Explainable neural kernel logistic regression : application to precision medicine

Marie Guyomard

Régression Logistique à Noyau Neural Explicable : Application à la Médecine de Précision

Marie Guyomard

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis

Présentée en vue de l'obtention du grade de docteur en automatique, traitement du signal et des images d'Université Côte d'Azur

Dirigée par : Lionel Fillatre
Professeur des universités, CNRS, I3S

Co-dirigée par : Nicolas Glaichenhaus
Professeur des universités, CNRS, IPMC

Soutenue le : 29 novembre 2023

Devant le jury, composé de :
Florence d'Alché-Buc, Professeure, LTCI, Télécom Paris
Paul Honeine, Professeur, LITIS, Université de Rouen Normandie
Blaise Hanczar, Professeur, IBISC, Université Paris-Saclay
Grégoire Montavon, Professeur associé, Freie Universität Berlin

cnrs i3s sophia antipolis ipmc UNIVERSITÉ CÔTE D'AZUR ÉCOLE UNIVERSITAIRE DE RECHERCHE SYSTÈMES NUMÉRIQUES POUR L'HUMAIN FRANCE 2030 Initiative d'Excellence

Régression Logistique à Noyau Neural Explicable : Application à la Médecine de Précision

Explainable Neural Kernel Logistic Regression : Application to Precision Medicine

Marie Guyomard

Direction de thèse :

Directeur :
M. Lionel FILLATRE, Professeur, Université Côte d'Azur, CNRS, Laboratoire I3S.

Co-directeur :
M. Nicolas GLAICHENHAUS, Professeur, Université Côte d'Azur, CNRS, IPMC.

Membres du jury :

Rapporteurs :

M. Paul HONEINE, Professeur, Université de Rouen Normandie, LITIS.

M. Blaise HANCZAR, Professeur, Université Paris-Saclay, IBISC.

M. Grégoire MONTAVON, Professeur, Freie Universität Berlin, BIFOLD.

Examinatrice :

Mme Florence D'ALCHÉ BUC, Professeure, Télécom Paris, LTCI.

Résumé

Régression Logistique à Noyau Neural Explicable : Application à la Médecine de Précision

L'utilisation de l'Intelligence Artificielle pour la médecine de précision ne cesse de progresser. L'apprentissage statistique automatique est de plus en plus employé afin de personnaliser les parcours de soin des patients, comme par exemple pour la prédiction de pathologies ou la prescription de traitements adéquats. Les algorithmes de support à la décision développés dans ce but prennent en compte les caractéristiques clinico-biologiques propres à chaque patient pour émettre un diagnostic. En pratique, ces outils statistiques permettent pour un grand nombre de patients de s'affranchir de méthodes invasives, telles que les biopsies qui sont à la fois lourdes pour les patients et coûteuses pour les systèmes hospitaliers. Les méthodes développées doivent fournir nécessairement des performances satisfaisantes et fiables mais aussi des prédictions interprétables par les experts du secteur médical. Néanmoins, les méthodes les plus performantes en apprentissage automatique sont souvent aussi les plus complexes et donc les plus difficiles à interpréter.

Cette thèse est consacrée au développement d'une nouvelle méthode de classification supervisée explicable, telle que la règle de décision qui en découle soit interprétable et fiable, répondant aux enjeux de la médecine de précision. Nos recherches ont été menées en étroite collaboration avec l'Institut de Pharmacologie Moléculaire et Cellulaire (IPMC) et le Service d'Hépatologie du CHU de Nice.

Notre première contribution est l'introduction du modèle SATURNN (Splines Approximation Throught Understandable ReLU Neural Network). Il s'agit d'un réseau de neurones composé d'une seule couche cachée et dont la couche de sortie sigmoïde est appliquée à une fonction de score. L'architecture de ce réseau est contrainte afin que la règle de décision se réécrive comme une somme additive de splines univariées facilement interprétable. Néanmoins, comme tout réseau de neurones, nous ne disposons pas de garantie de convergence du processus d'apprentissage et donc d'unicité des estimations.

Notre seconde contribution vise à s'affranchir de cette limite et proposer une méthode explicable avec une phase d'apprentissage fiable. Nous proposons dans un premier temps de linéariser localement, au voisinage de ses initialisations, la fonction de score du SATURNN. Nous démontrons qu'il est alors équivalent d'entraîner un SATURNN composé d'un grand nombre de neurones ou une régression logistique appliquée aux données préalablement transformées par la fonction de score linéarisée. Dans un second temps, nous établissons que cette transformation peut se réécrire sous la forme d'un noyau qui converge asymptotiquement vers une limite finie. Nous proposons alors un nouveau noyau déterministe qui découle directement de l'architecture du SATURNN mais qui est indépendant de son initialisation.

Notre troisième contribution est l'introduction d'une Régression Logistique appliquée aux données préalablement transformées par le noyau déterministe. La segmentation des variables opérée par le noyau est elle aussi déterministe ; elle ne dépend d'aucun paramètre à apprendre mais seulement de l'ensemble de l'échantillon d'apprentissage. La règle de décision qui en résulte se réécrit comme une somme additive de splines univariées facilement interprétable. Pour l'application médicale, ces splines estimées expliquent l'impact

des variables clinico-biologiques sur la variable à prédire, tel que le risque de développer la pathologie ou la réponse à un traitement. Contrairement aux modèles additifs généralisés ou aux forêts aléatoires, la règle de décision qui découle de la Régression Logistique à noyau est unique conditionnellement à l'échantillon d'apprentissage, ce qui rend son interprétation fiable. Nous proposons de ce fait un algorithme de support à la décision explicable (interprétable et fiable) adapté aux enjeux de la médecine de précision.

Mots Clés : Réseaux de Neurones, Régression Logistique à Noyau, Noyau Neural Tangent, Modèles Additifs Généralisés, Splines Univariées, Apprentissage Automatique Interprétable, Intelligence Artificielle Explicable, Médecine de Précision.

Abstract

Explainable Neural Kernel Logistic Regression : Application to Precision Medicine

The use of Artificial Intelligence for precision medicine is constantly rising. Machine Learning is increasingly employed to personalize patient care pathways, such as predicting pathologies or prescribing appropriate medical treatments. Algorithmic Decision Systems developed for this purpose take into account the specific clinical and biological characteristics of each patient to make a diagnostic. In practice, for a significant number of patients, these statistical tools enable the avoidance of invasive methods, such as biopsies, which are both burdensome for patients and costly for healthcare systems. Proposed methods must not only produce satisfying and reliable performance but also provide predictions that are easily interpretable by medical experts. However, the best-performing Machine Learning algorithms are often the most complex, making them challenging to interpret.

This thesis focuses on the development of a novel explainable supervised classification method that produces an interpretable and reliable decision rule, addressing the challenges of precision medicine. Our research is conducted in close collaboration with the Institute of Molecular and Cellular Pharmacology (IPMC) and the Hepatology Department of Nice University Hospital.

Our primary contribution is the introduction of the SATURNN (Splines Approximation Throught Understandable ReLU Neural Network) model. This is a single-layer neural network designed for classification tasks, which a sigmoid output layer applied to a score function. The neural architecture is constrained to model the decision rule as an additive sum of univariate spline functions, which is easily interpretable. Nevertheless, like any neural network, there is no guarantee that the learning process will converge and produce unique estimates.

Our secondary contribution aims to overcome this limitation and to propose an explainable method with a reliable learning process. First, we propose to locally linearize the score function in the neighborhood of its initializations. We then demonstrate that it is equivalent to training a SATURNN composed of a large number of neurons or a logistic regression applied to data previously transformed by the linearized score function. Second, we establish that this mapping function can be reformulated as a kernel that asymptotically converges to a finite limit. As a result, we propose a new deterministic kernel, derived directly from the SATURNN architecture but entirely independent of its initializations.

Our third contribution is the introduction of a Logistic Regression applied to data previously transformed by the deterministic kernel. The partitioning of the features produced by the kernel is also deterministic ; it does not depend on any trainable parameters, but only on the training samples as a whole. The resulting decision rule is modeled as an additive sum of univariate spline functions, making it easily interpretable. In medical applications, these splines reflect the impact of clinical or biological characteristics on the target variable, such as the risk of developing a pathology or the response to a treatment. Unlike generalized additive models or random forests, the decision rule produced by the proposed kernel logistic regression is unique conditional on the training samples, ensuring

the reliability of its interpretation. We therefore propose an explainable (interpretable and reliable) decision support algorithm that addresses the challenges of precision medicine.

Keywords : Neural Networks, Kernel Logistic Regression, Neural Tangent Kernel, Generalized Additive Models, Univariate Splines, Interpretable Machine Learning, Explainable Artificial Intelligence, Precision Medicine.

Remerciements

Je souhaite exprimer mes sincères remerciement à l'ensemble de mon jury de thèse, Messieurs Hanczar, Honeine et Montavon pour leurs rapports détaillés concernant ce manuscrit et Madame d'Alché-Buc d'avoir présidé le jury de ma soutenance. Les échanges fructueux que nous avons eus ont été d'une grande valeur pour l'avancement de mes travaux.

Je tiens particulièrement à remercier mes encadrants de thèse Messieurs Fillatre et Glaichenhaus. Je suis consciente de la chance que j'ai eue de bénéficier de leur encadrement. La confiance que vous m'avez témoignée, la grande motivation dont vous avez fait part ainsi que le temps précieux que vous avez consacré à ce projet ont été d'une importance capitale.

Monsieur Fillatre, ce fut un réel plaisir de travailler avec vous. Je suis sincèrement reconnaissante pour votre incitation constante à aller plus loin, à améliorer notre travail et à développer mes compétences. Chaque réunion, que ce soit en visioconférence ou en présentiel a été source de motivation pour moi, et c'est avec le sourire et beaucoup d'idées que chacune d'elles se terminait.

Monsieur Glaichenhaus, je vous remercie pour le temps que vous avez consacré à nos échanges, toujours enrichissants et dans la bonne humeur. Votre expertise médicale a non seulement contribué à faire avancer notre projet, mais m'a aussi beaucoup apporté sur le plan personnel. J'ai appris grâce à votre grande pédagogie l'existence de mécanismes biologiques jusqu'alors insoupçonnés se déroulant dans mon corps !

Je suis reconnaissante pour ces trois années passées ensemble et pour toutes les opportunités qui m'ont été offertes. Si c'était à refaire, je le referais dix fois sans hésiter. J'espère sincèrement que l'on aura le plaisir de continuer à collaborer ensemble dans les années à venir.

Je souhaite également remercier l'ensemble des personnes avec qui j'ai eu l'occasion de travailler.

Cyprien, un simple merci ne suffirait pas. C'est grâce à toi que j'ai eu envie de poursuivre en thèse et que j'ai pu m'épanouir ces trois dernières années. Tu as toujours su faire preuve de patience durant mon stage, lorsque tu me chargeais d'écrire la partie expérimentale de nos articles et que suite à mon intervention sur Overleaf, 55 erreurs rouges apparaissaient. Tu m'as appris la rigueur et le souci du détail, jusqu'au choix des couleurs des figures. Merci pour l'ami formidable que tu es, j'ai hâte de pouvoir continuer à collaborer avec toi. J'espère suivre ton exemple et un jour être une chercheuse épanouie comme tu l'es. Merci pour tout mon Cyp'.

Je remercie le Professeur Rodolphe Anty du Service d'Hépatologie du CHU de Nice qui a fait part dès le début de notre collaboration d'une grande confiance à mon égard. Ses précieux conseils ont contribué à éclairer de nombreuses problématiques spécifiques au domaine médical liées à l'utilisation de l'apprentissage statistique. J'espère sincèrement que nos projets aboutiront positivement dans les semaines à venir.

Un grand merci également à Susana Barbosa qui a consacré beaucoup de son temps à chacune de nos réunions à l'IPMC pour que le projet avance dans les meilleurs conditions. Tu as toujours pris soin de répondre à nos questions et su nous conseiller quant à la méthodologie à suivre.

Mes remerciements vont aussi naturellement à l'ensemble des personnes que j'ai rencontré au laboratoire I3S, personnels administratifs, permanents et doctorants, avec qui j'ai passé de super moments. Nadia, je souhaite te remercier pour toute l'aide que tu m'as apporté. Malgré nos (trop nombreuses) sollicitations, tu nous aides toujours dans la bonne humeur et avec une grande bienveillance. Pierre, merci d'avoir toujours pris part à nos idées stupides et rigolé à toutes nos mauvaises blagues. Un grand merci à Xavier et Éric pour nos échanges, Diana, Lina, Laëtitia, Romain, François, Margaux, Guillaume et tous les doctorants du 2e étage pour nos repas décontractants le midi et les nombreux afterworks animés.

Je tiens particulièrement à remercier Cédric et Bastien pour leurs bonnes humeurs et leur décoration travaillée et soignée des bureaux 116 et 122 qui les caractérise si bien. Ce fut un plaisir de partager tous ces moments avec vous les amis, il faut tout de même avouer que vous êtes un public facile pour les blagues... Un grand merci à Baptiste pour sa bonne humeur et sa gentillesse. Je me dois de souligner son soutien sans faille tous les soirs de soumission, à rester jusqu'au bout pour vérifier que je n'écrives pas de bêtises sur les derniers instants. Ça a été un véritable plaisir de partager son bureau pendant un an, j'espère qu'il pourra en dire autant !

Enfin, je souhaite finir par adresser des remerciements à mes amis et ma famille. Solène, merci de m'avoir fait aimer les maths dès mon plus jeune âge et d'avoir été une des seules à apprécier toutes ces équations mathématiques lors de ma soutenance de thèse. Je souhaite grandement remercier mes grands-parents, c'est aussi grâce à eux que j'ai pu m'épanouir dans mes études. Ils ont toujours veillé à ce que je reste motivée et réussisse.

Loïc, je te remercie d'avoir toujours soutenu mes choix, notamment celui de poursuivre en thèse. Un grand merci à mes parents pour avoir toujours tout mis en oeuvre pour que je puisse réaliser au mieux mes études. J'ai énormément de chance d'avoir des parents si compréhensifs et dévoués pour leurs enfants. Si j'ai pu m'épanouir dans la réalisation de cette thèse c'est avant tout grâce à eux. Maman, je tenais tout particulièrement à te remercier pour les heures passées à relire tous mes rapports de projets, de stage mais aussi à m'écouter répéter mes présentations orales. Ce manuscrit, personne ne le connaît mieux que toi. S'il est aussi bien écrit c'est grâce à toi et à toutes tes corrections, à la virgule près. Cette thèse, elle est à toutes les deux.

Enfin, merci à Marius d'avoir partagé ces trois années de thèse avec moi.

Acronymes

ADS Algorithmes de Support à la Décision. 19–29, 31, 115, 120, 128–130, 132, 140–143, 148, 149, 151, 152

AUC Area Under the Curve. 23, 67, 132, 133, 135

BCE Cross-Entropie Binaire. 39, 42, 48, 62, 143

CV Validation Croisée. 22, 27, 67, 68, 90, 118, 135, 142

DT Arbre de décision. 36, 43, 46, 47, 62, 75

EBM Machines Explicables Boostées. 34, 36, 46, 47, 66, 69, 72, 117, 118, 120–122, 132, 133, 138, 144, 148, 154

EKLR Régression Logistique à Noyau Déterministe. 33, 102–105, 107, 108, 110–113, 117, 118, 120, 123–126, 133, 134, 138, 139, 142, 144–148, 153–155, 200

GA^{2}M Modèle Additif Généralisé avec Intéractions. 46, 53

GAM Modèle Additif Généralisé. 34–36, 43–47, 52, 53, 56, 57, 59, 60, 62, 66, 69, 70, 72, 75, 104, 117, 118, 121–123, 125, 154

GP Processus Gaussien. 93, 94

IA Intelligence Artificielle. 19, 141

KLR Régression Logistique à Noyau. 32, 33, 96–98, 100, 102–105, 107, 108, 110–113, 117, 118, 120, 123–126, 132, 134, 138, 139, 142, 144–148, 153–155, 200

LR Régression Logistique. 34, 35, 37–40, 42, 53, 59, 69, 76, 82, 83, 102, 103, 132, 133, 135, 137, 138, 140–145, 148, 152, 153

LR NCS Régression Logistique à Splines Naturelles Cubiques. 41, 53, 59, 66, 69, 72

LR PSI LIN Régression Logistique appliquée à la fonction de score linéarisée. 32, 82–91, 94–96, 98, 102, 103, 105, 107, 117, 118, 120, 123–125, 132–134, 138, 139, 144–147, 153

MAP Maximum a Posteriori. 37, 44, 45, 67

MARS Régression Multivariée par Splines Adaptatives. 34, 36, 43, 44, 47, 53, 56, 57, 59, 62, 66, 69, 72, 75, 117, 118, 120, 154

MAS Max-Affine Spline. 50

MASO Opérateurs Max-Affine Spline. 50

ML Machine Learning. 19–22, 24–28, 31, 34, 151

NAM Modèles Additifs Neuronaux. 52, 53, 56, 57, 62, 66, 69, 70, 72, 117, 118, 120, 121, 123, 125

NCS Splines Naturelles Cubiques. 40–43, 45, 69

NTK Neural Tangent Kernel. 93, 94

RF Forêt Aléatoire. 35, 36, 46, 47, 66, 69, 70, 117, 118, 120, 154

RN Réseau de Neurones. 34, 35, 48–54, 56–62, 66, 69, 71, 72, 75, 93, 94, 117, 120, 152, 154

RN-MARS Réseau de Neurones MARS. 32, 55–59, 64, 66, 69–72, 152

SATURNN Splines Approximation Throught Univariate ReLU Neural Network. 32, 33, 55, 56, 59–64, 66, 69–72, 74–100, 102–108, 110, 113, 117, 118, 120–125, 132–134, 138, 139, 144, 145, 148, 152–156, 185–187

SGD Descente de Gradient Stochastique. 39, 48, 49, 59, 64, 75, 93

SVM Machines à Vecteurs de Support. 35, 93, 105, 108, 110, 111

XAI Intelligence Artificielle Explicable. 27

Notations

$\mathbb{R}$	Espace des réels
$\mathcal{D}$	Ensemble de données
X	Base de données de variables explicatives
x	Ensemble des variables explicatives pour un échantillon donné
Y	Base de données des étiquettes binaires à prédire
y	Étiquette d'appartenance pour un échantillon donné
N	Nombre d'échantillons
d	Nombre de variables explicatives
$\hat{Y}$	Ensemble des étiquettes prédites
$\hat{y}$	Étiquette prédite pour un échantillon donné
$\mathbb{P}$	Probabilité
$\hat{\mathbb{P}}$	Probabilité estimée
θ	Paramètres du SATURNN
p	Nombre de neurones composant le SATURNN
$\psi(x, \theta)$	Fonction de score du SATURNN appliquée à l'échantillon x avec les paramètres θ
σ	Sigmoïde
$\phi(\cdot)$	Activation ReLU
$\mathbb{P}$	Probabilité
L	Fonction de coût Entropie Croisée Binaire
$\mathcal{B}_2^d(0, r)$	Boule ouverte dans $\mathbb{R}^d$ centrée en 0 et de rayon $r > 0$
O	Notation grand O

Table des matières

Chapitre 1

Introduction et contexte

Dans ce chapitre, nous présentons le contexte de notre recherche. En Section 1.1, nous définissons le principe de l'Intelligence Artificielle pour la médecine de précision. Ensuite, la Section 1.2 introduit les différents challenges du *Machine Learning* pour ce domaine d'application. Dans un premier temps nous introduisons les enjeux techniques (sous-section 1.2.1) auxquels les algorithmes de *Machine Learning* sont confrontés lorsqu'ils sont appliqués à la médecine de précision, avant d'en définir dans un second temps (sous-section 1.2.2) les problématiques éthiques et sociales. La Section 1.3 présente les différentes collaborations médicales qui ont permis de mener à bien ces recherches. Les échanges avec des Professeurs d'Immunologie et d'Hépatologie nous ont permis de cerner les besoins et les attentes des praticiens et ainsi développer des méthodes de *Machine Learning* adaptées à la réalité terrain. Enfin la Section 1.4 introduit les différentes contributions présentées dans ce manuscrit.

Sommaire

1.1 Principe de l'Intelligence Artificielle pour la médecine de précision

Principe général de la médecine de précision

La médecine personnalisée, également appelée médecine de précision, est une nouvelle discipline médicale qui vise à proposer des mesures préventives ciblées, à affiner les diagnostics et à personnaliser les traitements en utilisant les données biologiques, environnementales ou sociales, spécifiques au patient. La médecine personnalisée a ainsi pour but d'adapter les parcours de soin des patients en fonction de leurs caractéristiques spécifiques. La Figure 1.1 illustre le principe de médecine de précision et plus précisément celui visant à personnaliser les traitements. À gauche, nous retrouvons une médecine non-personnalisée, pour laquelle tous les patients reçoivent le même traitement quelles que soient leurs caractéristiques (bleu, orange ou rose). Les effets de la thérapie divergent alors en fonction des profils de patients : les bleus guérissent tandis que l'état de santé des oranges ne s'améliore pas et pire encore, celui du patient rose se dégrade. Lorsque nous comparons avec la médecine de précision qui propose à chaque catégorie de patients un traitement personnalisé en fonction de ses caractéristiques, nous pouvons voir que l'état de santé de tous les patients s'améliore.

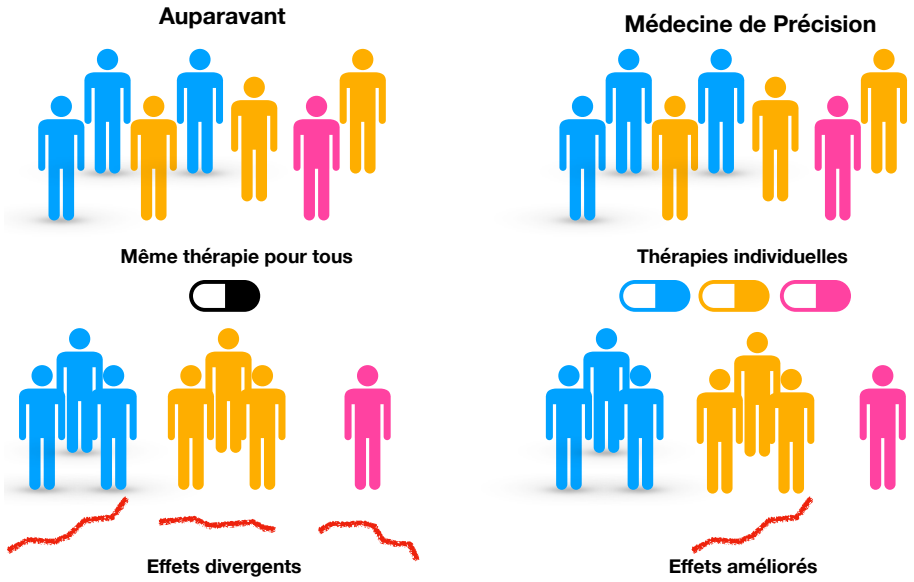

FIGURE 1.1 – Schéma illustrant les avantages du passage de la médecine traditionnelle (Gauche) à la médecine de précision (Droite) pour l'exemple de la personnalisation de thérapies. Pour la médecine traditionnelle les patients reçoivent le même traitement quelles que soient leurs caractéristiques (bleu, orange, rose) et ainsi l'effet sur la pathologie que l'on souhaite soigner diverge. En médecine de précision, le traitement est personnalisé et nous obtenons ainsi des effets améliorés pour tous les profils de patients.

En 2010, le Docteur Leroy Hood (*Institute for Systems Biology*) a prôné la transformation de la recherche en biologie en un système P4 où la médecine se doit d'être Prédictive, Préventive, Personnalisée et Participative (*predictive, preventative, personalized, participatory*) [Hood et Flores, 2012]. Le terme P4 est alors un nouveau paradigme pour la re-

cherche bio-médicale ayant pour but d'utiliser un large réseau d'information, menant à une recherche inter-disciplinaire entre biologistes, physiciens, mathématiciens, etc. Dès l'Introduction du Plan stratégique de l'Inserm [1] de 2020 [2], le constat d'une recherche évolutive pour le domaine bio-médical est fait, nous pouvons notamment lire :

> *"Nous allons vers (...) une médecine enrichie par la recherche fondamentale, multidisciplinaire et translationnelle, le biologiste et le clinicien faisant désormais appel aux apports du physicien, du chimiste, du mathématicien, du bioinformaticien, de l'ingénieur, de l'écologiste et du chercheur en sciences humaines et sociales et environnementales."*

Algorithme de Support à la Décision

Depuis plusieurs années, l'Intelligence Artificielle (IA) est de plus en plus employée pour l'application médicale dans le but de fournir des algorithmes de support à la décision pour les cliniciens. Les Algorithmes de Support à la Décision (ADS) sont des méthodes d'apprentissage automatique permettant de soutenir et d'aider considérablement les cliniciens en matière de diagnostic, de pronostic et de traitement. Ces algorithmes n'ont pas pour objectif de remplacer les médecins mais de les aiguiller quant à leur prise de décision. L'interprétation et l'explication de résultats complexes sont des tâches essentielles qui dépassent la portée de tout algorithme d'apprentissage automatique [Eyal *et al.*, 2019].

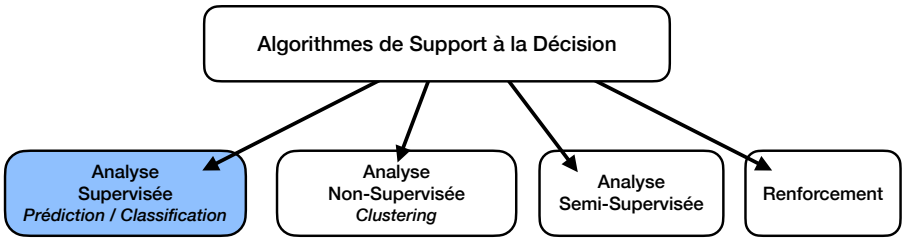

FIGURE 1.2 – Schéma des différentes catégories d'Algorithme de Support à la Décision. La catégorie bleue est celle d'intérêt pour la suite de ce manuscrit.

Le *Machine Learning* (ML) également appelé apprentissage automatique est un champ d'étude de l'IA permettant de découvrir des "patterns", à savoir des motifs récurrents dans un ensemble de données. Ainsi, le ML est un outil intéressant de développement d'ADS dans le but d'améliorer le parcours de soin de patients et promouvoir la médecine de précision. Pour l'application médicale, ces méthodes sont souvent employées sur des données cliniques (genre, âge, poids, taille, etc.), des données biologiques (taux de cholestérol, glucose, hémoglobines rouges, etc.), des images (IRM, scanner, radiographie, etc.) ou encore plus récemment les données génétiques (séquences ARN ou ADN). Selon les objectifs visés, quatre grandes catégories d'analyse peuvent être réalisées sur ces données (Figure 1.2) [Nayyar *et al.*, 2021, Habehh et Gohel, 2021, Shailaja *et al.*, 2018] :

— Analyse supervisée : l'objectif des méthodes supervisées de ML est de développer un algorithme de prédiction d'une variable cible pouvant être numérique (régression) ou catégorielle (classification) à partir de l'ensemble des données d'apprentissage mises à disposition.

1. Institut National de la Santé et de la Recherche Médicale (Inserm).
2. Plan stratégique de l'Inserm, 2020.

— Analyse non supervisée : l'objectif des méthodes non supervisées de ML est de regrouper les patients en différentes catégories (*clusters*) selon des caractéristiques communes. Ces techniques peuvent identifier des similarités entre les patients que les praticiens ne soupçonnaient à priori pas.
— Analyse semi-supervisée : l'objectif des méthodes semi-supervisées est de développer des algorithmes de prédiction à partir de variables cibles connues et manquantes.
— Apprentissage par Renforcement : le *Reinforcement Learning* dépend de séquences de récompenses similaires aux mécanismes de conditionnement en psychologie. Bien que cette application au domaine bio-médical soit limitée de part la structure des données nécessaires pour son apprentissage [Riachi *et al.*, 2021] elle présente un potentiel important pour faire progresser les soins de santé de manière significative.

Dans la suite de ce manuscrit, nous allons plus particulièrement nous intéresser aux méthodes de classification supervisée appliquées aux données clinico-biologiques. L'apprentissage supervisé est très répandu pour le domaine bio-médical et notamment pour répondre à deux objectifs principaux :

1. Détection de pathologies : des approches de ML ont notamment déjà été employées dans le but par exemple de prédire l'épilepsie [Siddiqui *et al.*, 2020], le diabète [Woldaregay *et al.*, 2019], le cancer de la peau [Esteva *et al.*, 2017] ou encore la pneumonie [Rajpurkar *et al.*, 2017].
2. Personnalisation de traitements : dans [Costello *et al.*, 2014], les auteurs ont comparé 44 ADS visant à adapter les thérapies pour le cancer du sein selon les caractéristiques génétiques spécifiques des patients à l'aide de données génomiques, épigénomiques et protéomiques. Plus récemment, les auteurs de [Katzman *et al.*, 2018] ont développé un ADS pour la personnalisation de traitements et l'ont appliqué à différentes bases de données cliniques réelles dans le but d'augmenter les chances de survie des patients ayant subi une attaque cardiaque ou développé un cancer du sein.

1.2 Challenges du Machine Learning pour la médecine de précision

Ces dernières années, l'utilisation grandissante du ML pour l'application médicale s'est révélée très prometteuse dans l'amélioration des parcours de soin de patients mais a aussi fait émerger un grand nombre de challenges [Aung *et al.*, 2021, Xing *et al.*, 2020, Rajpurkar *et al.*, 2022, Adam *et al.*, 2020]. En effet, les ADS développés pour la médecine de précision font face à deux grandes limites. La première difficulté est de l'ordre technique et concerne naturellement la performance des modèles (sous-section 1.2.1). La deuxième contrainte est quant à elle plutôt sociale et relève de l'ordre de l'éthique (sous-section 1.2.2) à savoir la confiance que l'on a envers les ADS développés.

1.2.1 Techniques : performances prédictives

De nombreux facteurs techniques altèrent ou limitent la performance prédictive des ADS développés pour l'application médicale. Nous pouvons en dénombrer notamment trois principaux : la disponibilité des données, la limitation du sur-apprentissage des ADS mais aussi leur équité. Comme illustré sur la Figure 1.3, ces trois facteurs impactent directement la performance des ADS (flèches noires), mais interagissent aussi entre eux (flèches pointillé gris).

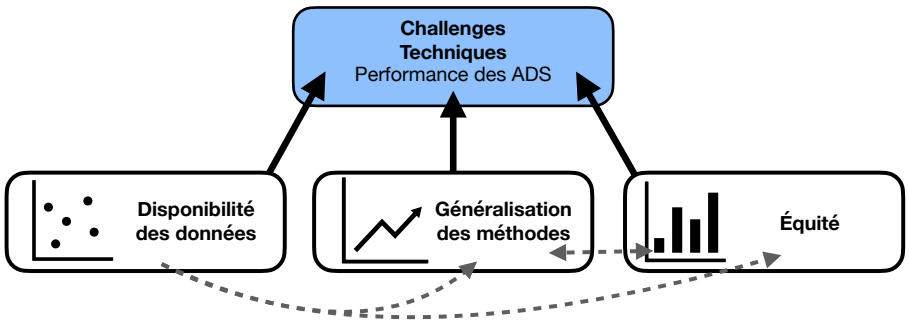

FIGURE 1.3 – Schéma des grands challenges techniques rencontrés pour l'application de méthodes de *Machine Learning* à la médecine de précision. Les trois grands challenges impactent directement la performance prédictive des ADS (flèches noires) mais aussi interagissent entre eux (flèches pointillé gris).

Disponibilité des données

Plus les modèles de ML disposent de données d'apprentissage, plus les ADS issus ont tendance à être robustes. Néanmoins, les données de santé sont souvent disponibles en petite quantité. Dans le contexte de la médecine de précision, il y a bien évidemment des considérations éthiques, légales et morales quant à la collecte de données de santé, le stockage, mais aussi leur transfert pour être analysées. Par exemple, la confidentialité des données médicales (*Privacy*) implique la non-identifiabilité des patients de part l'anonymisation des données, mais aussi la suppression de toutes informations personnelles pouvant amener à retrouver l'identité des patients figurant dans les bases de données utilisées pour l'apprentissage d'ADS. Le recueil de données prend bien évidemment du temps mais leur qualité est aussi souvent limitée [Aung *et al.*, 2021]. Les données médicales sont souvent incohérentes, parfois inexactes et la façon dont elles sont stockées et formatées n'est pas normalisée [Sun et Medaglia, 2019, Wiens et Shenoy, 2018]. Le pré-traitement des bases de données médicales prend ainsi du temps et nécessite de gérer de nombreuses données manquantes, entraînant des incertitudes quant à l'exactitude des données d'apprentissage en cas d'imputation des données manquantes ou bien des bases de données très petites. Ces lacunes dans les données d'entraînement ont naturellement un impact direct sur la performance prédictive des ADS développés, que ce soit en terme de généralisation (limiter le sur-apprentissage) ou bien d'équité des règles de décision qui en découlent.

Sur-Apprentissage

Du fait de la petite taille des bases de données médicales disponibles pour entraîner les ADS, les règles de décision issues ont tendance à être moins généralisables. Un modèle est dit généralisable si la règle de décision estimée sur l'échantillon d'apprentissage s'applique aussi bien à de nouvelles observations test. En revanche, si elle n'est pas généralisable alors cette méthode sur-apprend. En pratique, nous avons tendance à sous-diviser les jeux de données à disposition afin de créer des échantillons à la fois d'entraînement et de validation dans le but d'estimer les performances de généralisation des modèles sur de nouvelles observations test. Sur la Figure 1.4, nous pouvons constater que la règle de décision généralisable apprise sur la base d'apprentissage (gauche - haut) s'applique tout aussi bien à l'échantillon de validation (gauche - bas). En revanche, lorsqu'une méthode sur-apprend la règle de décision qui en découle ne génère que peu, voire aucune erreur de classification

sur la base d'apprentissage (droite - haut) et un grand nombre sur l'échantillon de validation (droite - bas). Lorsque nous comparons le pouvoir de généralisation obtenu par les deux méthodes, nous constatons que seulement 3 échantillons de validation sont mal classifiés avec le modèle généralisable contre 9 pour celui ayant sur-appris.

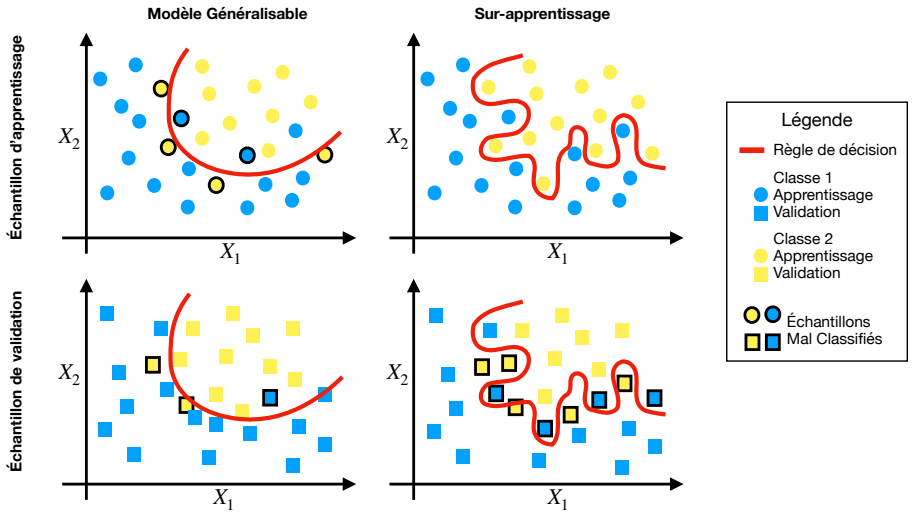

FIGURE 1.4 – Schéma illustrant le phénomène de sur-apprentissage. Nous comparons deux règles de décision (en rouge), la première est généralisable (gauche) tandis que la seconde sur-apprend (droite) sur les échantillons d'apprentissage (haut) et de validation (bas).

Ainsi, il est important de vérifier que les méthodes de ML développées puisse correctement prédire de nouveaux échantillons. Afin de vérifier qu'une méthode ne présente pas de sur-apprentissage, nous devons alors calculer les performances prédictives, c'est à dire les erreurs de classification issues des méthodes à la fois sur les échantillons d'apprentissage et de validation. Pour ce faire, nous calculons le risque d'erreur global c'est à dire le taux d'échantillons mal-classifiés. Nous pouvons constater sur la Figure 1.5 qu'au cours de l'apprentissage d'un ADS (itérations d'entraînement), l'erreur obtenue sur la base d'apprentissage (courbe bleue) tend à diminuer considérablement tandis que celle obtenue sur l'échantillon de validation (courbe orange) augmente. Ainsi, un compromis est à trouver afin qu'une méthode se généralise correctement. Généralement, pour se faire nous arrêtons le processus d'apprentissage à partir du moment où la courbe d'erreur de généralisation se met à croître (étoile noire sur la Figure 1.5).

Afin de disposer de garanties quant à la généralisation d'un modèle de ML, nous calculons les erreurs de classification sur différents échantillons d'apprentissage et de validation au travers d'un processus de Validation Croisée (CV). La CV k-folds consiste à séparer en k échantillons la base de données (Figure 1.6). À chaque fold, $k-1$ échantillons (bleus) sont utilisés pour apprendre le modèle, quant au dernier (orange), il permet de vérifier la stabilité du modèle, c'est à dire sa capacité à se généraliser sur un nouvel échantillon.

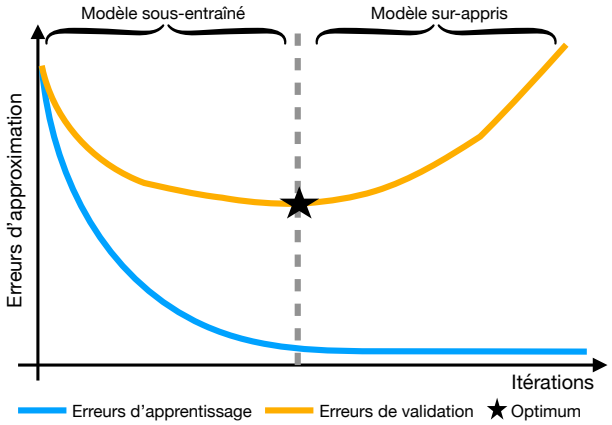

FIGURE 1.5 – Erreurs de classification sur les échantillons d'apprentissage (bleu) et de validation (orange) au cours des itérations d'entraînement. Le modèle optimal se situe au niveau de l'étoile noire. À gauche de cette étoile le modèle est sous-appris, tandis qu'à sa droite il sur-apprend.

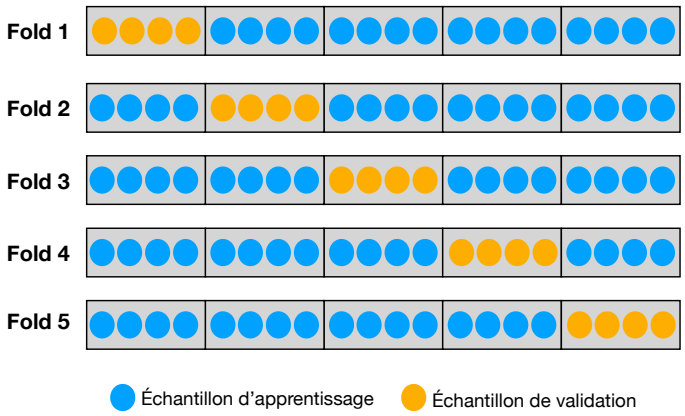

FIGURE 1.6 – Schéma explicatif de la Validation Croisée à 5-folds. La base de données est divisée en 5 sous-échantillons. À chaque fold, 4 sous-échantillons sont utilisés pour l'apprentissage de la méthode (en bleu) et sur le dernier (orange) le modèle estimé est appliqué et évalué.

Équité

Naturellement, plus les bases de données sont fournies, plus les méthodes apprises seront globalement performantes et généralisables. Néanmoins, si de gros déséquilibres entre différentes catégories d'échantillons sont présents dans la base de données, il peut arriver que ces populations sous-représentées soient lésées dans l'apprentissage des méthodes. Les bases de données médicales sont souvent grandement déséquilibrées : les patients d'intérêt, à savoir malades, sont fréquemment plus rares. Ainsi, du fait de leur sous-représentation, certains algorithmes ont tendance à les léser lors de l'apprentissage. Bien qu'il soit difficile de prédire correctement les patients d'intérêt dans ces cas-là, il est nécessaire de développer des ADS capables de surmonter cette difficulté. Afin de vérifier que la classe la moins représentée, qui plus est souvent celle d'intérêt en médecine soit bien prédite, l'*Area Under The Curve* (AUC, sous-section 3.4.3) est une métrique très informative. Si les patients issus de la classe minoritaire sont mal classifiés, la performance globale de l'algorithme pourra être très élevée, car la majorité des échantillons sera correctement prédite. En revanche la valeur de l'AUC sera quant à elle faible, cette

métrique prenant en compte la répartition de la performance globale entre les deux classes d'échantillon. Bien que cette problématique ne constitue pas l'objectif principal de nos recherches dans ce manuscrit, il nous est important de la mentionner. D'une part, c'est un problème récurrent en médecine de précision, auquel nous avons été confrontés dans nos travaux en collaboration avec le CHU de Nice. D'autre part, en parallèle de nos recherches nous avons eu l'opportunité de collaborer sur ce sujet avec Cyprien Gilet, comme détaillé plus précisément en annexe de ce manuscrit (G).

Les ADS peuvent par ailleurs être discriminatoires pour des sous-groupes de population selon leur étiquette d'appartenance mais aussi selon certaines caractéristiques telles que leur genre, leur tranche d'âge ou leurs antécédents médicaux par exemple. En raison des petites bases de données et de la sélection des patients parfois introduits dans les études, les ADS développés pour la médecine de précision ont tendance à être biaisés pour des sous-catégories de patients. Ces biais peuvent non seulement nuire à la précision et à la généralisation des modèles d'apprentissage automatique, mais aussi entraîner des discriminations [MacEachern et Forkert, 2021]. Ainsi, plus que de devoir être généralisables et applicables à de nouveaux patients, les ADS développés pour la médecine de précision se doivent d'être justes au sens de l'équité (*Fairness AI*). Afin de pouvoir vérifier qu'aucune sous-catégorie de population ne soit lésée par la règle de décision issue de l'ADS développé, nous pouvons calculer les performances prédictives au sein d'un sous-groupe de population (*accuracies* conditionnelles). Les performances conditionnelles informent quant à l'efficacité de l'algorithme selon certaines caractéristiques. Nous pouvons par exemple les calculer selon les antécédents médicaux, le genre des patients ou encore différentes tranches d'âge. Il est aussi possible d'entraîner différents modèles pour différentes catégories de population. Néanmoins, limiter les répercutions des sous-représentations dans la base de données en procédant de cette manière, nécessite de sous-échantillonner à nouveau la base de données initiale et donc d'entraîner les modèles sur des échantillons encore plus restreints. C'est pourquoi, il n'est parfois pas envisageable d'avoir recours à cette stratégie. Dans le Chapitre 7 nous mentionnerons à nouveau ces différentes difficultés lors de la présentation des résultats de nos collaborations médicales.

