrete ordered data. It infers a model, from a sample, to predict and to explain a real variable Y given an input variable $X \in \mathcal{X}$ (features). The algorithm extracts a sparse set of hyperrectangles $\mathbf{r} \subset \mathcal{X}$, which can be thought of as rules of the form *If-Then*. This set is then turned into a partition of the features space $\mathcal{X}$ of which each cell is explained as a list of rules with satisfied their *If* conditions.

The process of RIPE is illustrated on simulated datasets and its efficiency compared with that of other usual algorithms.

Key words: Machine learning, Data mining, Interpretable models, Rule induction, Data-Dependent partitioning, Regression models.

5.1 Introduction

To find an easy way to describe a complex model with a high accuracy is an important objective for machine learning. Many research fields such as medicine, marketing, or finance need algorithms able to give a reason for each prediction made. Until now, a common solution to achieve this goal has been to use induction rule to describe cells of a partition of the features space $\mathcal{X}$. A rule is an *If-Then* statement which is understood by everyone and easily interpreted by experts (medical doctors, asset managers, etc.). We focus on rules with a *If* condition defined as a hyperrectangle of $\mathcal{X}$.Sets of such rules have always been seen as decision trees, which means that there is a one-to-one correspondence between a rule and a generated partition cell. Therefore, algorithms for mining induction rules have usually been developed to solve the *optimal decision tree* problem [9]. Most of

1. vincent.margot@hotmail.com.
2. jean-patrick.baudry@upmc.fr.
3. frederic.guilloux@upmc.fr.
4. olivier.wintenberger@upmc.fr.

them use a greedy splitting technique [3, 12, 6, 5] whereas others use an approach based on Bayesian analysis [4, 10, 13].

RIPE (Rule Induction Partitioning Estimator) has been developed to be a *deterministic* (identical output for an identical input) and easily *understandable* (simple to explain and to interpret) predictive algorithm. In that purpose, it has also been based on rule induction. But, on the contrary to other algorithms, rules selected by RIPE are not necessarily disjoint and are independently identified. So, this set of selected rules does not form a partition and it cannot be represented as a decision tree. This set is then turned into a partition. Cells of this partition are described by a set of activated rules which means that their *If* conditions are satisfied. So, a same rule can explain different cells of the partition. Thus, RIPE is able to generate a fine partition whose cells are easily described, which would usually require deeper decision tree and less and less *understandable* rules. Moreover, this way of partitioning permits to have cells which are not a hyperrectangles.

The simplest estimator is the constant one which predicts the empirical expectation of the target variable. From it, RIPE searches rules which are *significantly* different. To identify these, RIPE works recursively, searching more and more complex rules, from the most generic to the most specific ones. When it is not able to identify new rules, it extracts a set of rules by an empirical risk minimization. To ensure a covering of $\mathcal{X}$, a *no rule satisfied* statement is added to the set. It is defined on the subset of $\mathcal{X}$ not covered by the union of the hyperrectangles of the extracted rules. At the end, RIPE generates a partition spanned by these selected rules and builds an estimator. But the calculation of a partition from a set of hyperrectangles is very complex. To solve this issue, RIPE uses what we called the *partitioning trick* which is a new algorithmic way to bypass this problem.

5.1.1 Framework

Let $(\mathbf{X}, Y) \in \mathcal{X} \times \mathbb{R}$, where $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, be a couple of random variables with unknown distribution P.

Definition 5.1.1. *1. Any measurable function $g : \mathcal{X} \to \mathbb{R}$ is called a* predictor *and we denote by $\mathbb{G}$ the set of all the predictors.*

2. The accuracy of g as a predictor of Y from $\mathbf{X}$ is measured by the quadratic risk, *defined by*

$$\mathcal{L}(g) = \mathbb{E}_{(\mathbf{X},Y)\sim\mathcal{P}}\left[(g(\mathbf{X}) - Y)^2\right]. \tag{5.1.1}$$

From the properties of the conditional expectation, the optimal predictor is the regression function (see [1, 7] for more details):

$$g^* := \mathbb{E}[Y|\mathbf{X}] = \arg\min_{g\in\mathbb{G}} \mathcal{L}(g) \text{ a.s.} \tag{5.1.2}$$

Definition 5.1.2. *Let $D_n = ((\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n))$ be a sample of independent and identically distributed copies of $(\mathbf{X}, Y)$.*

Definition 5.1.3. *The empirical risk of a predictor g on D_n is defined by:*

$$\mathcal{L}_n(g) = \frac{1}{n}\sum_{i=1}^{n} (g(\mathbf{X}_i) - Y_i)^2 \tag{5.1.3}$$

Equation (5.1.2) provides a link between prediction and estimation of the regression function. So, the purpose is to produce an estimator of g^* based on a partition of $\mathcal{X}$ that provides a good predictor of Y. However, the partition must be simple enough to be *understandable*.

5.1.2 Rule Induction Partitioning Estimator

The RIPE algorithm is based on rules. Rules considered in this paper are defined as follows:

Definition 5.1.4. *A rule is an* If-Then *statement such that its* If *condition is a hyperrectangle* $\mathbf{r} = \prod_{k=1}^{d} I_k$*, where each* I_k *is an interval of* $\mathcal{X}_k$*.*

Definition 5.1.5. *For any set* $E \subset \mathcal{X}$*, the empirical conditional expectation of* Y *given* $X \in E$ *is*

$$\mu(E, D_n) := \frac{\sum_{i=1}^{n} y_i \mathbf{1}_{\mathbf{x}_i \in E}}{\sum_{i=1}^{n} \mathbf{1}_{\mathbf{x}_i \in E}},$$

where, by convention, $\frac{0}{0} = 0$*.*

The natural estimator of g^* on any $\mathbf{r} \subset \mathcal{X}$ is the empirical conditional expectation of Y given $X \in \mathbf{r}$. A rule is completely defined by its condition $\mathbf{r}$. So, by an abuse of notation we do not distinguish between a rule and its condition.