1.2.2 Sociaux : confiance envers les algorithmes

Afin de rendre les ADS davantage performants, la complexité des méthodes de ML développées pour la médecine de précision ne fait qu'accroître ces dernières années. Néanmoins, une telle complexification des algorithmes entraîne une importante baisse de compréhension des règles de décision estimées pour les cliniciens. Cela pose alors des problèmes juridiques et éthiques quant à leur utilisation. Rappelons que les ADS ont pour vocation à aiguiller les cliniciens et non pas à les remplacer. Si les professionnels du secteur de la santé ne sont pas en mesure de comprendre et d'avoir confiance envers ces algorithmes, ces outils ne pourront alors pas être utilisés en pratique bien qu'étant très performants. Ainsi, les ADS développés pour la médecine de précision, plus que d'être performants, doivent d'une part être interprétables par les médecins et d'autre part faire preuve d'une grande fiabilité.

Interprétabilité des modèles

Dans de nombreux domaines d'application, notamment pour la médecine de précision, le besoin d'employer des méthodes de ML interprétables est de plus en plus im-

portant [Ahmad *et al.*, 2018]. Dans [Doshi-Velez et Kim, 2017] plusieurs raisons sont évoquées quant à cette nécessité. Tout d'abord, l'interprétabilité des méthodes employées est importante pour acquérir de nouvelles pistes de recherche médicale à partir des résultats des modèles de ML. De plus, du point de vue éthique il est nécessaire de développer des modèles dont les résultats pourront être facilement interprétés par les médecins, la santé des patients étant en jeu [Bharati *et al.*, 2023]. En effet, il est d'une part important pour un médecin de pouvoir repérer les différentes failles que pourrait rencontrer l'algorithme. Par exemple, un expert du domaine médical pourrait attester de la discrimination involontaire faite par l'algorithme, d'erreurs systématiques ou encore d'incohérences biologiques [Chaddad *et al.*, 2023, Albahri *et al.*, 2023]. D'autre part si la règle de décision issue de la méthode est interprétable, elle permettrait alors de justifier le diagnostic fait par l'ADS. Il est difficilement concevable d'utiliser un ADS pour lequel le diagnostic est inexplicable et donc injustifiable auprès des patients.

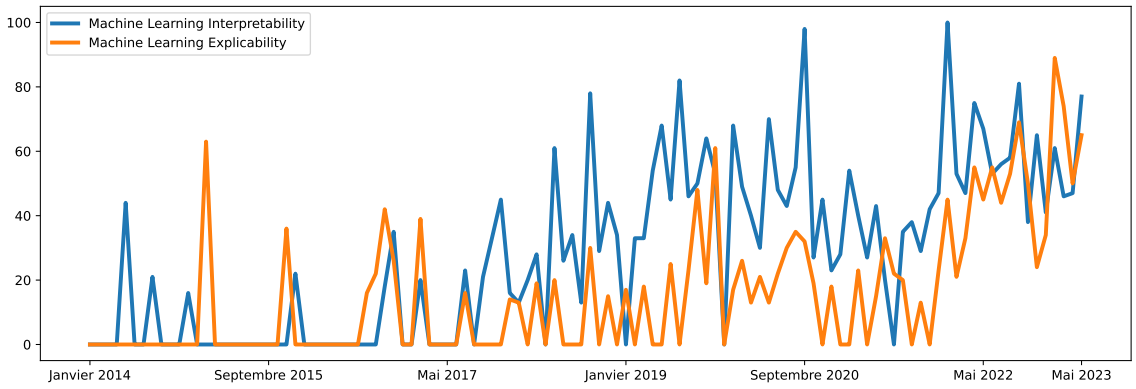

FIGURE 1.7 – Indice de popularité de Google Trends (Valeur maximale de 100) des termes "*Machine Learning Interpretability*" en bleu et "*Machine Learning Explicability*" en orange entre Janvier 2014 et Mai 2023.

Sur la Figure 1.7 nous pouvons remarquer que le nombre de recherches contenant l'expression "*Machine Learning Interpretability*" (courbe bleue) ne cesse d'augmenter depuis Janvier 2014. Néanmoins, la notion d'interprétabilité est large. D'après les exemples précédents, elle peut à la fois s'appuyer sur la possibilité d'analyser la règle de décision (interprétabilité du diagnostic) ou encore les transformations des données opérées par le modèle (cohérence biologique). D'après [Molnar, 2020], "*l'interprétabilité est le degré à quel point un humain peut expliquer de manière cohérente les prédictions du modèle*". Cette définition étant large, elle offre un grand panel d'appropriation de cette notion. Dans [Linardatos *et al.*, 2020] quatre grandes catégories de modèles pouvant être qualifiées d'interprétables sont définies. Tout naturellement la première regroupe les modèles "boîtes blanches", c'est à dire intrinsèquement interprétables. D'autre part un modèle expliquant une méthode complexe dite "boîte noire", peut être qualifié d'interprétable. Dans [Montavon *et al.*, 2017, Montavon *et al.*, 2018, Samek *et al.*, 2021], de nombreuses méthodes d'interprétation *post hoc* pour les réseaux de neurones, intervenant une fois le réseau entraîné sont proposées. Pour l'application médicale, certaines méthodologies s'intéressent directement à l'interprétation des prédictions issues de réseaux de neurones pour les experts du domaine, afin de lever de nouvelles hypothèses biologiques [Hanczar *et al.*, 2020]. Enfin, les travaux de [Gilpin *et al.*, 2018] tentent de définir des notions clé autour de l'interprétabilité du *Deep Learning* et proposent une taxonomie de

classification de l'interprétabilité des réseaux de neurones en trois catégories. La première regroupe les méthodes faisant apparaître les connexions entre les entrées et les sorties du modèle. La seconde contient les approches tentant d'expliquer la représentation des données d'entrée au sein du réseau. Enfin, la dernière se compose des réseaux de neurones s'expliquant eux-mêmes.

Puisque la notion même d'interprétabilité est ambiguë et propre à chacun il est davantage compliqué de développer des métriques universelles afin de pouvoir estimer le degré d'interprétabilité des méthodes. D'autres raisons expliquant le manque de formalisme mathématique réside dans le type de méthode considérée. Tout d'abord la notion d'interprétabilité pour des tâches d'analyse d'images est évidemment très différente de celle employée pour l'application à des données cliniques. Là encore, les potentielles métriques développées différeront. De plus, la notion d'interprétabilité dépend aussi du type de modèle considéré. Si l'application est restreinte à une certaine famille d'algorithme nous considérerons alors une interprétabilité spécifique au modèle (*Model specific*). En revanche, si elle s'applique à tous les algorithmes possibles alors elle sera qualifiée d'agnostique (*Model Agnostic*). Enfin, un des aspects crucial divisant la famille de méthodes interprétables est fondé sur l'échelle d'interprétation. Si la méthode permet une explication seulement d'une instance spécifique alors elle sera locale, tandis que si la méthode explique l'entièreté du modèle on la qualifiera de globale [Beaudouin *et al.*, 2020].

Pour plus de clarté dans la suite du manuscrit, nous allons définir la notion d'interprétabilité globale que nous considérons.

Définition 1 (Méthode interprétable)**.**
Une méthode est dite interprétable si d'une part sa règle de décision est interprétable et si d'autre part les connections entre les entrées et les sorties (transformations au sein du modèle) sont identifiables.

Ainsi, la définition de l'interprétabilité que nous retenons est à la fois cohérente avec celles de [Linardatos *et al.*, 2020] et [Guidotti *et al.*, 2018] puisque le modèle peut être caractérisé de "boîte blanche" ou de "boîte noire transparente" pour reprendre leur taxonomie respective. De plus, notre définition s'applique aussi aux réseaux de neurones puisqu'elle réunit les trois catégories définies par [Linardatos *et al.*, 2020]. La régression logistique (Section 2.2.1) donne des informations aux praticiens en estimant l'impact de chaque caractéristique clinico-biologique sur la variable à prédire. Il est alors possible de comprendre pourquoi un diagnostic est favorisé, ce qui rend la méthode interprétable. Les arbres de décision (Section 2.3.3) ou encore les modèles additifs généralisés (Section 2.3.2) sont aussi d'après cette définition des méthodes interprétables ; il est possible de justifier la prédiction de ces algorithmes en étudiant respectivement le partitionnement de l'espace d'entrée ou les splines estimées par variable. En revanche les règles de décision issues des forêts aléatoires (Section 2.3.3) ou des réseaux de neurones (Section 2.4) sont trop complexes pour pouvoir être analysées, ces méthodes de ML ne réunissent pas les conditions requises pour être qualifiées d'interprétables.

Explicabilité des méthodes

L'interprétabilité telle que définie précédemment ne fait pas intervenir de notion de fiabilité de la méthode développée par rapport à la tâche visée. Dans [Kulesza *et al.*, 2013], l'interprétation faite d'un ADS est dite fidèle ("*faithful*") si l'explication décrit l'entière

dynamique du modèle et ne révèle aucune incohérence biologique. Il est donc important de différencier les termes “interprétabilité” et “explicabilité” des méthodes, bien que souvent utilisés simultanément [Cabitza *et al.*, 2019, Miller, 2019]. Nous pouvons remarquer sur la Figure 1.7 que les mots clés “*Machine Learning Interpretability*” (courbe bleue) et “*Machine Learning Explicability*” (courbe orange) vont de paires. Ces dernières années le domaine de l’Intelligence Artificielle Explicable (XAI) [Gunning et Aha, 2019] prend de l’ampleur. En 2023 la recherche concernant l’explicabilité tend même à surpasser celle d’interprétabilité. Une fois encore, l’explicabilité des méthodes de ML est une notion très vague, dont la définition manque de formalisme [Adadi et Berrada, 2018], et qui souvent est confondue à celle d’interprétabilité. Quatre grandes catégories d’explicabilité des méthodes sont identifiées dans [Guidotti *et al.*, 2018], en fonction du type de problème pour lequel elles ont été créées : une catégorie pour expliquer les modèles de “boîte noire”, une pour les inspecter, une pour expliquer leurs résultats et enfin, une pour créer des modèles “boîtes noires transparents”. Dans [Guidotti *et al.*, 2018] un modèle est explicable s’il est à la fois interprétable et fidèle à la tâche pour laquelle il a été développé.

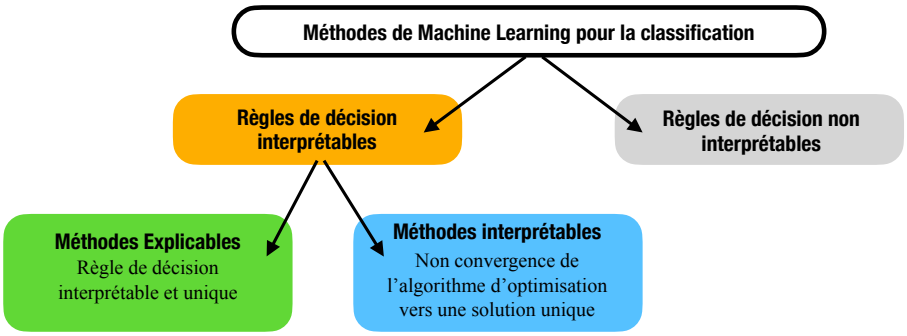

FIGURE 1.8 – Définitions proposées pour l’interprétabilité et l’explicabilité dans le *Machine Learning*.

La confiance et la fidélité d’un algorithme ne se résument pas seulement à son pouvoir de généralisation [Pineau *et al.*, 2021, Ali *et al.*, 2023]. Bien que les performances prédictives évaluées par CV soient un bon moyen d’évaluer la fiabilité d’un ADS sur différents échantillons dans sa globalité mais aussi au niveau de sous-groupes de population, cela n’est pas suffisant. Certaines méthodes de ML lors de leur apprentissage ne sont pas en mesure de fournir des règles de décision uniques. L’échantillon d’apprentissage utilisé pour apprendre la règle de décision aura naturellement un impact sur son apprentissage. Néanmoins, si nous supposons que la base de données à disposition pour apprendre les modèles est représentative de l’ensemble de la population, ce facteur ne pose pas de problèmes éthiques. En revanche, des facteurs aléatoires, tels que les initialisations des paramètres pour le processus d’apprentissage des modèles peuvent eux aussi avoir un impact non-négligeable sur la règle de décision développée. Plus précisément, il arrive souvent que des ADS entraînés sur les mêmes échantillons d’apprentissage mais avec de nouveaux paramètres initialisés aléatoirement mènent à des règles de décision différentes. Par exemple les réseaux de neurones, bien que très performants ne sont pas fiables : leur processus d’optimisation n’offre pas de garantie de convergence vers une solution unique. En médecine de précision, cela pose un véritable problème éthique ; il est inconcevable que des évènements aléatoires, que nous ne contrôlons pas puissent intervenir dans les décisions prises pour personnaliser les parcours de soin de patients. Nous ne pouvons pas envisager d’utiliser un ADS pour lequel d’un

processus d'entraînement à un autre, les règles de décision estimées soient relativement différentes : comment expliquer par exemple à un médecin que selon l'apprentissage de l'algorithme l'impact du poids des patients sur la prédiction de la pathologie puisse avoir un impact contradictoire ? Il est donc nécessaire de fournir aux médecins des ADS interprétables pour qu'ils puissent expliquer les résultats, comprendre le cheminement menant à la prise de décision et pour lesquels la règle de décision est unique conditionnellement à l'échantillon d'apprentissage et ne dépende pas de l'aléatoire intervenant au départ du processus d'entraînement des méthodes. Ainsi, pour la suite du manuscrit nous considérerons la définition suivante afin de qualifier une méthode comme explicable :

Définition 2 (Méthode explicable)**.**
Une méthode est dite explicable si sa règle de décision est d'une part interprétable conformément à la Définition 1, mais aussi unique conditionnellement à l'échantillon d'entraînement utilisé pour son apprentissage.

Ainsi, d'après notre définition, une méthode est dite explicable si elle est interprétable et offre des garanties d'unicité de ses estimations. Par rapport aux méthodes interprétables citées précédemment, seule la régression logistique (Section 2.2.1) est explicable. Les arbres de décision (Section 2.3.3) et les modèles additifs généralisés (Section 2.3.2) bien qu'interprétables ne sont pas explicables car il n'est pas possible de garantir la convergence de leur processus d'optimisation.

1.3 Collaborations médicales

Tout au long de la thèse, deux étroites collaborations médicales ont permis de comprendre les besoins et les attentes des praticiens. Tout au long des échanges, de nouveaux challenges et objectifs sont apparus afin de mener à bien le projet de développer des méthodes de ML destinées à améliorer les parcours de soin des patients.

1.3.1 Collaboration avec l'IPMC sur les maladies mentales

Notre recherche a été menée en collaboration avec l'Institut de Pharmacologie Moléculaire et Cellulaire de Sophia-Antipolis et plus particulièrement avec Nicolas Glaichenhaus (Professeur d'Immunologie) et Susana Barbosa (Ingénieure de recherche). Nous avons principalement collaboré ensemble sur deux domaines d'application de maladies mentales chroniques.

Le diagnostic de la bipolarité

Les troubles bipolaires touchent entre 1 et $2,5\%$ de la population, soit entre 650 000 et 1 650 000 personnes en France[3]. La bipolarité se caractérise chez les patients par l'alternance de phases euphoriques et dépressives pouvant durer chacune d'entre elles quelques jours à quelques mois. Les périodes dépressives peuvent souvent amener à des tentatives de suicide : 20% des patients bipolaires non traités décèdent par suicide. Cette maladie contraignante affecte la vie quotidienne des patients bipolaires, si bien que l'Organisation Mondiale de la Santé (OMS) la place au 6ème rang mondial des handicaps. C'est pourquoi il est important de pouvoir la détecter le plus rapidement possible. Or, il est estimé qu'en moyenne 10 ans séparent le premier épisode et l'instauration d'un traitement. Ce retard

3. Fondation Fondamental : https://www.fondation-fondamental.org/les-maladies-mentales/les-troubles-bipolaires

de diagnostic pénalise le patient et s'explique par la difficulté de dissocier les symptômes de la bipolarité de ceux de la dépression. Les patients bipolaires ont tendance à consulter lorsqu'ils traversent une phase de dépression et non d'euphorie. Ainsi, il est estimé que 40% des patients détectés comme dépressifs sont en réalité bipolaires.

Quand bien même le diagnostic de la bipolarité est établi, il est très difficile d'instaurer un traitement médical adapté aux caractéristiques spécifiques des patients. Il est à noter que la bipolarité est une maladie chronique dont on ne peut guérir mais dont on peut atténuer les symptômes au travers de médicaments, de psychothérapies et du respect d'une bonne hygiène de vie. Le traitement thérapeutique (thymorégulateur) vise à stabiliser l'humeur des patients bipolaires, à savoir leur éviter l'alternance de phases de dépression et d'euphorie. Il est estimé qu'entre 25 et 30% des bipolaires suivent un traitement à base de lithium, reconnu comme étant le thymorégulateur le plus efficace. Néanmoins adapter son dosage est très difficile : un dosage trop petit ne permet pas la stabilisation des humeurs euphoriques tandis qu'un dosage trop élevé aurait pour risque d'entraîner des effets secondaires incommodants comme le dérèglement de la thyroïde ou de la fonction filtrante des reins.

Tout au cours de notre collaboration, nous nous sommes intéressés plus particulièrement au diagnostic de la bipolarité, plus précisément au développement d'un ADS permettant de différencier les patients dépressifs des bipolaires à l'aide de variables clinico-biologiques telles que l'âge, le poids, le cholestérol ou encore certaines cytokines (protéines). Pour établir ce pronostic, nous disposons de données cliniques et biologiques hétérogènes (catégorielles et numériques). De plus, cette base de données est relativement petite. Ainsi, cette étude est un parfait exemple des challenges techniques et sociaux détaillés précédemment. Les résultats de cette analyse sont présentés dans le Chapitre 7.

La personnalisation de traitements pour la schizophrénie

La Schizophrénie est une maladie chronique sévère dont souffre en France 600 000 personnes [4]. Afin de la détecter, il existe trois groupes de symptômes présents chez les patients simultanément. Les symptômes dits "positifs" s'ajoutent aux perceptions ordinaires et se caractérisent notamment par des hallucinations auditives ou visuelles. La réduction de l'ensemble des activités, comme la difficulté à se concentrer et mener une action ou le manque d'énergie sont appelés symptômes "négatifs". Enfin, les patients schizophrènes souffrent d'une désorganisation de la pensée et du comportement ayant pour conséquence par exemple de rendre leur discours flou ou incohérent. Tout comme pour la bipolarité, l'OMS classe la schizophrénie dans le groupe des 10 maladies entraînant le plus d'invalidité.

Lorsque cette maladie est prise en charge tôt, il est estimé que 15 à 20% des schizophrénies débutantes évoluent favorablement. De plus, lorsqu'un traitement thérapeutique adapté est rapidement prescrit, une rémission satisfaisante et une réinsertion sociale partielle voire totale est permise dans la moitié des cas. Les antipsychotiques (neuroleptiques) ont pour but d'estomper les symptômes chez les patients à savoir la réduction des hallucinations, des idées délirantes ou de la désorganisation de la pensée ainsi que de permettre aux patients de retrouver une activité sociale. Pour que ce traitement soit efficace, il convient d'adapter son dosage mais parfois aussi de le combiner à d'autres médicaments. Il arrive que l'association des antipsychotiques avec des antidépresseurs,

4. Fondation Fondamental : https://www.fondation-fondamental.org/les-maladies-mentales/schizophrenie

des anxiolytiques ou des thymorégulateurs soit nécessaire pour que les symptômes de la schizophrénie soit atténués chez certains patients.

Ainsi, notre collaboration avec l'IPMC a visé à prédire l'efficacité des traitements dans le cas de la schizophrénie. Contrairement au cas de la bipolarité, nous ne disposons dans la base de données cette fois que de patients schizophrènes, pour lesquels il convient de personnaliser le traitement.

1.3.2 Collaboration avec le service d'Hépatologie du CHU de Nice

Le Professeur Rodolphe Anty (Gastro-entérologue et hépatologue du CHU de Nice) consacre depuis plusieurs années ses recherches à la prédiction de la NASH (*Non-Alcoholic SteatoHepatitis*) Fibrosante.

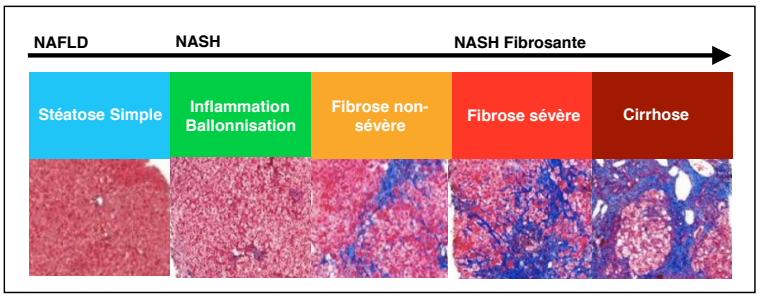

FIGURE 1.9 – Dégradation du foie menant à la Cirrhose non-alcoolique.

La stéatose hépatique (NAFLD : *Non-Alcoholic fatty liver disease*) est une surcharge en graisse du foie, sans rapport avec l'alcool (Groupe bleu sur la Figure 1.10). Parmi les personnes en surpoids atteintes de cette maladie le plus souvent asymptomatique, deux groupes peuvent être distingués : ceux ayant une NAFLD sans NASH (Groupe vert sur la Figure 1.10) et ceux avec NASH (Groupe orange). La présence de NASH permet de déterminer la gravité de l'impact de la graisse sur le bon fonctionnement du foie. Afin de déterminer la présence de NASH, le score NAS (*Non-alcoholic fatty liver diseasse Activity Score*) a été élaboré. S'il est supérieur à 4 on peut alors dire que la personne est atteinte de NASH. Ces personnes sont généralement atteintes de stéatose accompagnée de ballonnements et d'inflammation. Selon la présence supplémentaire d'une fibrose importante (supérieure à 2), les patients sont alors atteints de ce qui est couramment appelée NASH Fibrosante (Groupe rouge sur la Figure 1.10). En cas de non prise en charge de ces patients et de non traitement, sa progression peut conduire à la cirrhose. Il est donc important pour les médecins de diagnosticr correctement les personnes atteintes de NASH Fibrosante.

De nombreuses recherches [R. Anty et all., 2010, J. Boursier et all, 2018, Philip N Newsome et all., 2020] ont été menées dans le but de diagnosticr la NASH Fibrosante sans avoir à utiliser des méthodes invasives (biopsies) à la fois lourdes pour les patients et coûteuses pour le système hospitalier. Malgré de belles avancées, les techniques utilisées restent parfois trop complexes pour les médecins. Dans l'article [R. Anty et all., 2010] par exemple, la prise en compte de la cytokeratin-18 dans le modèle peut être considérée comme une limite. En effet, mesurer cette variable est complexe et coûteuse. Dans l'article [J. Boursier et all, 2018] la base de données sur laquelle le modèle

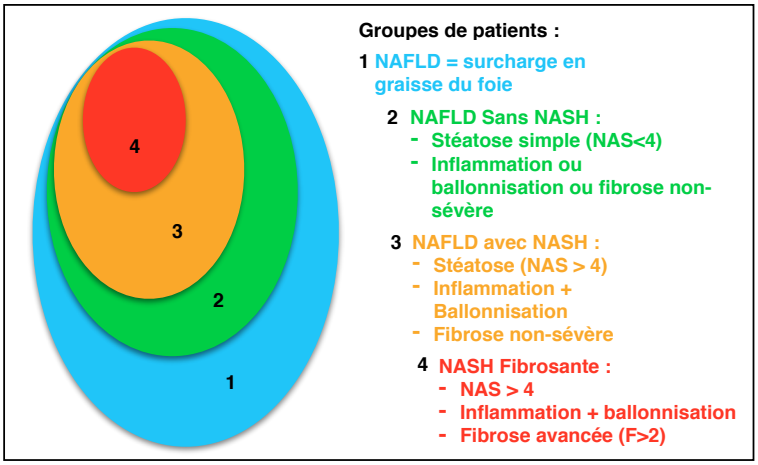

FIGURE 1.10 – Les différentes catégories de stéatose hépatique. En bleu nous retrouvons les patients présentant une surcharge en graisse du foie (NAFLD) que nous pouvons diviser en deux sous groupes : ceux sans NASH en vert et avec NASH en orange. Finalement la catégorie rouge représente la population atteinte d'une NASH Fibrosante.

a été entraîné ne reflète pas la population totale (beaucoup de patients malades présents), ce qui rend le modèle difficilement généralisable et donc peu fiable. Enfin, dans les articles [J. Boursier et all, 2018] et [Philip N Newsome et all., 2020] les techniques développées proposent trois classes : les malades, les non-malades et une zone d'ombre ("*Grey Zone*"). Cette classe regroupant plus de 34% des patients du panel pour le premier article cité recommande la vérification par biopsie. Ainsi, il semble très difficile de diagnosticr la NASH Fibrosante de manière non-invasive.

L'objectif de notre collaboration est alors de développer un ADS afin de diagnosticr la NASH Fibrosante ne nécessitant ni méthode invasive, ni caractéristiques difficiles à définir et à relever. Pour établir ce pronostic, nous disposons de données cliniques et biologiques hétérogènes (catégorielles et numériques). Comme pour la grande majorité des bases de données médicales, les patients d'intérêt à savoir ayant développé une NASH Fibrosante sont minoritaires dans ce jeu de données déséquilibré.

1.4 Organisation du manuscrit

L'objectif principal de cette Thèse a donc été de développer un ADS pour la prédiction de pathologie ou la personnalisation de traitements adaptés aux besoins des cliniciens. Tout ce travail a été mené en étroite collaboration avec les praticiens mentionnés précédemment afin de répondre au mieux à leurs attentes et d'être en mesure de fournir un ADS utilisable par les spécialistes des différents domaines d'application.

Dans le Chapitre 2 nous conceptualisons dans un premier temps le problème de classification supervisée. Dans un second temps, nous présentons différentes méthodes de ML de l'état de l'art ayant inspiré nos travaux. Nous nous sommes concentrés sur les méthodes de classification non linéaires. D'après les experts du domaine bio-médical il est nécessaire de pouvoir modéliser des effets non linéaires tels que des effets de seuil pour prédire correctement les pathologies ou adapter les traitements aux caractéristiques des

patients. La grande difficulté réside dans le fait d'obtenir à la fois des règles de décision performantes, interprétables et fiables. Les algorithmes les plus performants sont aussi les plus complexes et dont les règles de décision sont parfois difficilement interprétables. Les méthodes sont détaillées en fonction de leur degré d'explicabilité. Nous commençons par introduire les méthodes explicables, et plus précisément la Régression Logistique. Ensuite, nous nous intéressons aux méthodes interprétables, produisant des règles de décision interprétables mais pour lesquelles nous ne disposons d'aucune garantie d'unicité des estimations, à savoir la Régression Multivariée par Splines Additives, les Modèles Additifs Généralisés et les Machines Explicables Boostées. Enfin, nous introduisons les Réseaux de Neurones, des méthodes "Boîtes Noires", dont il est très difficile, si ce n'est impossible d'interpréter les prédictions.

Dans le Chapitre 3 nous introduisons nos premières contributions. Le Réseau de Neurones MARS (RN-MARS) et le modèle *Splines Approximation Throught Univariate ReLU Neural Network* (SATURNN) sont fortement inspirés des méthodes de l'état de l'art. En effet, nous nous sommes inspirés des avantages d'interprétabilité de certaines d'entre elles tout en s'affranchissant de leurs contraintes d'optimisation. Le SATURNN est un Réseau de Neurones contraint afin de comprendre les transformations opérées sur les données (cheminement entre les entrées et les sorties du réseau) et ainsi modéliser une règle de décision interprétable. Pour ce faire, l'architecture du SATURNN est contrainte telle que chaque variable explicative est segmentée indépendamment les unes des autres. La règle de décision qui en découle se modélise comme une somme additive de splines univariées, facilement interprétable. En effet, il est possible de visualiser l'impact estimé de chaque variable d'entrée sur la prédiction. Cette méthode bien qu'interprétable au sens de la Définition 1, n'est cependant pas explicable. L'apprentissage des réseaux de neurones nécessite de minimiser un problème d'optimisation non convexe, ne fournissant aucune garantie de convergence des estimations vers une solution unique.

Afin de palier à cette limite, nous proposons dans le Chapitre 4 une Régression Logistique appliquée aux données préalablement transformées par une application non linéaire découlant de l'architecture du SATURNN. Dans un premier temps, nous linéarisons localement la couche cachée du SATURNN, au voisinage de ses initialisations. Ensuite, nous démontrons qu'il est équivalent d'entraîner une Régression Logistique appliquée aux données préalablement transformées par la couche cachée linéarisée, notée LR PSI LIN, ou un SATURNN composé d'un grand nombre de neurones. En réinjectant les paramètres estimés par la LR PSI LIN dans le SATURNN, nous pouvons retrouver la règle de décision se modélisant comme une somme additive de splines univariées. En revanche, le problème d'optimisation de la LR PSI LIN est fortement convexe et ainsi nous disposons de paramètres estimés uniques. Le SATURNN retrouvé est donc explicable conformément à la Définition 2 ; la règle de décision qui en découle est interprétable et unique. Néanmoins, le processus d'entraînement ainsi que les estimations de cette méthode restent dépendants du processus d'initialisation aléatoire des paramètres du SATURNN. Ainsi, l'unicité de la règle de décision issue est conditionnelle aux initialisations du SATURNN.

Dans le Chapitre 5 nous montrons que la transformation non linéaire appliquée aux données par la LR PSI LIN se réécrit sous la forme d'un noyau. Ainsi un premier noyau découlant directement de l'architecture du SATURNN et dépendant de ces initialisations est introduit. Il devient alors équivalent d'entraîner la LR PSI LIN ou une Régression Logistique appliquée à ce noyau, notée KLR. Le noyau opère un partitionnement de l'espace

d'entrée multivarié, qui au premier abord est difficilement interprétable. En revanche, en réinjectant les paramètres appris par la KLR dans un SATURNN nous retrouvons une règle de décision explicable, se modélisant comme une somme additive de splines univariées et uniques conditionnellement aux initialisations du SATURNN considérées. Nous démontrons dans un second temps que ce noyau converge asymptotiquement vers une limite finie. La dernière contribution de ce manuscrit est donc l'introduction d'une Régression Logistique à Noyau Déterministe, notée EKLR. La transformation opérée sur les données par ce noyau ne dépend cette fois d'aucun paramètre aléatoire, mais seulement de l'ensemble de l'échantillon d'apprentissage. Nous pouvons réécrire la règle de décision issue de l'EKLR comme celle du SATURNN, facilement interprétable. De plus, entraîner l'EKLR revient à optimiser un problème d'optimisation fortement convexe, les estimations qui en résultent sont donc uniques. Cette méthode de classification est donc explicable et a pour avantage d'estimer une règle de décision unique conditionnellement seulement à l'échantillon d'apprentissage.

Diverses expériences numériques menées sur des données simulées sont présentées dans les chapitres de contribution du manuscrit pour valider et confirmer les différents résultats théoriques établis. Le Chapitre 6 est consacré à la mise en valeur de nos différents travaux sur des données réelles médicales. Nous comparons les résultats des contributions de ce manuscrit aux méthodes de l'état de l'art sur trois bases de données réelles publiques afin qu'ils puissent être répliqués. Les caractéristiques des bases de données sont toutes différentes les unes des autres, tant sur le nombre d'échantillons et de variables clinico-biologiques à disposition. Nous comparons les performances prédictives de nos contributions aux méthodes de l'état de l'art et mettons en avant l'avantage de leur explicabilité par rapport à la fiabilité des interprétations.

Le Chapitre 7 est quant à lui consacré aux travaux réalisés en collaboration étroite avec Nicolas Glaichenhaus et Susana Barbosa de l'IPMC. Nous détaillons dans un premier temps les différentes étapes de préparation des données. Ensuite nous proposons des modèles estimés sur l'ensemble des variables clinico-biologiques à disposition. L'instabilité des résultats nous a conduit à réaliser au préalable une sélection des variables les plus pertinentes. Nous présentons la méthodologie employée afin de limiter le problème de *Confonding* souvent rencontré en application médicale du fait de la forte corrélation des caractéristiques clinico-biologiques entre elles. Les modèles entraînés avec un nombre de variables restreint sur-apprennent beaucoup moins, bien que restant relativement instables pour la prédiction de la bipolarité. C'est pourquoi nous introduisons la notion de Zone Grise, une zone d'incertitude pour laquelle les algorithmes ne prennent aucune décision. Les Zones Grises obtenues étant larges, ce qui rend difficile l'utilisation de ces modèles, nous développons dans un dernier temps des modèles sexe-spécifiques. Nous savons que le genre des patients influe sur leurs parcours de soin. Néanmoins, apprendre des modèles sexes-spécifiques revient à sous-diviser la base de données qui, initialement est déjà petite. Enfin, nous concluons ce chapitre par une discussion autour de l'acquisition de ces données et les conséquences engendrées sur l'apprentissage des méthodes.

Enfin, le Chapitre 8 conclut les travaux présentés dans ce manuscrit. Nous apportons des pistes de recherche intéressantes pour améliorer la personnalisation des parcours de soin et plus précisément le SATURNN et ses méthodes d'approximation. Pour finir, nous introduisons une liste détaillée des articles de recherche présentés en référence avec nos travaux de recherche lors de diverses conférences nationales et internationales.

Chapitre 2

Positionnement du problème et état de l'art

Dans ce chapitre, nous conceptualisons dans un premier temps le problème de classification supervisée (Section 2.1). Ensuite, nous présentons diverses méthodes de ML ayant inspiré nos travaux en fonction de leur degré d'explicabilité. La section 2.2.1 est dédiée à la présentation des méthodes explicables pour la classification supervisée et plus précisément la Régression Logistique (LR). Dans la Section 2.3 nous présentons trois méthodes interprétables plus performantes que les précédentes : la Régression Multivariée par Splines Additives (MARS, sous-section 2.3.1), les Modèles Additifs Généralisés (GAM, sous-section 2.3.2) et les Machines Explicables Boostées (EBM, sous-section 2.3.3). Enfin nous introduirons des méthodes qualifiées de "boîtes noires" mais dont les performances prédictives pour les problèmes de classification non linéaire ne sont plus à prouver : les Réseaux de Neurones (RN, Section 2.4).

Sommaire

2.1 La classification binaire supervisée

Nous considérons un problème de classification binaire supervisée. Nous supposons un ensemble de données $\mathcal{D} = \left\{x^{(i)}, y^{(i)}\right\}_{i=1}^{N}$ composé de N couples indépendants et identiquement distribués, où $x^{(i)} \in \mathbb{R}^d$ est le vecteur des variables explicatives de l'observation i et $y^{(i)} \in \{0, 1\}$ est l'étiquette binaire à prédire. La notation (X, Y) désigne le couple de variables aléatoires dont sont issus les couples $(x^{(i)}, y^{(i)})$. Nous cherchons alors une fonction $f : \mathbb{R}^d \to \{0, 1\}$ telle que :

$$Y = f(X). \tag{2.1}$$

En d'autres termes, nous cherchons alors à estimer une fonction discriminante afin d'identifier les échantillons issus des deux classes. De nombreuses méthodes existent et se différencient de part leurs approches respectives. Tout d'abord, nous pouvons discerner les classifieurs linéaires et non linéaires.

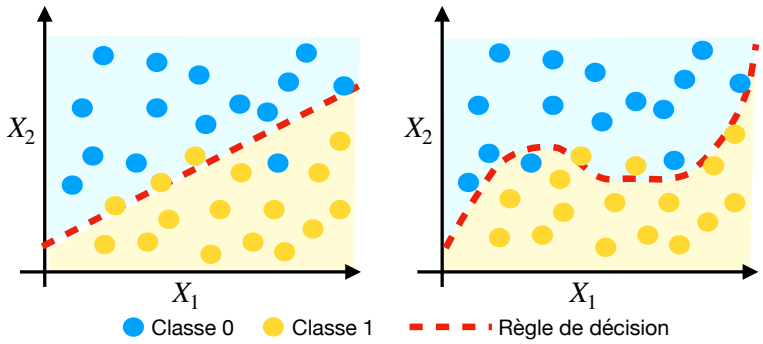

FIGURE 2.1 – Illustration d'une règle de décision linéaire (gauche) et non linéaire (droite). Les échantillons appartenant à la classe 0 sont en bleu et ceux issus de la classe 1 en jaune. Les règles de décision sont en rouge.

Sur la Figure 2.1-gauche nous pouvons visualiser l'exemple d'un classifieur linéaire, c'est à dire un classifieur estimant une fonction discriminante linéaire. Tandis que sur la Figure 2.1-droite, une règle de décision non linéaire est représentée. Dans de nombreux domaines d'application, notamment en médecine de précision, l'introduction de phénomènes non-linaires permet d'obtenir un gain de performance prédictive. Néanmoins, cela a pour inconvénient d'ajouter de la complexité dans la règle de décision et peut donc rapidement amener à des modèles difficilement interprétables [Wanner *et al.*, 2021].

Sur la Figure 2.2-gauche nous comparons différentes méthodes de Machine Learning pour la classification en fonction de leurs performances prédictives et du niveau d'interprétabilité de leurs règles de décision. Nous pouvons constater que les modèles non linéaires tels que les Forêts Aléatoires (RF) [Breiman, 2001, Hastie *et al.*, 2009], les Modèles Additifs Généralisés (GAM) [Hastie, 2017, Hastie *et al.*, 2009] ou encore les Machines à Vecteurs de Support (SVM) [Cortes et Vapnik, 1995, Hastie *et al.*, 2009] sont plus performants que la Régression Logistique (LR) [Hastie *et al.*, 2009, Murphy, 2012], méthode linéaire très reconnue en médecine de précision. En revanche, l'interprétabilité de leurs règles de décision est moins évidente. Les Réseaux de Neurones (RN) [LeCun *et al.*, 2015] sont réputés pour être très performants car leur architecture leur permet d'introduire de nombreux effets non linéaires dans la modélisation. Néanmoins, les RN sont souvent qualifiés de "boîtes noires" [Fel et Vigouroux, 2020] car leur règle de

décision est difficilement, si ce n'est impossible à interpréter.

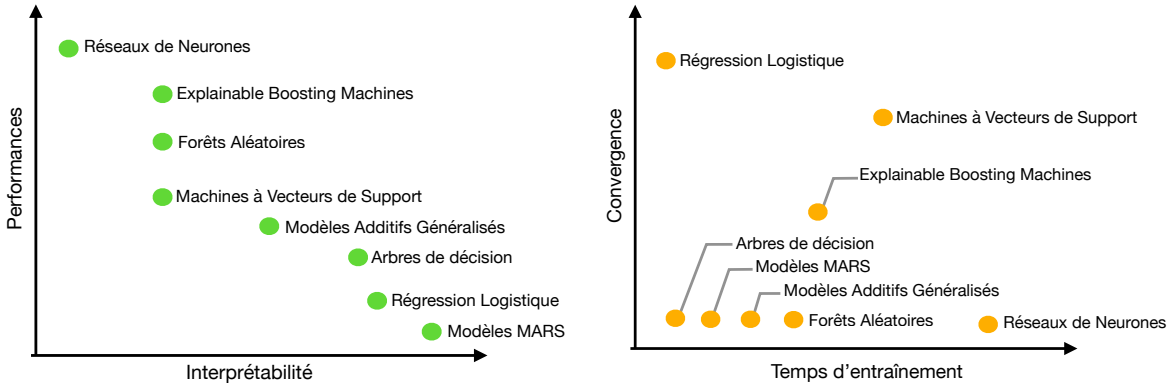

FIGURE 2.2 – Schémas comparatifs de méthodes de Machine Learning couramment utilisées pour les tâches de classification. Sur la Figure de gauche nous comparons les performances prédictives des ces méthodes et le niveau d'interprétabilité de leurs règles de décision. À droite, nous opposons les méthodes en fonction de leur temps d'apprentissage et de leur garantie de convergence vers des solutions uniques.

La deuxième grande difficulté quant à l'intégration de phénomènes non linéaires dans la modélisation réside dans l'apprentissage de la méthode. Sur la Figure 2.2-droite nous comparons des méthodes de Machine Learning couramment employées en fonction du temps nécessaire à leur apprentissage et des garanties de convergence et donc d'unicité de leurs estimations dont nous disposons. La LR est la méthode la plus rapide à entraîner et peut offrir des garanties quant à l'unicité de ces résultats. De nombreuses méthodes non linéaires telles que les Arbres de Décision (DT) [Hastie *et al.*, 2009], la Régression Multivariée par Splines Adaptatives (MARS) [Friedman, 1991] ou encore les GAM ajoutent itérativement des effets de seuils à leurs règles de décision. De part leur apprentissage itératif, qualifié de glouton, aucune garantie d'optimalité, de convergence et donc de robustesse ne peut être établie pour ces algorithmes gloutons. Les RF en agrégeant les DT, tout comme les Machines Explicables Boostées (EBM) [Lou *et al.*, 2012] qui boostent l'apprentissage des RF, offrent un gain de performance non négligeable. Néanmoins ils diminuent l'interprétabilité de la règle de décision estimée et augmentent le temps de calcul nécessaire à leur apprentissage sans offrir de garanties de convergence et d'optimalité. De plus, si elles ne sont pas calibrées correctement, comme par exemple à l'aide d'un algorithme de *gridsearch* des paramètres optimaux, ces méthodes ont tendance à facilement sur-apprendre la base d'apprentissage et donc à moins bien se généraliser.

Ainsi, il existe un compromis important entre performance, interprétabilité, robustesse et rapidité d'entraînement. Parmi les méthodes de classification non linéaires interprétables, il convient dans un premier temps de distinguer celles qui sont convergentes (Section 2.2) et celles qui nécessitent un algorithme d'apprentissage itératif et sous-optimal (Section 2.3). Pour les méthodes non linéaires interprétables convergentes nous détaillons plus précisément les avantages et les inconvénients de la Régression Logistique à Splines Naturelles Cubiques (sous-Section 2.2.2). Tandis que pour les classifieurs non linéaires interprétables à algorithme itératif nous présentons les MARS (sous-Section 2.3.1) et les Modèles Additifs Généralisés ainsi que ses extensions (sous-Section 2.3.2). Dans un second temps, parmi les méthodes performantes et dont l'interprétabilité de la règle de décision est difficile si ce

n'est impossible, plusieurs noyaux couramment utilisés pour les Machines à Support Vectoriels sont détaillés dans le chapitre 5. Dans ce chapitre, nous nous concentrons sur les Réseaux de Neurones (Section 2.4).

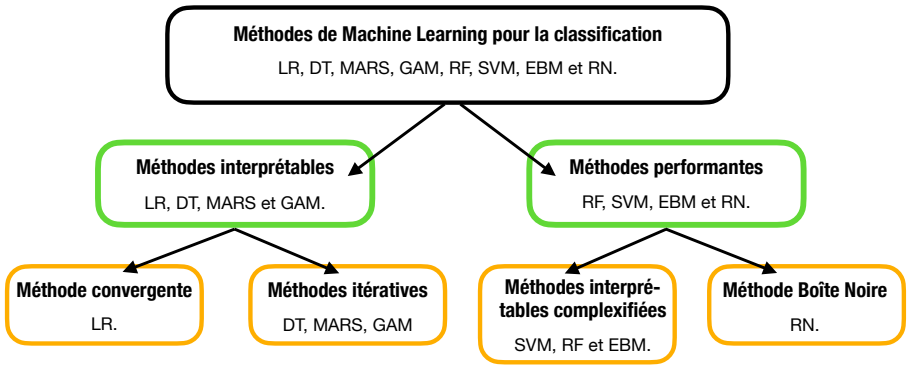

FIGURE 2.3 – Schéma des différentes grandes catégories de méthodes de classification mentionnées précédemment selon leur performance, leur interprétabilité et leur convergence. Les catégories en vert sont issues de la Figure 2.2-gauche et celles en orange de la Figure 2.2-droite.