A set of rules $\mathcal{S}_n$ is selected based on the sample D_n. Then $\mathcal{S}_n$ is turned into a partition of $\mathcal{X}$ denoted by $\mathcal{K}(\mathcal{S}_n)$ (see Section 5.2.1). To make sure to define a covering of the features space the *no rule satisfied* statement is added to the set of hyperrectangles.

Definition 5.1.6. *The* no rule satisfied *statement for a set or rules* $\mathcal{S}_n$*, is an* If-Then *statement such that its* If *condition is the subset of* $\mathcal{X}$ *not covered by the union of the hyperrectangles of* $\mathcal{S}_n$*.*

One can notice that it is not a rule according to the definition 5.1.4 because it is not necessarily defined on a hyperrectangle.

Finally, an estimator $\hat{g}^{\mathcal{S}_n}$ of the regression function g^* is defined:

$$\hat{g}^{\mathcal{S}_n} : (\mathbf{x}, D_n) \in \mathcal{X} \times (\mathcal{X} \times \mathbb{R})^n \;\mapsto\; \mu(K_n(\mathbf{x}), D_n), \tag{5.1.4}$$

with $K_n(\mathbf{x})$ the cell of $\mathcal{K}(\mathcal{S}_n)$ which contains $\mathbf{x}$.

The partition itself is *understandable*. Indeed, the prediction $\hat{g}^{\mathcal{S}_n}(\mathbf{x}, D_n)$ of Y is of the form "*If rules . . . are satisfied, then* Y *is predicted by . . .* " The cells of $\mathcal{K}(\mathcal{S}_n)$ are explained by sets of satisfied rules, and the values $\mu(K_n(\mathbf{x}), D_n)$ are the predicted values for Y .

5.2 Fundamental Concepts of RIPE

RIPE is based on two concepts, the *partitioning trick* and the *suitable rule*.

5.2.1 Partitioning Trick

The construction of a partition from a set of R hyperrectangles is time consuming and it is an exponential complexity operation and this construction occurs several times in the algorithm. To reduce the time and complexity we have developed the *partitioning trick*.

First, we remark that to calculate $\mu(K_n(\mathbf{x}), D_n)$, it is not necessary to build the partition, it is sufficient to identify the cell which contains $\mathbf{x}$. Figure 5.1 is an illustration of this process. To do that, we first identify rules activated by $\mathbf{x}$, i.e that $\mathbf{x}$ is in their hyperrectangles (Fig 5.1, to the upper left). And we calculate the hyperrectangle defined by their intersection (Fig 5.1, to the lower left). Then, we calculate the union of hyperrectangles of rules which are not activated (Fig 5.1, to the upper right). To finish, we calculate the cell by difference of the intersection and the union (Fig 5.1, to the lower right). The generated subset is the cell of the partition $\mathcal{K}(\mathcal{S}_n)$ containing $\mathbf{x}$.