2.2 Méthodes explicables : Régressions Logistiques

Dans cette partie, nous nous intéressons à la LR reconnue pour son explicabilité. La LR appliquée à une combinaison linéaire de splines naturelles cubiques a pour avantage de fournir des garanties de convergence et donc d'unicité de ces estimateurs.

2.2.1 La Régression Logistique linéaire

La LR [Hastie *et al.*, 2009, Murphy, 2012] est couramment employée pour l'application médicale, sa règle de décision étant très informative pour les experts du domaine et facilement interprétable. Un classifieur du Maximum a Posteriori (MAP) attribue à un échantillon $x = [x_1, \ldots, x_d]$ l'étiquette $\hat{y} = \{0, 1\}$ la plus probable :

$$\hat{y} = \underset{y \in \{0,1\}}{\arg\max}\, \hat{\mathbb{P}}(Y = y | X = x), \tag{2.2}$$

avec $\hat{\mathbb{P}}(Y = y|X = x)$ la probabilité estimée d'appartenir à la classe y sachant l'ensemble des variables d'entrée x. Puisque nous considérons un cas de classification binaire, nous avons $\hat{\mathbb{P}}(Y = 0|X = x) = 1 - \hat{\mathbb{P}}(Y = 1|X = x)$. Ainsi, le classifieur MAP défini à l'équation (2.2) peut se réécrire de la manière suivante :

$$\hat{y} = \begin{cases} 1 & \text{si } \hat{\mathbb{P}}(Y = 1|X = x) \geq 0.5 \\ 0 & \text{sinon.} \end{cases} \tag{2.3}$$

En d'autres termes, un classifieur MAP attribue l'étiquette 1 si la probabilité conditionnelle a posteriori d'appartenir à cette classe $\hat{\mathbb{P}}(Y = 1|X = x)$ est supérieure ou égale à 0.5, comme illustré par la Figure 2.4. Cette probabilité est estimée par une règle de décision que l'on note $\delta : \mathbb{R}^d \to [0, 1]$. La LR est un classifieur MAP dont la règle de décision se

modélise de la manière suivante :

$$\delta^{\mathrm{LR}}(x, \beta) = \sigma(f_\beta(x)) = \frac{1}{1 + \exp(-f_\beta(x))}. \tag{2.4}$$

La LR applique la sigmoïde σ (illustrée par la Figure 2.4) à la fonction de score $f_\beta(x)$. Traditionnellement, lorsque l'on applique la LR linéaire, $f_\beta(x)$ est la combinaison linéaire des variables descriptives :

$$f_\beta(x) = \beta_0 + \sum_{i=1}^{d} \beta_i x_i, \tag{2.5}$$

avec, $\beta = [\beta_0, \beta_1, \ldots, \beta_d] \in \mathbb{R}^{d+1}$ le vecteur de coefficients à estimer. Les coefficients β_i, $i \in \{1, \ldots, d\}$ quantifient l'impact de chaque composante du vecteur x_i sur la probabilité d'appartenir à la classe $y = 1$. Si nous estimons $\beta_i < 0$, alors plus x_i augmente, moins $\hat{\mathbb{P}}(Y = 1|X = x)$ sera élevée. Prenons un exemple médical et supposons que la classe des patients malades soit représentée par l'étiquette $y = 1$. Si la variable âge obtient un coefficient $\beta_{\text{âge}} > 0$ alors le modèle estime que vieillir augmente les risques d'être malade. À l'inverse si $\beta_{\text{sport}} < 0$ alors faire de l'activité physique semble protéger des risques de développer la pathologie. Ainsi, cette règle de classification est facilement interprétable par les experts du domaine sur lequel on l'applique.

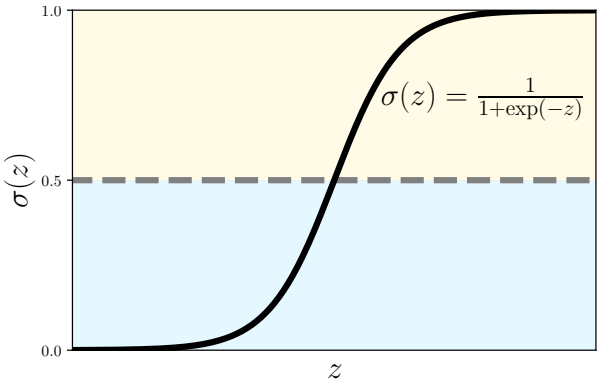

FIGURE 2.4 – Illustration de la règle de décision issue d'une Régression Logistique binaire. La sigmoïde définie à l'équation (2.4) est tracée en noir. La règle de décision MAP définie à l'équation (2.3) est illustrée par les encadrés de couleur. En bleu nous avons $\hat{\mathbb{P}}(Y = 1|X = x) < 0.5$ et donc $\hat{y} = 0$. A contrario, en jaune nous avons $\hat{\mathbb{P}}(Y = 1|X = x) \geq 0.5$ et donc $\hat{y} = 1$.

Afin de modéliser la probabilité $\hat{\mathbb{P}}(Y = 1|X = x)$ dans (2.3), il convient d'estimer le vecteur de paramètres $\beta \in \mathbb{R}^{d+1}$ composant la fonction de score $f_\beta(x)$ (2.5) à laquelle la sigmoïde (2.4) est appliquée. Pour ce faire, la méthode du maximum de vraisemblance conditionnelle est utilisée. Nous cherchons à maximiser la vraisemblance qui se définit comme étant la probabilité conditionnelle jointe $L_\beta(\mathcal{D}) = \mathbb{P}_\beta\left(Y_1 = y^{(1)}, ..., Y_N = y^{(N)}|X_1 = x^{(1)}, ..., X_N = x^{(N)}\right)$. Les échantillons $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^N$ sont supposés indépendants, la vraisemblance du problème de classification se réécrit alors :

$$\mathfrak{L}(\beta, \mathcal{D}) = \prod_{i=1}^{N} \hat{\mathbb{P}}_\beta\left(Y_i = y^{(i)}|X_i = x^{(i)}\right). \tag{2.6}$$

Rappelons que nous considérons une tâche de classification binaire, tel que $y \in \{0, 1\}$, et

$\hat{\mathbb{P}}(Y=0|X=x) = 1 - \hat{\mathbb{P}}(Y=1|X=x)$. Ainsi nous pouvons facilement vérifier que nous avons pour tout échantillon fixé $i \in \{1, \dots, N\}$:

$$\hat{\mathbb{P}}_\beta\left(Y_i = y^{(i)}|X_i = x^{(i)}\right) = \hat{\mathbb{P}}_\beta\left(Y_i = 1|X_i = x^{(i)}\right)^{y^{(i)}} \times \left(1 - \hat{\mathbb{P}}_\beta\left(Y_i = 1|X_i = x^{(i)}\right)\right)^{1-y^{(i)}}.$$

En effet nous retrouvons bien avec cette réécriture si $y^{(i)} = 0$: $\hat{\mathbb{P}}_\beta\left(Y_i = y^{(i)}|X_i = x^{(i)}\right) = \hat{\mathbb{P}}(Y_i = 0|X_i = x^{(i)})$. Cette égalité reste vraie dans le cas où $y^{(i)} = 1$. Ainsi, nous pouvons réécrire la vraisemblance (2.6) que nous souhaitons maximiser comme ci-dessous :

$$\mathfrak{L}(\beta, \mathcal{D}) = \prod_{i=1}^{N} \mathbb{P}_\beta\left(Y_i = 1|X_i\right)^{y^{(i)}} \times \left(1 - \mathbb{P}_\beta\left(Y_i = 1|X_i\right)\right)^{1-y^{(i)}}. \tag{2.7}$$

Pour faciliter la résolution du problème (2.7), il convient de considérer une fonction de coût, c'est à dire une fonction à minimiser, nous passons à la log-vraisemblance négative car $\arg\max_\beta \mathfrak{L}(\beta, \mathcal{D}) \Leftrightarrow \arg\min_\beta - \log(\mathfrak{L}(\beta, \mathcal{D}))$, avec

$$\log(\mathfrak{L}(\beta, \mathcal{D})) = \sum_{i=1}^{N} y^{(i)} \log \hat{\mathbb{P}}_\beta(Y_i = 1|X_i) + (1 - y^{(i)}) \log(1 - \hat{\mathbb{P}}_\beta(Y_i = 1|X_i)). \tag{2.8}$$

Ainsi, maximiser la vraisemblance $L(\beta)$ définie à l'équation (2.7) dans le cas de la classification binaire revient à minimiser le négatif du logarithme de cette vraisemblance. À partir de l'équation (2.8), nous pouvons remarquer que cette log-vraissemlance négative est égale à la fonction de coût Entropie Croisée Binaire (BCE) [Goodfellow *et al.*, 2020, Murphy, 2012]. Ainsi, maximiser la vraisemblance (2.7) revient à minimiser la BCE. La BCE appliquée à une règle de décision $\delta : \mathbb{R}^d \to [0, 1]$ se définit comme :

$$L(\delta(\beta, x), y) = -y \log\left(\delta(x, \beta)\right) + (1 - y) \log\left(1 - \delta(x, \beta)\right). \tag{2.9}$$

Finalement, le problème d'optimisation de la LR (δ^{LR} définie à l'équation (2.4)) permettant d'estimer les coefficients $\beta \in \mathbb{R}^{d+1}$ revient à minimiser la BCE moyenne :

$$\hat{\beta}^{\text{LR}} = \underset{\beta \in \mathbb{R}^{d+1}}{\arg\min}\, \mathcal{L}^{\text{LR}}(\beta, \mathcal{D}), \tag{2.10}$$

avec

$$\mathcal{L}^{\text{LR}}(\beta, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} L\left(\delta^{\text{LR}}(\beta, x^{(i)}), y^{(i)}\right). \tag{2.11}$$

Afin d'optimiser le problème défini à l'équation (2.10), il convient d'annuler le gradient de la fonction de coût $\mathcal{L}^{\text{LR}}(\beta, \mathcal{D})$ définie à l'équation (2.11). Les dérivées partielles étant non linéaires en β la résolution du problème nécessite alors un apprentissage itératif, à l'aide d'une descente de gradient [Hastie *et al.*, 2009, Murphy, 2012]. Différents algorithmes d'optimisation ont été développés et sont comparés dans [Minka, 2003]. Le plus couramment utilisé est celui de descente de gradient stochastique (SGD) [Schmidt *et al.*, 2017]. Enfin, puisque le problème d'optimisation (2.10) est strictement convexe, il existe au plus un minimum qui en cas d'existence sera global [Boyd *et al.*, 2004]. Ainsi les $\hat{\beta}$ sont optimaux, au sens qu'ils sont estimés par une SGD convergente. Afin de garantir l'existence d'un minimum global unique, il est possible d'ajouter une pénalisation ℓ_2 au problème d'optimisation

(détaillée à la Section 3.3.3).

2.2.2 La Régression Logistique à Splines Naturelles Cubiques

Comme illustré sur la Figure 2.1, modéliser des effets non linéaires ajoute de la complexité dans les règles de décision mais peut entraîner un gain de performance non négligeable dans de nombreux domaines d'application réelle, notamment en médecine de précision. En effet, la modélisation de la LR (2.4), avec seulement des effets linéaires n'est pas suffisamment performante lorsque les données ne sont pas linéairement séparables. La LR non linéaire est très intéressante dans ce contexte car il est possible d'intégrer des effets non linéaires tout en conservant une grande interprétabilité de la règle de décision estimée. Sur la Figure 2.5-gauche, nous retrouvons la règle de décision estimée par la LR linéaire δ^{LR} (2.4). De part la contrainte de linéarité, la LR ne permet pas d'attribuer la bonne étiquette aux échantillons issus de la classe 0 ayant une valeur de X élevée. En intégrant des effets de seuils, comme illustrés sur la Figure 2.5-droite, la règle de décision issue de cette modélisation classifie correctement ces échantillons.

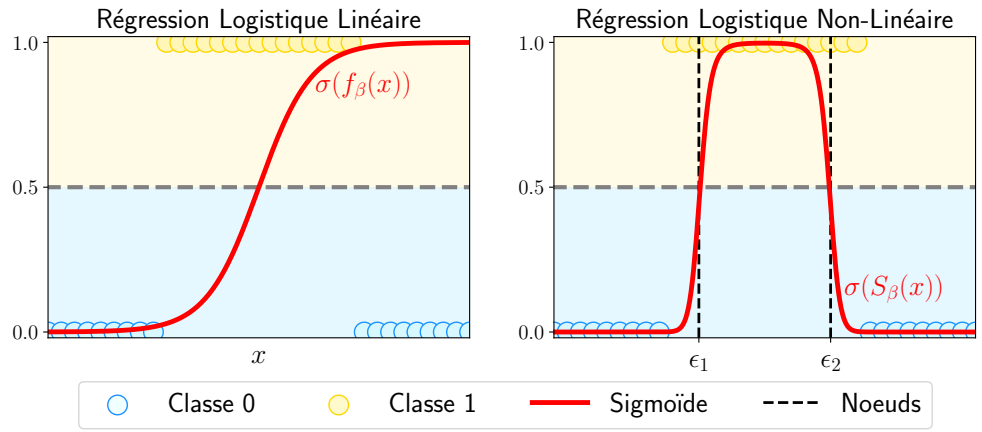

FIGURE 2.5 – Illustration d'une règle de décision issue d'une Régression Logistique linéaire (gauche) et d'une Régression Logistique non linéaire (droite). Pour la Régression Logistique linéaire (2.4), la sigmoïde est appliquée à f_β la combinaison linéaire des variables (2.5), où $f_\beta(x) = \beta_0 + \beta_1 \times x$. La Régression non linéaire (2.12) applique la sigmoïde à S_β une fonction de *mapping* non linéaire des variables (2.13), ici dans ce cas $S_\beta(x) = \beta_0 + \beta_1 \times x^2$.

La Régression Logistique non linéaire applique la sigmoïde aux variables préalablement transformées. Pour ce faire, $f_\beta(x)$ dans (2.4) est remplacée par une fonction de *mapping* non linéaire des variables descriptives que l'on note $S : \mathbb{R}^d \to \mathbb{R}$. Sur la Figure 2.6 nous pouvons retrouver quatre exemples de modélisations non linéaires couramment employées [Hastie *et al.*, 2009]. Tout d'abord il existe des splines constantes par morceau (Figure 2.6 - haut gauche) réalisant un partitionnement par morceau et modélisant des effets constants sur chacun des intervalles créés. Les splines linéaires par morceau (Figure 2.6 - haut droit) quant à elles modélisent des effets linéaires non constants sur chaque partition. Il est possible d'ajouter une contrainte de continuité entre ces segments, comme illustré avec la spline linéaire et continue par morceau (Figure 2.6 - bas gauche). Finalement, les Splines Naturelles Cubiques (NCS) (Figure 2.6 - bas droit) segmentent l'espace d'entrée et modélisent sur chacun des intervalles des comportements polynomiaux.

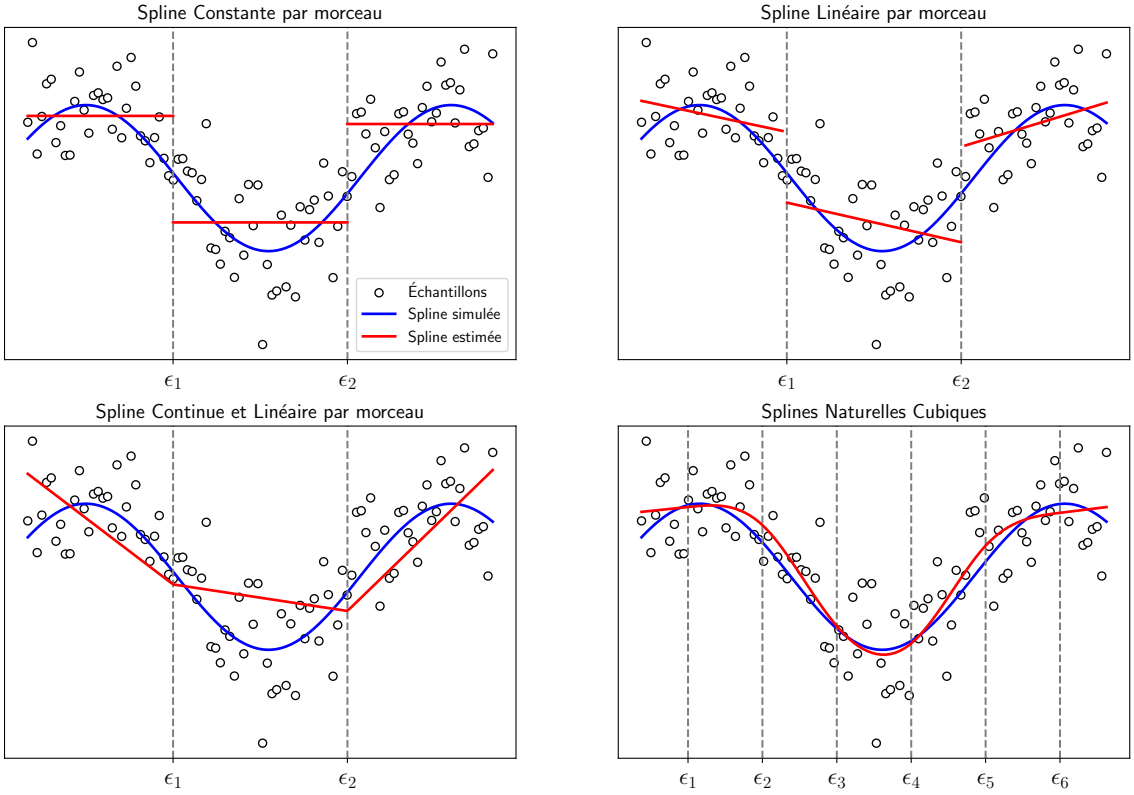

FIGURE 2.6 – Illustrations de différentes splines univariées : les splines constantes par morceau (haut gauche), les linéaires par morceau (haut droit), les linéaires et continues par morceau (bas - gauche) et enfin les splines naturelles cubiques (bas droit). À partir des échantillons (noirs), nous avons appris différentes approximations (en rouge) de la spline simulée (en bleu).

Nous allons principalement nous intéresser à la Régression Logistique à Splines Naturelles Cubiques (LR NCS) [Hastie *et al.*, 2009]. Cette modélisation revient à appliquer la sigmoïde à une somme de Splines Naturelles Cubiques (NCS) univariées :

$$\delta^{\text{LR NCS}}(x, \beta) = \sigma(S_\beta(x)), \tag{2.12}$$

avec $\beta \in \mathbb{R}^{d \times K-1}$ et S_β la somme de NCS univariées s_K à $K \geq 3$ degrés de définition définie par :

$$\begin{aligned} S_\beta : \ & \mathbb{R}^d \to \mathbb{R} \\ & x \mapsto \sum_{i=1}^{d} s_K(x_i, \beta_i), \end{aligned} \tag{2.13}$$

tel que

$$s_K(x_i, \beta_i) = \sum_{j=1}^{K-1} \beta_i N_j(x_i), \tag{2.14}$$

avec $\beta_i \in \mathbb{R}^{K-1}$, $i \in \{1, \ldots, d\}$ les coefficients à estimer par variable descriptive et N_j, $j \in \{1, \ldots, K-1\}$ les bases de splines. Chaque variable est transformée comme étant la

combinaison linéaire de bases de NCS (2.14). Soit $\bar{x} = x_i$ une variable fixée. Les NCS introduisent deux grands types de non-linéarité pour chaque variable descriptive. Premièrement, des effets de seuils sont créés. Les NCS à K degrés de définition segmentent en $K+1$ intervalles l'espace d'entrée en faisant intervenir K bornes d'intervalle que l'on appelle noeuds et que l'on note $\epsilon = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_K\}$. Deuxièmement, sur chaque intervalle créé, une fonction non linéaire à base de polynômes est modélisée. Les NCS à K degrés de définition font intervenir $K-1$ bases de splines $N(\bar{x}) = [N_1(\bar{x}), \ldots, N_{K-1}(\bar{x})]$, avec $N_1(\bar{x}) = \bar{x}$. Pour $N_{k+1}(\bar{x})$, avec $k \in \{1, \ldots, K-2\}$ des bases réduites sont créées avec des contraintes telles qu'aux bornes des intervalles nous modélisons des effets linéaires afin de ne pas partir dans les extrêmes, contrairement aux polynômes traditionnels. Finalement, les bases de splines $N(\bar{x}) = [N_1(\bar{x}), \ldots, N_{K-1}(\bar{x})]$ dans (2.14) sont définies par :

$$N_1(\bar{x}) = \bar{x}, \quad N_{k+1}(\bar{x}) = d_k(\bar{x}) - d_{K-1}(\bar{x}), \text{ pour } k \in \{1, \ldots, K-2\}, \tag{2.15}$$

avec

$$d_k(\bar{x}) = \frac{[\bar{x} - \epsilon_k]_+^3 - [\bar{x} - \epsilon_{K-1}]_+^3}{\epsilon_k - \epsilon_{K-1}}, \tag{2.16}$$

et

$$[x]_+ = \begin{cases} x & \text{si } x \geq 0 \\ 0 & \text{sinon.} \end{cases} \tag{2.17}$$

L'interprétabilité de la règle de décision $\delta^{\text{LR NCS}}$ (2.12) est donc conservée. En effet, pour chaque intervalle créé un coefficient est estimé, et les NCS univariées peuvent-être visualisées (un exemple est disponible à la Figure 2.6 - bas droite). Tout comme pour la LR linéaire, le problème d'optimisation revient à minimiser la BCE (2.9) appliquée à cette règle de décision :

$$\hat{\beta}^{\text{LR NCS}} = \underset{\beta \in \mathbb{R}^{d \times K-1}}{\arg\min}\, \mathcal{L}^{\text{LR NCS}}(\beta, \mathcal{D}), \tag{2.18}$$

avec

$$\mathcal{L}^{\text{LR NCS}}(\beta, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} L\left(\delta^{\text{LR NCS}}(\beta, x^{(i)}), y^{(i)}\right). \tag{2.19}$$

Cette modélisation ne permet cependant pas d'optimiser à la fois les noeuds ϵ et les coefficients β composant la règle de décision $\delta^{\text{LR NCS}}$ (2.12). La maximisation standard de la vraisemblance ne peut pas être utilisée lorsque les noeuds doivent être optimisés. En effet, la log-vraisemblance du problème devient seulement différentiable par morceau, les conditions de régularité ne sont alors pas réunies [Hawkins, 1972, Tishler et Zang, 1981a, Tishler et Zang, 1981b, Muggeo, 2003]. Afin d'utiliser l'estimateur de Maximum de Vraisemblance, [Hawkins, 1972] propose d'utiliser une approximation linéaire. D'autres propositions de linéarisation ont ensuite été formulées dans [Tishler et Zang, 1981a, Tishler et Zang, 1981b] qui proposent de lisser les opérateurs min et max à l'aide d'une approximation différentielle. Enfin [Hawkins, 1972] emploie les expansions de Taylor d'ordre 1 dans ce but. Ainsi ces méthodes proposent d'estimer conjointement les noeuds et les coefficients respectifs de chaque segment créé en minimisant la BCE (2.19) appliquée à la somme de NCS univariées (2.13) linéarisées. La fonction de coût minimisée perd néanmoins sa propriété de stricte convexité. En effet, le problème d'optimisation qui en résulte n'est plus convexe et n'offre ainsi aucune garantie de convergence vers un minimum global. Il se peut que la descente de gradient utilisée pour optimiser (2.18) soit bloquée en un minimum local. Pour palier à ce problème et garantir l'unicité

des estimateurs par Maximum de Vraisemblance, il est possible de fixer a-priori les noeuds. Pour ce faire, des méthodes computationnellement coûteuses de *gridsearch* peuvent être utilisées. [Ulm, 1991] propose de minimiser le problème d'optimisation (2.18) avec différentes valeurs de noeuds et garder celles qui maximisent le plus la vraisemblance. Dans [Stasinopoulos et Rigby, 1992, Küchenhoff, 1996] des algorithmes sont proposés pour ce faire. Lorsque plusieurs variables sont segmentées à plusieurs reprises, la combinatoire de ces algorithmes devient trop grande, ils deviennent alors difficilement employables. Si les experts du domaine d'application ne sont pas en mesure de fournir les points de changement, une solution raisonnable est de les fixer selon les quantiles uniformes des variables descriptives (voir exemple 5.2.2 dans [Hastie *et al.*, 2009]). Une fois le nombre de degrés de définition K des NCS arbitrairement choisi, il suffit de calculer les K quantiles uniformes de chaque variable pour obtenir les différents segments.

2.3 Méthodes interprétables : MARS et GAMs

Dans cette section, nous allons nous intéresser plus particulièrement aux classifieurs non linéaires interprétables utilisant un algorithme d'entraînement glouton. Dans la suite du manuscrit, nous comparerons nos contributions à certaines de ces méthodes, notamment les MARS et les GAM.

2.3.1 Régression Multivariée par Splines Additives (MARS)

Les MARS pour Régression Multivariée par Splines Adaptatives ont été introduits dans [Friedman, 1991] pour répondre à des problèmes de régression. Leur modélisation non linéaire se définie comme la somme de $M > 0$ splines adaptatives d'ordre q :

$$\psi^{\text{MARS}}(x) = \beta_0 + \sum_{m=1}^{M} h_m(x), \tag{2.20}$$

avec

$$h_m(x) = \beta_m [s_m(x_{v(m)} - b_m)]_+^q, \tag{2.21}$$

tel que $[t]_+ = \max\{0, t\}$ est défini à l'équation (2.17). Chaque base de splines $h_m(x)$, $m = \{1, \ldots, M\}$ dépend uniquement d'une seule variable $x_{v(m)}$ avec $v(m) = \{1, \ldots, d\}$ l'indice de la variable traitée par la base de splines m. Si $v(m) \neq k$ pour tout $m \in \{1, \ldots, M\}$ alors la composante k de x ne sera jamais incluse dans le modèle. Le réel b_m est le noeud de la spline et l'entier $s_m \in \{-1, 1\}$ indique si l'effet non linéaire est créé à gauche ou à droite du noeud. Enfin β_m caractérise l'impact de ce segment sur la règle de décision. Plusieurs exemples de bases de splines univariées (2.21) introduites par cette modélisation sont illustrés à la Figure 2.7. Cette méthode segmente chaque variable indépendamment des autres et réalise un partitionnement de l'espace d'entrée à l'aide d'hyperplans, comme le feraient les DT. La règle de décision qui en résulte est alors facilement interprétable.

Afin d'entraîner ces modèles, un algorithme récursif glouton est nécessaire. Tout d'abord il convient de fixer à priori le degré maximal q et le nombre de splines M que l'on souhaite estimer. Il est possible d'optimiser ces hyperparamètres en réalisant un algorithme de *gridsearch*. Pour ce faire, le modèle est entraîné avec les différents couples d'hyperparamètres que l'on souhaite tester. Le couple obtenant la meilleure performance

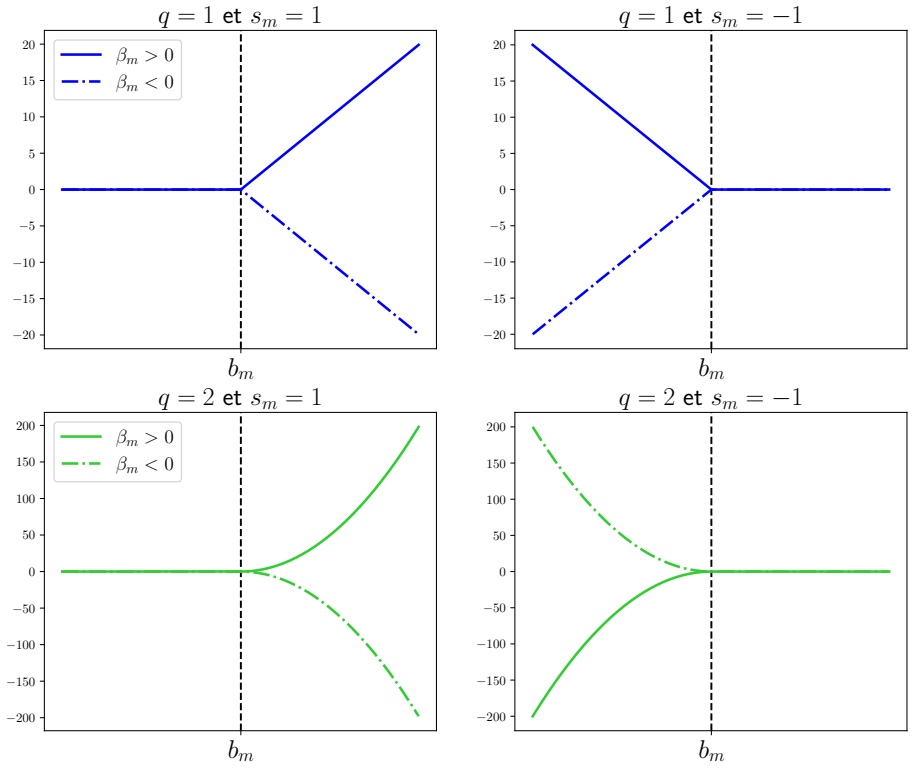

FIGURE 2.7 – Illustrations de splines univariées $h_m(x)$ définies à l'équation (2.21), selon les signes de s_m ($s_m = 1$ à gauche et $s_m = -1$ à droite) et β_m ($\beta_m > 0$ en trait continu et $\beta_m < 0$ en pointillés). Les splines d'ordre $q = 1$ sont en bleu et celles d'ordre $q = 2$ sont en vert.

prédictive sur l'échantillon de validation est conservé. À chaque itération, l'ensemble des paires de bases de splines possibles $\mathcal{C} = \{[x_j - t]_+^q, [t - x_j]_+^q\}$ avec $j \in \{1, \ldots, d\}$ et $t \in \{x_j^{(1)}, \ldots, x_j^{(N)}\}$ sont testées. Ainsi, chaque valeur prise par les variables descriptives constitue un potentiel noeud. Tant que le nombre de bases de splines M n'est pas atteint, la paire de bases de splines dont l'introduction dans le modèle réduit le plus l'erreur d'apprentissage est ajoutée (procédure *forward*). D'autre part, afin de limiter le sur-apprentissage, la base de splines dont la suppression du modèle entraîne la plus petite augmentation d'erreur est retirée à chaque itération (procédure *backward*). Cependant, la récursivité du modèle, illustrée à la Figure 2.8 rend incontrôlable la segmentation des variables. Il se peut qu'une même variable soit segmentée un grand nombre de fois. De plus, l'optimalité globale de cet algorithme d'optimisation glouton ne peut être établie. Ainsi, bien que ce modèle soit interprétable, nous ne disposons d'aucune garantie de convergence.

Bien que les MARS aient été initialement développés pour les tâches de régression, il est possible d'adapter cette méthode aux problèmes de classification. Nous pouvons considérer un classifieur MAP binaire (2.3) tel que $\psi^{\text{MARS}}(x)$ (2.20) modélise $\hat{\mathbb{P}}(Y = y|X = x)$. Une manière plus rigoureuse d'adapter les MARS à la classification est de considérer les GAM développés dans la prochaine sous-section.

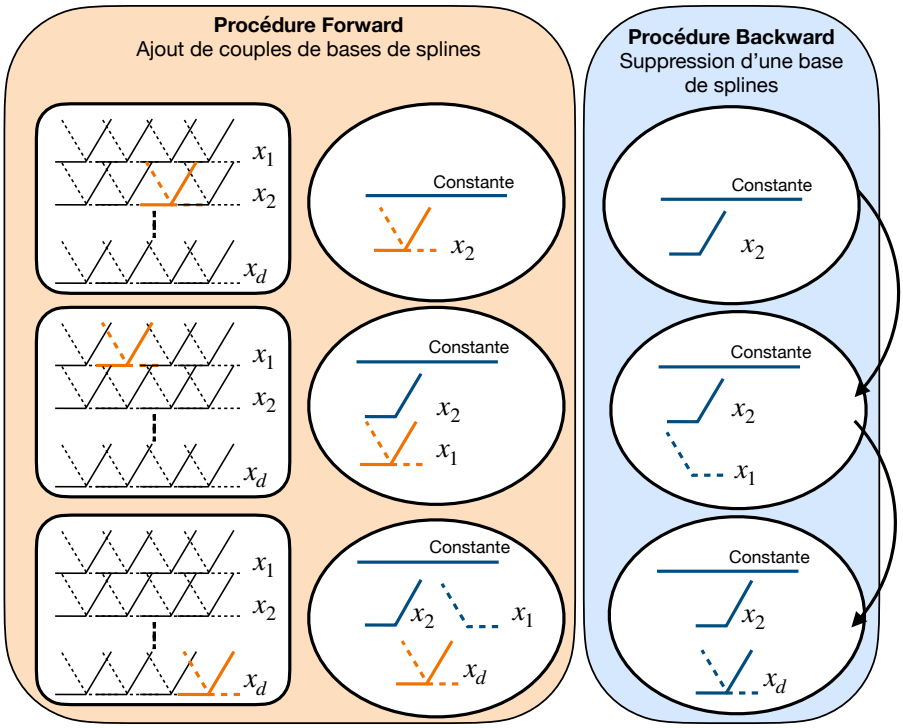

FIGURE 2.8 – Schéma illustratif de l'algorithme glouton utilisé pour entraîner le MARS (2.20) composé de $M = 3$ bases de splines. En orange on retrouve la procédure *forward* qui ajoute au modèle le couple de base de splines entraînant la plus grande diminution de l'erreur et en bleu la procédure *backward* enlevant la base de spline dégradant le moins la performance prédictive. Nous pouvons constater que la spline ajoutée lors de la 2e itération, jugée la plus discriminante à ce moment là, est finalement supprimée à l'itération 3. Il s'agit d'une des conséquences de l'algorithme glouton et de sa sous-optimalité : le partitionnement de l'espace d'entrée est incontrôlable.

2.3.2 Modèles Additifs Généralisés (GAMs) et ses extensions

Les Modèles Additifs Généralisés (GAM) sont des méthodes non linéaires couramment utilisées pour la classification. Cette méthode a été introduite dans une série d'articles [Hastie et Tibshirani, 1987a, Hastie et Tibshirani, 1987b, Stone, 1985] puis décrite dans [Hastie, 2017]. Les GAM pour la classification binaire sont des classifieurs MAP (2.3) modélisant la probabilité d'appartenir à la classe 1 à partir des caractéristiques données $\hat{\mathbb{P}}(Y = 1|X = x)$ par :

$$\delta^{\text{GAM}}(x, \beta) = \sigma(\beta_0 + f_1(x_1) + \dots f_d(x_d)), \tag{2.22}$$

avec σ la sigmoïde définie à l'équation (2.4) et des fonctions de splines univariées $f_i(x_i)$, pour $i \in \{1, \dots, d\}$. Ces fonctions peuvent très bien être modélisées par des Splines Naturelles Cubiques (2.15), des splines cubiques [Hastie, 2017], des MARS (2.20), des B-splines [De Boor, 1978], des *shape functions* [Lou *et al.*, 2012] ou des fonctions non linéaires continues par morceau. Cette modélisation (2.22) est couramment appelée Régression Logistique Additive Généralisée et est un cas spécifique des GAM pour la classification. Par la suite lorsque nous ferons référence aux GAM, nous considérerons ce modèle de classification. Afin d'entraîner cette méthode [Hastie, 2017] propose un algorithme glouton de *backfitting* dont l'optimalité globale ne peut être établie. En effet, à

chaque itération les splines $f_i(x_i)$ pour $i \in \{1, \ldots, d\}$ sont mises à jour par un algorithme local ("*Local Scoring*"), une variante du score de Fisher utilisée localement, combiné au Maximum de Vraisemblance.

De plus, il est possible d'étendre cette méthode afin de prendre en compte des effets d'intéraction entre les variables. Dans [Lou *et al.*, 2013] les GA^{2}Ms sont introduits. Il s'agit de GAM faisant intervenir des splines bidimensionnelles :

$$\delta^{\mathrm{GA^2M}}(x, \beta) = \sigma(\beta_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{i \neq j} f_{i,j}(x_i, x_j)). \tag{2.23}$$

Le principal défi dans la construction du modèle (2.23) réside dans le grand nombre de paires de variables à considérer. En effet, plus la base de données contient de variables, plus il y aura d'effets d'intéractions $f_{i,j}(x_i, x_j)$ pour $i \neq j$, $(i, j) \in \{1, \ldots, d\}^2$ tel que $i \neq j$ à intégrer. Pour cela, [Lou *et al.*, 2013] propose une méthode d'identification des intéractions importantes à inclure dans le modèle, appelée *FAST*. Cette méthode gloutonne est néanmoins très coûteuse computationnellement. Deux grandes étapes sont nécessaires pour entraîner les GA^{2}Ms. Tout d'abord il convient d'entraîner le GAM sans effet d'intéraction avec la procédure de *backfitting* introduite précédemment. Ensuite, nous considérons ce premier modèle fixé et nous entraînons les effets d'intéraction que la méthode *FAST* a identifié comme étant bénéfiques. Cette méthode utilise aussi un modèle glouton dont l'optimalité globale ne peut être établie. Il est possible de combiner ce processus d'apprentissage avec des méthodes de *boosting* [Schapire et Freund, 2012].

2.3.3 Machines Explicables Boostées

Dans [Lou *et al.*, 2012] les Machines Explicables Boostées (EBM) sont développées afin de combiner performance prédictive et interprétabilité. Ce modèle estime une règle de décision se modélisant comme celle des GAM ou des GA^{2}Ms. Chaque fonction non linéaire univariée $f_i(x_i)$, $i \in \{1, \ldots, d\}$ composant le GAM (2.22) est estimée par l'agrégation d'arbres de décision boostés.

Les arbres de décision (DT) segmentent les variables descriptives itérativement avec un processus glouton. À chaque itération, la variable dont la segmentation différencie au mieux les deux classes est seuillée, de sorte que les échantillons soient à nouveau divisés en deux catégories en fonction de leur valeur par rapport au seuil retenu. Au fur et à mesure des itérations, le quadrillage de l'espace d'entrée, opéré par des orthotopes, devient alors de plus en plus précis. Pour chaque feuille terminale, ou région résultante, la décision est prise par vote majoritaire : l'étiquette attribuée est la plus représentée au sein des échantillons compris dans cette région. Il est possible d'agréger les DT de sorte à obtenir une Forêt Aléatoire (RF) comme illustré par la Figure 2.9. Plutôt que d'apprendre un seul arbre, plusieurs arbres sont entraînés sur différents sous-échantillons de la population globale (*boostraping*). La règle de décision est toujours prise par vote majoritaire, mais cette fois elle dépend des décisions prises par chaque arbre composant la Forêt. Sur la Figure 2.9 nous illustrons ce principe avec trois DT composant la RF. Pour un échantillon donné, nous illustrons la règle de décision qui en découle. Puisqu'un arbre (bleu) prédit l'appartenance à la classe 2 et deux (jaune et vert) associent l'étiquette 1, la RF composée de ces arbres classifie l'échantillon comme appartenant à la classe 1.

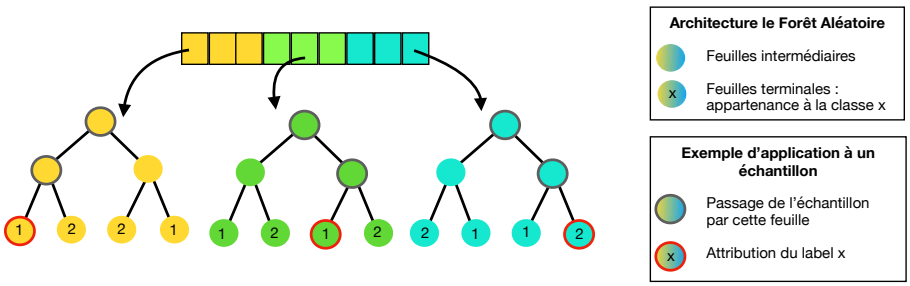

FIGURE 2.9 – Construction d'une Forêt Aléatoire par l'agrégation de plusieurs arbres de décision entraînés sur différents sous-échantillons. La règle de décision résultante se fait par vote majoritaire par rapport aux décisions de tous les arbres composant la forêt. Pour un échantillon donné, nous retrouvons le passage dans chaque feuille (ronds encadrés gris) et la décision qui en découle (ronds encadrés rouges).

Ainsi les EBM entraînent autant de RF qu'il y a de variables explicatives. À chaque itération et pour chaque variable successivement, un nouvel DT est ajouté à la RF comme illustré par la Figure 2.10. Bien que l'idée soit d'apprendre des fonctions univariées, la règle de décision prend en compte toutes les caractéristiques. La règle de décision est alors mise à jour à chaque ajout d'un nouvel arbre par variable et boostée de sorte que les échantillons mal classifiés se voient accorder un poids plus important et aussi que la segmentation suivante permette de les prédire correctement. Les auteurs préconisent l'utilisation d'un petit pas d'entraînement afin que l'ordre d'ajout des DT par variable n'ait pas d'importance : que nous traitons la variable X_1 puis ensuite la variable X_2 et non pas l'inverse n'a pas d'importance. Une fois les RF complètes et entraînées, il est possible de dessiner les splines univariées qui en résulte.

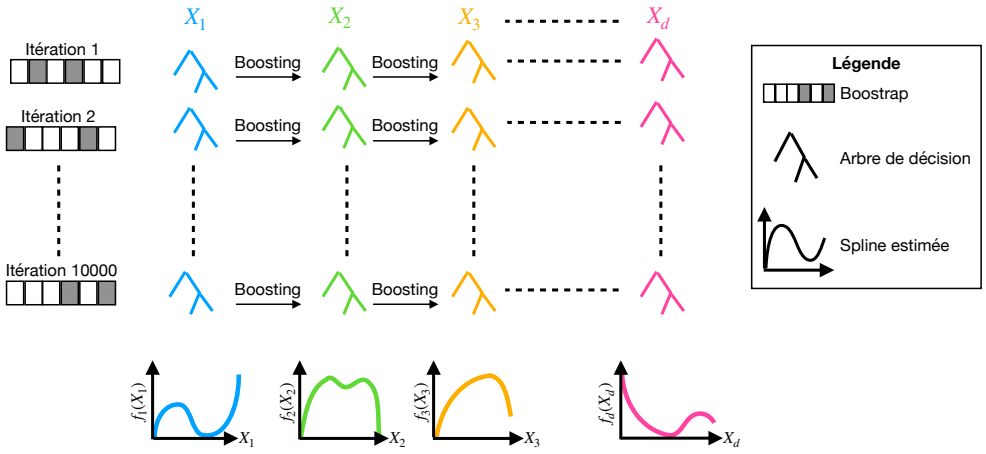

FIGURE 2.10 – Illustration du processus glouton d'entraînement des EBMs. À chaque itération et pour chaque variable successivement un nouvel arbre de décision est appris sur un sous-ensemble de la base de données. À chaque ajout d'arbres, la fonction de coût est mise à jour et boostée. Finalement, nous obtenons des splines univariées.

Tout comme pour les MARS et GAM présentés précédemment, cette méthode bien que performante et interprétable, utilise un processus d'optimisation séquentiel glouton dont l'optimalité globale ne peut être établie.

2.4 Boîtes Noires : Réseaux de Neurones ReLU

2.4.1 Modélisation et Apprentissage

Les Réseaux de Neurones (RN) [Goodfellow *et al.*, 2020] sont couramment utilisés tant leurs performances prédictives pour les tâches de régression et de classification sont significatives [Meijering *et al.*, 2022]. Pour les tâches de classification les RN composés de couches cachées d'activation ReLU (*Rectified Linear Unit*) sont le plus souvent utilisés. Un RN ReLU pour la classification (Figure 2.11) composé de $C > 1$ couches cachées et d'une couche de sortie sigmoïde se définit comme suivant :

$$\begin{aligned} \Phi^{\text{ReLU}}(x, \theta) : &\mathbb{R}^d \to \{0, 1\} \\ &x \mapsto \sigma \circ h^C \circ \cdots \circ h^1(x), \end{aligned} \tag{2.24}$$

telles que les couches cachées h^j, $j \in \{1, \ldots, C\}$ sont composées de n_j neurones auxquels est appliquée l'activation ReLU (2.17) :

$$\begin{aligned} h^1(x) &= \beta_{1,0} + \beta_{1,\cdot}^T [W_1 x + b_1]_+ \\ h^j(x) &= \beta_{j,0} + \beta_{j,\cdot}^T [W_j h^{j-1}(x) + b_j]_+ \quad \text{pour} \quad j \in \{2, \ldots, C\}. \end{aligned} \tag{2.25}$$

Chaque couche du RN se caractérise comme une combinaison linéaire de transformation ReLU. On note θ l'ensemble des paramètres estimés par le RN. Dans (2.25), pour $j \in \{2, \ldots, C\}$, $b_j \in \mathbb{R}^{n_j}$ sont les biais, $\beta_{j,0} \in \mathbb{R}$ et $\beta_{j,\cdot} \in \mathbb{R}^{n_j}$ sont les coefficients, telle que $\beta_{j,\cdot}^T$ est sa transposée. Enfin, $W_1 \in \mathbb{R}^{n_1 \times d}$ et $W_j \in \mathbb{R}^{n_j \times n_{j-1}}$ sont les matrices de poids. Dès la première couche, la matrice de poids W_1 mélangent les variables d'entrée entre elles. Dans les couches suivantes, les matrices de poids vont quant à elles mélanger toutes les transformations non linéaires opérées dans les précédentes. Interpréter la règle de décision qui en découle est alors impossible. Sur les Figures A.2 et A.3 nous pouvons constater que le partitionnement réalisé en régions obliques pour les RN ReLU à 1 et 2 couches est trop complexe pour être analysé. La couche de sortie du RN définie à l'équation (2.24) et illustrée en vert sur la Figure 2.11, applique quant à elle la sigmoïde (2.4) sur les données transformées par la dernière couche cachée :

$$\Phi^{\text{ReLU}}(x, \theta) = \sigma(h^C(x)). \tag{2.26}$$

Les RN sont donc composés d'un grand nombre de paramètres qu'il convient d'estimer. Contrairement aux méthodes non linéaires présentées précédemment qui utilisent des algorithmes itératifs gloutons ou nécessitent de spécifier à priori certaines caractéristiques de la règle de décision, les RN optimisent un critère global afin d'estimer conjointement tous les paramètres. Pour ce faire, la BCE est minimisée :

$$\hat{\theta}^{\text{RN ReLU}} = \arg\min_{\theta} \mathcal{L}^{\text{RN ReLU}}(\theta, \mathcal{D}), \tag{2.27}$$

avec,

$$\mathcal{L}^{\text{RN ReLU}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(\Phi^{\text{ReLU}}(x^{(i)}, \theta), y^{(i)}). \tag{2.28}$$

Afin de minimiser la fonction de coût (2.28), une SGD stochastique est le plus couramment

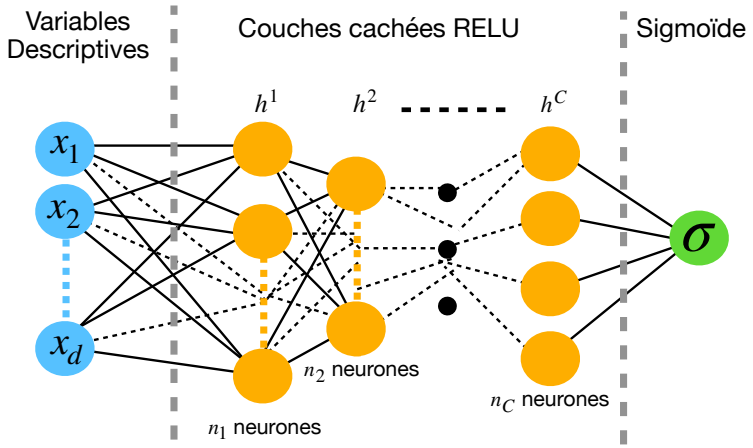

FIGURE 2.11 – Schéma illustratif d'un Réseau de Neurones ReLU pour la classification binaire (2.24). En bleu nous retrouvons les entrées du réseau, les variables descriptives. Les couches cachées (2.25) sont représentées en orange. La couche de sortie, appliquant la sigmoïde aux données transformées (2.26) est en vert.

employée [Kingma et Ba, 2017]. Néanmoins, l'apprentissage des RN est rendue instable de part l'architecture de cette méthode. Étant donné que l'activation ReLU (2.17) est non linéaire, le problème d'optimisation défini ci-dessus n'est pas convexe. Ainsi, il se peut que la SGD soit bloquée dans un minimum local et que l'algorithme d'optimisation n'ait pas convergé. Nous ne disposons donc pas de garanties quant à l'unicité des résultats. Pour différentes initialisations $\theta^{(0)}$ du RN, nous pouvons obtenir des estimations des paramètres $\hat{\theta}^{\text{RN ReLU}}$ très différentes. L'initialisation du RN va donc avoir un impact important sur la règle de décision estimée.

2.4.2 Interprétabilité

L'architecture complexe des RN rend l'interprétation à la fois de leurs prédictions mais aussi des transformations opérées sur les données très difficile. Ces dernières années de nombreux travaux ont été menés afin de permettre l'analyse des sorties des RN. Plusieurs méthodes *Post-Hoc* sont proposées dans la littérature [Dwivedi *et al.*, 2023, Ali *et al.*, 2023] dans ce but. Les techniques d'importance des caractéristiques (*Feature Importance Techniques*) [Fisher *et al.*, 2018] sont appliquées une fois le modèle entraîné afin de mettre en évidence l'impact d'une caractéristique sur la prédiction. L'Explication Locale Interprétable de Modèle-Agnostique (*Local Interpretable Model-agnostic Explanation*, LIME) [Ribeiro *et al.*, 2016] et son modèle dérivé Anchors [Ribeiro *et al.*, 2018] introduisent des perturbations dans les données d'entrée, en supprimant, changeant ou masquant des échantillons, et examinent comment les prédictions changent. La méthode *SHapley Additive exPlanations* (SHAP) [Lundberg et Lee, 2017], quant à elle s'appuie sur la théorie des jeux afin de fournir des informations sur l'impact de chaque variable sur l'estimation de la sortie d'un RN. Enfin, les graphiques de Dépendance Partielle (*Partial Dependence Plots*, PDP) [Friedman, 2001] et ses extensions, les Espérances Conditionnelles Individuelles (*Individual Conditional Expectations*, ICE) [Goldstein *et al.*, 2015] et les Effets Locaux Accumulés (*Accumulated Local Effects*, ALE) [Apley et Zhu, 2020], sont des méthodes de visualisation pour expliquer comment une seule caractéristique influence les prédictions. D'une part, ces méthodes sont coûteuses en termes de calcul [Zintgraf *et al.*, 2017], mais elles ne per-

mettent pas de fournir une compréhension globale de la dynamique du RN. Des travaux ont aussi été menés dans le but de comprendre les transformations intrinsèques opérées par les couches cachées des RN et en proposer une interprétation. Dans la prochaine sous section, les travaux de [Balestriero *et al.*, 2019] sont introduits et ont permis de démontrer théoriquement que les RN sont des approximateurs de splines.

2.4.3 Approximateurs de Splines

Les couches cachées des RN introduisent de la non-linéarité dans la modélisation. Il est établi dans [Hornik *et al.*, 1989, Leshno *et al.*, 1993] que les RN sont des approximateurs universels de fonctions. Des études ont été menées afin de quantifier le pouvoir d'approximation des RN ReLU [Yarotsky, 2017]. De plus, il a été démontré que les RN ReLU introduisent un partitionnement de l'espace d'entrée [Montufar *et al.*, 2014]. Plus le réseau est composé de couches ReLU, plus le partitionnement est précis (voir Figure 2 dans [Serra *et al.*, 2018]). Il est établi dans [Balestriero *et al.*, 2019] que le partitionnement d'une couche ReLU peut être visualisé par un Diagramme de Laguerre–Voronoi (*Power Diagram*). Il est proposé un processus de sous-division afin de visualiser l'évolution de l'espace d'entrée au fur et à mesure que l'on ajoute des couches ReLU à l'aide de Diagrammes de Laguerre–Voronoi. Plus généralement, les RN ReLU peuvent approximer des fonctions continues et linéaires par morceau (Figure 2.6 bas gauche). En effet, un pont rigoureux a récemment été construit entre les RN ReLU et l'approximation de fonctions multidimensionnelles par des splines [Daubechies *et al.*, 2022, Balestriero *et al.*, 2018]. Les auteurs dans [Balestriero *et al.*, 2018] prouvent notamment que des RN ReLU peuvent être interprétés comme des opérateurs de splines multivariées. Ces travaux s'appuient sur les résultats établis dans [Magnani et Boyd, 2009, Hannah et Dunson, 2013]. Il y est démontré que si les splines continues et linéraires par morceau sont contraintes à être globalement convexes alors elles peuvent se réécrire comme des *Max-Affine Splines* (MAS) :

$$s[W, b, \Omega](x) = \max_{r=1,\dots,R} W_r x + b_r, \tag{2.29}$$

avec Ω l'espace d'entrée partitionné en $R > 0$ régions tel que $\Omega = \cup_{r=1}^{R} \omega_r$. Respectivement W_r et b_r sont les $r^{\text{ème}}$ composantes des vecteurs $W \in \mathbb{R}^R$ et $b \in \mathbb{R}^R$. Comme illustré sur la Figure 2.12, l'espace d'entrée Ω est partitionné et sur chaque segment ω_r, $r \in \{1, \dots, R\}$, une fonction affine est créée. Lorsque la spline que l'on cherche à approximer (courbe bleue) est globalement convexe, on peut l'approximer en prenant le maximum de chaque fonction affine (courbes vertes) par morceau.

À partir de ces résultats, [Balestriero *et al.*, 2018] ont démontré que chaque couche cachée ReLU (2.25) du RN défini à l'équation (2.24) peut se réécrire comme des opérateurs max-affine spline (MASO). Chaque couche ReLU va ainsi réaliser un partitionnement de l'espace d'entrée et pouvoir approximer une spline affine continue par morceau multivariée. Pour la couche j du réseau de neurones composée de n_j neurones, il y aura alors n_j régions créées. Nous pouvons alors réécrire les couches ReLU comme des opérateurs MASO :

$$\begin{aligned} h^1(x) &= \max_{r=1,\dots,n_1} W_{1,r} x + b_{1,r} \\ h^j(x) &= \max_{r=1,\dots,n_j} W_{j,r} h^{j-1}(x) + b_{j,r}, \quad j = \{2, \dots, C\} \end{aligned} \tag{2.30}$$

avec respectivement $W_{j,r}$ et $b_{j,r}$ le poids et le biais du $r^{\text{ème}}$ neurone de la couche j.

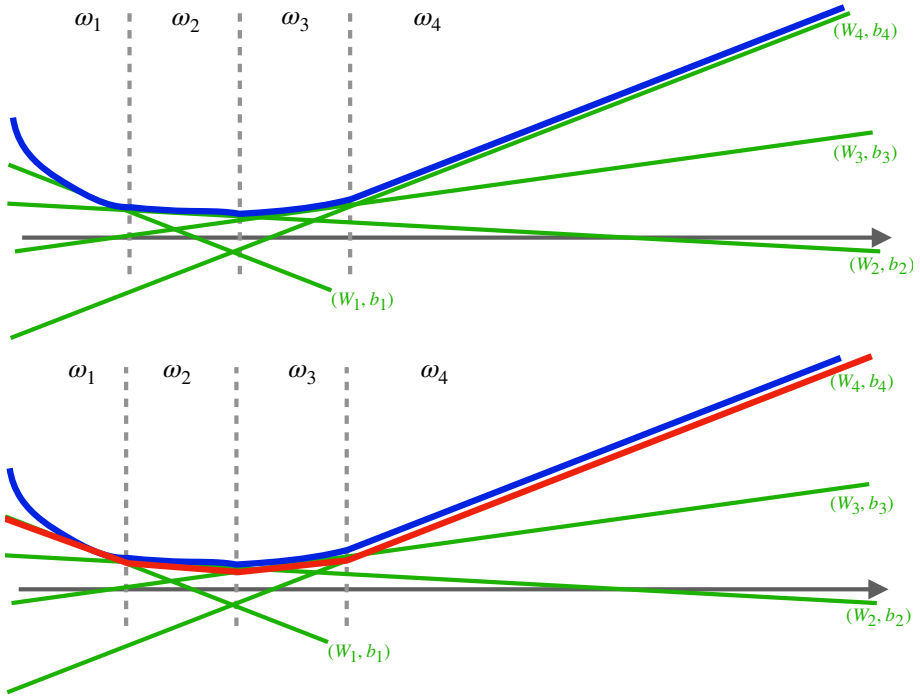

FIGURE 2.12 – Schéma illustratif de l'approximation d'une spline continue et linéraire par morceau univariée par une Max-Affine Splines (2.29). La segmentation de l'espace $\Omega = \{\omega_1, \ldots, \omega_4\}$ en 4 régions distinctes est représentée par les traits en pointillés gris. La spline continue et linéraire par morceau univariée que nous cherchons à approximer est en bleu. Enfin, sur la figure du dessous, nous retrouvons son approximation en rouge qui est définie comme étant le maximum des 4 fonctions affines en vert.

Ainsi, tous ces travaux démontrent d'une part que les RN ReLU pour la classification (2.24) peuvent alors être considérés comme une régression logistique non linéaire définie dans la sous-Section 2.2.2. D'autre part, ils permettent de déterminer à la fois les noeuds et les splines. Néanmoins, le partitionnement de l'espace d'entrée étant multivarié, l'interprétabilité de la règle de décision induite est difficile si ce n'est impossible. La Figure 2.13-gauche illustre la règle de décision (courbe rouge) résultant du partitionnement opéré (courbes noires) par un RN à une couche cachée composée de 10 neurones. La segmentation est caractérisée par des régions obliques, difficilement explicables. Lorsque nous ajoutons une deuxième couche à RN ReLU, l'interprétation devient encore plus compliquée. Sur la Figure 2.13-droite est affiché le partitionnement induit par un RN ReLU à deux couches composées chacune de 5 neurones. En rouge nous retrouvons la règle de décision finale, en noir le partitionnement opéré par la première couche et en vert celui induit par la seconde. Au plus nous ajoutons des couches, au plus il est difficile de fournir une interprétation de la segmentation de l'espace d'entrée.

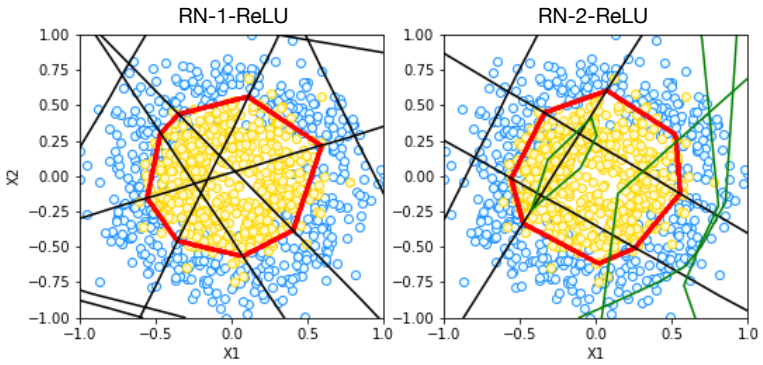

FIGURE 2.13 – Règles de décision (courbes rouges) résultant de deux réseaux de neurones ReLU visant à séparer les deux classes (échantillons jaunes et bleus). À gauche, le partitionnement (courbes noires) d'un RN à une couche composée de 10 neurones est affiché. À droite est illustrée la segmentation de l'espace d'entrée opérée par un RN à deux couches cachées composées chacune de 5 neurones, les courbes noires représentant le partitionnement de la première couche et celles en vertes celui de la seconde couche.

2.4.4 Modèles Additifs Neuronaux

Les Modèles Additifs Neuronaux (NAM), développés dans [Agarwal *et al.*, 2021] modélisent un GAM (2.22) par des RN ReLU. Chaque fonction spline univariée $f_i(x_i)$, $i \in \{1, \ldots, d\}$ composant la règle de décision est estimée par un sous-réseau de neurones comme illustré sur la Figure 2.14. Ainsi si nous disposons de d variables explicatives, d sous-réseaux seront estimés. Enfin un dernier réalise la combinaison linéaire de leurs sorties respectives pour finalement retrouver une somme additive de fonctions univariées.

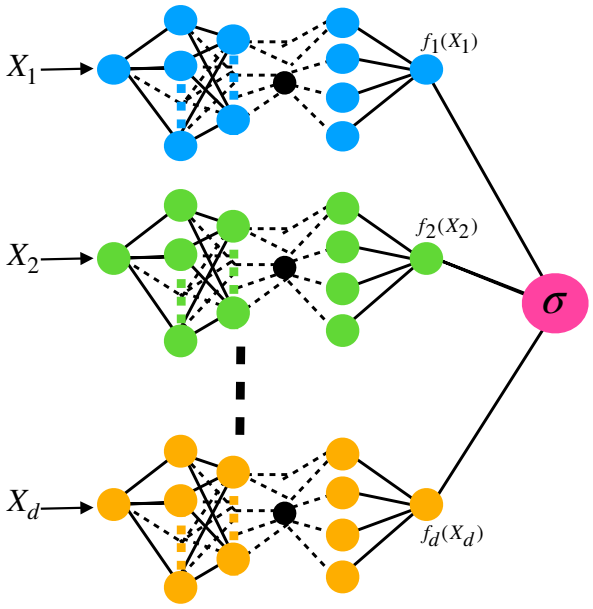

FIGURE 2.14 – Construction d'un Modèle Additif Neuronal composé de d variables explicatives. Chaque variable est transformée non linéairement par un sous réseau de neurones. En sortie la sigmoïde est appliquée à la combinaison linéaire des sorties des d réseaux.

La méthode permet aussi d'ajouter des effets d'interactions afin de modéliser la règle de décision d'un GA^{2}M (2.23). Plusieurs options peuvent être ajoutées lors de l'apprentissage afin de sélectionner les neurones importants pour chaque sous réseau, de sorte à ne conserver que les partitions univariées importantes. Néanmoins, pour pouvoir apprendre la règle de décision issue d'un NAM réalisant à la fois de la sélection de variables et de segmentations par variable de nombreux hyper-paramètres sont à définir et à optimiser. L'optimisation d'un tel modèle est d'une part complexe mais aussi comme pour tout réseau de neurones, ne fournit aucune garantie de convergence et d'unicité des résultats.

2.4.5 Le cas particulier du Réseau de Neurones ReLU pour la classification à une couche cachée

Par la suite, nous considérerons le cas particulier des RN ReLU pour la classification à une seule couche cachée composée de p neurones. Nous pouvons alors réécrire le modèle :

$$\Phi^{\mathrm{ReLU}(p)}(x, \theta) = \sigma(\psi^{\mathrm{ReLU}(p)}(x)), \tag{2.31}$$

telle que la fonction de score se définie par :

$$\psi^{\mathrm{ReLU}(p)}(x) = \beta_0 + \sum_{k=1}^{p} \beta_k \phi\left(W_k x + b_k\right), \tag{2.32}$$

avec $\beta = (\beta_0, \beta_1, \ldots, \beta_p) \in \mathbb{R}^{p+1}$, $b = (b_1, \ldots, b_p) \in \mathbb{R}^p$ et $W_1, \ldots, W_p \in \mathbb{R}^{1\times d}$ les paramètres à estimer. La fonction $\phi(\cdot) = \mathrm{ReLU}(\cdot)$ est l'activation ReLU (2.17).

2.5 Synthèse

Le tableau 2.1 illustre les avantages et les inconvénients des méthodes de l'état de l'art précédemment présentées. Comme illustré par la Figure 2.2, les méthodes se distinguent par leur degré d'interprétabilité et de performances mais aussi par leur méthode d'optimisation à savoir le temps nécessaire pour estimer leur règle de décision et leur convergence. Ces critères nous permettent de définir le caractère d'interprétabilité des méthodes considérées conformément aux définitions 1 et 2. Ainsi, trois grandes catégories de méthodes peuvent être distinguées.

— Tout d'abord nous avons les méthodes explicables mais peu performantes. La Régression Logistique linéaire (LR (2.4)) ne modélisant pas d'effets non linéaires n'est pas en mesure d'identifier des classes non linéairement séparables. La Régression Logistiques à Splines Naturelles Cubiques (LR NCS (2.12)) estime quant à elle des effets de seuils mais ne permet pas de les optimiser et perd donc en efficacité. Néanmoins, ces deux méthodes sont explicables car elles estiment des règles de décision uniques et interprétables mais aussi très rapides à entraîner.

— La deuxième catégorie de méthodes de Machine Learning identifiable se compose de la Régression Multivariée à Splines Additives (MARS (2.20)) et des Modèles Additifs Généralisés (GAM (2.22)). Ces méthodes sont plus performantes que celles citées précédemment car elles optimisent des effets non linéaires. Néanmoins, leur règle de décision est estimée par un algorithme glouton n'offrant aucune garantie de convergence. Elles ne remplissent alors pas les conditions d'explicabilité définies par la Définition 2 mais seulement celles qualifiant les méthodes d'interprétables.

— Enfin, la dernière catégorie se caractérise par une grande performance prédictive. Contrairement aux méthodes de la deuxième catégorie, les Réseaux de Neurones (RN (2.31)) modélisent des effets non linéaires et les estiment en minimisant un critère global. Néanmoins, cet algorithme est d'une part très long à entraîner et n'offre aucune garantie de convergence. D'autre part, la règle de décision qui en résulte est impossible à interpréter. Ces méthodes sont qualifiées alors de "Boîtes Noires".

Méthodes	**RL Linéaire** $\delta^{\mathbf{LR}}$ **(2.4)**	**RL NCS** $\delta^{\mathbf{LR\ NCS}}$ **(2.12)**	**MARS** $\psi^{\mathbf{MARS}}$ **(2.20)**	**GAM** $\delta^{\mathbf{GAM}}$ **(2.22)**	**RNs** $\Phi^{\mathbf{ReLU}}$ **(2.31)**
Performances Prédictives	-	- +	+	+ +	+ + +
Interprétabilité de la règle de décision	✔	✔	✔	✔	✖
Temps de Calcul	- - -	- -	- +	- +	+ + +
Convergence de l'algorithme	✔	✔	✖	✖	✖
Interprétabilité vs Explicabilité	Explicable	Explicable	Interprétable	Interprétable	Boîte Noire

TABLE 2.1 – Tableau comparatif des méthodes de classification non linéaires développées précédemment en fonction de leurs performances prédictives, de la possibilité de donner une interprétation à la règle de décision estimée, du temps de calcul nécessaire pour leur optimisation et la convergence de l'algorithme utilisé. La dernière ligne les différencie selon leur degré d'interprétabilité.

Toutes ces méthodes seront des outils de comparaison en terme de performances prédictives, d'interprétabilité et d'explicabilité pour la suite du manuscrit. Dans le chapitre suivant, nous allons introduire le modèle SATURNN, fortement inspiré des méthodes développées dans cet état de l'art.

Chapitre 3

SATURNN

Dans ce chapitre, nous introduisons nos premières contributions. Tout d'abord, en Section 3.1 nous présentons les motivations nous amenant à développer le Réseau de Neurones MARS (RN-MARS, Section 3.2). Ce modèle est fortement inspiré des méthodes présentées précédemment tout en s'affranchissant de leurs principales contraintes. Ensuite, la Section 3.3 introduit la généralisation du RN-MARS appelée SATURNN (*Splines Approximation Throught Understandable ReLU Neural Network*). Nous détaillons sa modélisation, son processus d'initialisation ainsi que sa méthode d'apprentissage. Enfin la Section 3.4 contient des expériences sur deux bases de données simulées. Nous comparons la performance ainsi que l'interprétabilité du RN-MARS et du SATURNN aux méthodes de l'état de l'art présentées précédemment.

Sommaire

3.1 Motivations

Comme détaillées dans le Chapitre 2 de nombreuses méthodes ont été développées pour les tâches de classification non linéaires. Néanmoins, elles comportent toutes des limites dont nous aimerions nous affranchir. La Régression Multivariée par Splines Additives (MARS, Section 2.3.1) et les Modèles Additifs Généralisés (GAM, Section 2.3.2) partitionnent l'espace d'entrée par des orthotopes, rendant leur règle de décision facilement interprétable. En revanche, leur algorithme glouton implique une segmentation des variables descriptives incontrôlable et sous-optimale. Les Réseaux de Neurones (RN) quant à eux optimisent un critère global afin d'entraîner conjointement l'ensemble des paramètres et donc la règle de décision dans sa globalité. Néanmoins, ils réalisent un partitionnement de l'espace d'entrée à l'aide de régions obliques, rendant difficile si ce n'est impossible l'interprétation de la règle de décision.

Nous souhaitons alors développer une nouvelle méthode de classification modélisant des effets non linéaires s'affranchissant des contraintes énoncées des méthodes de l'état de l'art, tout en s'inspirant fortement de leurs avantages. De récents travaux de recherche ont été menés afin de combiner l'architecture des méthodes non linéaires interprétables et l'optimisation des RN. Dans [Eckle et Schmidt-Hieber, 2019] les auteurs proposent une architecture de RN approximant les MARS. Ils démontrent que n'importe quelle fonction pouvant être exprimée sous la forme des MARS (2.20) peut être correctement approximée par un RN ReLU multi-couches (2.24). Tout ce travail est néanmoins entièrement théorique et aucun algorithme n'est proposé afin d'entraîner cette architecture de RN. Les Modèles Additifs Neuronaux (NAM) développés dans [Agarwal *et al.*, 2021] approximent quant à eux les GAM (2.22). Il s'agit d'une combinaison de $d+1$ RN (Figure 2.14). Les d premiers sous-réseaux composant les NAM ont pour but d'entraîner les différentes splines $f_i(x_i)$, $i \in \{1, \dots, d\}$ composant les GAM. Ensuite, un dernier réseau permet d'appliquer la sigmoïde à la combinaison linéaire des d sous-réseaux. Cette architecture est alors très complexe et il est difficile d'établir des garanties d'optimalité de convergence de leur processus d'optimisation. De plus, pour estimer la règle de décision issue des NAM, de nombreux hyper-paramètres sont à définir. Afin d'optimiser au mieux tous ces paramètres, un algorithme glouton de *gridsearch* est développé et proposé par les auteurs [1]. Cette méthode, de part son architecture et son nombre d'hyper-paramètres à optimiser, est très coûteuse computationnellement.

Nous proposons des RN dont la modélisation d'effets non linéaires est directement inspirée par les MARS. Le Réseau de Neurones MARS (RN-MARS) détaillé dans la Section 3.2 réalise un partitionnement de l'espace d'entrée telle que chaque variable descriptive est segmentée indépendamment des autres, comme le font les MARS et les GAM. Contrairement à ces méthodes gloutonnes, le partitionnement induit par le RN-MARS est contrôlé. Le nombre d'effets de seuils introduits par variable est limité à 2, en accord avec les recommandations faites par les experts du domaine médical. Le *Splines Approximation Throught Understandable ReLU Neural Network* (SATURNN) détaillé en Section 3.3 est une généralisation du RN-MARS. Ce réseau de neurones réalise aussi un partitionnement de l'espace d'entrée univarié ; chaque variable est segmentée indépendamment des autres. Sa règle de décision se modélise alors comme une somme additive de splines continues et linéaires par morceau. Le nombre de splines estimées par variable est cette fois moins contrôlé, bien

1. NAM : https://github.com/google-research/google-research/tree/master/neural_additive_models

qu'une contrainte sur l'architecture du réseau impose le même nombre de segmentation par variable. Contrairement aux travaux de [Eckle et Schmidt-Hieber, 2019], un algorithme est proposé pour les entraîner. De plus la globalité des règles de décision est estimée en optimisant un unique critère, ce qui les différencie des NAM [Agarwal *et al.*, 2021].

3.2 Modèle RN-MARS

Nous proposons un RN ReLU à une couche cachée pour la classification dont la fonction de score imite celle des MARS. Le RN-MARS estime alors une règle de décision inspirée de celles des GAM, se modélisant comme une somme additive de splines univariées :

$$\Phi^{\text{RN-MARS}}(x, \theta) = \sigma(\psi^{\text{RN-MARS}}(x, \theta)), \tag{3.1}$$

$$\psi^{\text{RN-MARS}}(x, \theta) = \beta_0 + \sum_{j=1}^{d} g_j(x_j). \tag{3.2}$$

Chaque variable x_j du vecteur $x \in \mathbb{R}^d$ est transformée par $g_j : \mathbb{R} \to \mathbb{R}$. Contrairement aux MARS et GAM, le RN-MARS contrôle le partitionnement de l'espace d'entrée. En effet, les experts du domaine bio-médical sont persuadés que des effets non linéaires interviennent dans la modélisation du problème de classification mais restent limités. Ils estiment que segmenter plus de deux fois la même variable clinique n'est biologiquement pas très pertinent. Ainsi, la fonction g_j dans (3.2) est composée par deux activations ReLU (2.17) de sorte à ce qu'au plus deux intervalles maximum par variable soient créés :

$$g_j(t) = \beta_{j1}[b_{j1} - t]_+ + \beta_{j2}[t - b_{j2}]_+, \quad t \in \mathbb{R}. \tag{3.3}$$

Cette fonction correspond à une paire de neurones. Le premier neurone est une spline non-nulle avant la valeur du noeud b_{j1} et le second neurone est une spline non-nulle après la valeur du noeud b_{j2}. De ce fait, la fonction g_j modélise des fonctions avec un profil composé de 3 segments linéaires comme illustré dans l'encadré gris sur la Figure 3.1.

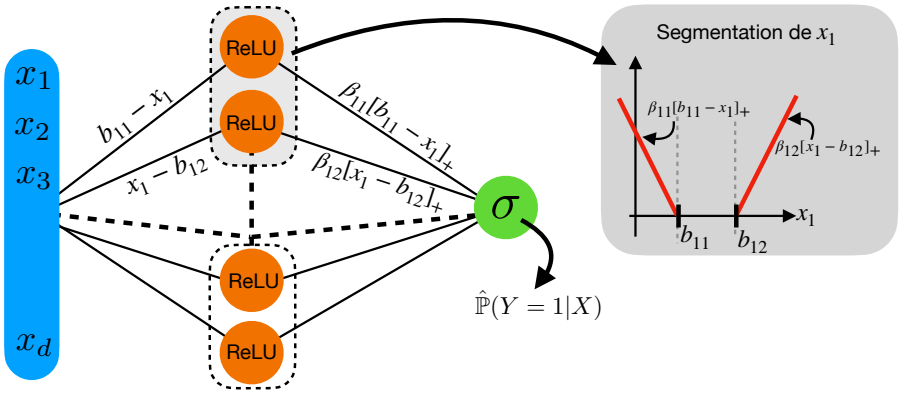

FIGURE 3.1 – Architecture du RN-MARS (3.1) : les variables descriptives en entrée du réseau sont en bleu, la couche cachée modélisant la fonction de score définie à l'équation (3.2) en orange et la couche de sortie appliquant la sigmoïde (2.4) est en vert. L'encadré en gris illustre le partitionnement résultant pour la variable x_1.

Sur cette figure, la variable X_1 peut par exemple représenter le poids d'un patient. Être en

sous-poids ($X_1 < b_{11}$) ou en sur-poids ($X_1 > b_{12}$) augmente la probabilité de développer la pathologie. En revanche, entre ces deux intervalles, l'impact du poids sur la maladie est négligeable.

Le RN-MARS réalise ainsi un partitionnement de l'espace d'entrée contrôlé. Rappelons que la matrice de poids W dans les RN ReLU traditionnels (2.32) mélange toutes les variables d'entrée entre elles. Afin de modéliser le partitionnement de l'espace d'entrée par des orthotopes, l'architecture du RN-MARS est en partie fixée :

$$\psi^{\text{RN-MARS}}(x, \theta) = \beta^T [\bar{W}x + b]_+, \tag{3.4}$$

avec $\theta = [\beta, b]$ l'ensemble des paramètres qu'il convient d'estimer. La couche cachée est composée de $2 * d$ neurones ainsi $\beta \in \mathbb{R}^{2d}$ et $b \in \mathbb{R}^{2d}$. Afin que chaque variable soit traitée indépendamment des autres par des couples de neurones (comme illustré par la Figure 3.1), la matrice de poids $\bar{W} \in \mathbb{R}^{2d \times d}$ est fixée de la manière suivante :

$$\bar{W} = \begin{pmatrix} -1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \tag{3.5}$$

La matrice de poids $\bar{W} = [W_{11}, W_{12}, \dots, W_{j1}, W_{j2}, \dots, W_{d1}, W_{d2}]$ attribue à chaque variable x_j, $j \in \{1, \dots, d\}$ deux vecteurs tel que seul l'indice j associé à la variable en question est non-nul. De ce fait, uniquement cette variable est traitée par le couple de neurones. Le premier neurone introduira un effet linéaire à gauche du seuil (car $W_{j1} = (0, \dots, 0, -1, 0, \dots, 0)$), tandis que le second le fera à droite ($W_{j2} = (0, \dots, 0, 1, 0, \dots, 0)$). Contrairement aux RN traditionnels, le RN-MARS réalise ainsi un partitionnement interprétable de l'espace d'entrée. Il s'appuie sur des hyperplans orthogonaux à la base canonique de l'espace $\mathbb{R}^d$, tout comme le font les arbres de décisions ou les MARS. Le découpage de l'espace $\mathbb{R}^d$ s'effectue alors avec des hypercubes et non des polyèdres aux formes complexes.

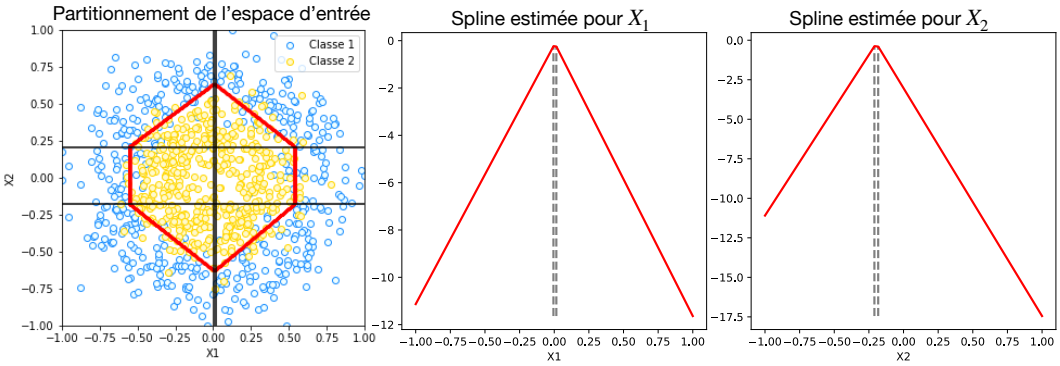

FIGURE 3.2 – Gauche - Règle de décision (courbe rouge) résultant du partitionnement (hyperplans noirs) d'un RN-MARS visant à séparer les deux classes (échantillons bleus et jaunes). Centre et Droite - splines estimées respectivement pour les variables X_1 et X_2.

La Figure 3.2-Gauche ci-dessus illustre la règle de décision (courbe rouge) résultant du partitionnement (hyperplans noirs) d'un RN-MARS. Le partitionnement est opéré par des orthotopes produisant des régions en hypercubes. La règle de décision obtenue est facilement interprétable puisque la fonction de score est linéaire sur chaque hypercube.

Ainsi, la fonction $\psi^{\text{RN-MARS}}(x)$ segmente les variables descriptives afin de faire intervenir des effets de seuils dans la règle de décision. Chaque composante de x va donc avoir un profil non linéaire spécifique comme illustré par la Figure 3.2-Centre et Droite pour respectivement les variables X_1 et X_2. En pratique, cela revient à effectuer une LR non linéaire comme définie à l'équation (2.12) ou un GAM (2.22). Contrairement à la LR appliquée à des Splines Naturelles Cubiques, les seuils déterminant le partitionnement des variables sont optimisés par le RN-MARS et ne doivent pas être définis *à priori*. Ainsi le RN-MARS estime à la fois les différents seuils b ainsi que l'impact de chacun d'entre eux β sur la règle de décision. L'apprentissage de l'ensemble des paramètres θ se fait avec une SGD utilisant l'entropie croisée binaire (2.9) comme fonction de perte. L'avantage du RN-MARS par rapport aux GAM réside alors dans son apprentissage. Le RN-MARS utilise un critère global pour entraîner la règle de décision, quand le GAM ajoute de manière itérative et donc sous-optimale des effets non linéaires. Le RN-MARS s'inspire ainsi des méthodes de classification non linéaires de l'état de l'art tout en s'affranchissant de leurs principales contraintes : (i) le RN est interprétable, (ii) l'apprentissage permet d'estimer à la fois les seuils délimitant les segments et les coefficients associés contrairement à la LR NCS, (iii) l'ensemble de la règle de décision est estimée par optimisation d'un critère global, par opposition aux MARS et GAM qui utilisent des méthodes itératives.

3.3 Le Modèle SATURNN

3.3.1 Modélisation

Dans cette section nous introduisons le SATURNN pour *Splines Approximation Throught Understandable ReLU Neural Network*. Il s'agit de la généralisation du modèle RN-MARS. Tout comme le modèle précédemment présenté, le SATURNN réalise un partitionnement de l'espace d'entrée à l'aide d'orthotopes. Cette fois le nombre de splines univariées ne dépend plus du nombre de variables descriptives. La fonction de score de ce RN ReLU pour la classification composé de p neurones se définie de la manière suivante :

$$\Phi^{\text{SATURNN}}(x, \theta) = \sigma(\psi(x, \theta)), \tag{3.6}$$

$$\psi(x, \theta) = \frac{1}{\sqrt{p}} \left[\beta_0 + \sum_{k=1}^{p} \beta_k \phi(s_k x_{\upsilon(k)} + b_k) \right], \tag{3.7}$$

où $\theta = [\beta^T, b^T]^T \in \mathbb{R}^{2p+1}$ est le vecteur des paramètres à estimer avec $\beta = [\beta_0, \beta_1, \ldots, \beta_p]$ et $b = [b_1, \ldots, b_p]$. $\psi(\cdot)$ représente l'activation ReLU définie à l'équation (2.17). Tout comme pour le RN-MARS, la matrice de poids W composant les RN ReLU (2.31) est dans la modélisation du SATURNN un paramètre fixé, directement inspiré des modèles MARS (Section 2.3.1). Chacun des p neurones composant la couche cachée $h_k(x) = \phi(s_k x_{\upsilon(k)} + b_k)$ traite une unique variable. Le k-ème neurone $h_k(x)$ traite $x_{\upsilon(k)}$ où $\upsilon : \{1, \ldots, p\} \to \{1, \ldots, d\}$ est un sélecteur indiquant quelle variable d'entrée est transformée par ce neurone. Puisque la fonction ReLU est non-décroissante, le signe $s_k \in \{-1, 1\}$ précise si $h_k(x)$ est

une fonction non-décroissante ou non-croissante de $x_{\upsilon(k)}$. Ainsi, la matrice de poids dans le RN ReLU à une couche (2.31) est considérée fixée telle que :

$$\bar{W}_k = s_k e_k, \quad \text{avec} \quad e_k = (0, \dots, 0, \underbrace{1}_{\text{indice } \upsilon(k)}, 0, \dots, 0). \tag{3.8}$$

Enfin, le biais b_k indique le noeud, c'est à dire le point à partir duquel $h_k(x)$ devient linéaire : $h_k(x) = s_k x_{\upsilon(k)} + b_k$ quand $s_k x_{\upsilon(k)} > -b_k$. Cette construction de la couche cachée réalise un partitionnement de l'espace d'entrée avec des orthotopes, comme illustré dans l'encadré gris de la Figure 3.3 , aboutissant à une règle de décision interprétable. Puisqu'un orthotope est le produit d'intervalles fermés, il est facile de savoir quelles variables sont impliquées dans cet orthotope et quels sont leurs domaines de définition.

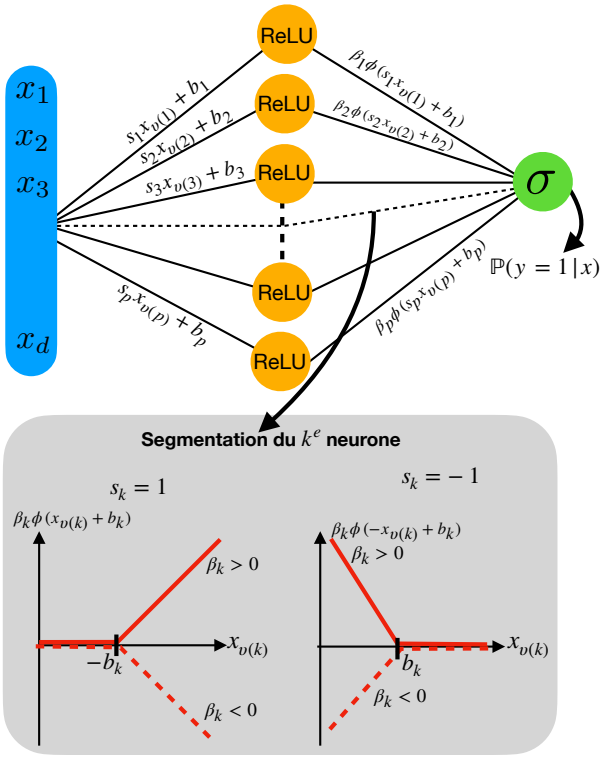

FIGURE 3.3 – Architecture du SATURNN (3.6) : les variables descriptives en entrée du réseau sont en bleu, la couche cachée modélisant la fonction de score (3.7) en orange et la couche de sortie appliquant la sigmoïde (2.4) est en vert. L'encadré en gris illustre le partitionnement résultant du neurone k.