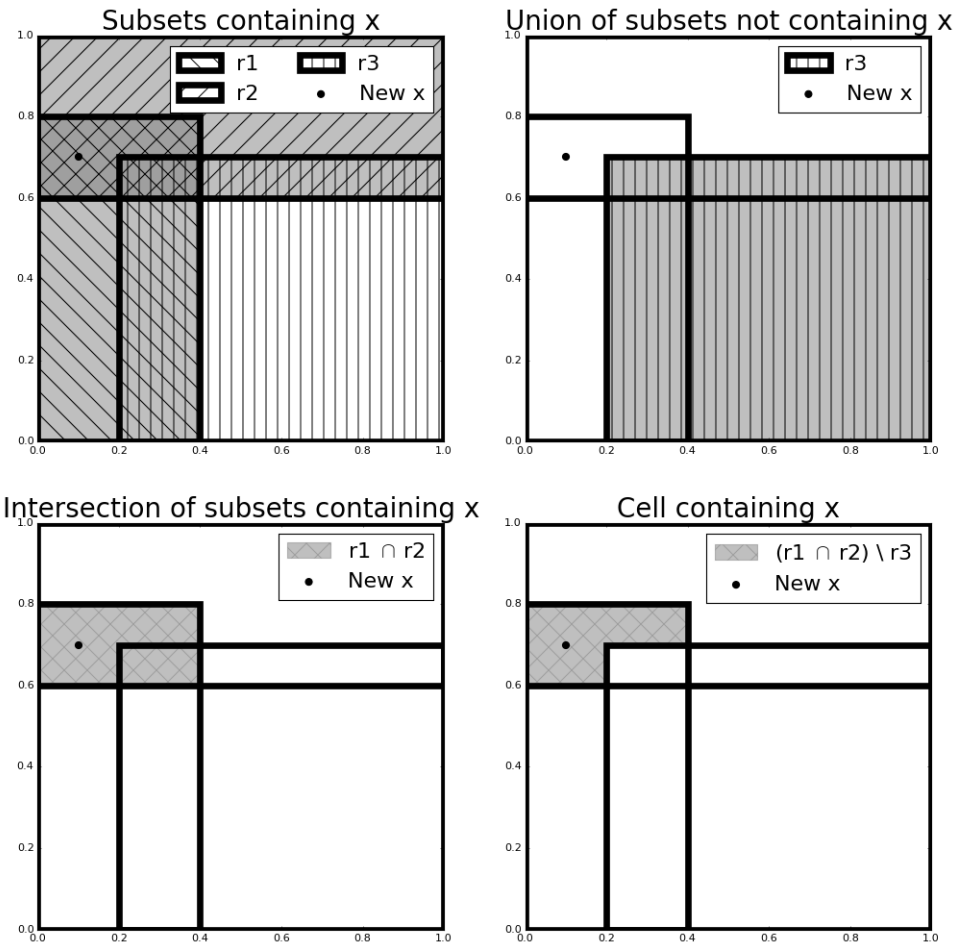


Figure 5.1 – Different steps of the *partitioning trick* for a set of three hyperrectangles $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$ of $[0,1]^2$ and a new observation $\mathbf{x} = (0.1, 0.7)$. It is important to notice that the cell containing $\mathbf{x}$, $(\mathbf{r}_1 \cap \mathbf{r}_2) \setminus \mathbf{r}_3$, is not a hyperrectangle so it does not define a rule.

Proposition 5.2.1. *Let $\mathcal{S}_n$ be a set of R rules selected from a sample D_n. Then, the complexity to calculate $\mu(K_n(\mathbf{x}), D_n)$ for a new observation $\mathbf{x} \in \mathcal{X}$ is $O(nR)$.*

Proof. It is sufficient to notice that $\mu(K_n(\mathbf{x}), D_n)$ can be express as follows:

$$\mu(K_n(\mathbf{x}), D_n) = \frac{\sum_{j=1}^{n} y_j k(\mathbf{x}, \mathbf{x}_j, \mathcal{S}_n)}{\sum_{j=1}^{n} k(\mathbf{x}, \mathbf{x}_j, \mathcal{S}_n)}, \tag{5.2.1}$$

with $k(\mathbf{x}, \mathbf{x}_j, \mathcal{S}_n) = \prod_{i=1}^{R} \left(\mathbf{1}_{\mathbf{x} \in \mathbf{r}_i} \mathbf{1}_{\mathbf{x}_j \in \mathbf{r}_i} + \mathbf{1}_{\mathbf{x} \notin \mathbf{r}_i} \mathbf{1}_{\mathbf{x}_j \notin \mathbf{r}_i} \right)$.

In (5.2.1), the complexity $O(nR)$ appears immediately. ■ □

5.2.2 Independent Suitable Rules

Each dimension of $\mathcal{X}$ is discretized into m_n classes such that

$$\frac{(m_n)^d}{n} \to 0, \qquad n \to \infty. \tag{5.2.2}$$

To do so empirical quantiles of each variable are considered (when it has more than m_n different values). Thus, each class of each variable covers about $100/m_n$ percent of the sample. This discretization is the reason why RIPE deals with continuous and ordered discrete variables only.

It is a theoretical condition. However, it indicates that m_n must be inversely related to d: The higher the dimension of the problem, the smaller the number of modalities. It is a way to avoid *overfitting*.
We first define two crucial numbers:

Definition 5.2.1. *Let* $\mathbf{r} = \prod_{k=1}^{d} I_k$ *be a hyperrectangle.*

1. *The* number of activations *of* $\mathbf{r}$ *in the sample* D_n *is*

$$n(\mathbf{r}, D_n) := \sum_{j=1}^{n} \mathbf{1}_{\mathbf{x}_j \in \mathbf{r}}. \tag{5.2.3}$$

2. *The* complexity *of* $\mathbf{r}$ *is*

$$cp(\mathbf{r}) = d - \#\{1 \leq k \leq d; I_k = \mathcal{X}_k\}, \tag{5.2.4}$$

We are now able to define a *suitable rule*.

Definition 5.2.2. *A rule* $\mathbf{r}$ *is a* suitable rule *for a sample* D_n *if and only if it satisfies the two following conditions:*

1. ***Coverage condition.***

$$\frac{n(\mathbf{r}, D_n)}{n} \leq \frac{1}{\ln(m_n)}, \tag{5.2.5}$$

2. ***Significance condition.***

$$|\mu(\mathbf{r}, D_n) - \mu(\mathcal{X}, D_n)| \geq z(\mathbf{r}, D_n), \tag{5.2.6}$$

for a chosen function z.

The coverage condition (5.2.5) ensures that the coverage ratio $n(\mathbf{r}, D_n)/n$ of a rule tends toward 0 for $n \to \infty$. It is a necessary condition to prove the consistency of the estimator which it is the purpose of a companion paper.
The threshold in the significance condition (5.2.6) is to ensure that the local estimators defined on subsets $\mathbf{r}$ is different than the simplest estimator which is the one who is identically equal to $\mu(\mathcal{X}, D_n)$.

RIPE generates rules of complexity $c \geq 2$ by a *suitable intersection* of rules of complexity 1 and rule of complexity $c - 1$.

Definition 5.2.3. *Two rules* $\mathbf{r}_i$ *and* $\mathbf{r}_j$ *define a* suitable intersection *if and only if they satisfy the two following conditions:*

1. ***Intersection condition:***

$$\begin{aligned} &\mathbf{r}_i \cap \mathbf{r}_j \neq \emptyset, \\ &n(\mathbf{r}_i \cap \mathbf{r}_j, D_n) \neq n(\mathbf{r}_i, D_n), \\ &n(\mathbf{r}_i \cap \mathbf{r}_j, D_n) \neq n(\mathbf{r}_j, D_n) \end{aligned} \tag{5.2.7}$$

2. ***Complexity condition:***

$$cp(\mathbf{r}_i \cap \mathbf{r}_j) = cp(\mathbf{r}_i) + cp(\mathbf{r}_j). \tag{5.2.8}$$

The intersection condition (5.2.7) avoids adding a useless condition for a rule. In other words, to define a *suitable intersection* $\mathbf{r}_i$ and $\mathbf{r}_j$ must not be satisfied for the same observations of D_n. And the complexity condition (5.2.8) means that $\mathbf{r}_i$ and $\mathbf{r}_j$ have no marginal index k; $\mathbf{1}_k \subsetneq \mathcal{X}_k$, in common of $\mathcal{X}$.