Il est intéressant de constater que le SATURNN peut s'apparenter à un GAM (2.22). En réécrivant (3.7), nous obtenons :

$$\psi(x, \theta) = \frac{1}{\sqrt{p}} \left[\beta_0 + \sum_{i=1}^{d} f_i(x_i) \right], \tag{3.9}$$

avec $f_i(x_i)$ une fonction de spline univariée de x_i, composée de la somme des transformations non linéaires ReLU appliquées à chaque variable. Afin de visualiser les splines univariées estimées par le SATURNN, il convient pour la variable $i = \{1, \dots, d\}$ de som-

mer les transformations appliquées par les neurones k tel que $\upsilon(k) = i$, ce qui conduit à :

$$f_i(x_i) = \sum_{1 \leq k \leq p \,:\, \upsilon(k)=i} \beta_k \phi\left(s_k x_i + b_k\right). \tag{3.10}$$

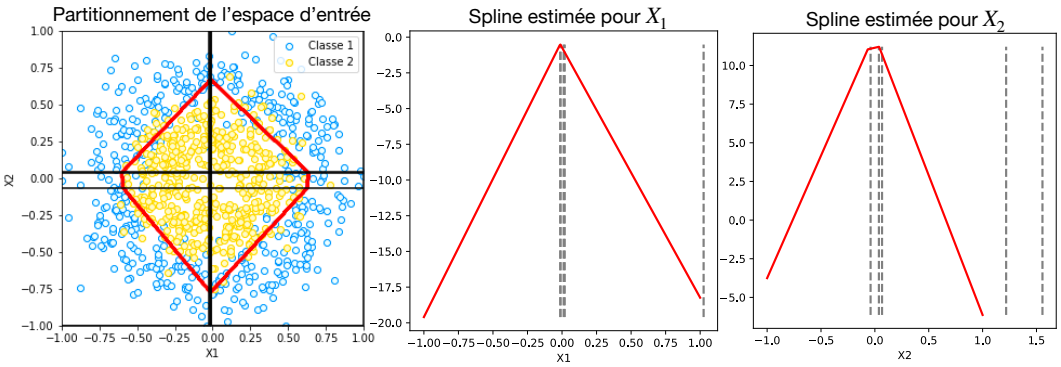

FIGURE 3.4 – Gauche - Règle de décision (courbe rouge) résultant du partitionnement (hyperplans noirs) d'un SATURNN composé de 10 neurones visant à séparer les deux classes (échantillons bleus et jaunes). Centre et Droite - splines estimées respectivement pour les variables X_1 et X_2.

Sur la Figure 3.4 - Gauche nous constatons que les régions résultant d'un SATURNN composé de 10 neurones sont des hypercubes facilement interprétables. De plus, il est possible de visualiser l'impact estimé par le SATURNN pour chaque variable descriptive en affichant les splines (3.4-Centre et Droite). Conformément à la Définition 1, le SATURNN est un modèle intrinsèquement interprétable. D'une part, nous pouvons retrouver toutes les transformations non linéaires opérées sur les variables et les interpréter. Ainsi, le SATURNN est un modèle blanc, ou "transparent" au sens que nous pouvons analyser le cheminement des entrées vers la sortie. De plus, la règle de décision étant composée d'une somme additive de splines univariées, nous pouvons facilement l'interpréter. Dans le cadre de l'application médicale pour la prédiction d'une pathologie par exemple, nous pouvons interpréter l'impact des différents effets de seuils des caractéristiques cliniques et biologiques sur le diagnostic.

3.3.2 Initialisations

Afin d'apprendre la règle de décision du SATURNN et donc d'estimer ses paramètres, il convient comme pour tous les RN d'initialiser l'ensemble de ses paramètres, y compris ceux qui nécessitent d'être appris et ceux qui sont considérés comme fixes. Dans un premier temps nous nous concentrons sur l'initialisation et la fixation de la matrice de poids $\bar{W}$ définie à l'équation (3.8). Pour ce faire, les sélecteurs de variables $\upsilon(k)$ dans (3.7) sont tirés aléatoirement selon une loi uniforme $\upsilon(k) \sim \mathcal{U}[\![1; d]\!]$, pour tout $k \in \{1, \ldots, p\}$. Puisque $\mathbb{E}(\upsilon(k)) = \frac{1}{d}$ toutes les variables descriptives sont sélectionnées en moyenne le même nombre de fois, soit p/d fois. Les $s_k = \{-1, 1\}$, pour tout $k \in \{1, \ldots, p\}$ sont quant à eux distribués aléatoirement selon une loi de Bernoulli de paramètre $1/2$, puis fixés. Ainsi, en moyenne autant d'effets non linéaires seront créés à droite et à gauche d'un seuil pour les variables.

Dans un second temps, nous initialisons les paramètres estimés par le SATURNN à savoir les biais b_k et les coefficients β_0, β_k, pour tout $k \in \{1, \ldots, p\}$. Les coefficients $\beta_0^{(0)}, \beta_k^{(0)}$

suivent une distribution normale centrée réduite tel que $\beta_0^{(0)}, \beta_k^{(0)} \sim \mathcal{N}(0,1)$,. Concernant l'initialisation des seuils et donc des biais b_k nous utilisons une loi uniforme. Nous supposons que les variables descriptives sont dans une boule ouverte $\mathcal{B}_2^d(0,r)$ telle que $\mathcal{B}_2^d(c,r) := \{x \in \mathbb{R}^d : \|x-c\|_2\}$ est une boule dans $\mathbb{R}^d$ centrée en $c \in \mathbb{R}^d$ et de rayon $r > 0$ avec $\|x\|_2^2 = \sum_{i=1}^d x_i^2$ la norme euclidienne de $x = (x_1, \ldots, x_d)$. Ainsi, les variables traitées par le SATURNN varient chacune dans l'intervalle de valeurs $[-r, r]$. Nous initialisons les biais b_k à partir d'une loi uniforme sur ce même intervalle : $b_k^{(0)} \sim \mathcal{U}[-r, r]$. De ce fait, les valeurs des seuils segmentant les variables se trouvent bien à l'initialisation dans l'intervalle de valeurs prises par les données. Ces hypothèses d'initialisation sont très importantes pour la suite du manuscrit et nécessitent ainsi d'être retenues :

Hypothèse 1 (Initialisations des paramètres du SATURNN)**.**
Soient N couples $\{x^{(i)}, y^{(i)}\}_{i=1}^N$ indépendants et identiquement distribués avec $x^{(i)} \in \mathcal{B}_2^d(0,r)$, $r > 0$, le vecteur des variables explicatives et $y^{(i)} = \{0,1\}$ l'étiquette binaire à prédire. Soit $\theta^{(0)} = [\beta_0^{(0)}, \beta_1^{(0)}, \ldots, \beta_p^{(0)}, b_1^{(0)}, \ldots, b_p^{(0)}]$ le vecteur des paramètres initialisés du SATURNN (3.6) tels que $\beta_0^{(0)}, \beta_k^{(0)} \sim \mathcal{N}(0,1)$ et $b_k^{(0)} \sim \mathcal{U}[-r, r]$, pour tout $k \in \{1, \ldots, p\}$.

3.3.3 Apprentissage

Afin d'estimer les paramètres $\theta = [\beta_0, \beta_1, \ldots, \beta_p, b_1, \ldots, b_p]$ composant le SATURNN, il convient comme pour tous les RN d'initialiser les paramètres selon l'Hypothèse 1. Contrairement aux GAM [Hastie, 2017] et aux méthodes non linéaires gloutonnes telles que les MARS [Friedman, 1991] ou DT [Breiman, 2017] dont s'inspire grandement le SATURNN, l'ensemble des paramètres est optimisé simultanément. De plus, son apprentissage nécessite d'entraîner un seul RN pour apprendre l'ensemble de la règle de décision tandis que les NAM [Agarwal *et al.*, 2021] nécessitent l'entraînement de $d+1$ RN :

$$\hat{\theta}^{\text{SATURNN}} = \underset{\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)}{\arg\min} \mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D})), \tag{3.11}$$

avec

$$\mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D})) = \frac{1}{N} \sum_{i=1}^{N} L\left(\sigma(\psi(x^{(i)}, \theta)), y^{(i)}\right). \tag{3.12}$$

Le problème d'optimisation défini à l'équation (3.11) permet d'apprendre l'ensemble des paramètres composant SATURNN. $\hat{\theta}^{\text{SATURNN}}$ est ainsi obtenu en minimisant la fonction de coût définie à l'équation (3.12), telle que $L(\cdot)$ désigne la fonction de perte, à savoir la BCE (2.9) [Goodfellow *et al.*, 2020].

Une régularisation ℓ_2 est ajoutée au problème d'optimisation du SATURNN (3.11). Cette contrainte, aussi appelée pénalisation de Ridge a été introduite dans [Hoerl et Kennard, 1970b] et [Hoerl et Kennard, 1970a] et vise à réduire le sur-apprentissage des modèles. Dans notre cas, l'introduction de cette pénalité a pour fonction principale de garantir que les paramètres estimés $\hat{\theta}^{\text{SATURNN}}$ ne soient pas trop éloignés de ceux initialisés. Plus précisément, l'apprentissage est contraint de sorte que la distance euclidienne entres les paramètres finaux et initiaux soit au plus égale à $R > 0$. Cette contrainte est très importante pour la suite des recherches menées et constituera une condition importante pour l'établissement des futurs résultats.

Hypothèse 2 (Contrainte sur les paramètres estimés)**.**
Soient $\hat{\theta}$ le vecteur des paramètres estimés du SATURNN en optimisant le problème défini à l'équation (3.11) et $\theta^{(0)}$ leurs valeurs initiales (Hypothèse 1). Les paramètres estimés par le SATURNN $\hat{\theta}$ ne sont pas trop éloignés de ceux initialisés, soit à une distance $R > 0$ près :

$$\hat{\theta} \in \mathbb{R}^{2p+1} : \|\hat{\theta} - \theta^{(0)}\|_2 \leq R. \tag{3.13}$$

Le problème d'optimisation du SATURNN (3.11) peut alors se réécrire :

$$\hat{\theta}^{\text{SATURNN}} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L\left(\sigma(\psi(x^{(i)}, \theta)), y^{(i)}\right) + \lambda \|\theta - \theta^{(0)}\|_2^2, \tag{3.14}$$

avec $\lambda > 0$ le paramètre de régularisation. Nous pouvons remarquer que lorsque $\lambda = 0$, cela revient à minimiser la fonction de coût sans pénalisation définie à l'équation (3.12). Plus le paramètre de régularisation λ est grand dans (3.14), plus la pénalisation est forte et ainsi la distance entre les paramètres estimés et leur valeur initiale diminue. La Figure 3.5 illustre la répartition des distances moyennes D_k (3.15) par paramètre en fonction de différents paramètres de régularisation λ.

$$D_k = \sum_{i \in \mathcal{F}_k} \frac{\|\hat{\theta} - \theta^{(0)}\|_2}{|\mathcal{F}_k|}, \tag{3.15}$$

avec $\mathcal{F}_k$ le k-ème fold et $|\mathcal{F}_k|$ sa taille. Nous pouvons constater que plus le paramètre de régularisation est grand, plus la répartition des distances se recentre en zéro. Ce résultat se retrouve sur la Figure 3.6 puisque la courbe orange représentant les distances moyennes décroît lorsque la valeur du paramètre de régularisation λ augmente. Ainsi, plus la régularisation est importante plus les paramètres estimés $\hat{\theta}^{\text{SATURNN}}$ restent proches de ceux initialisés $\theta^{(0)}$.

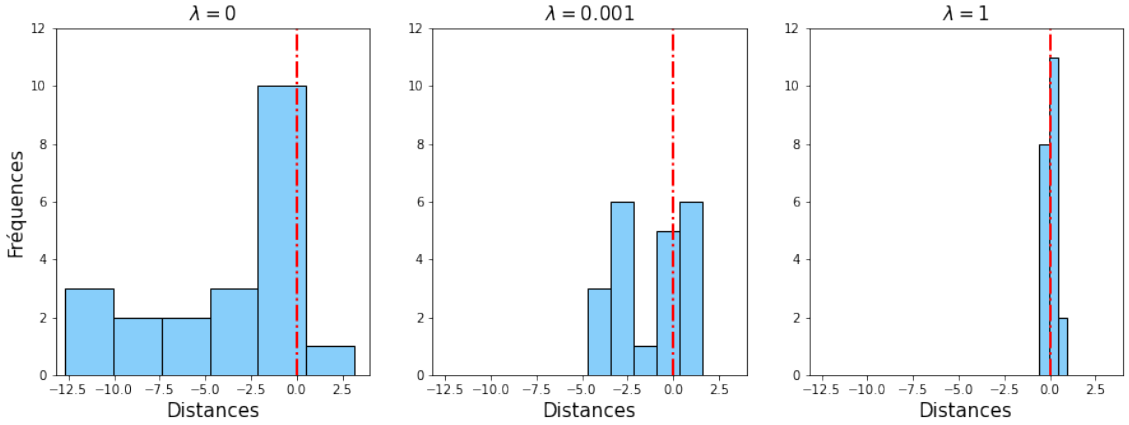

FIGURE 3.5 – Histogrammes des distances moyennes (3.15) obtenues pour chaque paramètre par Validation Croisée 5 folds, pour $\lambda = 0$ (gauche), $\lambda = 0.001$ (milieu) et $\lambda = 1$ (droite). Les résultats sont issus de l'entraînement du SATURNN composé de 10 neurones sur la base de données Circle (Figure 3.7-Gauche.

Puisque nous forçons les paramètres à rester proches des initialisations, l'ensemble image

est restreint. En d'autres termes, l'ensemble des valeurs parcourues et donc des règles de décision potentiellement estimées est limité. Ainsi, la règle de décision estimée avec pénalisation peut potentiellement être moins performante, comme nous pouvons le constater sur la Figure 3.6-Gauche. Entraîner le SATURNN avec $\lambda = 0.001$ augmente légèrement la performance moyenne obtenue par Validation Croisée $5-$folds par rapport à un modèle estimé sans pénalisation. Néanmoins, une trop grande pénalisation peut très vite détériorer les scores de classification. Lorsque nous considérons le problème d'optimisation (3.14) avec un paramètre de régularisation $\lambda = 1$, la performance prédictive est en moyenne de 50% sur l'échantillon de validation alors qu'elle était de 90% sans pénalisation. Ainsi, il y a un compromis important à prendre en compte lors de l'entraînement entre la distance des paramètres estimés et initialisés, et la performance prédictive. De plus, sur la Figure 3.6-Droite nous pouvons constater que la fonction de coût pour une pénalisation grande est plus instable (zone bleue).

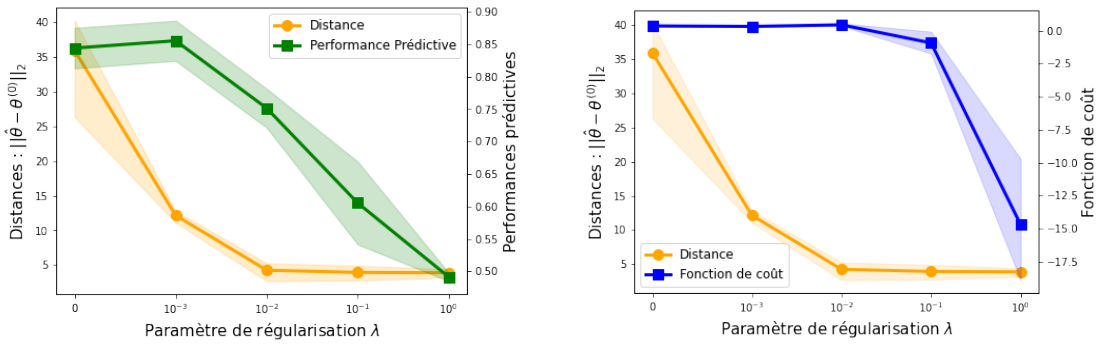

FIGURE 3.6 – Résultats des expériences de régularisation pour un SATURNN à 10 neurones entraîné par Validation Croisée 5 folds sur la base de données Circle (Figure 3.7-Gauche). Gauche : comparaison des distances moyennes (3.15) en orange et des performances prédictives moyennes sur l'échantillon de validation en vert pour différentes valeurs de λ. Droite : comparaison des distances moyennes (3.15) en orange et des fonctions de coût moyennes sur l'échantillon d'apprentissage en bleu pour différentes valeurs de λ.

Enfin, il est important de remarquer que l'application $\psi(x, \theta)$ composant la fonction de coût à minimiser pour estimer les paramètres du SATURNN est non linéaire à la fois en x et en θ. Ainsi, le problème d'optimisation défini à l'équation (3.11) n'est pas convexe. De ce fait, aucune garantie de convergence ne peut être établie. Il se peut tout à fait que l'algorithme de SGD soit bloqué dans un minimum local et que plusieurs minimums locaux existent. Ainsi pour des initialisations $\theta^{(0)}$ différentes, l'algorithme convergera potentiellement vers différents minimums locaux. Nous n'avons alors aucune garantie d'unicité des résultats.

3.4 Expériences numériques sur données simulées

Dans cette section, nous comparons les performances du RN-MARS et SATURNN aux méthodes de l'état de l'art sur deux bases de données simulées. Les codes pour pouvoir entraîner le RN-MARS [2] et le SATURNN [3] sont disponibles sur le Github . Nous avons décidé de considérer deux variables descriptives afin de pouvoir visualiser les règles de décision estimées ainsi que le partitionnement induit par les méthodes.

2. Marie Guyomard - Dépôt NN-MARS : https://github.com/GuyomardMarie/NN-MARS
3. Marie Guyomard - Dépôt SATURNN : https://github.com/GuyomardMarie/SATURNN

3.4.1 Bases de données simulées

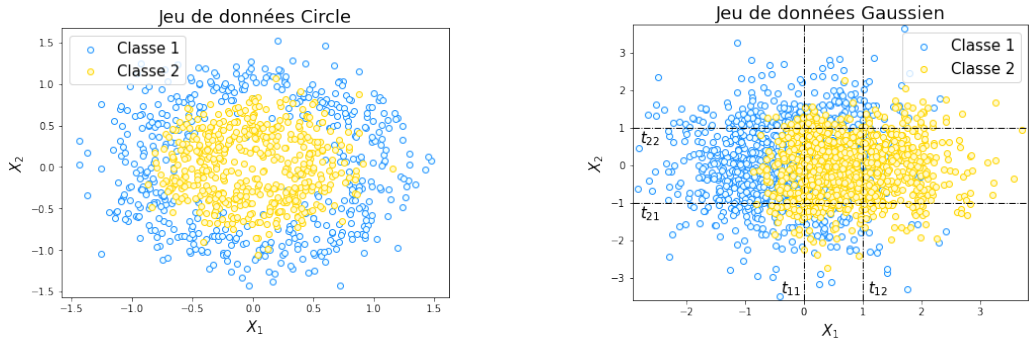

FIGURE 3.7 – Gauche - Jeu de données simulé Circle. En bleu nous retrouvons les échantillons de la Classe 1 et en jaune ceux de la Classe 2. Droite - Jeu de données simulé Gaussien. En bleu nous retrouvons les échantillons de la Classe 1 et en jaune ceux de la Classe 2. Les courbes noires représentent les valeurs des seuils (équations (3.17) et (3.18)).

Jeu de données Circle :
Nous générons deux cercles bruités[4] de données chacun composé de 500 échantillons. Le jeu de données qui en résulte, illustré par la Figure 3.7 - Gauche est très pertinent lorsque nous souhaitons comparer des méthodes de classification non linéaires.

Jeu de données Gaussien :
Nous simulons 2000 échantillons $x^{(i)} = (x_1^{(i)}, x_2^{(i)}) \in \mathbb{R}^2$ selon une loi normale. Puisqu'en réalité la règle de décision est bruitée, nous définissons les étiquettes $y^{(i)}$ à partir d'une distribution de Bernoulli :

$$y^{(i)} \sim \mathcal{B}(p(x^{(i)})) \text{ avec } p(x^{(i)}) = \sigma(f_1(x_1^{(i)}) + f_2(x_2^{(i)})). \tag{3.16}$$

La probabilité générant la distribution de Bernoulli est construite à partir de la sigmoïde appliquée à une fonction de score non linéaire. Les fonctions f_1 et f_2 sont définies de manière à obtenir différentes régions, affichées sur la Figure 3.7-Droite, grandement inspirées par l'application médicale. Nous supposons qu'avoir une valeur de X_1 inférieure à un certain seuil ($X_1 < t_{11}$) diminue la probabilité de développer une pathologie, tandis qu'un niveau élevé ($X_1 > t_{12}$) l'augmente. X_1 pourrait ainsi représenter le taux de cholestérol : avoir un niveau faible protège contre certaines maladies tandis qu'un taux élevé représente un risque. Nous avons alors pour un échantillon $i \in \{1, \dots, N\}$:

$$f_1(x_1^{(i)}) = \beta_{11} \times x_1^{(i)} \mathbb{1}_{\{x_1^{(i)} < t_{11}\}} + \beta_{12} \times x_1^{(i)} \mathbb{1}_{\{x_1^{(i)} > t_{12}\}}, \tag{3.17}$$

avec $\beta_{11} \in \mathbb{R}^-$ et $\beta_{12} \in \mathbb{R}^+$. Pour ce qui est de la variable X_2 nous supposons qu'un taux trop faible ($X_2 < t_{21}$) ou trop élevé ($X_2 > t_{22}$) augmente la probabilité de Bernouilli. Cette variable est inspirée par le poids : être en sous ou sur-poids est un facteur à risque pour la santé. Ainsi la fonction f_2 se définit par :

$$f_2(x_2^{(i)}) = \beta_{21} \times x_2^{(i)} \mathbb{1}_{\{x_2^{(i)} < t_{12}\}} + \beta_{22} \times x_2^{(i)} \mathbb{1}_{\{x_2^{(i)} > t_{22}\}}, \tag{3.18}$$

4. Méthode *make_ circles* disponible dans la librairie Python Scikit-Learn [Pedregosa *et al.*, 2011].

avec $(\beta_{21}, \beta_{22}) \in \mathbb{R}^{+2}$. Les différents seuils t_{11}, t_{12}, t_{21} et t_{22} peuvent être visualisés en gris sur la Figure 3.7-Droite.

Les méthodes comparées sont apprises sur les données normalisées. La normalisation des données est essentielle [Raschka et Mirjalili, 2019] car elle permet (i) de stabiliser les algorithmes d'apprentissage automatique en réduisant l'amplitude des valeurs des caractéristiques, (ii) leur convergence plus rapide et (iii) d'améliorer leurs performances prédictives en aidant les algorithmes à mieux se généraliser sur de nouvelles données, en particulier dans les modèles sensibles à l'échelle des caractéristiques tels que ceux optimisés par descente de gradient. Nous avons opté pour une normalisation des données $\tilde{X}$ entre $[-1, 1]$, ce choix est grandement motivé par des résultats théoriques établis dans la suite du manuscrit. Pour ce faire, nous appliquons aux données les transformations suivantes :

$$\tilde{X} = 2 * \left(\frac{X - \min(X)}{\max(X) - \min(X)} \right) - 1. \tag{3.19}$$

3.4.2 Méthodes comparées

Nous comparons la performance du RN-MARS (Section 3.2) et du SATURNN (Section 3.3) à de nombreuses méthodes de l'état de l'art pour la classification. Pour la Régression Logistique à Splines Naturelles Cubiques (LR NCS, Section 2.2.2) [Hastie *et al.*, 2009] nous avons utilisé 5 seuils par variable, fixés par quantiles uniformes.
Pour les méthodes gloutonnes, nous avons choisi d'utiliser un algorithme de *gridsearch* afin d'optimiser les hyper-paramètres des Forêts Aléatoires (RF) [Breiman, 2001] et des Machines Explicables Boostées (EBM) [Lou *et al.*, 2012]. Concernant les méthodes de Régression Multivariée par Splines Adaptatives (MARS, Section 2.3.1) [Friedman, 1991] et les Modèles Additifs Généralisés (GAM, Section 2.3.2) [Hastie, 2017] nous n'avons pas fixé le nombre de splines que les modèles doivent comporter. En effet, les packages *py-earth*[5] et *pygam*[6] proposent d'arrêter le processus itératif dès que l'ajout d'une nouvelle base de splines n'apporte plus de gain de performance. Afin de pouvoir comparer justement les performances prédictives de nos contributions avec les méthodes de l'état de l'art, les GAM et EBM ne prennent pas en compte d'effets d'intéraction entre les variables descriptives.
Nous avons aussi confronté nos contributions à divers Réseaux de Neurones (RN, Section 2.4) [LeCun *et al.*, 2015]. Nous avons entraîné des RN ReLU à une couche cachée composés d'autant de neurones que les SATURNN, soit $p = 10$ neurones. Nous comparons aussi les performances de nos contributions à des RN ReLU à 2 couches cachées composés pour les deux couches de 5 neurones. Pour ce qui est des Réseaux de Neurones Additifs (NAM) [Agarwal *et al.*, 2021], l'algorithme de *gridsearch* proposé par les auteurs[7] afin d'optimiser les hyper-paramètres étant trop long à entraîner, nous avons décidé de ne pas l'utiliser. Nous avons fixé à 5 le nombre de bases de splines apprises par variable. Ainsi, nous pouvons comparer équitablement les SATURNN et les NAM. Les RN, à savoir les RN-MARS, les SATURNN, les RN ReLU à 1 et 2 couches cachées, ainsi que les NAM sont entraînés sur 30000 itérations. Le pas de gradient a été choisi de sorte que la fonction de coût sur l'échantillon d'apprentissage décroisse et converge.

5. La méthode *Earth* développée dans le package *Py-earth* permet l'apprentissage des modèles MARS.
6. La méthode *LogisticGAM* dans le package *Pygam* permet l'apprentissage des GAMs.
7. https://github.com/google-research/google-research/tree/master/neural_additive_models

3.4.3 Métriques de Performance

Afin de comparer à la fois les performances moyennes ainsi que la stabilité des modèles, les résultats présentés dans les tableaux 3.1, 3.2 et 3.3 sont issus d'une Validation Croisée (CV) sur 5 folds (Figure 1.6). Nous utilisons deux métriques principales pour comparer les méthodes testées. La performance prédictive (*accuracy*) établit le ratio entre le nombre de prédictions correctes par rapport au nombre total de prédictions. Pour un ensemble d'échantillons $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, tel que $y^{(i)} = \{0, 1\}$, la performance globale d'un problème de classification binaire est calculée de la manière suivante :

$$\text{Performance Globale } = \sum_{i=1}^{N} \sum_{k=0}^{1} \frac{\mathbb{1}_{\{\hat{y}^{(i)}=k,\, y^{(i)}=k\}}}{N}. \tag{3.20}$$

Il est possible de la calculer sur un sous-ensemble, nous pouvons par exemple l'examiner par classe afin de vérifier qu'une classe ne soit pas discriminée par la règle de décision. Nous pouvons définir l'*accuracy* conditionnelle comme suivant :

$$\text{Performance Classe } k = \sum_{i=1}^{N} \frac{\mathbb{1}_{\{\hat{y}^{(i)}=k,\, y^{(i)}=k\}}}{n_k}, \tag{3.21}$$

avec $n_k = \sum_{i=1}^{N} \mathbb{1}_{\{y^{(i)}=k\}}$. Pour les tâches de classification binaire appliquées au domaine médical, la Sensibilité désigne la performance prédictive de la classe d'intérêt ($y = 1$), les patients malades ou répondant au traitement par exemple, tandis que la Spécificité correspond à la performance de l'autre classe ($y = 0$). L'*Area Under the Curve* (AUC) [Saporta, 2006, Murphy, 2012] est très employée en médecine de précision [Delacour *et al.*, 2005]. Cette métrique calcule l'aire sous la courbe ROC. Sur la Figure 3.8 diverses courbes ROC sont dessinées. En abscisse nous retrouvons le taux d'erreur associé à la Classe 0 (1−Spécificité) et en ordonnée le taux de performance de la classe d'intérêt.

Afin de maximiser à la fois la Spécificité et la Spécificité du modèle, nous pouvons jouer sur le seuil à partir duquel les patients seront classifiés comme appartenant à la classe 1. La règle de décision MAP binaire définie à l'équation (2.3) peut se réécrire :

$$\hat{y} = \begin{cases} 1 & \text{si } \hat{\mathbb{P}}(Y = 1 | X = x) \geq \tau \\ 0 & \text{sinon,} \end{cases} \tag{3.22}$$

avec τ le seuil que nous pouvons faire varier. Afin de favoriser la Sensibilité du modèle, un petit τ est recommandé afin que les patients soient plus facilement classifiés comme appartenant à la classe 1. Néanmoins, un compromis est nécessaire car favoriser la Sensibilité d'un modèle entraîne une détérioration de la Spécificité. Pour différentes valeurs de τ testées, nous appliquons la règle de décision et traçons la courbe reliant les différentes performances obtenues. Puisque nous aimerions maximiser à la fois la Sensibilité et la Spécificité du modèle, nous cherchons une courbe ROC qui soit le plus proche du point $(0, 1)$. Puisque la courbe ROC est nécessairement croissante, l'aire sous la courbe (AUC) donne ainsi une information quand à la performance prédictive globale du modèle mais aussi conditionnelle à chaque classe. Sur la Figure 3.8, le modèle associé à la courbe ROC bleue a donc une AUC supérieure à ceux associés aux courbes vertes et oranges.

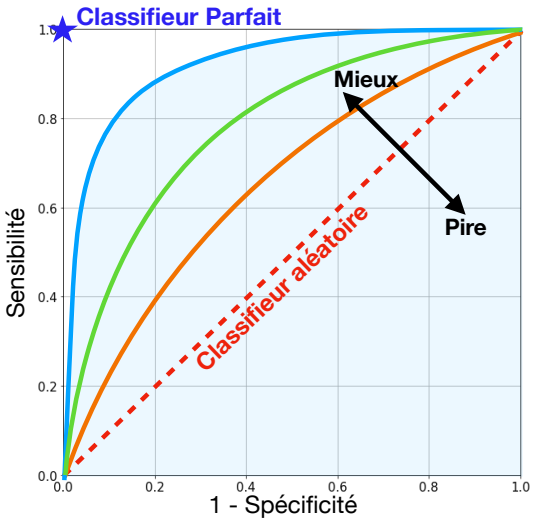

FIGURE 3.8 – Différentes courbes ROC. En abscisse nous retrouvons le taux d'erreur associé à la classe 0 (1−Spécificité) et en ordonnée la Sensibilité du modèle. L'AUC associée à la courbe ROC bleue est supérieure à celle des courbes verte et orange. Le modèle parfait est représenté par l'étoile bleue au point (0,1).

Les tableaux 3.1 et 3.2 résument les performances moyennes (performances globales (3.20) et AUC) obtenues par CV sur 5-folds pour chaque méthode testée à la fois sur les échantillons d'apprentissage et de validation. Le tableau 3.3 quant à lui, illustre les performances conditionnelles (3.21).

3.4.4 Performances prédictives

Dans le tableau 3.1 nous retrouvons les résultats moyennisés (écart-types) des performances prédictives globales (3.20) par les différentes méthodes sur 5-folds sur la base de données Circle.

		Échantillon d'apprentissage		**Échantillon de validation**		**Temps**
	Méthodes	**Perf.**	**AUC**	**Perf.**	**AUC**	
RL	RL	0.55 (0.08)	0.51 (0.01)	0.5 (0.09)	0.5 (0.01)	0.002
	RL NCS	0.88 (0.01)	0.94 (0.01)	0.86 (0.01)	0.93 (0.01)	0.004
Méthodes itératives	MARS	0.9 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.01)	0.324
	GAM	0.9 (0.01)	0.96 (0.01)	0.88 (0.02)	0.95 (0.01)	0.033
	EBM	0.89 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.01)	0.221
	RF	0.9 (0.01)	0.97 (0.01)	0.87 (0.01)	0.94 (0.02)	0.103
RNs ReLU	RN 1-ReLU	0.9 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.01)	702
	RN 2-ReLU	0.9 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.01)	695
	NAM	0.9 (0.01)	0.96 (0.01)	0.88 (0.02)	0.95 (0.01)	314
Nos contributions	RN-MARS	0.88 (0.03)	0.94 (0.04)	0.86 (0.05)	0.93 (0.06)	692
	SATURNN	0.87 (0.03)	0.94 (0.02)	0.85 (0.02)	0.93 (0.01)	687

TABLE 3.1 – Résultats des expériences sur le jeu de données simulé Circle : performances et AUC moyennes (écart-types) obtenus par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation et temps d'entraînement moyen (en secondes). Nos contributions sont en bleu.

Ce jeu de données étant hautement non linéaire il était attendu que la LR ne soit pas en mesure de différencier les deux classes correctement (50% $\pm$ 9 sur l'échantillon de validation). Lorsque nous intégrons à cette modélisation des effets de seuils et plus particulièrement des Splines Naturelles Cubiques (NCS), l'AUC atteint 93% sur l'échantillon de validation. Il est à noter ici que la LR NCS est très performante car les quantiles uniformes dans ce cas bien précis sont avantageux pour dessiner le partitionnement de l'espace ; le partitionnement uniforme se recentre bien sur la classe 2 dans ce cas (Annexe A, Figure A.3-(b)). Dans des applications réelles, les seuils ne sont pas distribués uniformément et il donc est plus difficile si ce n'est impossible d'obtenir de bonnes performances prédictives avec cette méthode. Lorsque nous nous intéressons aux méthodes non linéaires optimisant les seuils, nous constatons que les méthodes gloutonnes ont des performances prédictives aussi élevées que les RN traditionnels et le NAM. Les MARS, GAM, EBM, RN 1-ReLU, RN 2-ReLU et le NAM obtiennent 88% de performance prédictive et 95% d'AUC sur l'échantillon de test. Lorsque nous nous focalisons sur nos contributions, le RN-MARS obtient des performances prédictives légèrement moins élevées que les méthodes non linéaires de l'état de l'art, mais surtout moins stables (respectivement 86% $\pm$5 et 93% $\pm$6 de performance globale et d'AUC sur l'échantillon de validation). La volatilité de ces résultats provient du nombre insuffisant de neurones composant le RN-MARS. En effet, le SATURNN obtient quant à lui des performances certes un peu moins élevées que les autres méthodes non linéaires de l'état de l'art mais stables. Il est tout à fait acceptable dans le contexte de l'application médicale de perdre légèrement en performance globale pour gagner en interprétabilité. De plus, nous savons que des effets non linéaires interviennent en médecine mais restent néanmoins bien moins importants que ceux intervenant sur la base de données Circle.

Nous pouvons vérifier la cohérence de nos méthodes sur le jeu de données Gaussien, dont les effets non linéaires à estimer sont davantage réalistes biologiquement. Les résultats obtenus par les différentes méthodes apparaissent dans le Tableau 3.2.

		Échantillon d'apprentissage		**Échantillon de validation**		**Temps**
	Méthodes	**Perf.**	**AUC**	**Perf.**	**AUC**	
RL	RL	0.72 (0.01)	0.82 (0.01)	0.71 (0.01)	0.81 (0.02)	0.002
	RL NCS	0.76 (0.01)	0.85 (0.01)	0.75 (0.01)	0.83 (0.01)	0.006
Méthodes itératives	MARS	0.78 (0.01)	0.86 (0.01)	0.77 (0.01)	0.84 (0.01)	0.12
	GAM	0.78 (0.01)	0.87 (0.01)	0.77 (0.01)	0.84 (0.01)	0.04
	EBM	0.79 (0.01)	0.87 (0.01)	0.77 (0.01)	0.84 (0.01)	0.09
	RF	0.8 (0.01)	0.89 (0.01)	0.77 (0.02)	0.84 (0.02)	0.83
RNs ReLU	RN 1-ReLU	0.75 (0.01)	0.84 (0.01)	0.73 (0.01)	0.82 (0.01)	1392
	RN 2-ReLU	0.76 (0.01)	0.85 (0.01)	0.74 (0.01)	0.83 (0.01)	1364
	NAM	0.76 (0.01)	0.85 (0.01)	0.75 (0.02)	0.83 (0.01)	571
Nos contributions	RN-MARS	0.74 (0.01)	0.84 (0.01)	0.72 (0.01)	0.82 (0.01)	1315
	SATURNN	0.74 (0.01)	0.83 (0.01)	0.72 (0.01)	0.82 (0.01)	1315

TABLE 3.2 – Résultats des expériences sur le jeu de données simulé gaussien : performances et AUC moyennes (écart-types) obtenues par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation et temps d'entraînement moyen (en secondes). Nos contributions sont en bleu.

Les méthodes les plus performantes sont celles utilisant un algorithme glouton : les MARS, GAM, EBM et RF atteignent 77% de performance prédictive sur l'échantillon de validation. Néanmoins, sur la Figure 3.9 nous pouvons remarquer que l'EBM et surtout le RF isolent

certains échantillons et estiment des règles de décision discontinues et parfois trop spécifiques à l'échantillon d'apprentissage. Il s'agit d'une des principales limites à l'utilisation d'un algorithme itératif.

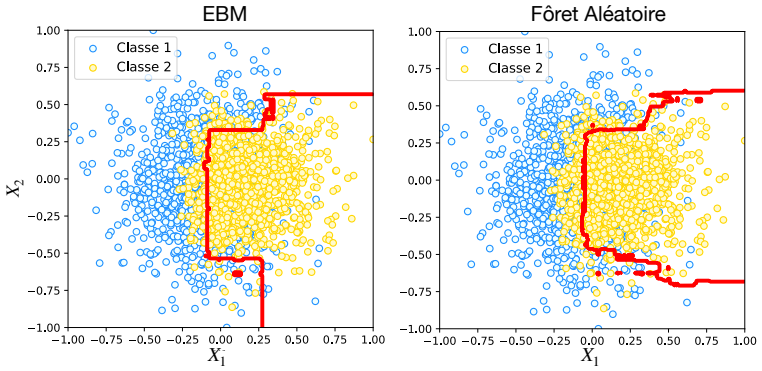

FIGURE 3.9 – Règles de décisions estimées (en rouge) sur le jeu de données simulé gaussien pour l'EBM et le RF. Les règles de décision affichées sont issues des modèles ayant obtenu la meilleure AUC sur l'échantillon de validation lors de la CV.

Lorsque nous nous concentrons sur la métrique AUC, prenant en compte la sensibilité et la spécificité des modèles, l'écart de performance entre les méthodes gloutonnes et nos contributions diminue. Cela peut s'expliquer du fait que le RN-MARS et le SATURNN égalisent davantage les performances conditionnelles (3.21). Dans le tableau 3.3, nous pouvons constater que nos contributions tendent à aussi bien prédire les 2 classes. Quand le SATURNN obtient respectivement 71% et 72% de performances conditionnelles pour les classes 1 et 2, le GAM et le RF atteignent respectivement 67% et 87%. Les échantillons de la classe 1 sont beaucoup moins bien prédits par les méthodes gloutonnes. Le NAM quant à lui obtient des performances par classe déséquilibrées, mais aussi très volatiles pour la classe la mieux prédite (80% $\pm$ 7% d'accuracy pour la classe 2 sur l'échantillon de validation).

		Échantillon d'apprentissage		**Échantillon de validation**	
	Méthodes	**Perf. Cl.1**	**Perf. Cl.2**	**Perf. Cl.1**	**Perf. Cl.2**
RL	RL	0.73 (0.01)	0.71 (0.01)	0.7 (0.02)	0.71 (0.02)
	RL NCS	0.74 (0.01)	0.77 (0.02)	0.71 (0.01)	0.79 (0.03)
Méthodes itératives	MARS	0.69 (0.01)	0.87 (0.01)	0.66 (0.01)	0.87 (0.02)
	GAM	0.7 (0.01)	0.87 (0.01)	0.67 (0.02)	0.87 (0.02)
	EBM	0.71 (0.02)	0.87 (0.01)	0.69 (0.01)	0.85 (0.01)
	RF	0.71 (0.01)	0.87 (0.01)	0.67 (0.04)	0.87 (0.03)
RNs ReLU	RN 1-ReLU	0.73 (0.02)	0.76 (0.03)	0.71 (0.02)	0.75 (0.03)
	RN 2-ReLU	0.76 (0.01)	0.76 (0.01)	0.73 (0.01)	0.75 (0.03)
	NAM	0.73 (0.02)	0.8 (0.04)	0.7 (0.04)	0.8 (0.07)
Nos contributions	RN-MARS	0.75 (0.01)	0.73 (0.01)	0.72 (0.03)	0.72 (0.01)
	SATURNN	0.75 (0.01)	0.73 (0.01)	0.72 (0.01)	0.73 (0.02)

TABLE 3.3 – Performances et AUC moyennes (écart-types) pour chaque classe par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation de la base de données Gaussienne.

3.4.5 Interprétabilité des méthodes

Partitionnement de l'espace d'entrée

Un des grands avantages de nos contributions par rapport aux RN traditionnels réside dans la facilité d'interpréter leurs règles de décision. Tout d'abord le partitionnement des RN-MARS et des SATURNN se fait par orthotopes. Nous pouvons le visualiser par les courbes noires sur les Figures 3.10-(a,b,e,f). Ce partitionnement univarié est bien plus facilement interprétable que celui issu des RN. Sur les Figures 3.10-(c) et (g) nous retrouvons le découpage de l'espace opéré par un RN ReLU à 1 couche cachée à l'aide de régions obliques. Lorsque nous ajoutons une deuxième couche cachée, il devient encore plus difficile, et même impossible d'interpréter les différentes régions. Le partitionnement de l'espace opéré par la deuxième couche cachée des RN ReLU est illustré par les courbes vertes sur les Figures 3.10-(d) et (h).

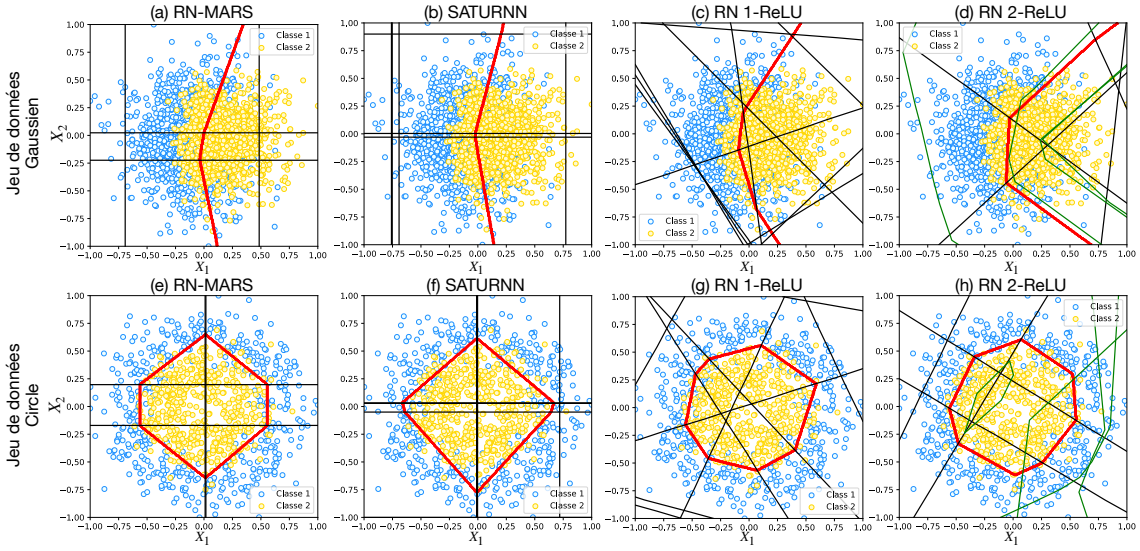

FIGURE 3.10 – Règles de décisions estimées (en rouge) sur les jeux de données simulés Gaussien et Circle pour les RN-MARS (a & e), les SATURNN (b & f), les RNs ReLU à 1 couche (c & g) et 2 couches (d & h). Pour les RNs ReLU à 1 couche, les RN-MARS ainsi que les SATURNN le partitionnement est affiché en traits noirs. Pour les RNs ReLU à 2 couches il est affiché la segmentation opérée par la première couche en traits noirs et la seconde en traits verts. Les règles de décision affichées sont issues des modèles ayant obtenu la meilleure AUC sur l'échantillon de validation lors de la Validation Croisée 5−folds.