5.3 RIPE Algorithm

We now describe the methodology of RIPE for designing and selecting rules. The *Python* code is available at **https://github.com/VMargot/RIPE**.

The main algorithm is described as Algorithm 10. The methodology is divided into two parts. The first part aims at finding all suitable rule and the second one aims at selecting a small subset of suitable rules that estimate accurately the objective g^*.

The parameters of the algorithm are:

— m_n, the sharpness of the discretization, which must fulfill (5.2.2);
— $\alpha \in [0, 1]$, which specifies the false rejecting rate of the test;
— z, the significance function of the test;
— and $M \in \mathbb{N}$, the number of rules of complexity 1 and $c-1$ used to define the rules of complexity c.

Algorithm 10: Main

Global parameters: m_n, α, z and M;
Input:
— $(\mathbf{X}, \mathbf{y}) \in \mathbb{R}^{n(d+1)}$: data
Output:
— $\mathcal{S}$: the set of selected rules

Set $h_n = 1/\ln(m_n)$ the maximal coverage ratio of a rule;
$\tilde{\mathbf{X}} \leftarrow Discretize(\mathbf{X}, m_n)$ discretization in m_n modalities;
$\mathcal{R} \leftarrow Calc\_cp1((\tilde{\mathbf{X}}, \mathbf{y}), h_n)$;
for $c = 2, \ldots, d$ **do**
    $\mathcal{R}' \leftarrow Calc\_cpc((\tilde{\mathbf{X}}, \mathbf{y}), \mathcal{R}, c, h_n)$;
    if $len(\mathcal{R}') = 0$ **then**
        Break;
    else
        $\mathcal{R} \leftarrow append(\mathcal{R}, \mathcal{R}')$;
end
$\mathcal{R} \leftarrow Sort\_by\_risk(\mathcal{R}, (\tilde{\mathbf{X}}, \mathbf{y}))$;
$\mathcal{S} \leftarrow Select(\mathcal{R}, (\tilde{\mathbf{X}}, \mathbf{y}))$;
Return $\mathcal{S}$;

5.3.1 Designing Suitable Rules

The design of suitable rules is made recursively on their complexity. It stops at a complexity c if no rule is suitable or if the maximal complexity $c = d$ is achieved.

Complexity 1:

The first step is to find suitable rules of complexity 1. This part is described as Algorithm 11. First notice that the complexity of evaluating all rules of complexity 1 is $O(ndm_n^2)$.

Rules of complexity 1 are the base of RIPE search heuristic. So all rules are considered and just suitable are kept, i.e rules that satisfied the coverage condition (5.2.5) and the significance condition (5.2.6). Since rules are considered regardless of each others, the search can be parallelized.

At the end of this step, the set of suitable rules is sorted by their empirical risk (5.1.3), $\mathcal{L}_n(\hat{g}^{\{\mathbf{r}\}})$, with $\hat{g}^{\{\mathbf{r}\}}$ the predictor based on exactly one rule $\mathbf{r}$.

Algorithm 11: Calc_cp1

Global parameters: m_n, α and z;
Input:
— $(\mathbf{X}, \mathbf{y}) \in \mathbb{R}^{n(d+1)}$: data
— h_n: parameter
Output:
— $\mathcal{R}$: the set of all suitable rules of complexity 1;

$\mathcal{R} \leftarrow \emptyset$;
for $i = 1, \ldots, d$ **do**
  $\mathbf{x}_i \leftarrow \mathbf{X}[i]$, the ith feature ;
  for $b_{min} = 0, \ldots, m_n$ **do**
    for $b_{max} = b_{min}, \ldots, m_n$ **do**
      Set $\mathbf{r} = \prod_{k=1}^{d} I_k$ with $\begin{cases} I_k = [0, m_n],\ k \neq i \\ I_i = [b_{min}, b_{max}] \end{cases}$;
      if *is_suitable*$(\mathbf{r},\ (\mathbf{X}, \mathbf{y}),\ h_n,\ z,\ \alpha)$ **then**
        $\mathcal{R} \leftarrow append(\mathcal{R}, \mathbf{r})$;
    end
  end
end
Return $\mathcal{R}$

Complexity c:

Among the suitable rules of complexity 1 and $c-1$ sorted by their empirical risk (5.1.3), RIPE selects the M first rules of each complexity (1 and $c-1$). Then it generates rules of complexity c by pairwise *suitable intersection* according to the definition 5.2.3. It is easy to see that the complexity of evaluating all rules of complexity c obtained from their intersections is then $O(nM^2)$.

The parameter M is to control the computing time and it is fixed by the statistician. This part is described as Algorithm 12.