Étude des splines

De plus, nous pouvons de part le partitionnement univarié opéré par nos contributions afficher les splines univariées estimées. Les splines obtenues sur le jeu de données Gaussien sont illustrées par la Figure 3.11 et celles obtenues pour le jeu Circle se trouvent en Annexe A - Figure A.1. Lorsque nous nous intéressons aux splines estimées sur le jeu de données Gaussien, nous pouvons constater que pour la variable X_1, toutes les splines augmentent quand X_1 augmente. Ce comportement est justifié puisque lorsque l'on regarde la répartition des deux classes (Figure 3.7-Droite), nous pouvons constater que la classe 2 apparaît au-delà d'une certaine valeur de X_1. En ce qui concerne la variable X_2, presque toutes les

courbes ont la même tendance. Par exemple nous remarquons que l'EBM (courbe bleue claire), le NAM (en rose) et nos contributions (RN-MARS en orange et le SATURNN en rouge) modélisent une spline croissante jusqu'à un certain seuil avant de décroître. Ainsi, ces méthodes ainsi que la LR NCS (en pointillé marron) et le MARS (en pointillé bleu foncé) estiment des splines cohérentes avec le jeu simulé. Nous pouvons aussi constater sur la Figure 3.7-Droite illustrant le jeu de données Gaussien que les échantillons de la classe 2 sont peu présents pour une petite et une grande valeur de X_2. En revanche, la spline estimée par le GAM (en pointillé vert) pour cette variable ne cesse d'augmenter. Il s'agit là encore d'une des conséquences de la nature gloutonne de son algorithme d'optimisation. D'une part la segmentation des variables ne peut être contrôlée mais aussi, aucune garantie de convergence du processus d'estimation des paramètres et d'unicité des résultats ne peut être établi. Bien que ces méthodes soient performantes et interprétables, nous ne sommes pas en mesure de pouvoir interpréter leurs résultats de manière certaine. Ainsi, nos contributions RN-MARS et SATURNN semblent être le meilleur compromis entre performances prédictives et interprétabilité des résultats.

3.5 Synthèse

Dans ce chapitre, nous avons alors présenté nos deux premières contributions. Le RN-MARS (Section 3.2) est un RN interprétable fortement inspiré par les Modèles Additifs Généralisés. Sa règle de décision se réécrit comme une somme additive de fonctions non linéaires univariées. Cette méthode a été développée en étroite collaboration avec des médecins, de sorte qu'elle n'estime que deux fonctions non linéaires (splines) par variable descriptive. Lors des expériences numériques, et plus précisément celles réalisées sur le jeu de données Circle, nous avons néanmoins constaté que parfois, ce nombre de neurones imposé est trop restrictif et ne permet pas l'estimation d'un modèle stable.

Aussi, nous avons introduit en Section 3.3, le SATURNN pour *Splines Approximation Throught Understandable ReLU Neural Network*, une généralisation du RN-MARS. Contrairement aux RN traditionnels, le SATURNN réalise un partitionnement de l'espace interprétable à l'aide d'orthotopes. Pour ce faire, son architecture est contrainte (sous-section 3.3.1) afin que les variables soient traitées indépendamment les unes des autres. Lors des expériences numériques (Section 3.4) nous avons constaté que ces contraintes imposées au réseau le rend certes légèrement moins performant que ceux qui mélangent les variables entre elles, mais permet néanmoins de gagner en interprétabilité. Enfin, l'interprétation de la règle de décision estimée par le SATURNN est plus fiable que celle des méthodes de l'état de l'art telles que les RFs, les MARS ou encore les GAMs. Ces méthodes gloutonnes ne disposent d'aucune garantie de convergence et l'interprétation de ces splines estimées peuvent parfois être incohérente. Ainsi, le SATURNN est un bon compromis entre performance et interprétabilité.

Bien que le SATURNN optimise un critère global, nous ne disposons pas de garantie d'unicité des résultats estimés, le modèle n'est donc qu'interprétable (Définition 1). Afin que notre méthode soit explicable (Définition 2), c'est à dire estime une règle de décision interprétable et unique, nous proposons dans le chapitre suivant de linéariser partiellement le SATURNN afin de pouvoir l'approximer par une Régression Logistique explicable.

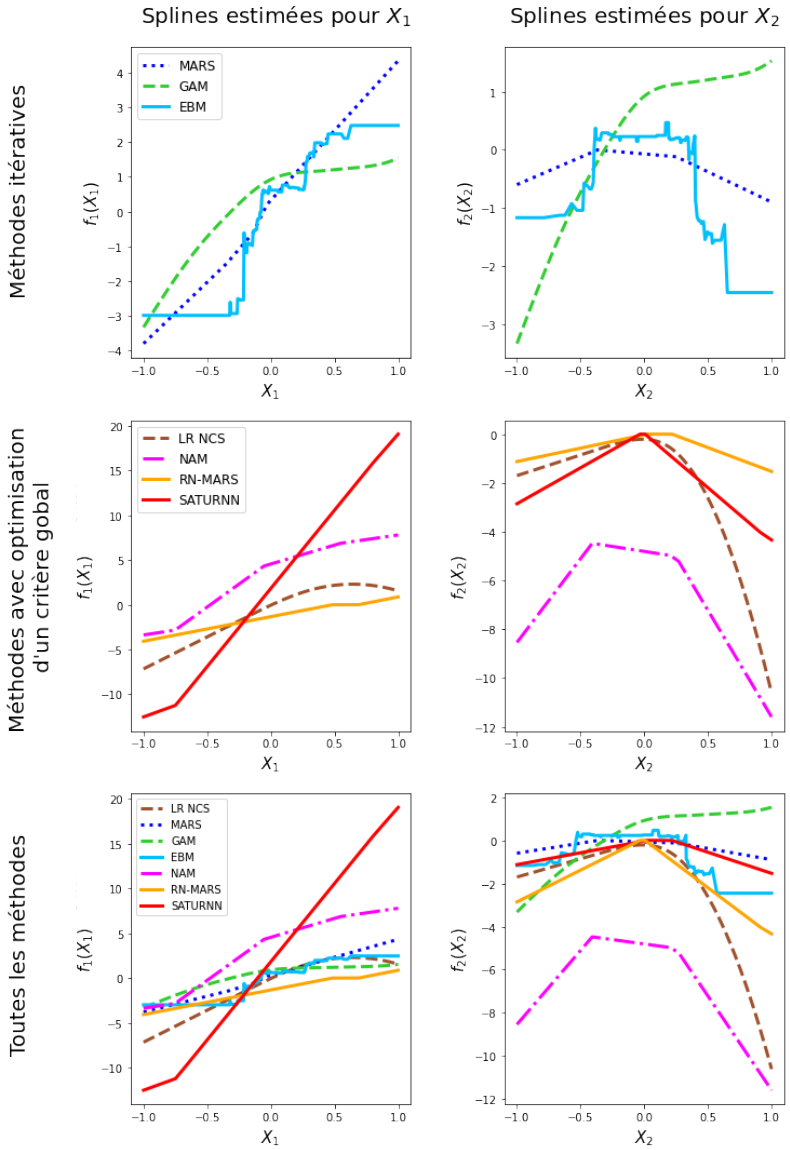

FIGURE 3.11 – Splines estimées sur le jeu de données simulé Gaussien par les différentes méthodes testées estimant des splines univariées : LR NCS (marron), MARS (bleu foncé), GAM (vert), EBM (bleu clair), NAM (rose), RN-MARS (orange) et SATURNN (rouge). À gauche nous retrouvons la spline estimée pour X_1 et à droite celle pour X_2. Sur la figure du haut nous retrouvons les méthodes itératives, sur celle du milieu les méthodes utilisant un critère d'optimisation globale et enfin sur la figure du bas nous retrouvons les splines estimées par l'ensemble des méthodes. Les splines univariées affichées sont issues des modèles ayant obtenu la meilleure AUC sur l'échantillon de validation lors de la Validation Croisée 5−folds.

Chapitre 4

Approximation du SATURNN par Régression Logistique

Dans ce chapitre, nous introduisons une méthode d'approximation du SATURNN qui est interprétable. Tout d'abord, en Section 4.1 nous présentons des travaux de l'état de l'art réalisés pour la linéarisation des Réseaux de Neurones. Bien que ces approximations linéaires facilitent l'étude de la convergence de leur processus d'apprentissage, elles reposent sur des hypothèses fortes non réunies par le SATURNN. Nous nous sommes alors inspirés de ces travaux afin de réaliser en Section 4.2 une linéarisation partielle du SATURNN. En Section 4.3, nous démontrons à partir de ces résultats qu'il est possible d'approximer correctement le SATURNN par une Régression Logistique appliquée à une transformation non linéaire des variables découlant directement de l'architecture du SATURNN. Enfin la Section 4.4 contient des expériences numériques sur les deux bases de données simulées présentées précédemment afin de valider numériquement les différents Lemmes, Propositions et Théorèmes établis dans ce chapitre.

Sommaire

4.1 La linéarisation des Réseaux de Neurones

Bien que les Réseaux de Neurones ReLU (RN ReLU, Section 2.4) optimisent un critère global pour estimer la règle de décision, leur architecture non convexe ne permet d'établir aucune garantie de convergence et donc d'unicité des résultats. Du fait de la non-convexité de la fonction de score $\psi(x, \theta)$ (3.7), la règle de décision estimée par le SATURNN ne peut être définie comme étant explicable au sens de la Définition 2. De nombreux travaux ont été menés afin d'étudier la possibilité d'optimiser des RN convexes. Dans [Bengio *et al.*, 2005], les auteurs proposent un algorithme d'optimisation progressif applicable aux petites bases de données. Tant qu'un critère de convergence n'est pas atteint, des couches cachées sont ajoutées. Cet apprentissage est sous-optimal ; à chaque nouvelle couche ajoutée seulement la dernière est à nouveau optimisée. Puisque notre motivation principale est de développer un modèle de classification non linéaire explicable, cet algorithme ne convient pas à notre objet d'étude. D'une part, l'ajout de couches cachées rend la règle de décision estimée par le SATURNN difficilement interprétable. D'autre part, s'affranchir des limites des algorithmes gloutons tels que les DT, MARS ou encore GAM, en estimant itérativement un RN et non dans sa globalité n'est pas la solution idéale. Les *Input Convex Neural Networks* (ICNNs) développés dans [Amos *et al.*, 2017] optimisent une fonction de coût convexe en contraignant la matrice de poids W dans (2.31) à être non-négative. L'architecture du SATURNN ne remplie pas les critères nécessaires pour pouvoir s'appuyer sur les ICNNs. En effet, pour modéliser des effets non linéaires à droite ou à gauche d'un certain seuil (Encadré gris de la Figure 3.3) la matrice de poids fixée prend trois valeurs possibles $\{-1, 0, 1\}$.

Une autre approche afin d'établir la convergence de la SGD consiste à sur-paramétrer les RN. Les RN sur-paramétrés, contrairement à ce que cela laisserait supposé, ne sur-apprennent pas et obtiennent des bonnes performances de généralisation [Li et Liang, 2018, Brutzkus *et al.*, 2017]. De plus dans [Brutzkus *et al.*, 2017], il est démontré que l'optimisation par SGD des RN sur-paramétrés converge vers un minimum global. Néanmoins, cette théorie a été seulement développée pour des RN composés de deux couches, appliqués sur des données linéairement séparables. Ainsi, elle ne peut ni s'appliquer aux données bio-médicales, ni au SATURNN. Bien qu'établir la convergence des RN sur-paramétrés est difficile ou repose sur des hypothèses restrictives, les RN composés d'un nombre infini de neurones ont l'avantage de pouvoir être linéarisés [Neal, 2012, Lee *et al.*, 2019, Jacot *et al.*, 2018, Liu *et al.*, 2020b]. En linéarisant les RN, la convergence de la SGD peut ainsi être garantie. Les linéarisations proposées dans ces articles ne peuvent être directement appliquées au SATURNN. Dans [Jacot *et al.*, 2018] le modèle considéré n'inclut pas les paramètres β_0 et β composant la fonction de score du SATURNN (3.7), tandis que dans [Liu *et al.*, 2020b] ils sont considérés fixés. De plus, le SATURNN ne satisfait pas les hypothèses sur les matrices de poids, notre modèle n'en disposant pas. Tous les paramètres considérés dans ces études sont initialisés selon une loi normale centrée réduite. Or, nous avons choisi d'utiliser des distributions plus réalistes pour les initialisations du SATURNN (Hypothèse 1). En effet, les biais sont générés selon une loi uniforme et ainsi sont compris dans un intervalle de valeur pris par les données considérés. Enfin, il a été démontré dans [Liu *et al.*, 2020a] que la linéarisation globale d'un RN composé d'une couche de sortie non linéaire n'est pas possible. Puisque la couche de sortie du SATURNN est caractérisée par l'activation Sigmoïde, il n'est donc pas possible de le linéariser dans sa globalité.

Ainsi, nous proposons dans un premier temps de linéariser partiellement le SATURNN à travers la linéarisation de sa fonction de score (3.7). Nous pouvons aussi démontrer que la composition par la sigmoïde de la fonction de score linéarisée ne change pas la qualité de l'approximation (Section 4.2). Enfin, nous proposons d'approximer le SATURNN par une Régression Logistique (LR) appliquée à la fonction de score linéarisée (Section 4.3). Cette transformation est assimilable à un pré-traitement non linéaire des données lié directement à l'architecture du SATURNN.

4.2 Linéarisation locale du SATURNN

Puisqu'il n'est pas possible de linéariser dans sa globalité le SATURNN, nous nous concentrons sur sa fonction de score $\psi(x, \theta)$ définie par l'équation (3.7). L'activation ReLU (2.17) n'est pas différentiable en tout point ; il existe une valeur pour laquelle la dérivée n'est pas continue à gauche et à droite. L'application ReLU dépend seulement des paramètres b_k, $k \in \{1, \dots, p\}$ du SATURNN, s_k et υ_k étant initialisés puis fixés. Lorsque nous étudions la dérivée de $\psi(x, \theta)$ par rapport aux paramètres b_k nous obtenons une discontinuité pour deux scénarios : lorsque $b_k = x_{\upsilon(k)}$ si $s_k = -1$ et $b_k = -x_{\upsilon(k)}$ si $s_k = 1$. En effet, sur la Figure 4.1 les pointillés rouge illustrent les discontinuités de la dérivée de la fonction de score par rapport au paramètre b_k dans les deux scénarios.

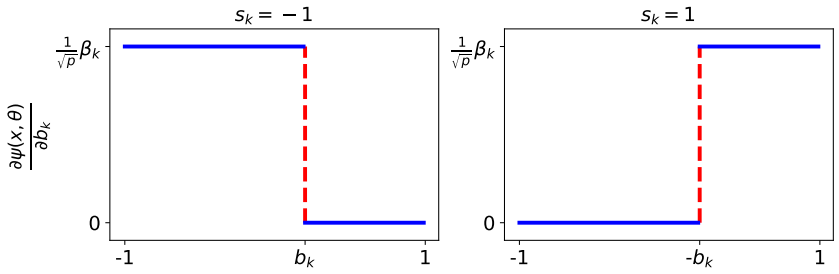

FIGURE 4.1 – Dérivée de la fonction de score $\psi(x, \theta)$ par rapport au paramètre b_k pour $s_k = -1$ (Gauche) et $s_k = 1$ (Droite).

Pour linéariser la fonction de score, nous allons devoir calculer le gradient de $\psi(x, \theta)$ au point $\theta^{(0)}$. Nous pouvons vérifier que pour $b_k^{(0)}$ continu, la probabilité qu'un des deux scénarios se produise, c'est à dire que $b_k^{(0)}$ prennent les valeurs $\{-x_{\upsilon(k)}, x_{\upsilon(k)}\}$ est nulle :

$$\mathbb{P}(b_k^{(0)} \in \{-x_{\upsilon(k)}, x_{\upsilon(k)}\}) = 0. \tag{4.1}$$

Ainsi, il est possible de choisir une initialisation $\theta^{(0)}$ telle que $\psi(x, \theta)$ soit différentiable partout avec une probabilité certaine (Annexe B.1.2). Ainsi, pour la suite du manuscrit nous pouvons alors omettre les points pour lesquels la fonction de score peut ne pas être différentiable. Nous supposons que $\psi(x, \theta)$ est $\mathcal{C}^2$ presque partout et prend des valeurs dans la boule ouverte $\mathcal{B}_2^{2p+1}(\theta, R) := \{\theta \in \mathbb{R}^{2p+1} : \|\theta - \theta^{(0)}\|_2 < R\}$.

4.2.1 Approximation de Taylor d'ordre 2

Nous supposons l'échantillon x fixé, nous linéarisons la fonction de score du SATURNN aux alentours de ses initialisations. D'après le Théorème d'approximation d'ordre 2 de Taylor (Annexe B.1.1 - Théorème 4), il existe $\tau \in \{0,1\}$ telle que la fonction de score peut se réécrire :

$$\begin{aligned}\tilde{\psi}(x,\theta) &= \psi(x,\theta^{(0)}) + \nabla_\theta \psi(x,\theta^{(0)})^T(\theta-\theta^{(0)}) \\ &+\frac{1}{2}(\theta-\theta^{(0)})^T H_\theta \psi((x,(1-\tau)\theta^{(0)}+\tau\theta))(\theta-\theta^{(0)}),\end{aligned} \tag{4.2}$$

avec $\nabla_\theta \psi(x,\theta^{(0)})$ le vecteur de gradient de $\psi(x,\theta)$ par rapport à θ calculé au point $\theta^{(0)}$. Il se définit comme étant l'ensemble des dérivées par rapport aux paramètres $\theta = [\beta^T, b^T]^T \in \mathbb{R}^{2p+1}$ de la fonction de score au point $\theta^{(0)}$:

$$\begin{aligned}\nabla_\theta \psi(x,\theta^{(0)}) &= \left[\frac{\partial\psi(x,\theta^{(0)})}{\partial\beta_0}, \frac{\partial\psi(x,\theta^{(0)})}{\partial\beta_1}, \ldots, \frac{\partial\psi(x,\theta^{(0)})}{\partial\beta_p}, \frac{\partial\psi(x,\theta^{(0)})}{\partial b_1}, \ldots, \frac{\partial\psi(x,\theta^{(0)})}{\partial b_p}\right]^T \\ &= \frac{1}{\sqrt{p}}\Big[1,\ \phi(s_1 x_{\upsilon(1)} + b_1^{(0)}), \phi(s_2 x_{\upsilon(2)} + b_2^{(0)}) \ , \ldots,\ \phi(s_p x_{\upsilon(p)} + b_p^{(0)}), \\ &\qquad \beta_1^{(0)} \mathbb{1}_{\{s_1 x_{\upsilon(1)} + b_1^{(0)} > 0\}},\ \ldots,\ \beta_p^{(0)} \mathbb{1}_{\{s_p x_{\upsilon(p)} + b_p^{(0)} > 0\}}\Big]^T,\end{aligned} \tag{4.3}$$

où $\mathbb{1}_A$ désigne l'indicatrice de l'évènement A. L'annexe B.2.1 détaille les dérivées partielles obtenues par paramètre. Nous pouvons établir que lorsque le nombre p de neurones composant la couche cachée du SATURNN augmente, le gradient (4.3) reste constant.

Lemme 1 (Constance du gradient $\nabla_\theta \psi(x,\theta^{(0)})$)**.**
Soit $\psi(x,\theta)$ la fonction de score du SATURNN définie à l'équation (3.7) et x le vecteur de variables descriptives tel que $x \in \mathcal{B}_2^d(0,r)$. Nous supposons que le vecteur de paramètres $\theta = [\beta_0, \beta_1, \ldots, b_1, \ldots, b_p]$ respecte l'Hypothèse 2 ($\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)$ avec $R > 0$) et que leurs initialisations $\theta^{(0)}$ respectent l'Hypothèse 1 ($\beta_k \sim \mathcal{N}(0,1)$ et $b_k \sim \mathcal{U}[-r,r]$, pour $k \in \{1,\ldots,p\}$ et $r > 0$). Soit $\nabla_\theta \psi(x,\theta^{(0)})$ le gradient de $\psi(x,\theta)$ par rapport à ses paramètres θ calculé au point $\theta^{(0)}$. Au fur et à mesure que le nombre de neurones composant la couche cachée du SATURNN augmente ($p \to \infty$), le gradient reste constant.

$$\sup_{\substack{x\in\mathcal{B}_2^d(0,r) \\ \theta\in\mathcal{B}_2^{2p+1}(\theta^{(0)},R)}} \nabla_\theta \psi\left(x,\theta^{(0)}\right)^T \nabla_\theta \psi\left(x,\theta^{(0)}\right) = O(1), \tag{4.4}$$

avec $O(\cdot)$ la notation grand O.

Démonstration.
La preuve du Lemme se trouve en Annexe B.2.2. □

Dans l'approximation de Taylor de la fonction de score (4.2), $H_\theta \psi((x,(1-\tau)\theta^{(0)}+\tau\theta))$ est la matrice Hessienne de la fonction de score au point $\tilde{\theta} := (1-\tau)\theta^{(0)} + \tau\theta$:

$$H_\theta(\psi(x,\tilde{\theta})) = \begin{pmatrix} H^{(1,1)} & H^{(1,2)} & \ldots & H^{(1,2p+1)} \\ H^{(2,1)} & H^{(2,2)} & \ldots & H^{(2,2p+1)} \\ \vdots & \vdots & \ddots & \vdots \\ H^{(2p+1,1)} & H^{(2p+1,2)} & \ldots & H^{(2p+1,2p+1)} \end{pmatrix}. \tag{4.5}$$

avec chaque élément de la matrice hessienne $H^{(i,j)} := \frac{\partial^2 \psi(x,\tilde{\theta})}{\partial \theta_i \partial \theta_j}$, $i,j \in \{1,\dots,2p+1\}$ correspondant à le dérivée partielle seconde de $\psi(x,\tilde{\theta})$ par rapport à ces paramètres $\theta = [\beta_0, \beta_1, \dots, \beta_p, b_1, \dots, b_p]$, tel que $\theta_0 = \beta_0$ et $\theta_{2p+1} = b_p$ par exemple. Toutes les dérivées secondes sont détaillées en Annexe B.3.1. Ainsi, $H_\theta(\psi(x,\tilde{\theta}))$ se définit comme étant :

$$\left(\begin{array}{c|ccc|ccc} 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \hline \vdots & & & & \ddots & & 0 \\ \vdots & & 0 & & & \frac{\tau^2}{\sqrt{p}} \mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}} & \\ \vdots & & & & 0 & & \ddots \\ \hline \vdots & \ddots & & 0 & & 0 & \\ \vdots & & \frac{\tau^2}{\sqrt{p}} \mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}} & & & & \\ 0 & 0 & & \ddots & & & \end{array}\right). \tag{4.6}$$

Nous remarquons que matrice hessienne ne prend que deux valeurs possibles à savoir 0 ou $\frac{\tau^2}{\sqrt{p}}$. Bien que dépendante des variables d'entrée x de part l'indicatrice $\mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}}$, la hessienne ne fait pas apparaître directement x. Cette architecture de la matrice hessienne résulte de la composition de la fonction de score par l'activation ReLU. Grâce à cette architecture épurée et la non-dépendance directe des valeurs des variables d'entrée, nous pouvons établir que la matrice hessienne $H_\theta((1-\tau)\theta^{(0)} + \tau\theta)$ tend à devenir nulle lorsque le nombre p de neurones composant le SATURNN devient suffisamment large.

Lemme 2 (Comportement asymptotique de la hessienne $H_\theta((1-\tau)\theta^{(0)} + \tau\theta)$)**.**
Soit $\psi(x,\theta)$ la fonction de score du SATURNN définie à l'équation (3.7) et x le vecteur de variables descriptives tel que $x \in \mathcal{B}_2^d(0,r)$. Nous supposons que le vecteur de paramètres $\theta = [\beta_0, \beta_1, \dots, b_1, \dots, b_p]$ respecte l'Hypothèse 2 ($\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)$ avec $R > 0$) et que leurs initialisations $\theta^{(0)}$ respectent l'Hypothèse 1 ($\beta_k \sim \mathcal{N}(0,1)$ et $b_k \sim \mathcal{U}[-r,r]$, pour $k \in \{1,\dots,p\}$ et $r > 0$). Soit $H_\theta((1-\tau)\theta^{(0)} + \tau\theta)$ la matrice hessienne de $\psi(x,\theta)$ par rapport à ses paramètres θ calculée au point $(1-\tau)\theta^{(0)} + \tau\theta$ avec $\tau \in \{0,1\}$. Au fur et à mesure que le nombre de neurones composant la couche cachée du SATURNN augmente ($p \to \infty$), la matrice hessienne $H_\theta((1-\tau)\theta^{(0)} + \tau\theta)$ s'approche de zéro :

$$\sup_{\substack{x \in \mathcal{B}_2^d(0,r) \\ \theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)},R)}} \left\| H_\theta(\psi(x, (1-\tau)\theta^{(0)} + \tau\theta)) \right\|_2 = O\left(\frac{1}{\sqrt{p}}\right). \tag{4.7}$$

Démonstration.
La preuve du Lemme se trouve en Annexe B.3.2. □

4.2.2 Linéarisation de la fonction de score

Ainsi, lorsque le nombre p de neurones est suffisamment grand, la deuxième partie du développement de Taylor définit à l'équation (4.2) est négligeable (Lemme 2). De plus, nous avons établi que lorsque la couche cachée du SATURNN est suffisamment profonde, le

gradient dans le développement de Taylor reste constant (Lemme 1). Ainsi, nous proposons de linéariser $\psi(x, \theta)$ par le modèle linéaire $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ suivant :

$$\psi^{\text{lin}}(x, \theta, \theta^{(0)}) = \psi(x, \theta^{(0)}) + \nabla_\theta \psi(x, \theta^{(0)})^T (\theta - \theta^{(0)}), \tag{4.8}$$

tel que $\nabla_\theta \psi(x, \theta^{(0)})$ est le gradient de $\psi(x, \theta)$ par rapport à ses paramètres θ calculé au point $\theta^{(0)}$. Il est à noter que le modèle $\psi^{\text{lin}}(x, \theta^{(0)})$ (4.8) est une linéarisation partielle de la fonction de score $\psi(x, \theta)$ (3.7). En effet, la fonction ne dépend plus des paramètres θ mais reste néanmoins non linéaire en x à travers le gradient $\nabla_\theta \psi(x, \theta^{(0)})$. De plus, cette linéarisation reste fortement liée à l'apprentissage du SATURNN car elle dépend des initialisations $\theta^{(0)}$ du réseau. Il est établi dans le Théorème 1 que lorsque le nombre p de neurones composant le SATURNN est suffisamment grand alors l'erreur d'approximation de la fonction de score du SATURNN $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^0)$ tend à être nulle.

Théorème 1 (Erreur d'approximation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$)**.**
Soit $\psi(x, \theta)$ la fonction de score du SATURNN définie à l'équation (3.7) et x le vecteur de variables descriptives tel que $x \in \mathcal{B}_2^d(0, r)$. Nous supposons que le vecteur de paramètres $\theta = [\beta_0, \beta_1, \ldots, b_1, \ldots, b_p]$ respecte l'Hypothèse 2 ($\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)$ avec $R > 0$) et que leurs initialisations $\theta^{(0)}$ respectent l'Hypothèse 1 ($\beta_k \sim \mathcal{N}(0, 1)$ et $b_k \sim \mathcal{U}[-r, r]$, pour $k \in \{1, \ldots, p\}$ et $r > 0$). L'erreur d'approximation de $\psi(x, \theta)$ par $\psi^{lin}(x, \theta, \theta^{(0)})$ définie à l'équation (4.8) est uniformément bornée par :

$$\sup_{\substack{x \in \mathcal{B}_2^d(0,r) \\ \theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)}} \left| \psi(x, \theta) - \psi^{lin}(x, \theta, \theta^{(0)}) \right| \leq \frac{R^2}{2\sqrt{p}} = O\left(\frac{1}{\sqrt{p}}\right). \tag{4.9}$$

Démonstration.
Nous supposons que la fonction de score $\psi(x, \theta)$ définie à l'équation (3.7) est $\mathcal{C}^2$ presque partout. De plus, nous supposons que les paramètres θ appartiennent à la boule ouverte $\mathcal{B}(\theta^{(0)}, R)$ avec $R > 0$ (Hypothèse 2). D'après l'approximation de Taylor d'ordre 2 (4.2), il existe $\tau \in [0, 1]$ tel que $\psi(x, \theta)$ peut être correctement approximée par :

$$\begin{aligned} \psi(x, \theta, \theta^{(0)}) &= \psi(x, \theta^{(0)}) + \nabla_\theta \psi(x, \theta^{(0)})^T (\theta - \theta^{(0)}) \\ &+ \frac{1}{2} (\theta - \theta^{(0)})^T H_\theta(\psi(x, (1 - \tau)\theta^{(0)} + \tau\theta))(\theta - \theta^{(0)}), \end{aligned} \tag{4.10}$$

avec $\nabla_\theta \psi(x, \theta^{(0)})$ (4.3) le gradient de $\psi(x, \theta^{(0)})$ par rapport à ses paramètres θ pris au point $\theta^{(0)}$, les initialisations des paramètres respectant l'Hypothèse 1. $H_\theta(\psi(x, (1 - \tau)\theta^{(0)} + \tau\theta))$ (4.6) désigne la matrice hessienne de $\psi(x, \theta)$ par rapport à ses paramètres θ calculée au point $(1 - \tau)\theta^{(0)} + \tau\theta$. À partir de l'équation (4.8), nous savons que nous pouvons réécrire cette expansion de Taylor d'ordre 2 comme suivant :

$$\psi(x, \theta, \theta^{(0)}) = \psi^{\text{lin}}(x, \theta, \theta^{(0)}) + \frac{1}{2} (\theta - \theta^{(0)})^T H_\theta(\psi(x, (1 - \tau)\theta^{(0)} + \tau\theta))(\theta - \theta). \tag{4.11}$$

Ainsi, l'erreur d'approximation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ se définie comme étant :

$$\left| \psi(x, \theta) - \psi^{\text{lin}}(x, \theta, \theta^{(0)}) \right| = \left| -\frac{1}{2} (\theta - \theta^{(0)})^T H_\theta(\psi(x, (1 - \tau)\theta^{(0)} + \tau\theta))(\theta - \theta^{(0)}) \right|. \tag{4.12}$$

La matrice hessienne $H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))$ admet les valeurs propres :

$$\lambda_k, \lambda_{k+p} = \frac{\tau^2}{p}\mathbb{1}_{\{s_k x_{\upsilon(k)}+(1-\tau)b_k^{(0)}+\tau b_k>0\}}, \tag{4.13}$$

pour $k \in \{1,\dots,p\}$ (Annexe B.3.2, équation (B.17)). Puisque la hessienne admet donc des valeurs propres ainsi que des vecteurs propres, elle est diagonalisable par $P^T\Lambda P$ avec $\Lambda = \text{diag}(\lambda_1,\dots,\lambda_{2p+1})$ la matrice diagonale de valeurs propres et P la matrice orthogonale composée des vecteurs propres associés. De ce fait, nous pouvons réécrire l'erreur d'approximation (4.12) de la manière suivante :

$$\begin{aligned}
\left|\psi(x,\theta) - \psi^{\text{lin}}(x,\theta,\theta^{(0)})\right| &= \left|-\frac{1}{2}(\theta-\theta^{(0)})^T P^T\Lambda P(\theta-\theta^{(0)})\right| \\
&= \left|-\frac{1}{2}z^T\Lambda z\right|, \text{ avec } z = P(\theta-\theta^{(0)}) \\
&= \left|-\frac{1}{2}\sum_{i=1}^{2p+1}\lambda_i z_i^2\right| \\
&\leq \frac{\lambda_{\max}}{2}\sum_{i=1}^{2p+1} z_i^2 \\
&\leq \frac{\lambda_{\max}}{2}\|z_i\|_2^2.
\end{aligned}$$

Nous savons que $\|\theta-\theta^{(0)}\|_2^2 = \|P(\theta-\theta^{(0)})\|_2^2 = \|z\|_2^2$. De plus, puisque $\|\theta-\theta^{(0)}\|_2 \leq R$ (Hypothèse 2), nous avons $\|z\|_2^2 \leq R^2$. Finalement, nous obtenons le résultat suivant :

$$\begin{aligned}
\left|\psi(x,\theta) - \psi^{\text{lin}}(x,\theta,\theta^{(0)})\right| &\leq \frac{\lambda_{\max}}{2}R^2 \\
&\leq \frac{R^2}{2\sqrt{p}} \quad \text{car} \quad \lambda_{\max} = \frac{1}{\sqrt{p}} > 0,\ R^2 > 0 \qquad (4.14) \\
&= O\left(\frac{1}{\sqrt{p}}\right). \qquad (4.15)
\end{aligned}$$

□

L'erreur d'approximation de la fonction de score $\psi(x,\theta)$ par $\psi^{\text{lin}}(x,\theta,\theta^{(0)})$ (4.8) établie dans le Théorème 1 ne dépend pas de x. Cette dépend directement de la régularisation utilisée pour entraîner le SATURNN (Hypothèse 2). Ainsi, l'approximation de la fonction de score par le modèle linéaire $\psi^{\text{lin}}(x,\theta,\theta^{(0)})$ est fortement dépendant du processus d'optimisation du SATURNN. Néanmoins, le pouvoir d'approximation de $\psi(x,\theta)$ par $\psi^{\text{lin}}(x,\theta^{(0)})$ dépend avant tout du nombre p de neurones composant le SATURNN. La régularisation ℓ_2 ajoutée au problème d'optimisation du SATURNN (3.12) à travers la contrainte $\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)$ (Hypothèse 2) garantit une erreur négligeable pour une valeur raisonnable de R et un nombre p de neurones composant la couche cachée du SATURNN suffisamment grand. Enfin il est important de remarquer que le Théorème 1 établit une borne d'erreur maximale qui en pratique peut être bien plus faible.

4.2.3 Linéarisation locale du SATURNN

Maintenant que nous avons démontré que la fonction de score $\psi(x, \theta)$ peut être correctement approximée par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ (Théorème 1) pour un nombre p de neurones suffisamment grand, nous pouvons linéariser partiellement le SATURNN. En effet, la composition par la sigmoïde ne change pas la qualité de l'approximation comme établi dans le théorème suivant.

Théorème 2 (Linéarisation locale du SATURNN)**.**
Soit $\psi(x, \theta)$ la fonction de score du SATURNN définie à l'équation (3.7), $\psi^{lin}(x, \theta, \theta^{(0)})$ son approximation linéaire (4.8) et x le vecteur de variables descriptives tel que $x \in \mathcal{B}_2^d(0, r)$. Nous supposons que le vecteur de paramètres $\theta = [\beta_0, \beta_1, \ldots, b_1, \ldots, b_p]$ respecte l'Hypothèse 2 ($\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)$ avec $R > 0$) et que leurs initialisations $\theta^{(0)}$ respectent l'Hypothèse 1 ($\beta_k \sim \mathcal{N}(0, 1)$ et $b_k \sim \mathcal{U}[-r, r]$, pour $k \in \{1, \ldots, p\}$ et $r > 0$). L'erreur d'approximation de $\sigma(\psi(x, \theta))$ par $\sigma\left(\psi^{lin}(x, \theta, \theta^{(0)})\right)$ est uniformément bornée par :

$$\sup_{\substack{x \in \mathcal{B}_2^d(0,r) \\ \theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)}} \left| \sigma(\psi(x, \theta)) - \sigma(\psi^{lin}(x, \theta, \theta^{(0)})) \right| \leq \frac{R^2}{8\sqrt{p}} = O\left(\frac{1}{\sqrt{p}}\right). \tag{4.16}$$

Démonstration.
Le Théorème 1 démontre que la fonction de score du SATURNN $\psi(x, \theta)$ peut être correctement approximée par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ définie à l'équation (4.8). Nous pouvons ainsi réécrire :

$$\psi(x, \theta) = \psi^{\text{lin}}(x, \theta, \theta^{(0)}) + \epsilon(x, \theta, \theta^{(0)}), \tag{4.17}$$

avec $\epsilon(x, \theta, \theta^{(0)})$ l'erreur d'approximation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ bornée par $\frac{R^2}{2\sqrt{p}}$ comme établi par le Théorème 1.

Nous supposons que la fonction de score $\psi(x, \theta)$ définie à l'équation (3.7) est $\mathcal{C}^2$ presque partout . De plus, nous supposons que les paramètres θ appartiennent à la boule ouverte $\mathcal{B}(\theta^{(0)}, R)$ avec $R > 0$ (Hypothèse 2). La série de Taylor d'ordre 1 garantit qu'il existe $\tau \in [0, 1]$ telle que $\sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)}))$ peut se réécrire :

$$\sigma(\psi(x, \theta)) = \sigma\left(\psi^{\text{lin}}(x, \theta, \theta^{(0)}) + \epsilon(x, \theta, \theta^{(0)})\right) \tag{4.18}$$

$$= \sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)})) + \sigma'(\psi^{\text{lin}}(x, \theta, \theta^{(0)} + \tau\epsilon(x, \theta, \theta^{(0)})))\epsilon(x, \theta, \theta^{(0)})), \tag{4.19}$$

avec $\sigma'(\cdot) = \sigma(\cdot)[1 - \sigma(\cdot)]$ la dérivée première de la sigmoïde, bornée par $\frac{1}{4}$.

L'erreur d'approximation de $\sigma(\psi(x, \theta))$ par $\sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)}))$ est alors égale à :

$$\begin{aligned}
\left| \sigma(\psi(x, \theta)) - \sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)})) \right| &= \left| \sigma'(\psi^{\text{lin}}(x, \theta) + \tau\epsilon(x, \theta, \theta^{(0)}))\epsilon(x, \theta, \theta^{(0)}) \right| \\
&= \Big| \sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)}) + \tau\epsilon(x, \theta, \theta^{(0)})) \\
&\quad \times \left[1 - \sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)}) + \tau\epsilon(x, \theta, \theta^{(0)})) \right] \epsilon(x, \theta, \theta^{(0)}) \Big|
\end{aligned}$$

$$\leq \left| \frac{1}{4} \epsilon(x, \theta, \theta^{(0)}) \right|$$

$$\leq \frac{R^2}{8\sqrt{p}} \tag{4.20}$$

$$= O\left(\frac{1}{\sqrt{p}}\right). \tag{4.21}$$

□

Ainsi, si la valeur de R est raisonnable et le nombre p de neurones composant le SATURNN est suffisamment grand, l'erreur d'approximation de $\sigma(\psi(x, \theta))$ par $\sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)}))$ est négligeable. Le Théorème 2 établit une borne supérieure d'erreur d'approximation ; ainsi en pratique il se peut que l'erreur soit bien plus petite (borne non-atteinte) même pour un petit nombre p de neurones.

4.3 Approximation du SATURNN par une Régression Logistique

D'après le Théorème 2, lorsque le nombre p de neurones composant le SATURNN est suffisamment grand alors le SATURNN peut être correctement approximé par $\sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)}))$. Dans cette section nous allons démontré qu'il est équivalent d'entraîner le SATURNN et une Régression Logistique (LR) appliquée aux données préalablement transformées.

4.3.1 Modélisation de la Régression Logistique appliquée à la fonction de score du SATURNN linéarisée

Soit $\delta^{\text{LR PSI LIN}}$ la LR appliquée aux données transformées $x \mapsto \psi^{\text{lin}}(x, \theta, \theta^{(0)})$:

$$\delta^{\text{LR PSI LIN}}(x, \eta) = \sigma(\psi(x, \theta^{(0)}) + g_0(x)^T(\theta - \theta^{(0)})) = \sigma(c_0(x) + g_0(x)^T\eta), \tag{4.22}$$

avec σ la sigmoïde définie par (2.4) et $\eta = \theta - \theta^{(0)}$ le vecteur de paramètres à apprendre. La LR est appliquée aux variables préalablement transformées par l'application non linéaire $g_0(x)$, le gradient de la fonction de score $\psi(x, \theta)$ par rapport à ses paramètres θ calculé au point $\theta^{(0)}$ défini à l'équation (4.3) :

$$g_0(x) = \nabla_\theta \psi(x, \theta^{(0)}). \tag{4.23}$$

Le terme $c_0(x) = \psi(x, \theta^{(0)})$ est constant par rapport à η et ne dépend que des initialisations du SATURNN $\theta^{(0)}$. Ainsi, en s'appuyant sur les hypothèses d'initialisation du SATURNN (Hypothèse 1), nous pouvons démontrer que $c_0(x)$ a une espérance nulle et une variance bornée par $4r^2 + \frac{1}{p}$ (Annexe B.4). Cette variance est négligeable par rapport à $g_0(x)^T\eta$ et donc la Régression Logistique appliquée à la fonction de score linéarisée (LR PSI LIN) définie à l'équation (4.22) peut être approximée par :

$$\delta^{\text{LR PSI LIN}}(x, \eta) = \sigma(g_0(x)^T\eta). \tag{4.24}$$

La LR PSI LIN applique la sigmoïde aux données préalablement transformées par l'application non linéaire $x \mapsto g_0(x)$. Ainsi, la segmentation de l'espace de l'entrée opéré par la LR PSI LIN est interprétable : les éléments du gradient $g_0(x)$ sont univariés, chaque variable sera donc segmentée indépendamment des autres. Plus nous considérons p élevée, plus le gradient est composé d'élément et donc naturellement plus la LR PSI LIN modélise d'effets non linéaires. Afin d'estimer les paramètres $\eta \in \mathbb{R}^{2p+1}$, il convient de minimiser la log vraisemblance du modèle $\mathcal{L}^{\text{LR}}(\eta, \mathcal{D})$:

$$\hat{\eta}^{\text{LR}} = \underset{\eta \in \mathcal{B}_2^{2p+1}(0,R)}{\arg\min} \mathcal{L}^{\text{LR}}(\eta, \mathcal{D}), \tag{4.25}$$

$$\mathcal{L}^{\text{LR}}(\eta, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} L\left(\delta^{\text{LR PSI LIN}}(x^{(i)}, \eta), y^{(i)}\right), \tag{4.26}$$

avec L l'entropie croisée binaire définie à l'équation (2.9). Le problème d'optimisation (4.25) est convexe par rapport aux paramètres η à estimer. Ainsi, l'étude de la convergence de l'entraînement et donc de l'unicité des résultats est possible. De plus, nous savons qu'entraîner une LR pour laquelle une contrainte ℓ_2 est ajoutée sur les paramètres à estimer revient à considérer la minimisation d'un problème d'optimisation fortement convexe (sous-section 2.2.1). Ainsi, la contrainte $\eta \in \mathcal{B}_2^{2p+1}(0, R) := \langle \eta \rangle_2$ utilisée pour apprendre la LR PSI LIN garantie la convergence du processus d'optimisation et l'unicité des paramètres estimés. Les paramètres estimés $\hat{\eta}^{\text{LR}}$ sont alors uniques conditionnellement aux initialisations $\theta^{(0)}$.

4.3.2 Équivalence avec le SATURNN

Nous pouvons démontrer que lorsque le nombre p de neurones composant le SATURNN est suffisamment grand, il est équivalent d'entraîner le SATURNN en minimisant $\mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D})$ (3.12) ou la LR PSI LIN en optimisant $\mathcal{L}^{\text{LR}}(\eta, \mathcal{D})$ (4.26).