Algorithm 12: Calc_cp$c$

[t]
Global parameters: α, z and M;
Input:
— $(\mathbf{X}, \mathbf{y}) \in \mathbb{R}^{n(d+1)}$: data
— $\mathcal{R}$: set of rules of complexity up to $c-1$
— c: complexity
— h_n: parameter
Output:
— $\mathcal{R}_c$: set of suitable rules of complexity c
$\mathcal{R}_c \leftarrow \emptyset$;
$\mathcal{R} \leftarrow Sort\_by\_risk(\mathcal{R}, (\mathbf{X}, \mathbf{y}))$;
$\mathcal{R}_1 \leftarrow$ the M first rules of complexity 1 in $\mathcal{R}$;
$\mathcal{R}_{c-1} \leftarrow$ the M first rules of complexity $c-1$ in $\mathcal{R}$;
if $\mathcal{R}_1 \neq \emptyset$ *and* $\mathcal{R}_{c-1} \neq \emptyset$ **then**
  for $\mathbf{r}_1$ *in* $\mathcal{R}_1$ **do**
    for $\mathbf{r}_2$ *in* $\mathcal{R}_{c-1}$ **do**
      if *is_suitable_intersection*($\mathbf{r}_1$, $\mathbf{r}_2$) **then**
        Set $\mathbf{r} = \mathbf{r}_1 \cap \mathbf{r}_2$;
        if *is_suitable*($\mathbf{r}$, $(\mathbf{X}, \mathbf{y})$, h_n, z, α) **then**
          $\mathcal{R}_c \leftarrow append(\mathcal{R}_c, \mathbf{r})$;
    end
  end
Return $\mathcal{R}_c$

5.3.2 Selection of Suitable Rules

After designing suitable rules, RIPE selects an optimal set of rules. Let $\mathcal{R}_n$ be the set of all suitable rules generated by RIPE. The optimal subset $\mathcal{S}_n^* \subset \mathcal{R}_n$ is defined by

$$\mathcal{S}_n^* := \underset{\mathcal{S} \subset \mathcal{R}_n}{\arg\min}\, \mathcal{L}_n(\hat{g}^{\{\mathcal{S}\}}) \tag{5.3.1}$$

is the empirical risk (5.1.3) of the predictor based on $\mathcal{S}$.

Each computation of the empirical risk (5.3.1) requires the partition from the set $\mathcal{S}$ of rules, as described in Section 5.2.1. The complexity to solve (5.3.1) naively, comparing all the possible sets of rules, is exponential in the number of suitable rules.

To work around this problem, RIPE uses Algorithm 13, a greedy recursive version of the naive algorithm: it does not explore all the subsets of $\mathcal{R}_n$. Instead, it starts with a single rule, the one with minimal risk, and iteratively keeps/leaves the rules by comparing the risk of a few combinations of these rules. More precisely, suppose that

— $\mathbf{r}_1, \dots, \mathbf{r}_N$ are the suitable rules, sorted by increasing empirical risk;
— $\mathbf{r}_1, \dots, \mathbf{r}_k$, $k < N$, have already been tested;
— j of them, say $\mathcal{S} \subset \{\mathbf{r}_1, \dots, \mathbf{r}_k\}$ have been kept, the $k - j$ other being left.

Then $\mathbf{r}_{k+1}$ is tested in the following way :

— Compute the risk of $\mathcal{S}$, $\mathcal{S} \cup \{\mathbf{r}_{k+1}\}$ and of all $\mathcal{S} \cup \{\mathbf{r}_{k+1}\} \setminus \{\mathbf{r}\}$ for $\mathbf{r} \in \mathcal{S}$;
— Keep the rules corresponding to the minimal risk;
— Possibly leave *once for all*, the rule in $\{\mathbf{r}_1, \dots, \mathbf{r}_{k+1}\}$ which is not kept at this stage.

Thus, instead of testing the 2^N subsets of rules, we make N steps and at the k^{th} step we test at most $k + 2$ (and usually much less) subsets, which leads to a theoretical overall maximum of $O(N^2)$ tested subsets. The heuristic of this strategy is that rules with low risk are more likely to be part of low risk subsets of rules; and the minimal risk is searched in subsets of increasing size.

Algorithm 13: Select

Input:
— $\mathcal{R}$: set of rules sorted by increasing risk
— $(\mathbf{X}, \mathbf{y}) \in \mathbb{R}^{n(d+1)}$: data

Output:
— $\mathcal{S}$: subset of selected rules approaching the argmin (5.3.1) over all subsets of $\mathcal{R}$;

```
Set S = {R(1)};
for i = 2, ..., len(R) do
    Set 𝔖 = { S ; S∪{R(i)} };
    for j=1,...,len(S) do
        𝔖 ← append( 𝔖 ; S∪{R(i)}\{S(j)} )
    end
    𝔖 ← Sort_by_risk(𝔖, (X, y));
    S ← 𝔖(1)
end
Return S;
```

5.4 Experiments

The experiments have been done with *Python*. To assure reproducibility the *random seed* has been set at 42. The codes of these experiments are available in **GitHub** with the package RIPE.

5.4.1 Artificial Data

The purpose here it is to understand the process of RIPE, and how it can explain a phenomenon. We generate a dataset of $n = 5000$ observations with $d = 10$ features. The target variable Y depends on two features X_1 and X_2 whose are identically distributed on $[-1, 1]$. In order to simulate features assimilated to white noise, the others variables follow a centered-reduced normal distribution $\mathcal{N}(0, 1)$. The model is the following

$$y_i = F^*(\mathbf{x}_i) + \epsilon_i \tag{5.4.1}$$

with $\epsilon_i \sim \mathcal{N}(0, 1)$ and

$$F^*(\mathbf{x}_i) = -2 \times \mathbf{1}_{\{x_{1,i}^2 + x_{2,i}^2 > 0.8\}} + 2 \times \mathbf{1}_{\{x_{1,i}^2 + x_{2,i}^2 < 0.5\}} \tag{5.4.2}$$

The dataset is randomly split into training set D_m^1 and test set D_l^2 such as D_m^1 represents 60% of the dataset. RIPE uses significance test based on (5.5.2) with a threshold $\alpha = 0.05$ and $m_n = 5$.

On Figure 5.2, we have on the left, the true model (5.4.2) according to X_1 and X_2 with the realization of Y not used during the learning. On the right, the model inferred by RIPE.

On Figure 5.1 we represent the set of selected rules. In this case rules form a covering of the features space. So it is not necessary to add the *no rule satisfied* statement (see Def 5.1.6).

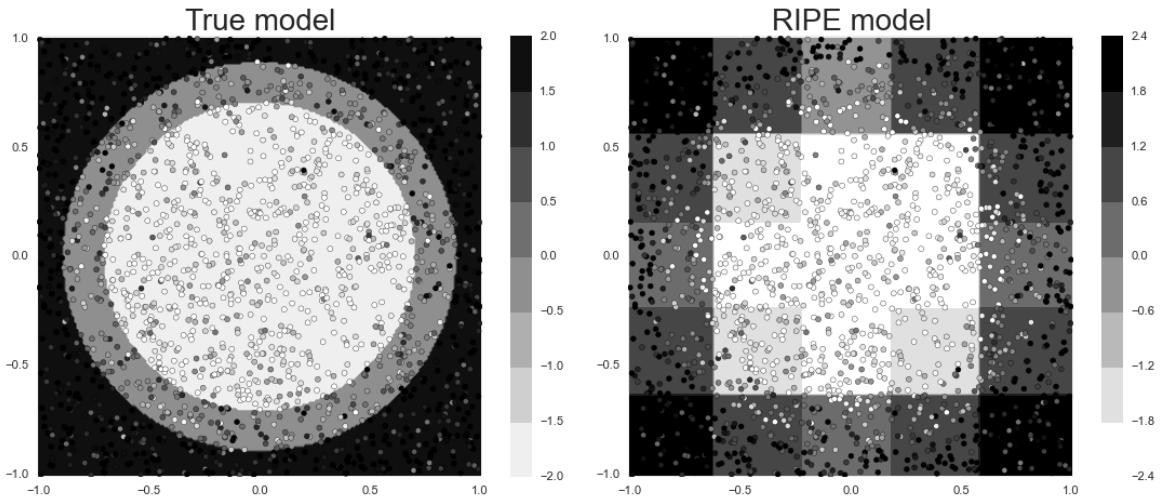


Figure 5.2 – The true model vs the model inferred by RIPE.

5.4.2 High Dimension Simulation

In this simulation, we use the function *make_ regression*, from the **Python** package *sklearn* ([54]), to generate a random linear regression model with n observations and d variables. Among these variables, p are informative and the rest are gaussian centered noise.

Rule	Conditions	Coverage	Prediction	Z	MSE
R 0(2)-	$X0 \in [1.0, 3.0]$ & $X1 \in [1.0, 3.0]$	0.36	-0.91	0.09	2.14
R 1(2)-	$X0 \in [1.0, 3.0]$ & $X1 \in [2.0, 4.0]$	0.36	-0.57	0.09	3.31
R 2(2)-	$X0 \in [2.0, 4.0]$ & $X1 \in [1.0, 3.0]$	0.35	-0.54	0.09	3.40
R 3(1)-	$X0 \in [2.0, 3.0]$	0.40	-0.46	0.08	3.48
R 4(1)-	$X0 \in [1.0, 2.0]$	0.40	-0.43	0.08	3.55
R 5(1)-	$X1 \in [1.0, 2.0]$	0.40	-0.43	0.08	3.55
R 6(1)+	$X0 = 4.0$	0.20	0.63	0.11	3.65
R 7(1)+	$X1 = 4.0$	0.20	0.56	0.12	3.74
R 8(1)+	$X1 \in [0.0, 1.0]$	0.40	0.19	0.08	3.95
R 9(1)+	$X0 \in [0.0, 1.0]$	0.40	0.15	0.08	3.99

Table 5.1 – Summary of selected rules with conditions interval express as modalities

Rule	Conditions	Coverage	Prediction	Z	MSE
R 1(2)-	$X_{976} \in [0.0, 2.0]$ & $X_{298} \in [0.0, 2.0]$	0.35	-0.83	0.28	7808.24
R 2(1)-	$X_{976} = 0.0$	0.20	-1.07	0.46	8907.38
R 3(1)+	$X_{976} = 4.0$	0.20	0.88	0.41	10081.34
R 4(2)-	$X_{976} \in [0.0, 1.0]$ & $X_{336} \in [0.0, 1.0]$	0.19	-0.89	0.40	10245.30
R 5(2)+	$X_{298} \in [2.0, 4.0]$ & $X_{976} \in [2.0, 3.0]$	0.24	0.65	0.27	10781.00
R 6(1)+	$X_{298} = 4.0$	0.20	0.73	0.43	10813.83
R 7(1)-	$X_{298} = 0.0$	0.20	-0.73	0.42	10822.09
R 8(2)-	$X_{298} \in [0.0, 1.0]$ & $X_{336} \in [0.0, 1.0]$	0.20	-0.66	0.38	11109.50
R 9(2)+	$X_{976} \in [2.0, 4.0]$ & $X_{564} = 4.0$	0.14	0.77	0.45	11253.10
R 10(2)+	$X_{976} \in [2.0, 4.0]$ & $X_{163} = 4.0$	0.13	0.75	0.44	11419.93
R 11(2)-	$X_{976} \in [0.0, 1.0]$ & $X_{945} = 2.0$	0.10	-0.87	0.51	11427.16
R 12(2)-	$X_{976} \in [0.0, 1.0]$ & $X_{733} = 1.0$	0.10	-0.84	0.58	11524.60
R 13(1)+	$X_{976} = 3.0$	0.20	0.55	0.31	11548.05
R 14	No rule activated	0.02	-0.35	0.45	12440.40

Table 5.2 – Summary of selected rules with conditions interval express as modalities

In this example we take $n = 500$, $d = 1000$ and $p = 5$ to simulate a noisy high dimensional problem. The data are randomly split into a training set and a test set, with a ratio of 60% \40%, respectively.