Théorème 3 (Équivalence entre le SATURNN et LR PSI LIN)**.**
Soient $\mathcal{L}^{SATURNN}(\theta, \mathcal{D})$ la fonction de coût du SATURNN (3.12) et $\mathcal{L}^{LR}(\eta, \mathcal{D})$ (4.26) celle de la Régression Logistique appliquée à la fonction de score linéarisée du SATURNN (4.22). Nous supposons que le vecteur de paramètres du SATURNN $\theta = [\beta_0, \beta_1, \ldots, b_1, \ldots, b_p]$ respecte l'Hypothèse 2 ($\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)$ avec $R > 0$) et que leurs initialisations $\theta^{(0)}$ respectent l'Hypothèse 1 . Lorsque le nombre p de neurones composant le SATURNN est suffisamment grand, il est équivalent d'entraîner le SATURNN ou la Régression Logistique appliquée à la fonction de score linéarisée du SATURNN :

$$\sup_{\substack{\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R) \\ \eta \in \mathcal{B}_2^{2p+1}(0,R)}} \left| \mathcal{L}^{SATURNN}(\theta, \mathcal{D}) - \mathcal{L}^{LR}(\eta, \mathcal{D}) \right| \leq \frac{R^2}{2\sqrt{p}}. \tag{4.27}$$

Éléments de preuve.
La démonstration détaillée se trouve en Annexe B.5. Supposons que nous disposons de N échantillons tels que pour chaque échantillon, les variables descriptives $x^{(i)}$ prennent des valeurs dans une boule ouverte de rayon $r > 0$: $x \in \mathcal{B}_2^d(0, r)$. Soit $\Phi^{\text{SATURNN}}(x, \theta) = \sigma(\psi(x, \theta))$ le SATURNN, tel que σ réfère à la sigmoïde (2.4). Nous supposons que les initialisations du SATURNN pour son processus d'optimisation respectent l'Hypothèse 1, à savoir $\theta^{(0)} = [\beta_0^{(0)}, \beta_1^{(0)}, \ldots, \beta_p^{(0)}, b_1^{(0)}, \ldots, b_p^{(0)}]$ avec $\beta_k^{(0)} \sim \mathcal{N}(0, 1)$ et $b_k^{(0)} \sim \mathcal{U}[-r, +r]$.

De plus, nous supposons que l'apprentissage du SATURNN est contraint de sorte que les paramètres estimés $\hat{\theta}$ ne s'éloignent pas trop de ceux initialisés $\theta^{(0)}$, soit à une distance $R > 0$ maximale (Hypothèse 2). Puisque nous nous intéressons à l'équivalence entre le SATURNN et la LR PSI LIN, nous cherchons donc à étudier :

$$\left|\mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D}) - \mathcal{L}^{\text{LR}}(\eta, \mathcal{D})\right| = \left| \frac{1}{N} \sum_{i=1}^{N} L\left(\sigma\left(\psi(x^{(i)}, \theta)\right), y^{(i)}\right) - L\left(\sigma\left(\psi^{\text{lin}}(x^{(i)}, \theta, \theta^{(0)})\right), y^{(i)}\right)\right|. \tag{4.28}$$

Nous avons démontré précédemment que la fonction de score du SATURNN $\psi(x, \theta)$ peut être correctement approximée par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$, telle que l'erreur d'approximation $\epsilon(x, \theta, \theta^{(0)})$ est bornée par $|\epsilon(x, \theta, \theta^{(0)})| \leq \frac{R^2}{2\sqrt{p}}$. Nous pouvons alors réécrire le problème que nous considérons de la manière suivante :

$$\left|\mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D}) - \mathcal{L}^{\text{LR}}(\eta, \mathcal{D})\right| = \left| \frac{1}{N} \sum_{i=1}^{N} L\left(\sigma\left(\psi^{\text{lin}}(x^{(i)}, \theta, \theta^{(0)}) + \epsilon(x^{(i)}, \theta, \theta^{(0)})\right), y^{(i)}\right) - L\left(\sigma\left(\psi^{\text{lin}}(x^{(i)}, \theta, \theta^{(0)})\right), y^{(i)}\right)\right|. \tag{4.29}$$

Lorsque nous raisonnons pour un échantillon i fixé et notons $z = \psi^{\text{lin}}(x, \theta, \theta^{(0)})$ nous pouvons démontrer que :

$$0 < |L(\sigma(z + \epsilon), y) - L(\sigma(z), y)| < \epsilon \leq \frac{R^2}{2\sqrt{p}} \ \forall z \in \mathbb{R}, \text{ quand } y = 0. \tag{4.30}$$

et

$$0 < |L(\sigma(z + \epsilon), y) - L(\sigma(z), y)| < \epsilon \leq \frac{R^2}{2\sqrt{p}} \ \forall z \in \mathbb{R}, \text{ quand } y = 1. \tag{4.31}$$

Finalement, lorsque nous repartons du cas général (4.29), nous pouvons ainsi établir :

$$\sup_{\substack{\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R) \\ \eta \in \mathcal{B}_2^{2p+1}(0, R)}} \left|\mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D}) - \mathcal{L}^{\text{LR}}(\eta, \mathcal{D})\right| \leq \frac{R^2}{2\sqrt{p}}. \tag{4.32}$$

□

D'après le Théorème 3, lorsque p tend vers l'infini alors la différence entre la fonction de coût du SATURNN et celle de la LR PSI LIN est nulle. Dans ce cas, les erreurs d'apprentissage issues de l'entraînement du SATURNN et de la LR PSI LIN sont équivalentes. Il est alors possible d'approximer le SATURNN par la LR PSI LIN : les règles de décision du SATURNN et de la LR PSI LIN obtiennent les mêmes performances prédictives lorsque nous considérons un nombre p de neurones suffisamment grand. De plus, à partir des paramètres estimés par la LR PSI LIN $\hat{\eta}^{\text{LR}}$, nous pouvons retrouver ceux du SATURNN approximé :

$$\tilde{\theta}^{\text{LR}} = \hat{\eta}^{\text{LR}} + \theta^{(0)}. \tag{4.33}$$

Ainsi, en calculant $\tilde{\theta}^{\mathrm{LR}}$, il est possible de retrouver la règle de décision du SATURNN approximé par la LR PSI LIN $\hat{\eta}^{\mathrm{LR}}$:

$$\tilde{\Phi}^{\mathrm{LR}}(x, \tilde{\theta}^{\mathrm{LR}}) = \sigma(\psi(x, \tilde{\theta}^{\mathrm{LR}})). \tag{4.34}$$

En approximant le SATURNN par la LR PSI LIN, nous nous affranchissons des contraintes valant la qualification de "boîtes noires" aux Réseaux de Neurones. En effet, d'une part lorsque nous réinjectons $\tilde{\theta}^{\mathrm{LR}}$ dans le SATURNN nous obtenons une règle de décision interprétable réalisant une segmentation de l'espace d'entrée à l'aide d'orthotopes et se modélisant comme une somme additive de fonctions de splines univariées. D'autre part, la contrainte ℓ_2 ajoutée lors de l'apprentissage des $\hat{\eta}^{\mathrm{LR}}$ (4.25) rend le problème d'optimisation fortement convexe. Ainsi, les paramètres estimés $\hat{\eta}^{\mathrm{LR}}$ et a fortiori $\tilde{\theta}^{\mathrm{LR}}$ sont uniques conditionnellement aux paramètres initialisés $\theta^{(0)}$. De ce fait la règle de décision du SATURNN est robuste et unique. D'après la Définition 2, la LR PSI LIN ainsi que le SATURNN associé sont donc des méthodes de classification explicables.

4.3.3 Implémentation de la Régression Logistique appliquée à la fonction de score du SATURNN linéarisée

Les codes pour pouvoir entraîner la LR PSI LIN sont disponibles sur le Github [1].

Algorithme 1 Apprentissage de $\delta^{\mathrm{LR}}(x, \eta)$

Entrées $X \in \mathbb{R}^{N \times d}$, $Y = \{0, 1\}^N$, p

1: Initialisation du SATURNN : $\theta^{(0)}$ ▷ Hypothèse 1
2: Normalisation des données : $X \to \tilde{X} \in \mathcal{B}_2^d(0, r)$, $r > 0$ ▷ Équation (3.19)
3: Transformation non linéaire des données : $\tilde{X} \mapsto g_0(\tilde{X})$ ▷ Équation (4.23)
4: Entraînement de la LR : estimation de $\hat{\eta}^{\mathrm{LR}}$ ▷ Équation (4.26)
5: Calcul de $\tilde{\theta}^{\mathrm{LR}}$: $\tilde{\theta} = \hat{\eta}^{\mathrm{LR}} + \theta^{(0)}$ ▷ Équation (4.33)
6: Injecter $\tilde{\theta}^{\mathrm{LR}}$ dans le SATURNN : $\tilde{\Phi}^{\mathrm{LR}}(\tilde{X}, \tilde{\theta}^{\mathrm{LR}})$ ▷ Équation (4.34)

Dans l'Algorithme 1, les grandes étapes de l'entraînement de la LR PSI LIN sont détaillées. Dans un premier temps, l'implémentation proposée initialise le SATURNN que nous souhaitons approximer en respectant l'Hypothèse 1. Ensuite les données sont normalisées et transformées par l'application non linéaire $g_0(x)$ (4.23). La Régression Logistique est alors entraînée sur ces données transformées. Pour ce faire, nous avons utilisé la méthode *LogisticRegression* [2] disponible dans la librairie Scikit-Learn. Une fois que les paramètres $\hat{\eta}^{\mathrm{LR}}$ (4.25) sont appris, nous pouvons retrouver l'architecture du SATURNN en calculant $\tilde{\theta}^{\mathrm{LR}}$ définie par l'équation (4.33). Il suffit de réinjecter ces paramètres dans le SATURNN initialisé à l'étape 1 afin de disposer d'un SATURNN entraîné $\tilde{\Phi}^{\mathrm{LR}}(x, \tilde{\theta}^{\mathrm{LR}})$ (4.34) pour lequel la règle de décision issue est interprétable et unique.

4.4 Résultats numériques sur données simulées

Dans cette section, nous présentons des résultats numériques obtenus sur les bases de données simulées Gaussienne et Circle présentées précédemment. Nous nous concentrons dans un premier temps sur la linéarisation locale du SATURNN et ensuite nous étudions

1. Marie Guyomard - Dépôt SATURNN : https://github.com/GuyomardMarie/SATURNN
2. Méthode *LogisticRegression* dans la librairie Scikit-Learn [Pedregosa *et al.*, 2011]

l'équivalence d'apprentissage entre le SATURNN et la LR PSI LIN. Les contributions théoriques établies par les Théorèmes 1, 2 et 3 de ce chapitre, sont présentées sous forme de bornes supérieures d'erreur d'approximation. Afin de vérifier ces théorèmes sur les jeux de données simulées, nous avons calculé à la fois les erreurs moyennes et maximales obtenues sur l'ensemble des échantillons. Ces expériences ont été répétées sur 5−folds. Les figures présentées dans cette section illustrent alors la moyenne ainsi que l'écart type des erreurs moyennes et maximales obtenus sur les 5−folds.

4.4.1 Linéarisation locale du SATURNN

En Section 4.2, nous avons proposé de linéariser la fonction de score du SATURNN par le modèle linéaire $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ défini à l'équation (4.8). Afin que la linéarisation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ soit correcte, deux propriétés doivent être vérifiées :

(i) le gradient de la fonction de score par rapport à ses paramètres calculé au point $\theta^{(0)}$ noté $\nabla_\theta \psi(x, \theta^{(0)})$ (4.3) devient constant à mesure que le nombre p de neurones composant le SATURNN augmente (Lemme 1),

(ii) la norme spectrale de la hessienne de la fonction de score par rapport à ses paramètres pris au point $(1-\tau)\theta^{(0)} + \tau\theta$ avec $\tau \in \{0, 1\}$ notée $H_\theta(\psi(x, (1-\tau)\theta^{(0)} + \tau\theta))$ (4.6) tend à être nulle à mesure que le nombre p de neurones composant le SATURNN augmente (Lemme 2).

Sur les deux bases de données, ces propriétés sont bien vérifiées et les résultats sont disponibles en Annexe C.1. Tout d'abord nous avons vérifié qu'à partir d'un certain p, les normes moyennes et maximales des gradients deviennent constantes (Annexe C.1.1, Figure C.1). De plus, nous avons affiché les histogrammes des normes moyennes et maximales obtenues pour chacun des échantillons (Figures C.2 et C.3). Nous pouvons constater qu'à partir d'une certaine profondeur p, les normes sont toutes inférieures à 1. Enfin, nous avons étudié les normes spectrales des hessiennes (Annexe C.1.2, Figure C.4), qui tendent à devenir nulles lorsque p augmente.

Erreur d'approximation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$

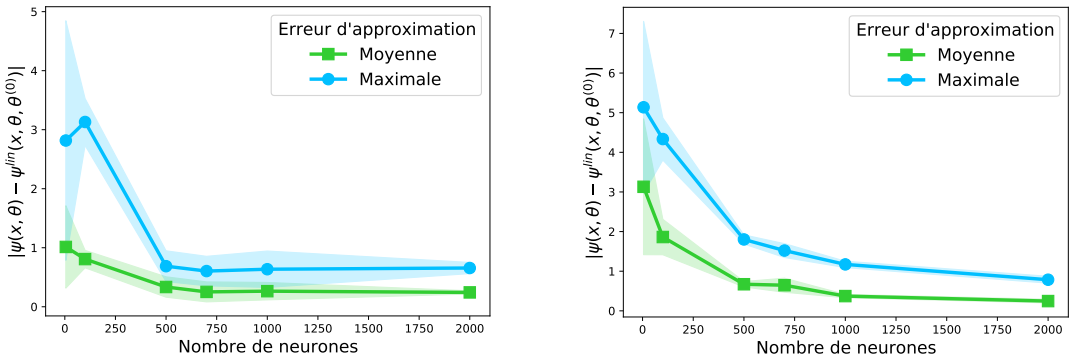

FIGURE 4.2 – Erreurs d'approximation $|\psi(x, \theta) - \psi(x, \theta, \theta^{(0)})|$ moyennes (courbes vertes) et maximales (courbes bleues) calculées pour différentes valeurs de p sur 5 sous-échantillons aléatoires des bases de données Gaussienne (Gauche) et Circle (Droite).

Après avoir vérifié les propriétés nécessaires pour que la linéarisation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ soit correcte, nous étudions son erreur d'approximation moyenne. La

Figure 4.2 illustre la moyenne ainsi que les écart-types des erreurs d'approximation $|\psi(x,\theta) - \psi(x,\theta,\theta^{(0)})|$ moyennes (courbes vertes) et maximales (courbes bleues) obtenues par Validation Croisée 5−folds sur les échantillons de validation des bases de données Gaussienne (Gauche) et Circle (Droite). Nous pouvons constater que les erreurs d'approximation diminuent mais sont aussi moins volatiles lorsque p est suffisamment grand. Ainsi, le Théorème 1 établissant que l'erreur d'approximation entre $\psi(x,\theta)$ et $\psi^{\text{lin}}(x,\theta,\theta^{(0)})$ diminue à une vitesse $\frac{1}{\sqrt{p}}$, est vérifié sur les deux jeux de données.

Erreur d'approximation de $\sigma(\psi(x,\theta))$ par $\sigma(\psi^{\text{lin}}(x,\theta,\theta^{(0)}))$

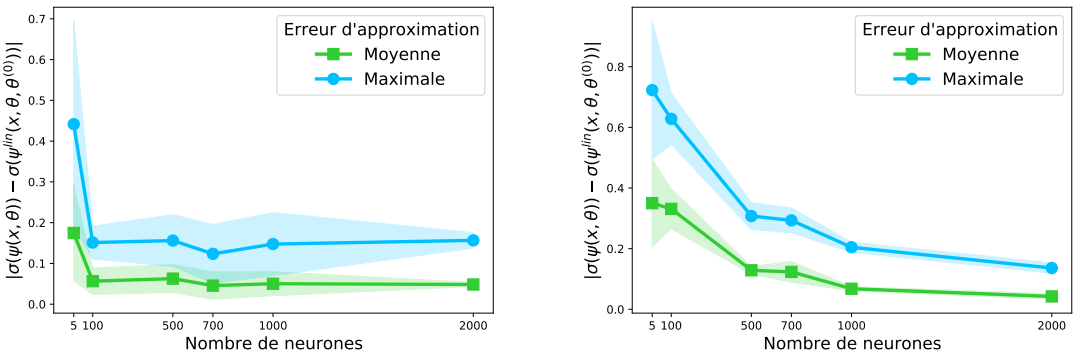

FIGURE 4.3 – Erreurs d'approximation $|\sigma(\psi(x,\theta)) - \sigma(\psi^{\text{lin}}(x,\theta,\theta^{(0)}))|$ moyennes (courbes vertes) et maximales (courbes bleues) obtenues pour différentes valeurs de p sur 5 sous-échantillons aléatoires des bases de données Gaussienne (Gauche) et Circle (Droite).

Nous avons démontré que la composition par la Sigmoïde ne change en rien la qualité de l'approximation. Sur la Figure 4.3, nous affichons les erreurs d'approximation de $\sigma(\psi(x,\theta))$ par $\sigma(\psi^{\text{lin}}(x,\theta,\theta^{(0)}))$ moyennes (courbes vertes) et maximales (courbes bleues). Nous avons calculé $|\sigma(\psi(x,\theta)) - \sigma(\psi^{\text{lin}}(x,\theta,\theta^{(0)}))|$ sur 5 sous -échantillons aléatoires. Nous pouvons constater que la moyenne des erreurs d'approximation maximales obtenues sur 5-folds décroissent à mesure que le nombre p de neurones augmente sur les deux bases de données. De plus, lorsque p augmente, les courbes moyennes d'erreur d'approximation (courbes vertes) tendent vers 0 à une vitesse $\frac{1}{\sqrt{p}}$. Enfin, nous pouvons remarquer que les écart-types de ces courbes diminuent. Ainsi comme établi par le Théorème 2, l'erreur d'approximation de $\sigma(\psi(x,\theta))$ par $\sigma(\psi^{\text{lin}}(x,\theta,\theta^{(0)}))$ tend à être nulle à mesure que la profondeur du SATURNN augmente, mais devient aussi moins volatile.

4.4.2 Équivalence avec la Régression Logistique

Nous avons démontré dans ce chapitre que lorsque p est suffisamment grand, il est équivalent d'apprendre le SATURNN $\Phi^{\text{SATURNN}}(x,\theta)$ (3.6) ou la LR PSI LIN $\delta^{\text{LR PSI LIN}}(x,\eta)$ (4.24). En effet, le Théorème 3 établit que les fonctions de coût minimisées par le SATURNN et la LR PSI LIN sont équivalentes. Dans un premier temps, nous proposons de vérifier l'équivalence établie dans ce théorème. De plus, nous avons admis que la LR PSI LIN admet une unique solution lorsqu'elle est optimisée avec une régularisation ℓ_2 sur les paramètres η. Dans un second temps nous étudions alors l'impact de cette régularisation

sur les paramètres estimés par LR PSI LIN. Puisque les conclusions sont les mêmes sur les deux bases de données simulées, nous présentons les résultats issus du jeu Gaussien. Ceux obtenus sur la base de données Circle sont disponibles en Annexe C.2.

Étude de l'impact de p

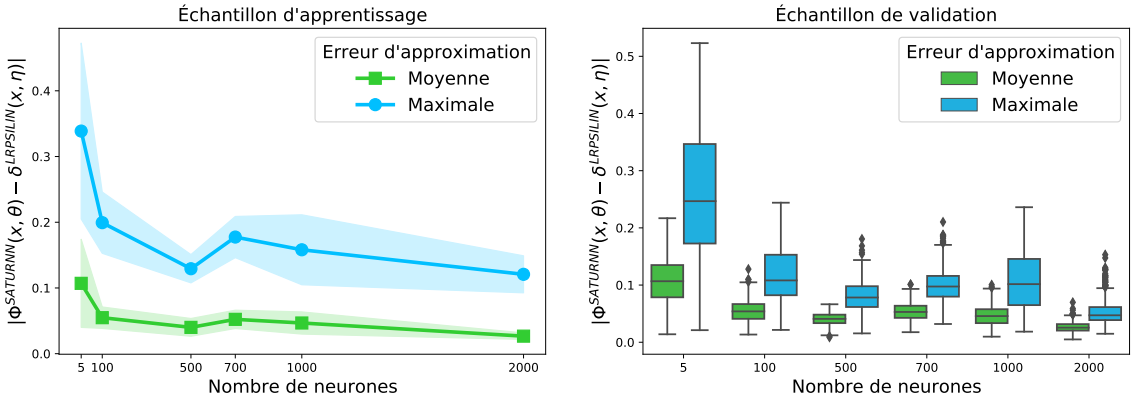

FIGURE 4.4 – Erreurs d'approximation $|\Phi^{\text{SATURNN}}(x,\eta) - \delta^{\text{LR PSI LIN}}(x,\eta)|$ moyennes (courbes vertes) et maximales (courbes bleues) pour différentes valeurs de p. Ces erreurs ont été calculées par Validation Croisée 5-folds sur les échantillons d'apprentissage (Gauche) et de validation (Droite).

La Figure 4.4 présente les moyennes ainsi que les écart-types des erreurs d'approximation de $\Phi^{\text{SATURNN}}(x,\eta)$ par $\delta^{\text{LR PSI LIN}}(x,\eta)$ moyennes (en vert) et maximales (en bleu) obtenues par Validation Croisée $5-$folds pour différentes valeurs de p. Nous constatons que les erreurs d'approximation $|\Phi^{\text{SATURNN}}(x,\eta) - \delta^{\text{LR PSI LIN}}(x,\eta)|$ moyennes et maximales diminuent lorsque p augmente sur les échantillons d'apprentissage (Figure 4.4-Gauche). Cette erreur d'approximation devient aussi moins volatile pour un p suffisamment grand, puisque nous pouvons constater que les écart-types diminuent. Ces résultats sont aussi confirmés sur les échantillons de validation. Sur la Figure 4.4-Droite, les boîtes à moustache d'erreur d'approximation moyennes (vert) et maximales (bleu) suivent la même tendance. À mesure que p augmente, les médianes d'erreurs moyennes et maximales diminuent. De plus, la répartition devient de plus en plus recentrée, ainsi les erreurs sont moins volatiles sur les échantillons de validation à mesure que p augmente. Les résultats théoriques établis par le Théorème 3 sont vérifiés : il est équivalent d'entraîner le SATURNN et la LR PSI LIN lorsque la profondeur p est suffisamment grande.

Lorsque nous nous intéressons à la performance prédictive (3.20) de la LR PSI LIN, nous constatons que plus p augmente, plus la méthode d'approximation du SATURNN est performante. Dans le Tableau 4.1, nous pouvons constater que la LR PSI LIN estimée pour $p = 5$ classifie correctement en moyenne 70% des échantillons de validation sur les $5-folds$ et 77% pour un $p = 2000$. Les AUCs moyennes obtenues sur les échantillons de test sont de 77% pour $p = 5$, 83% pour $p = 700$ et 84% pour la plus grande profondeur testée $p = 2000$.

	Apprentissage		Validation	
p	**Perf. Globale**	**AUC**	**Perf. Globale**	**AUC**
5	0.70 (0.09)	0.77 (0.12)	0.70 (0.07)	0.77 (0.1)
100	0.75 (0.01)	0.84 (0.01)	0.75 (0.01)	0.84 (0.01)
500	0.76 (0.01)	0.85 (0.01)	0.75 (0.02)	0.83 (0.01)
700	0.77 (0.01)	0.85 (0.01)	0.75 (0.02)	0.83 (0.02)
1000	0.77 (0.01)	0.85 (0.01)	0.75 (0.01)	0.84 (0.01)
2000	0.76 (0.01)	0.85 (0.01)	0.77 (0.02)	0.84 (0.02)

TABLE 4.1 – Résultats des LRs PSI LIN sur le jeu de données : performances globales et AUC moyennes (écart-types) obtenues par Validation Croisée 5-folds sur les échantillons d'apprentissage et de validation pour différentes valeurs de p.

Puisque la transformation non linéaire appliquée par la LR PSI LIN aux données n'est autre que le gradient de la fonction de score du SATURNN, elle dépend naturellement du nombre p de neurones considérés. Ainsi, plus p est grand, plus la LR PSI LIN modélise des effets non linéaires et gagne donc en performance prédictive. Au fur et à mesure que des transformations sont ajoutées ($p \to \infty$), la règle de décision devient de plus en plus précise comme illustré par la Figure 4.5 ci-dessus.

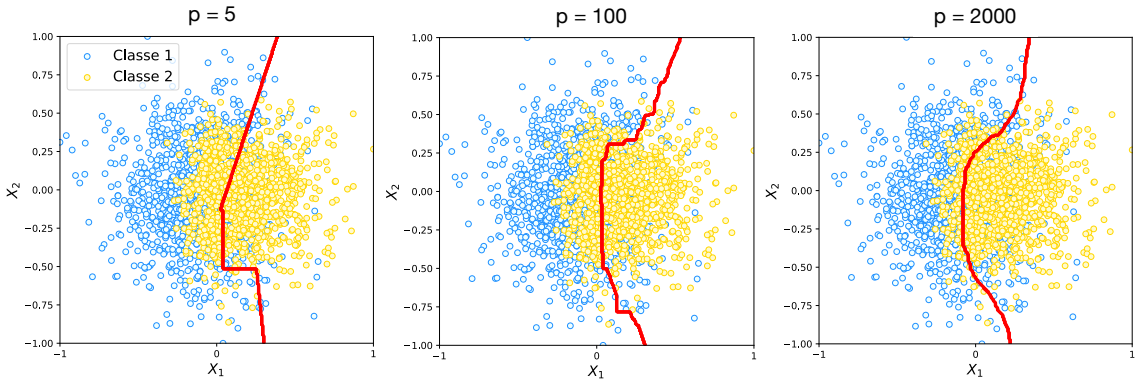

FIGURE 4.5 – Règles de décision (courbes rouges) obtenues par différentes LRs PSI LIN $\delta^{\text{LR PSI LIN}}(x, \eta)$ sur la base de données Gaussienne pour différentes valeurs de p.

Étude de l'impact de la régularisation ℓ_2 sur les paramètres $\hat{\eta}$

La profondeur du SATURNN a un impact non-négligeable sur la performance d'approximation des SATURNN par des LRs PSI LIN. Nous avons décidé pour cette partie expérimentale d'entraîner sur 5-folds un SATURNN composé de $p = 50\ 000$ neurones. Nous étudions l'erreur d'approximation de ce SATURNN par des LRs PSI LIN entraînées avec différentes valeurs de régularisation ℓ_2 sur les paramètres estimés $\hat{\eta}$.

La première chose que nous pouvons constater est que, plus le paramètre de régularisation λ est élevé (et donc R petit dans 4.25), moins les modèles ont tendance à sur-apprendre. La Figure 4.6 illustre les règles de décision issues des LRs PSI LIN apprises avec $\lambda = 0$ (Gauche), $\lambda = 0.01$ (Milieu) et $\lambda = 1$ (Droite). Nous constatons qu'à mesure que le paramètre de régularisation augmente, la frontière affichée en rouge est de moins en moins précise.

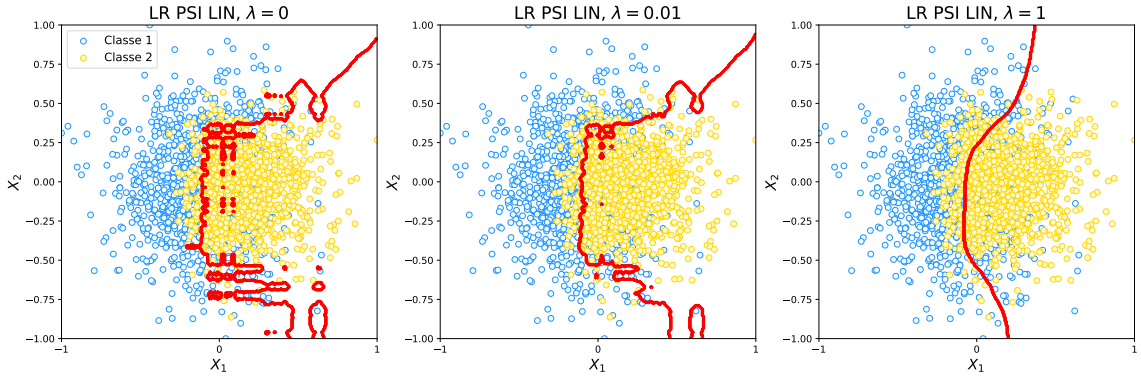

FIGURE 4.6 – Règles de décision (courbes rouges) issues des différentes LRs PSI LIN $\delta^{\text{LR PSI LIN}}(x, \eta)$ pour différentes valeurs de paramètres de régularisation $\lambda = [0, 0.01, 1]$.

Les splines estimées par les LRs PSI LIN semblent elles aussi directement impactées par le paramètre de régularisation. La Figure 4.7 illustre les splines estimées pour les variables X_1 (Gauche) et X_2 (Droite) en fonction du paramètre de régularisation utilisé lors de l'apprentissage des LRs PSI LIN. Nous remarquons que pour un grand paramètre de régularisation (courbes rouges), elles sont moins étendues bien qu'ayant la même forme. Cela s'explique par le fait que les paramètres estimés dans ce cas de figure sont plus proches de zéro.

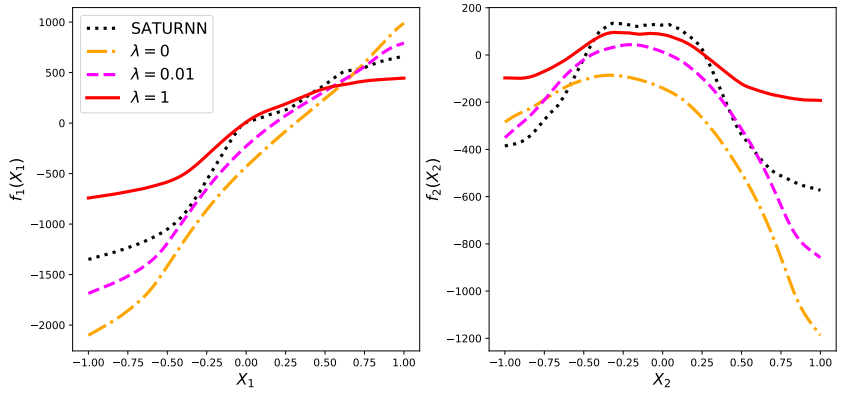

FIGURE 4.7 – Splines estimées pour X_1 (Gauche) et X_2 (Droite) par le SATURNN composé de $p = 50\,000$ neurones en noire, la LR PSI LIN avec $\lambda = 0$ en orange, celle entraînée avec $\lambda = 0.01$ en rose et finalement pour $\lambda = 1$ en rouge.

Plus que de limiter le sur-apprentissage, une régularisation forte a pour avantage de rendre les estimateurs $\hat{\eta}$ moins volatiles et moins dépendants de l'échantillon d'apprentissage sur lequel ils sont appris. La Figure 4.8 illustre les fréquences des normes des $\hat{\eta}$ obtenus par CV 5−folds. Nous pouvons constater que plus la régularisation est grande, plus la volatilité des estimateurs diminue. Ainsi, une forte régularisation rend la règle de décision des LRs PSI LIN unique. Conformément à la Définition 2, l'approximation du SATURNN par une LR PSI LIN entraînée avec une pénalisation suffisamment grande est explicable.

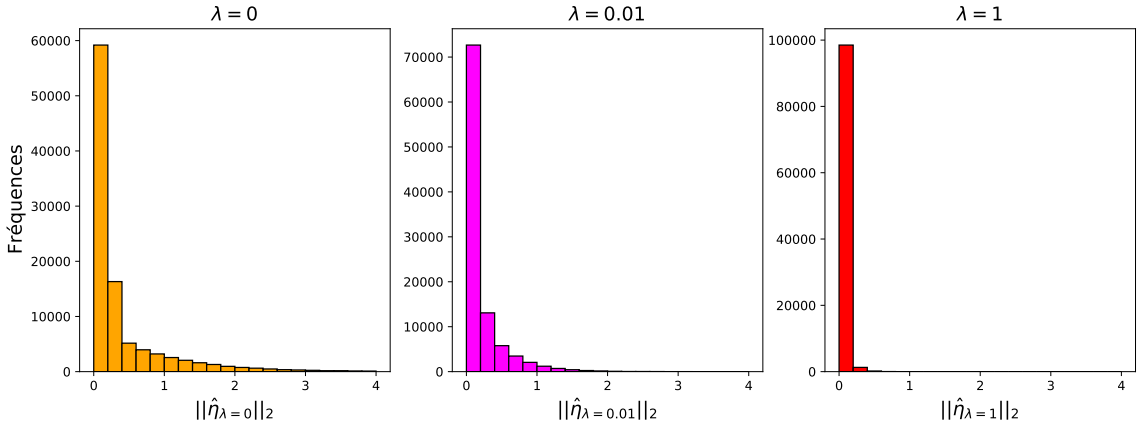

FIGURE 4.8 – Histogramme des normes des $\hat{\eta}$ obtenus par Validation Croisée 5−folds $(\langle \hat{\eta}_{\text{ fold 1}}, \hat{\eta}_{\text{ fold 2}}, \hat{\eta}_{\text{ fold 3}}, \hat{\eta}_{\text{ fold 4}}, \hat{\eta}_{\text{ fold 5}} \rangle)$.

4.5 Synthèse

Dans ce chapitre, nous nous sommes inspirés des travaux de l'état de l'art ayant pour but de linéariser les Réseaux de Neurones. Bien que le SATURNN ne respecte pas les conditions nécessaires pour que l'approximation par un modèle linéaire soit correcte, nous avons tout d'abord démontré en Section 4.2 que la fonction de score du SATURNN $\psi(x, \theta)$ pouvait être correctement approximée par le modèle linéaire $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ lorsque le nombre p de neurones composant le SATURNN est suffisamment grand (Proposition 1 et Théorème 2). Par la suite, nous avons montré que la composition par la sigmoïde ne change pas la qualité de cette approximation. Ainsi, nous avons établi par le Théorème 2 qu'un SATURNN profond (p grand) peut être partiellement linéarisé.

À partir de ce modèle linéaire, nous avons en Section 4.3 introduit une Régression Logistique $\delta^{\text{LR PSI LIN}}(x, \eta)$. La LR PSI LIN est appliquée à des données préalablement transformées par une fonction non linéaire découlant directement de l'architecture du SATURNN. Le Théorème 3 établit qu'il est équivalent d'entraîner un SATURNN composé d'un grand nombre p de neurones et la LR PSI LIN. De plus, lorsque nous ajoutons une régularisation ℓ_2 lors de l'entraînement de LR PSI LIN, nous obtenons une règle de décision unique. D'après la Définition 2, cette méthode d'approximation du SATURNN est explicable.

Il est à noter que cette méthode d'approximation $\delta^{\text{LR PSI LIN}}(x, \eta)$ est néanmoins dépendante du processus d'apprentissage du SATURNN que nous souhaitons approximer. En effet, la transformation non linéaire $g_0(x)$ appliquée aux variables d'entrée x est dépendante des initialisations du SATURNN $\theta^{(0)}$. Ainsi, l'unicité des résultats obtenus lorsque nous apprenons la règle de décision de la LR PSI LIN est conditionnelle à l'ensemble de paramètres initialisés. Dans le prochain chapitre, nous approximons la LR PSI LIN et donc le SATURNN par une méthode à noyau cette fois indépendante du vecteur de paramètres initialisés et donc du processus d'entraînement du SATURNN associé.

Chapitre 5

Approximation du SATURNN par une Régression Logistique à noyau

Dans ce chapitre, nous introduisons deux nouvelles méthodes d'approximation du SATURNN qui sont explicables. Dans un premier temps nous présentons en Section 5.1 les travaux de l'état de l'art qui nous ont inspirés. En Section 5.2 nous introduisons une première Régression Logistique à Noyau. Ce noyau découle directement de l'architecture du SATURNN mais reste dépendant de son processus d'apprentissage. En Section 5.3 nous étudions l'espérance de ce noyau et nous démontrons que le SATURNN peut être correctement approximé par une Régression Logistique à Noyau Déterministe. Bien que découlant directement de l'architecture du SATURNN que nous souhaitons approximer, ce noyau est néanmoins cette fois totalement indépendant de son processus d'initialisation. Enfin la Section 5.4 compare ces contributions aux méthodes à noyau traditionnelles sur les deux bases de données simulées présentées précédemment.

Sommaire

5.1 Équivalence entre les Réseaux de Neurones et les Méthodes à Noyau

Dans le chapitre précédant, nous avons démontré que lorsque les Réseaux de Neurones (RN) sont sur-paramétrés, c'est à dire composés d'un grand nombre de neurones et donc d'un nombre de paramètres estimés supérieur aux données disponibles pour l'entraînement, ils peuvent être correctement linéarisés. Récemment, les auteurs dans [Jacot *et al.*, 2018] ont étudié la convergence des RN au cours de leur processus d'apprentissage. Afin de comprendre l'évolution des paramètres estimés mais aussi leur impact sur la sortie des RN, ils ont étudié la dérivée de la fonction de coût minimisée par rapport au temps. Ils ont démontré qu'un noyau, appelé "Neural Tangent Kernel" (NTK) apparaît alors dans le gradient de la fonction de coût. Il est possible d'établir la convergence d'un RN en étudiant ce noyau. Bien que le NTK soit aléatoire à l'initialisation et varie pendant l'entraînement, lorsque les RN sont composés d'un nombre infini de neurones, le NTK converge vers un noyau explicite borné. De plus, ils ont établi que le NTK reste constant pendant l'apprentissage lorsque le RN considéré est suffisamment profond. Ainsi, en dimension infinie les RN convergent vers une solution finie. Néanmoins, toute cette théorie repose sur des hypothèses d'initialisations gaussiennes, ainsi elle ne peut pas s'appliquer au SATURNN.

De plus, il a été établi dans de nombreux papiers [Neyshabur *et al.*, 2014, Lee *et al.*, 2018, Novak *et al.*, 2018a, Novak *et al.*, 2018b, Neyshabur *et al.*, 2018] que les RN sur-paramétrés ont l'avantage de produire des fonctions se généralisant mieux. Sur le plan théorique, il a été démontré dans [Neal, 2012] et [Williams, 1996] que les RN profonds initialisés selon des lois gaussiennes convergent vers des Processus Gaussiens (GPs). Les auteurs dans [Lee *et al.*, 2018] ont vérifié numériquement que la performance prédictive d'un RN sur-paramétré entraîné avec des initialisations gaussiennes approche celle correspondant à un GP. De plus, une forte corrélation entre l'incertitude de GPs et l'erreur de prédiction du RN a pu être établie. Dans [Matthews *et al.*, 2018], les auteurs proposent une extension de cette théorie aux réseaux de neurones à plusieurs couches. La théorie de l'équivalence entre les RN sur-paramétrés et les GPs est très utilisée et étendue car elle démontre que paradoxalement il serait plus simple d'entraîner et généraliser des RN très profonds plutôt que composés d'un nombre fini de neurones.

Dans la théorie des GPs, les distributions gaussiennes peuvent se réécrire comme des noyaux. Ainsi, de nombreux travaux ont connecté la théorie des RN profonds aux méthodes à noyaux. Il a été établi dans [Lee *et al.*, 2018, Cho et Saul, 2009] que ces noyaux souvent utilisés en inférence bayésienne et pour les Méthodes à Support Vectoriel (SVMs) produisent des résultats comparables aux RN entraînés avec une Descente de Gradient (SGD). Enfin, les auteurs dans [Ortiz-Jiménez *et al.*, 2021] démontrent que les RN ne sont pas toujours plus performants que leurs approximations à noyau. Cet écart de performance dépend fortement de l'architecture du réseau, du nombre d'échantillons et de la tâche d'apprentissage. Ils démontrent notamment qu'au cours de leur apprentissage, les RN profonds intensifient la conformité de leur NTK empirique avec la tâche cible, ce qui explique pourquoi les approximations linéaires à la fin de la formation peuvent mieux expliquer la dynamique des réseaux profonds. Ils concluent que ce phénomène permet aux réseaux d'explorer des fonctions au-delà des limites imposées à leur linéarisation, et améliore leur vitesse d'apprentissage.

Toute cette théorie repose sur la linéarisation des RN. Ainsi, les travaux de [Jacot *et al.*, 2018] et ceux qui ont suivi [Neal, 2012, Lee *et al.*, 2018, Liu *et al.*, 2020a] ne peuvent s'appliquer directement au SATURNN. En effet, l'équivalence entre les RN profonds et les GPs repose notamment sur l'hypothèse d'initialisations gaussiennes et sur une architecture de RN différente de celle du SATURNN (Section 4.1). Bien que les conditions nécessaires à une linéarisation correcte énoncées dans [Liu *et al.*, 2020b] ne sont pas réunies, nous avons démontré qu'il est tout de même possible de linéariser partiellement le SATURNN dans le chapitre précédant. Ainsi, nous aimerions nous inspirer des travaux sur les NTKs afin de dériver une approche à noyau approximant le SATURNN lorsque le nombre de neurones le composant est suffisamment grand.

Dans ce chapitre nous dérivons directement de l'architecture du SATURNN des noyaux et démontrons que le SATURNN peut donc être correctement approximé par des Régression Logistiques à Noyau. Dans la Section 5.2, le noyau proposé est dépendant des initialisations du SATURNN. En effet, dans son architecture, nous retrouvons les paramètres initialisés du réseau. Afin de rendre totalement indépendant la Régression Logistique à Noyau de l'apprentissage du SATURNN, nous établissons en Section 5.3 que le noyau initialement proposé converge asymptotiquement vers son espérance. De ce fait, un noyau totalement indépendant des initialisations du SATURNN, bien que résultant directement de son architecture est proposé. Il devient alors équivalent d'entraîner le SATURNN ou la Régression Logistique appliquée aux données transformées par ce noyau déterministe. Enfin dans la Section 5.4 des expériences sur deux bases de données sont présentées afin de valider numériquement nos résultats.

5.2 Équivalence du SATURNN avec une Régression Logistique à Noyau

Dans le Chapitre 4 nous avons démontré que le SATURNN défini à l'équation (3.6) peut être correctement approximé par une Régression Logistique appliquée à une transformation non linéaire des variables descriptives. Nous considérons ce modèle $\delta^{\text{LR PSI LIN}}(x, \eta)$ défini par l'équation (4.22).