We use two others algorithms in this case: Decision Tree (DT) [3] without pruning and Random Forests (RF) [2], all from the package of python **sklearn** [11]. In order to evaluate the performance of our model, the normalized mean square error ($NMSE$) is computed.

Results are sum up in Table 5.3. Difference between the $NMSE$ of the training and the $NMSE$ of test indicates that Decision Tree and Random Forests overfit in this context. Conversely, RIPE infers a model which is more general (see Tab 5.3). Indeed, RIPE is able to describe the model with only 14 rules (see Tab 5.2) which have conditions on only seven variables from 1000. Among these selected variables only two are very important X_{976} and X_{298} (see Tab 5.4).

In this case, RIPE discretizes each variable in 5 modalities from 0 to 4. Table 5.2 presents the selected rules with their conditions. The rule $R14$ is the *no rule satisfied* statement (see Definition 5.1.6).

5. It is the mean of the number of rules of each tree

Algorithm	Parameters	$NMSE$ training	$NMSE$ test	Nb of rules	Complexity max
DT	/	0.0	0.46	350	14
RF	m_tree = 200 m_try = $d/3$	0.04	0.39	128.25 [5]	21
RIPE	M=300 z: see (5.5.2) α=0.05	0.13	0.30	14	2

Table 5.3 – Performance results of RIPE compared to two supervised learning algorithms: The Decision Tree (DT) and the Random Forests (RF).

Variable	X_{976}	X_{298}	X_{336}	X_{163}	X_{945}	X_{565}	X_{733}
Count	10	5	2	1	1	1	1

Table 5.4 – Count of variable occurencies in rules selected by RIPE.

5.4.3 Real Data

In this section, we present a quick overview of the use of the alogirthm RIPE on the well-known Kaggle's [6] dataset: *Titanic*. It is a binary classification problem. The goal is to predict which passengers survived the tragedy. We have kept only 7 features. We have dropped features *Name*, *Ticket Number*, and *Cabin Number* which are considered irrelevant for a first study, and we haven't done data engineering.

The accuracy rate given by Kaggle for RIPE's predictions on the test set is 0.765, but the most interesting output is the description of the model.

This can be sum up in Table 5.5 and with the two following figures. Figure 5.3 shows that the most important feature is the *fare* which appears seven times in the set of selected rules. The figure 5.4 permits to be more specific. Indeed, we can notice that the cheaper the ticket, the higher the risk to die.

This example shows the kind of interpretation that RIPE could offer to a statistical study on an unknown dataset.

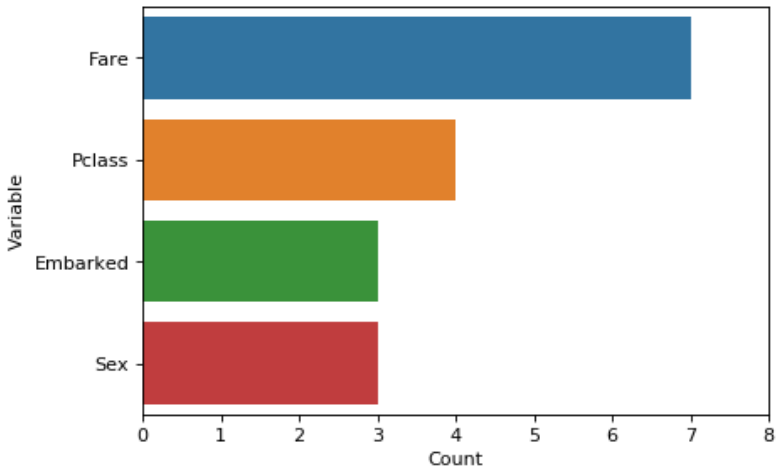


Figure 5.3 – Distribution of rules by variables

6. https://www.kaggle.com/

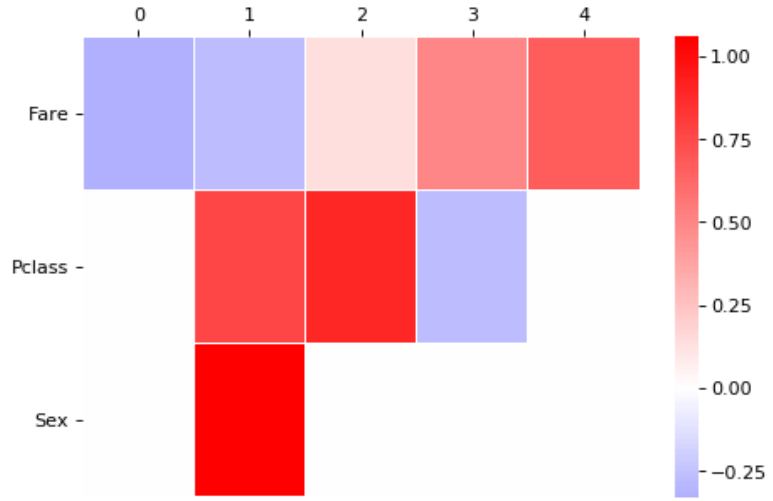


Figure 5.4 – Sum of the prediction of rules for each quantitative variable by modalities

Rule	Conditions	Coverage	Prediction	Z	MSE
R 0(2)+	$Sex = female$ & $Pclass \in [1.0, 2.0]$	0.19	1.16	0.26	0.32
R 1(2)+	$Sex = female$ & $Fare = 4.0$	0.11	1.09	0.35	0.41
R 3(2)+	$Sex = female$ & $Embarked \in [C, Q]$	0.12	0.93	0.32	0.42
R 2(1)+	$Pclass = 1.0$	0.24	0.51	0.21	0.43
R 5(1)-	$Fare \in [0.0, 1.0]$	0.38	−0.38	0.14	0.43
R 6(2)+	$Pclass \in [1.0, 2.0]$ & $Fare = 4.0$	0.18	0.63	0.25	0.43
R 4(2)-	$Pclass = 3.0$ & $Fare \in [0.0, 2.0]$	0.47	−0.27	0.13	0.44
R 7(2)+	$Fare \in [2.0, 4.0]$ & $Embarked \in [C, NaN]$	0.15	0.54	0.27	0.45
R 8(2)+	$Fare \in [2.0, 4.0]$ & $Embarked \in [C, Q]$	0.17	0.45	0.25	0.45
R 9(1)-	$Fare \in [1.0, 2.0]$	0.41	−0.16	0.14	0.46
R 10	No rule activated	0.09	−0.40	0.28	0.47

Table 5.5 – Summary of selected rules with conditions interval express as modalities

5.5 Conclusion

In this paper we present an *understandable* predictive algorithm, named RIPE. Considering the regression function is the best predictor RIPE has been developed to be a simple and accurate estimator of the regression function. The algorithm identified a set of *suitable rules*, not necessary disjoint, of the form *If-Then* such as their *If* conditions are hyperrectangles of the features space $\mathcal{X}$. Then, the estimator is built on the partition generated by the *partitioning trick*. Its computational complexity is linear in the data dimension $O(dn)$.

RIPE is different from existing methods which are based on a space-partitioning tree. It is able to generate a fine partition from a set of *suitable rules*, reasonably quickly such that their cells are explained as a list of *suitable rules*. Whereas there is a one-to-one correspondence between a rule and a cell of a partition provided by a decision tree. So to have a finer partition decision trees must be deeper and rules become less and less *understandable*. Furthermore, on the contrary to decision trees, the partition generated by RIPE can have cells which are not hyperrectangles.

Appendix: Examples of Significance Function

Here, we present three functions z used in practice.

1. The first one is based on the Hœffding's inequality [8]:

$$z(\mathbf{r}, D_n, \alpha) = \frac{(M-m)\sqrt{\ln(2/\alpha)}}{\sqrt{2n(\mathbf{r}, D_n)}}, \tag{5.5.1}$$

   where $M = \max_{i\in\{1,\ldots,n\}} y_i$ and $m = \min_{i\in\{1,\ldots,n\}} y_i$.

2. The second one is based on the Bernstein's inequality:

$$z(\mathbf{r}, D_n, \alpha) = \frac{1}{6n(\mathbf{r}, D_n)}\left(M\ln\left(\frac{2}{\alpha}\right) + \sqrt{M^2\ln\left(\frac{2}{\alpha}\right)^2 + 72v\ln\left(\frac{2}{\alpha}\right)}\right), \tag{5.5.2}$$

   where $M = \max_{i\in\{1,\ldots,n\}} y_i$ and $v = \sum_{i=1}^n y_i^2$.

3. And the last one is

$$z(\mathbf{r}, D_n) = \sqrt{\left(\beta_{\mathbf{r},n}\frac{1}{n-1}\sum_{i=1}^n \left(Y_i - \bar{Y}\right)^2 - \frac{1}{n(\mathbf{r}, D_n)-1}\sum_{i=1}^n \mathbf{1}_{X_i\in\mathbf{r}}\left(Y_i - \bar{Y}_{\mathbf{r}}\right)^2\right)}, \tag{5.5.3}$$

   where

$$\beta_{\mathbf{r},n} = \frac{n}{\sum_{\mathbf{r}'\in\mathcal{S}_n} n(\mathbf{r}', D_n)} \max_A \#\{\mathbf{r}' \in \mathcal{S}_n : \mathbf{r}' \cap A \neq \emptyset\}, \tag{5.5.4}$$

   with the max is taken upon the set of cells of $\mathcal{K}(\mathcal{S}_n)$ contained in $\mathbf{r}$. It means the set defined by

$$\{A \in \mathcal{K}(\mathcal{S}_n) : A \subseteq \mathbf{r}\}.$$

Bibliography

[1] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010. Published in Statistics Surveys.

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

[3] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* CRC press, 1984.

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

[5] K. Dembczyński, W. Kotłowski, and R. Słowiński. Solving regression by learning an ensemble of decision rules. In *International Conference on Artificial Intelligence and Soft Computing*, pages 533–544. Springer, 2008.

[6] J.H. Friedman and B.E. Popescu. Predective learning via rule ensembles. *The Annals of Applied Statistics*, 916–954, 2008.

[7] L. Györfi, M. Kohler, A. Krzyzak, and H Walk. *A Distribution-Free Theory of Nonparametric Regression.* Springer Science & Business Media, 2006.

[8] W Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[9] L. Hyafil and R.L. Rivest Constructing optimal binary decision trees is np-complete. *Information processing letters*, 5(1):15–17, 1976.

[10] B. Letham, C. Rudin, T.H. McCormick, and D. Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] J. R Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[13] H. Yang, C. Rudin, and M. Seltzer. Scalable bayesian rule lists. In *Proceedings of the 34th International Conference of Machine Learning (ICML'17)*, 2017.

List of Figures

List of Tables

"Odi panem quid meliora."
Kaamelott, Livre V, tome 1.