5.2.1 Modélisation de la Régression Logistique à Noyau

D'après le Théorème de Représentation [Schölkopf *et al.*, 2001], le vecteur de paramètres estimés par la LR PSI LIN $\hat{\eta}$ (4.25) peut se réécrire comme une combinaison linéaire des variables d'entrée, puisqu'il est optimisé avec une pénalisation ℓ_2. Plus précisément, le Théorème de Représentation établit qu'il existe $\{\alpha_j\}_{j=1}^N \in \mathbb{R}^N$ tel que :

$$\hat{\eta}^{\text{LR}} = \sum_{j=1}^{N} \alpha_j g_0(x^{(j)})^T. \tag{5.1}$$

La LR PSI LIN nécessite l'apprentissage de $2p+1$ paramètres ($\hat{\eta} \in \mathbb{R}^{2p+1}$). Or en réécrivant les paramètres estimés de la LR PSI LIN comme à l'équation (5.1), nous nous retrouvons avec N paramètres à apprendre : $\alpha = [\alpha_1, \ldots, \alpha_N] \in \mathbb{R}^N$. En réinjectant (5.1) dans le

problème d'optimisation de la LR PSI LIN défini par l'équation (4.25), un noyau apparaît naturellement :

$$\begin{aligned}\mathcal{L}^{\mathrm{LR}}(\eta^{\hat{\mathrm{LR}}}) &= -\frac{1}{N}\sum_{i=1}^{N} y^{(i)} \log\left(\sigma(\hat{\eta}^{\mathrm{LR}} g_0(x^{(i)}))\right) + (1-y^{(i)}) \log\left(1-\sigma(\hat{\eta}^{\mathrm{LR}} g_0(x^{(i)}))\right) \\ &= -\frac{1}{N}\sum_{i=1}^{N} y^{(i)} \log\left(\sigma\left(\sum_{j=1}^{N} \alpha_j \underbrace{g_0(x^{(j)})^T g_0(x^{(i)})}_{\kappa_0(x^{(j)},x^{(i)})}\right)\right) \\ &\qquad + (1-y^{(i)}) \log\left(1-\sigma\left(\sum_{j=1}^{N} \alpha_j \underbrace{g_0(x^{(j)})^T g_0(x^{(i)})}_{\kappa_0(x^{(j)},x^{(i)})}\right)\right). \end{aligned} \tag{5.2}$$

Soit $\kappa_0(x, \tilde{x})$ la fonction définie par :

$$\begin{aligned}\kappa_0 :\ & \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \\ & (x, \tilde{x}) \mapsto g_0(x)^T g_0(\tilde{x}).\end{aligned} \tag{5.3}$$

Un simple calcul montre que la fonction $\kappa_0(x, \tilde{x})$ définie par l'équation (5.3) peut se réécrire de la manière suivante :

$$\begin{aligned}\kappa_0(x, \tilde{x}) = \frac{1}{p}\Bigg[1 + \sum_{k=1}^{p} \phi\left(s_k x_{\upsilon(k)} + b_k^{(0)}\right) \phi\left(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}\right) \\ + {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}}\Bigg],\end{aligned} \tag{5.4}$$

avec $\phi(\cdot)$ l'activation ReLU (2.17) et $\beta_0, \beta_k, s_k, \upsilon(k), b_k$, pour tout $k \in \{1, \ldots, p\}$, l'ensemble des paramètres composant le SATURNN (Section 3.3.1). Rappelons que s_k et $\upsilon(k)$, pour tout $k \in \{1, \ldots, p\}$ sont des paramètres initialisés selon des distributions particulières puis fixés : $s_k \in \{-1, 1\} \sim \mathcal{B}(1/2)$ et $\upsilon(k) \sim \mathcal{U}[\![1, d]\!]$. De plus, $\theta^{(0)} = [\beta_0^{(0)}, \beta_1^{(0)}, \ldots, \beta_p^{(0)}, b_1^{(0)}, \ldots, b_p^{(0)}] \in \mathbb{R}^{2p+1}$ est le vecteur d'initialisation des paramètres appris par le SATURNN respectant l'Hypothèse 1. La fonction κ_0 (5.4) est symétrique, puisqu'il est clair que $\kappa_0(x, \tilde{x}) = \kappa_0(\tilde{x}, x)$. Soit $\mathcal{K}_0(x)$ la matrice suivante :

$$\mathcal{K}_0(X) = \begin{pmatrix} \kappa_0(x^{(1)}, x^{(1)}) & \ldots & \kappa_0(x^{(1)}, x^{(N)}) \\ \vdots & & \vdots \\ \kappa_0(x^{(N)}, x^{(1)}) & \ldots & \kappa_0(x^{(N)}, x^{(N)}) \end{pmatrix}. \tag{5.5}$$

La matrice $\mathcal{K}_0(X)$ définie par (5.5) est constituée des produits scalaires κ_0 pour tout couple de variables descriptives $\{x^{(i)}, x^{(j)}\}_{i=1,j=1}^{N}$, le Théorème d'Aronszajn [Aronszajn, 1950] nous permet d'affirmer que cette matrice est définie positive.

Définition 3 (Fonction noyau)**.**
Une fonction $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ est dite noyau si :

— *elle est symétrique :* $\forall x, y \in \mathcal{X},\ k(x, y) = k(y, x)$

— *elle respecte des conditions de positivité :* $\forall x^{(1)}, \ldots, x^{(N)} \in \mathcal{X}$*, la matrice de noyau* K *est définie positive, c-à-d :*

$$\forall \alpha \in \mathbb{R}^N,\ \alpha^T K \alpha = \sum_{i,j=1}^{N} \alpha_i \alpha_j k(x^{(i)}, x^{(j)}) \geq 0.$$

Ainsi, la fonction $\kappa_0(x, \tilde{x})$ définie à l'équation (5.4) et la matrice $\mathcal{K}_0(X)$ qui en découle définie par (5.5) réunissent toutes les conditions détaillées dans la Définition 3, pour affirmer d'une part que $\kappa_0(x, \tilde{x})$ est un noyau, et d'autre part que $\mathcal{K}_0(X)$ est la matrice noyau qui en résulte.

La LR PSI LIN peut alors se réécrire sous la forme d'une Régression Logistique à Noyau comme définie par la Proposition suivante :

Proposition 1 (Régression Logistique à Noyau)**.**
Soit $\delta^{LR\ PSI\ LIN}(x, \theta)$ *la Régression Logistique appliquée aux données préalablement transformées par* $g_0(x) = \nabla_\theta \psi(x, \theta^{(0)})$*, le gradient de la fonction de score du SATURNN par rapport à ses paramètres* θ *calculé au point* $\theta^{(0)}$*. Nous supposons que les initialisations* $\theta^{(0)}$ *du SATURNN respectent l'Hypothèse 1. Soient* $\sigma(\cdot)$ *la sigmoïde (2.4),* $\kappa_0(x, \tilde{x})$ *la fonction noyau définie par l'équation (5.4),* $\mathcal{K}_0(X)$ *la matrice noyau qui en découle (5.5) et* $K_0(x^{(j)})$ *le* j^e *vecteur composant la matrice noyau pour* $j \in \{1, \ldots, N\}$ *tel que*

$$K_0(x^{(j)}) = \left(\kappa_0(x^{(1)}, x^{(j)}), \ldots, \kappa_0(x^{(N)}, x^{(j)})\right)^T.$$

La LR PSI LIN est équivalente à la Régression Logistique à Noyau (KLR) suivante :

$$\delta^{KLR}(x, \alpha) = \sigma\left(\sum_{j=1}^{N} \alpha_j \kappa_0(x, x^{(j)})\right) = \sigma(K_0(x)^T \alpha). \tag{5.6}$$

Démonstration.
Puisque $\delta^{\text{KLR}}(x, \alpha)$ est fondée sur le Théorème de Représentation [Schölkopf *et al.*, 2001] appliqué à $\delta^{\text{LR PSI LIN}}(x, \eta)$, les deux modèles sont évidemment équivalents. □

Ainsi, LR PSI LIN peut être approximée par la KLR $\delta^{\text{KLR}}(x, \eta)$ définie à l'équation (5.6), puisque les deux méthodes sont équivalentes. Tout comme pour la LR PSI LIN, la KLR applique la Régression Logistique aux variables préalablement transformées. Pour $\delta^{\text{LR PSI LIN}}(x, \eta)$ la transformation non linéaire $x \mapsto g_0(x) = \nabla_\theta \psi(x, \theta^{(0)})$ est totalement indépendante du nombre d'échantillons composant la base d'apprentissage. En revanche pour la KLR, l'application non linéaire $x \mapsto \left(\kappa_0(x^{(i)}, x)\right)_{i=1}^{N}$ dépend de l'ensemble de l'échantillon d'apprentissage. Plus nous disposons d'échantillons pour l'apprentissage du modèle, plus la KLR modélise des effets non linéaires. Le noyau κ_0 mélange toutes les variables entre elles. Ainsi, les variables descriptives ne sont plus transformées indépendamment les unes des autres. Le partitionnement de l'espace d'entrée résultant n'est plus univarié et n'est donc plus opéré par des orthotopes.

Puisque (i) les fonctions de coût minimisées par le SATURNN et la LR PSI LIN sont équivalentes après entraînement pour un p suffisamment grand (Théorème 3), (ii) la LR

PSI LIN est équivalente à la KLR (Proposition 1), nous pouvons en déduire que les performances prédictives du SATURNN et de la KLR deviennent équivalentes à mesure que p augmente. Ainsi, le SATURNN composé d'un grand nombre p de neurones peut être correctement approximé par une Régression Logistique appliquée à un noyau directement inspiré de son architecture. En effet, si le nombre p de neurones composant le SATURNN est suffisamment grand, il est équivalent d'entraîner le SATURNN ou la KLR.

5.2.2 Apprentissage de la Régression Logistique à Noyau

Afin d'estimer les paramètres $\hat{\alpha}^{\text{KLR}} \in \mathbb{R}^N$ de la KLR, il suffit de minimiser la fonction de coût $\mathcal{L}^{\text{KLR}}$:

$$\hat{\alpha}^{\text{KLR}} = \underset{\alpha \in \mathcal{B}_2^N(0,R)}{\arg\min}\ \mathcal{L}^{\text{KLR}}(\alpha, \mathcal{D}), \tag{5.7}$$

définie par

$$\mathcal{L}^{\text{KLR}}(\alpha, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} L\left(\delta^{\text{KLR}}(x^{(i)}, \alpha), y^{(i)}\right), \tag{5.8}$$

avec L l'Entropie Croisée Binaire (2.9). La régularisation ℓ_2 ajoutée au problème d'optimisation de la KLR (5.7) assure que le processus d'optimisation converge vers une solution unique. Cette unicité reste néanmoins conditionnelle à l'échantillon d'apprentissage sur lequel le noyau est calculé et aux initialisations du SATURNN $\theta^{(0)}$.

Ainsi, le SATURNN composé d'un grand nombre p de neurones peut être correctement approximé par une Régression Logistique appliquée à un noyau directement inspiré de son architecture, et dont la règle de décision est unique. Il est important de noter que le noyau κ_0 mélange toutes les variables entre elles et qu'ainsi le partitionnement de l'espace d'entrée résultant de la KLR est initialement composé de régions obliques. Il est néanmoins possible de retrouver une règle de décision se modélisant comme une somme additive de splines univariées facilement interprétable. En effet, la KLR est construite à partir de certaines initialisations $\theta^{(0)}$ du SATURNN. À partir des paramètres appris par la KLR, $\hat{\eta}^{\text{KLR}}$ uniques conditionnellement aux initialisations $\theta^{(0)}$, nous pouvons retrouver la règle de décision d'un SATURNN initialisé avec $\theta^{(0)}$ en calculant :

$$\tilde{\theta}^{\text{KLR}} = \sum_{j=1}^{N} \hat{\alpha}_j^{\text{KLR}} g_0(x^{(j)}) + \theta^{(0)}. \tag{5.9}$$

Ainsi, la règle de décision issue de la KLR peut se réécrire comme une somme additive de splines univariées facilement interprétable en réinjectant $\tilde{\theta}^{\text{KLR}}$ (5.9) dans le SATURNN initialisé :

$$\tilde{\Phi}^{\text{KLR}}(x, \tilde{\theta}^{\text{KLR}}) = \sigma(\psi(x, \tilde{\theta}^{\text{KLR}})). \tag{5.10}$$

Il est alors possible de retrouver un partitionnement de l'espace d'entrée univarié opéré par des orthotopes, interprétable et unique ($\hat{\alpha}^{\text{KLR}}$ unique). Ainsi, conformément à la Définition 2, la KLR $\delta^{\text{KLR}}(x, \alpha)$ est une méthode de classification non linéaire explicable, puisque sa règle de décision est d'une part interprétable et d'autre part unique. Néanmoins, cette unicité est par construction conditionnelle à l'échantillon d'apprentissage (construction du noyau) mais aussi aux initialisations du SATURNN.

Dans l'Algorithme 2, les grandes étapes de l'entraînement de la KLR sont détaillées. Tout

comme pour la LR PSI LIN, nous initialisons le SATURNN que nous souhaitons approximer en respectant l'Hypothèse 1 et normalisons les données avant de les transformer par l'application non linéaire $x \mapsto \left(\kappa_0(x^{(i)}, x)\right)_{i=1}^{N}$ (5.4). La Régression Logistique est alors entraînée sur ces données transformées. Pour ce faire, nous avons utilisé la méthode *LogisticRegression*[1] disponible dans la librairie Scikit-Learn. Une fois que les paramètres $\hat{\alpha}^{\text{KLR}}$ (5.7) sont appris, nous pouvons retrouver l'architecture du SATURNN en calculant $\tilde{\theta}^{\text{KLR}}$ défini par l'équation (5.9). Il suffit alors de réinjecter ces paramètres dans le SATURNN initialisé à l'étape 1 afin de disposer d'un SATURNN entraîné $\tilde{\Phi}^{\text{LR}}(x, \tilde{\theta}^{\text{KLR}})$ pour lequel la règle de décision issue est interprétable et unique. Les codes pour pouvoir entraîner la KLR sont disponibles sur le Github[2].

Algorithme 2 Apprentissage de $\delta^{\text{KLR}}(x, \alpha)$

Entrées $X \in \mathbb{R}^{N \times d}$, $Y = \{0, 1\}^N$, p

1: Initialisation du SATURNN : $\theta^{(0)}$ ▷ Hypothèse 1
2: Normalisation des données : $X \to \tilde{X} \in \mathcal{B}_2^d(0, r)$, $r > 0$ ▷ Équation (3.19)
3: Transformation non linéaire des données : $\tilde{X} \mapsto \mathcal{K}_0(\tilde{X})$ ▷ Équation (5.5)
4: Entraînement de la LR : estimation de $\hat{\alpha}^{\text{KLR}}$ ▷ Équation (5.7)
5: Calcul de $\tilde{\theta}^{\text{KLR}}$: $\tilde{\theta}^{\text{KLR}} = \sum_{j=1}^{N} \hat{\alpha}_j^{\text{KLR}} g_0(\tilde{x}^{(j)})$ ▷ Équation (5.9)
6: Injecter $\tilde{\theta}^{\text{KLR}}$ dans le SATURNN : $\tilde{\Phi}^{\text{KLR}}(\tilde{X}, \tilde{\theta}^{\text{KLR}})$ ▷ Équation (5.10)

5.3 Équivalence du SATURNN avec une Régression Logistique à Noyau Déterministe

Tout comme la LR PSI LIN $\delta^{\text{LR PSI LIN}}(x, \eta)$ (4.24), la KLR $\delta^{\text{KLR}}(x, \alpha)$ (5.6) dépend des initialisations du SATURNN approximé : la règle de décision qui en découle est unique conditionnellement à l'échantillon d'apprentissage mais aussi aux initialisations $\theta^{(0)}$. Pour ces deux méthodes d'approximation, une transformation non linéaire est appliquée aux données avant d'apprendre la Régression Logistique. Les fonctions $g_0(x)$ (4.23) et $\kappa_0(x, \tilde{x})$ (5.4) pour $(x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d$ découlent directement de l'architecture du SATURNN et sont dépendantes de son processus d'entraînement au travers de $\theta^{(0)}$. Dans cette section, nous introduisons un nouveau noyau découlant directement de l'architecture du SATURNN et indépendant de ses initialisations.

5.3.1 Étude de $\kappa_0(x, \tilde{x})$

Nous nous concentrons dans un premier temps sur la fonction noyau $\kappa_0(x, \tilde{x})$ définie par l'équation (5.4). Le lemme 3 établit que cette fonction est d'espérance finie.

Lemme 3 (Espérance de $\kappa_0(x, \tilde{x})$)**.**
Soit la fonction $(x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \mapsto \kappa_0(x, \tilde{x}) \in \mathbb{R}$ *définie à l'équation (5.4). Cette fonction dépend des paramètres initialisés du SATURNN que nous souhaitons approximer par la KLR. Nous supposons que* $\theta^{(0)} = [\beta_0^{(0)}, \beta_1^{(0)}, \ldots, \beta_p^{(0)}, b_1^{(0)}, \ldots, b_p^{(0)}] \in \mathbb{R}^{2p+1}$ *respecte l'Hy-*

1. Méthode *LogisticRegression* dans la librairie Scikit-Learn [Pedregosa *et al.*, 2011]
2. Marie Guyomard - Dépôt SATURNN : https://github.com/GuyomardMarie/SATURNN

pothèse 1. L'espérance de $\kappa_0(x, \tilde{x})$ est finie :

$$\mathbb{E}(\kappa_0(x,\tilde{x})) = \frac{1}{p} + \frac{r^2}{6} + \frac{1}{4rd}\sum_{i=1}^{d} 2r(x_i\tilde{x}_i + 1) - |x_i - \tilde{x}_i| + \frac{1}{6}|x_i - \tilde{x}_i|^3. \tag{5.11}$$

Éléments de preuve.
La démonstration détaillée se trouve en Annexe D.1. Nous supposons que le vecteur de paramètres initialisés $\theta^{(0)}$ respecte l'Hypothèse 1, à savoir pour tout $k \in \{1, \dots, p\}$, $\beta_k^{(0)} \sim \mathcal{N}(0,1)$ et $b_k^{(0)} \sim \mathcal{U}[-r, r]$, avec $r > 0$ le rayon de la boule ouverte sur laquelle sont définies les variables descriptives $x \in \mathcal{B}_2^d(0, r)$. De plus, rappelons que les paramètres fixés du SATURNN sont initialisés selon certaines distributions : $s_k \in \{-1, 1\} \sim \mathcal{B}(1/2)$ et $\upsilon(k) \sim \mathcal{U}[\![1, d]\!]$, pour tout $k \in \{1, \dots, p\}$. Soit $\kappa_0 : (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ l'application suivante :

$$\kappa_0(x,\tilde{x}) = \frac{1}{p}\left[1 + \sum_{k=1}^{p} \phi\left(s_k x_{\upsilon(k)} + b_k^{(0)}\right)\phi\left(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}\right) + {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}}\right],$$

avec $\phi(\cdot) = \max\{0, \cdot\}$ l'activation ReLU.

Pour calculer l'espérance de l'application noyau $\mathbb{E}(\kappa_0(x, \tilde{x}))$, nous utilisons dans un premier temps les hypothèses $s_k \in \{-1, 1\} \sim \mathcal{B}(1/2)$ et $\upsilon(k) \sim \mathcal{U}[\![1, d]\!]$ pour tout $k \in \{1, \dots, p\}$. De plus, nous avons supposé les coefficients $\beta_k^{(0)} \sim \mathcal{N}(0, 1)$, pour tout $k \in \{1, \dots, p\}$, nous savons alors que $\mathbb{E}({\beta_k^{(0)}}^2) = 1$. Nous avons alors :

$$\begin{aligned}
\mathbb{E}(\kappa_0(x,\tilde{x})) = \frac{1}{p} + \frac{1}{2pd}\sum_{k=1}^{p}\sum_{i=1}^{d} &\mathbb{E}\left(\left[-x_i + b_k^{(0)}\right]\left[-\tilde{x}_i + b_k^{(0)}\right]\mathbb{1}_{\{b_k^{(0)} > x_i, b_k^{(0)} > \tilde{x}_i\}}\right) \\
&+ \mathbb{E}\left(\left[x_i + b_k^{(0)}\right]\left[\tilde{x}_i + b_k^{(0)}\right]\mathbb{1}_{\{b_k^{(0)} > -x_i, b_k^{(0)} > -\tilde{x}_i\}}\right) \\
&+ \mathbb{E}\left(\mathbb{1}_{\{b_k^{(0)} > x_i\}}\mathbb{1}_{\{b_k^{(0)} > \tilde{x}_i\}}\right) + \mathbb{E}\left(\mathbb{1}_{\{b_k^{(0)} > -x_i\}}\mathbb{1}_{\{b_k^{(0)} > -\tilde{x}_i\}}\right).
\end{aligned}$$

Dans un second temps, nous avons supposé pour tout $k \in \{1, \dots, p\}$, $b_k^{(0)} \sim \mathcal{U}[-r, +r]$, nous savons de ce fait que la fonction de densité de probabilité de b est égale à $\mathrm{f}_b(t) = \frac{1}{2r}$.

$$\begin{aligned}
\mathbb{E}(\kappa_0(x,\tilde{x})) &= \frac{1}{p} + \frac{1}{2pd}\sum_{k=1}^{p}\sum_{i=1}^{d}\int_{-r}^{r}[-x_i + t][-\tilde{x}_i + t]\mathrm{f}_b(t)\mathbb{1}_{\{t > x_i, t > \tilde{x}_i\}}\,\mathrm{d}t \\
&\qquad + \int_{-r}^{r}[x_i + t][\tilde{x}_i + t]\mathrm{f}_b(t)\mathbb{1}_{\{t > -x_i, t > -\tilde{x}_i\}}\,\mathrm{d}t \\
&\qquad + \int_{-r}^{r}\mathrm{f}_b(t)\mathbb{1}_{\{t > x_i, t > \tilde{x}_i\}}\,\mathrm{d}t + \int_{-r}^{r}\mathrm{f}_b(t)\mathbb{1}_{\{t > -x_i, t > -\tilde{x}_i\}}\,\mathrm{d}t \\
&= \frac{1}{p} + \frac{r^2}{6} + \frac{1}{4rd}\sum_{i=1}^{d}\Big[2r(x_i\tilde{x}_i + 1) + \underline{m}_i - \overline{m}_i - x_i\tilde{x}_i(\overline{m}_i - \underline{m}_i) \\
&\qquad + \frac{1}{2}(x_i + \tilde{x}_i)(\overline{m}_i^2 - \underline{m}_i^2) - \frac{1}{3}(\overline{m}_i^3 - \underline{m}_i^3)\Big],
\end{aligned}$$

avec les notations $\underline{m}_i = \min(x_i, \tilde{x}_i)$ et $\overline{m}_i = \max(x_i, \tilde{x}_i)$. Puisque $\max(a,b) = \frac{a+b+|a-b|}{2}$ et $\min(a,b) = \frac{a+b-|a-b|}{2}$ pour $a, b \in \mathbb{R}^2$, nous obtenons finalement :

$$\mathbb{E}(\kappa_0(x,\tilde{x})) = \frac{1}{p} + \frac{r^2}{6} + \frac{1}{4rd}\sum_{i=1}^{d} 2r(x_i\tilde{x}_i + 1) - |x_i - \tilde{x}_i| + \frac{1}{6}|x_i - \tilde{x}_i|^3. \tag{5.12}$$

□

Puisque $x \in \mathcal{B}_2^d(0,r)$, il est clair que l'espérance de $\kappa_0(x,\tilde{x})$ définie à l'équation (5.11) est finie. De plus, nous pouvons démontrer d'une part que la variance de $\kappa_0(x,\tilde{x})$ est bornée, et d'autre part qu'elle tend à être nulle lorsque le nombre p de neurones composant le SATURNN que l'on souhaite approximer augmente.

Lemme 4 (Variance de $\kappa_0(x,\tilde{x})$)**.**
Soit la fonction $(x,\tilde{x}) \in \mathbb{R}^d \mapsto \kappa_0(x,\tilde{x}) \in \mathbb{R}$ définie à l'équation (5.4). Cette fonction dépend des paramètres initialisés du SATURNN que nous souhaitons approximer par la KLR. Nous supposons que $\theta^{(0)} = [\beta_0^{(0)}, \beta_1^{(0)}, \dots, \beta_p^{(0)}, b_1^{(0)}, \dots, b_p^{(0)}] \in \mathbb{R}^{2p+1}$ respecte l'Hypothèse 1. La variance de $\kappa_0(x,\tilde{x})$ est bornée et tend vers 0 à mesure que la profondeur p du SATURNN approximé augmente :

$$\mathbb{V}(\kappa_0(x,\tilde{x})) \leq \frac{1}{p^2} + \frac{C}{p} = O\left(\frac{1}{p}\right), \tag{5.13}$$

avec $C \in \mathbb{R}$ une constante, indépendante de p telle que $C = 8r^4 + \frac{r^3}{3} + \frac{7r^2}{3} + 13$.

Éléments de preuve.
La démonstration détaillée se trouve en Annexe D.2. Nous supposons que le vecteur de paramètres initialisés $\theta^{(0)}$ respecte l'Hypothèse 1, à savoir pour tout $k \in \{1,\dots,p\}$, $\beta_k^{(0)} \sim \mathcal{N}(0,1)$ et $b_k^{(0)} \sim \mathcal{U}[-r,r]$, avec $r > 0$ le rayon de la boule ouverte sur laquelle sont définies les variables descriptives $x \in \mathcal{B}_2^d(0,r)$. De plus, rappelons que les paramètres fixés du SATURNN sont initialisés selon certaines distributions : $s_k \in \{-1,1\} \sim \mathcal{B}(1/2)$ et $\upsilon(k) \sim \mathcal{U}[\![1,d]\!]$, pour tout $k \in \{1,\dots,p\}$. Soit $\kappa_0 : (x,\tilde{x}) \in (\mathbb{R}^d, \mathbb{R}^d) \to \mathbb{R}$, l'application suivante :

$$\kappa_0(x,\tilde{x}) = \frac{1}{p}\left[1 + \sum_{k=1}^{p} \phi\left(s_k x_{\upsilon(k)} + b_k^{(0)}\right)\phi\left(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}\right) + {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}}\right],$$

avec $\phi(\cdot) = \max\{0, \cdot\}$ l'activation ReLU.

La variance de $\kappa_0(x,\tilde{x})$ s'écrit :

$$\mathbb{V}(\kappa_0(x,\tilde{x})) = \mathbb{E}(\kappa_0(x,\tilde{x})^2) - \mathbb{E}(\kappa_0(x,\tilde{x}))^2$$

Puisque $\mathbb{E}(\kappa_0(x,\tilde{x})) = 0$ (Proposition 3), nous avons finalement $\mathbb{V}(\kappa_0(x,\tilde{x})) = \mathbb{E}(\kappa_0(x,\tilde{x})^2)$. Dans un premier temps nous développons $\mathbb{E}(\kappa_0(x,\tilde{x})^2)$ et nous utilisons les hypothèses $s_k \in \{-1,1\} \sim \mathcal{B}(1/2)$ et $\upsilon(k) \sim \mathcal{U}[\![1,d]\!]$ pour tout $k \in \{1,\dots,p\}$ de la même manière que dans la preuve du Lemme 3. De plus, nous avons supposé les coefficients $\beta_k^{(0)} \sim \mathcal{N}(0,1)$, pour tout $k \in \{1,\dots,p\}$. Ainsi, nous savons que $\mathbb{E}({\beta_k^{(0)}}^2) = 1$ et $\mathbb{E}({\beta_k^{(0)}}^4) = 3$. Enfin, nous supposons conformément à l'Hypothèse 1 que

les biais b_k pour tout $k \in \{1, \dots, p\}$ sont indépendamment et identiquement distribués : pour tout $i, j \in \{1, \dots, p\}^2$ nous avons $\mathbb{E}(b_i) = \mathbb{E}(b_j)$, nous posons alors un b quelconque tiré selon une loi uniforme sur $[-r, r]$ et nous obtenons :

$$\begin{aligned}\mathbb{V}(\kappa_0(x, \tilde{x})) = \frac{1}{p^2} \sum_{i=1}^{d} \frac{1}{2pd} \Big[& \mathbb{E}\left(\phi(x_i + b)^2 \phi(\tilde{x}_i + b)^2 + \phi(-x_i + b)^2 \phi(-\tilde{x}_i + b)^2\right) \\ & + 3\mathbb{E}\left(\mathbb{1}_{\{x_i + b > 0\}} \mathbb{1}_{\{\tilde{x}_i + b > 0\}} + \beta_k^{(0)^4} \mathbb{1}_{\{-x_i + b > 0\}} \mathbb{1}_{\{-\tilde{x}_i + b > 0\}}\right)\Big] \\ + \frac{1}{pd} \Big[& 2\mathbb{E}\left(\phi(x_i + b)\phi(\tilde{x}_i + b) + \phi(-x_i + b)\phi(-\tilde{x}_i + b)\right) \\ & + \mathbb{E}\left(\mathbb{1}_{\{x_i + b > 0\}} \mathbb{1}_{\{\tilde{x}_i + b > 0\}} + \beta_k^{(0)^4} \mathbb{1}_{\{-x_i + b > 0\}} \mathbb{1}_{\{-\tilde{x}_i + b > 0\}}\right)\Big]\end{aligned}$$

Nous pouvons majorer cette équation puisque nous savons que le maximum atteint pour l'application ReLU est :

$$\max(\phi(x_i + b)) = \max(x_i + b) \leq r + b,$$

puisque les variables se situent dans la boule ouverte de rayon $r > 0$: $x \in \mathcal{B}_2^d(0, r)$. Ainsi, nous obtenons :

$$\mathbb{V}(\kappa_0(x, \tilde{x})) \leq \frac{1}{p^2} + \sum_{i=1}^{d} \frac{1}{2pd} \left[\mathbb{E}(2r^4 + 2r^2 b + 8r^2 b^2 + 2r^2 b + 2b^2) + 6\right] + \frac{2}{pd} \left[\mathbb{E}(2r^2 + 2b^2) + 2\right]$$

Enfin, puisque pour tout $k \in \{1, \dots, p\}$, $b_k^{(0)} \sim \mathcal{U}[-r, r]$, nous avons $\mathbb{E}(b) = 0$ et $\mathbb{E}(b^2) = \mathbb{V}(b) = \frac{r^2}{3}$. Nous pouvons alors majorer la variance par :

$$\mathbb{V}(\kappa_0(x, \tilde{x})) \leq \frac{1}{p^2} + \frac{1}{p}\left[8r^4 + \frac{r^3}{3} + \frac{7r^2}{3} + 13\right] \tag{5.14}$$

$$= O\left(\frac{1}{p}\right). \tag{5.15}$$

À mesure que $p \to \infty$, la variance de $\kappa_0(x, \tilde{x})$ tend à être nulle. □

5.3.2 Modélisation de la Régression Logistique à Noyau Déterministe

Dans la section précédente nous avons montré que l'espérance de $\kappa_0(x, \tilde{x})$ définie par l'équation (5.4) est finie (Lemme 3) et que sa variance est bornée et tend à être nulle lorsque p augmente (Lemme 4). Nous pouvons remarquer que $\kappa_0(x, \tilde{x})$ se définit comme une moyenne empirique. Ainsi, d'après la Loi Forte des Grands Nombres [Kolmogorov et Bharucha-Reid, 2018] nous pouvons affirmer que $\kappa_0(x, \tilde{x})$ converge presque sûrement vers son espérance :

$$\mathbb{P}\left((x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \,\middle|\, \lim_{p \to \infty} \kappa_0(x, \tilde{x}) = \mathbb{E}(\kappa_0(x, \tilde{x}))\right) = 1. \tag{5.16}$$

Puisque pour un p suffisamment grand, $\kappa_0(x, \tilde{x})$ devient équivalent à $\mathbb{E}(\kappa_0(x, \tilde{x}))$, un nouveau noyau peut être défini :

$$\begin{aligned} \kappa : \mathbb{R}^d \times \mathbb{R}^d &\to \mathbb{R} \\ (x, \tilde{x}) &\mapsto \mathbb{E}(\kappa_0(x, \tilde{x})), \end{aligned} \tag{5.17}$$

avec $\mathbb{E}(\kappa_0(x, \tilde{x}))$ défini à l'équation (5.11). Puisque les deux noyaux $\kappa_0(x, \tilde{x})$ et $\kappa(x, \tilde{x})$ sont équivalents pour un p suffisamment grand, nous proposons une nouvelle LR appliquée au noyau $\kappa(x, \tilde{x})$ notée $\delta^{\text{EKLR}}(x, \alpha)$ équivalente à $\delta^{\text{KLR}}(x, \alpha)$.

Proposition 2 (Régression Logistique à Noyau Déterministe)**.**
Soit $\delta^{KLR}(x, \alpha)$ la Régression Logistique appliquée aux données préalablement transformées par le noyau $\kappa_0 : (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ défini à l'équation (5.4). Nous savons que l'espérance de $\kappa_0(x, \tilde{x})$ est finie (Lemme 3) et que sa variance est bornée et tend à être nulle lorsque p est suffisamment grand (Lemme 4). Ainsi, d'après la Loi Forte des Grands Nombres nous pouvons établir que lorsque p est suffisamment grand ($p \to \infty$), la KLR $\delta^{KLR}(x, \alpha)$ converge presque sûrement vers une Régression Logistique à noyau déterministe (EKLR) définie par :

$$\delta^{EKLR}(x, \alpha) = \sigma\left(\sum_{j=1}^{N} \alpha_j \kappa(x, x^{(j)})\right) = \sigma(K(x)^T \alpha), \tag{5.18}$$

avec la matrice noyau

$$K(x^{(j)}) = \left(\kappa(x^{(1)}, x^{(j)}), \ldots, \kappa(x^{(N)}, x^{(j)})\right)^T,$$

et $\kappa : (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ l'application noyau définie par :

$$\kappa(x, \tilde{x}) = \frac{1}{p} + \frac{r^2}{6} + \frac{1}{4rd} \sum_{i=1}^{d} \varrho(x_i, \tilde{x}_i), \tag{5.19}$$

tel que

$$\varrho(x_i, \tilde{x}_i) = 2r(x_i \tilde{x}_i + 1) - |x_i - \tilde{x}_i| + \frac{1}{6}|x_i - \tilde{x}_i|^3. \tag{5.20}$$

Démonstration.
Puisque $\delta^{\text{EKLR}}(x, \alpha)$ est construite à partir de la Loi Forte des Grands Nombres [Kolmogorov et Bharucha-Reid, 2018], lorsque $p \to \infty$, il est évident que les Régressions Logistiques à Noyau $\delta^{\text{KLR}}(x, \alpha)$ et $\delta^{\text{EKLR}}(x, \alpha)$ sont équivalentes. □

Tout comme la LR PSI LIN et la KLR, l'EKLR $\delta^{\text{EKLR}}(x, \alpha)$ applique la Régression Logistique aux variables transformées préalablement par une fonction non linéaire découlant directement de l'architecture du SATURNN. Contrairement à la KLR, l'EKLR applique un noyau $\kappa(x, \tilde{x})$ totalement indépendant du processus d'apprentissage du SATURNN, et plus précisément de ses paramètres initialisés $\theta^{(0)}$. De plus, le noyau déterministe transforme les variables descriptives $x \mapsto (\kappa(x^{(i)}, x))_{i=1}^N$ de manière univariée : le partitionnement de chaque variable est indépendant des autres variables. En effet, l'application $\rho(x, \tilde{x})$ (5.20) composant le noyau $\kappa(x, \tilde{x})$ se modélise comme une somme additive de splines univariées.

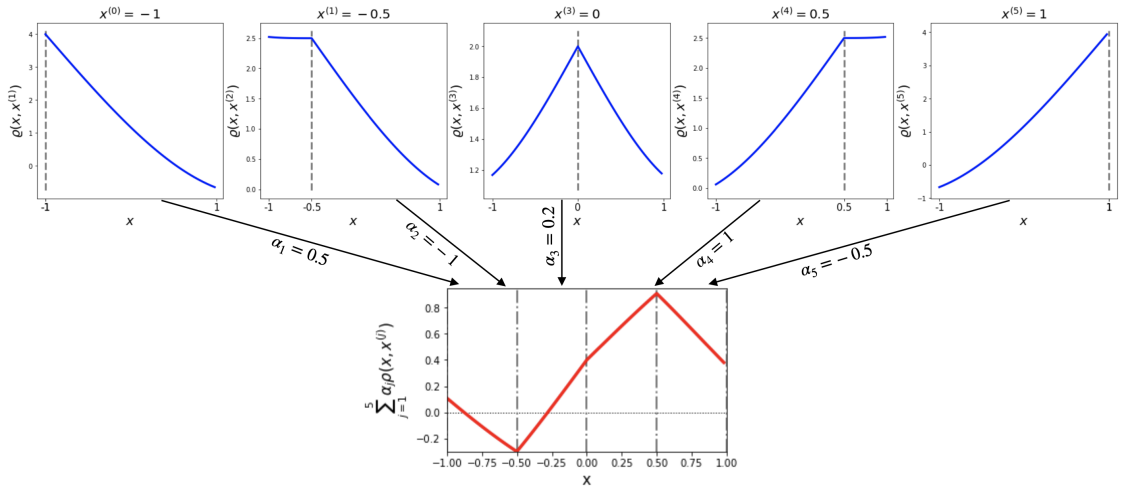

FIGURE 5.1 – Illustration de la transformation opérée par le noyau $\kappa(x, \tilde{x})$ (5.19) pour $x \in \mathcal{B}_2(0, 1)$ et 5 échantillons fixés : $x^{(0)} = -1,\ x^{(1)} = -0.5,\ x^{(2)} = 0,\ x^{(3)} = 0.5$ et $x^{(4)} = 1$. Sur la partie haute de la Figure (courbes en bleu) nous retrouvons les transformations opérées par l'application $\varrho(x, \tilde{x})$ (5.20) pour chacun des échantillons fixés $\tilde{x}$. Enfin la partie basse (courbe rouge) affiche la combinaison linéaire de ces transformation par le vecteur $\alpha = [\alpha_1 = 0.5,\ \alpha_2 = -1,\ \alpha_3 = 0.2,\ \alpha_4 = 1,\ \alpha_5 = -0.5]$ et donc la spline univariée résultante de l'application noyau $\kappa(x, \tilde{x})$.

La Figure 5.1 illustre le partitionnement induit par le noyau $\kappa(x, \tilde{x})$ sur un jeu de données composé de 5 échantillons d'apprentissage et d'une seule variable x. Nous pouvons analyser la transformation opérée par l'application $\varrho(x, \tilde{x})$ en nous intéressant aux courbes bleues. Nous supposons des valeurs de $\bar{x}$ fixées pour les 5 échantillons : $x^{(0)} = -1,\ x^{(1)} = -0.5,\ x^{(2)} = 0,\ x^{(3)} = 0.5$ et $x^{(4)} = 1$. Nous pouvons constater que la fonction $\varrho(x, \tilde{x})$ modélise des effets non linéaires. En fonction de la valeur prise $\bar{x}$, le degré de non-linéarité varie. En réalisant la combinaison linéaire de ces transformations, nous obtenons alors la variable x transformée par rapport à l'ensemble des valeurs prises dans la base de données. Le vecteur α dans (5.18) peut alors être interprété comme un coefficient d'importance de chacun des échantillons. Ainsi, l'EKLR applique la LR à des données transformées non linéairement par le noyau $x \mapsto (\kappa(x^{(i)}, x))_{i=1}^{N}$.

Puisque pour un p suffisamment grand (i) les fonctions de coût minimisées par le SATURNN et LR PSI LIN sont équivalentes après entraînement (Théorème 3), (ii) la LR PSI LIN est équivalente à la KLR (Proposition 1) et (iii) le noyau κ_0 converge asymptotiquement vers son espérance $\mathbb{E}(\kappa_0(x, \tilde{x}))$ (Proposition 2), il est alors équivalent d'entraîner le SATURNN ou l'EKLR. Ainsi, le SATURNN composé d'un grand nombre p de neurones peut être correctement approximé par l'EKLR, une Régression Logistique appliquée à un noyau directement inspiré de son architecture bien que totalement indépendant de son processus d'apprentissage.

5.3.3 Apprentissage de la Régression Logistique à Noyau Déterministe

Afin d'estimer la règle de décision de l'EKLR, il convient d'estimer les paramètres $\hat{\alpha}^{\text{EKLR}} \in \mathbb{R}^N$ en minimisant la fonction de coût $\mathcal{L}^{\text{EKLR}}$:

$$\hat{\alpha}^{\text{EKLR}} = \underset{\alpha \in \mathcal{B}_2^N(0,R)}{\arg\min}\ \mathcal{L}^{\text{EKLR}}(\alpha, \mathcal{D}), \tag{5.21}$$

définie par

$$\mathcal{L}^{\text{EKLR}}(\alpha, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} L\left(\delta^{\text{EKLR}}(x^{(i)}, \alpha), y^{(i)}\right), \tag{5.22}$$

avec L l'Entropie Croisée Binaire (2.9). La régularisation ℓ_2 ajoutée au problème d'optimisation de l'EKLR (5.21) assure que le processus d'optimisation converge vers une solution unique. Cette unicité reste néanmoins conditionnelle à l'échantillon d'apprentissage sur lequel le noyau est calculé. Néanmoins, contrairement à la KLR, le noyau $\kappa : (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ est indépendant des initialisations du SATURNN $\theta^{(0)}$. Ainsi, l'apprentissage de l'EKLR et par conséquent l'unicité de la règle de décision qui en découle, sont totalement indépendants des initialisations du SATURNN.

Le partitionnement opéré par le noyau $\kappa : (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (5.19) est univarié. En réinjectant les paramètres estimés par l'EKLR $\hat{\alpha}^{\text{EKLR}}$ (5.18) dans la règle de décision $\delta^{\text{EKLR}}(x, \alpha)$ (5.18), nous pouvons la réécrire :

$$\begin{aligned}
\delta^{\text{EKLR}}(x, \hat{\alpha}^{\text{EKLR}}) &= \sigma\left(\sum_{j=1}^{N} \hat{\alpha}_j^{\text{EKLR}} \left(\frac{1}{p} + \frac{r^2}{6} + \frac{1}{4rd}\sum_{i=1}^{d} \varrho(x_i^{(j)}, x_i)\right)\right) \\
&= \sigma\left(\left(\frac{1}{p} + \frac{r^2}{6}\right)\sum_{j=1}^{N} \hat{\alpha}_j^{\text{EKLR}} + \frac{1}{4rd}\sum_{i=1}^{d}\sum_{j=1}^{N} \hat{\alpha}_j^{\text{EKLR}} \varrho(x_i^{(j)}, x_i)\right).
\end{aligned} \tag{5.23}$$

Ainsi, la règle de décision estimée par l'EKLR peut se réécrire comme celles estimées par les GAM. En effet, la sigmoïde $\sigma(\cdot)$ (2.4) est appliquée à une somme additive de splines univariées où

$$\hat{\beta}_0 = \left(\frac{1}{p} + \frac{r^2}{6}\right)\sum_{i=1}^{N} \hat{\alpha}_j^{\text{EKLR}}, \tag{5.24}$$

est le biais et les splines univariées notées $f_i(x_i)$, pour tout $i \in \{1, \ldots, d\}$ pour les GAM ou le SATURNN se définissent par :

$$\hat{\beta}_i(x_i) = \frac{1}{4rd}\sum_{j=1}^{N} \hat{\alpha}_j^{\text{EKLR}} \varrho(x_i^{(j)}, x_i), \tag{5.25}$$

avec $\varrho(\cdot)$ défini à l'équation (5.20) et illustré par la Figure 5.1. Ainsi, l'EKLR estime une règle de décision modélisant une segmentation univariée des variables d'entrée. Ce découpage de l'espace est donc unique ($\hat{\alpha}^{\text{EKLR}}$ unique), interprétable et contrairement à celui opéré par la KLR indépendant des initialisations du SATURNN que l'on souhaite approximer. Conformément à la Définition 2, l'EKLR est une méthode de classification non linéaire explicable. En effet, en approximant le SATURNN par l'EKLR définie à l'équation

(5.18), nous obtenons une règle de décision interprétable et unique conditionnellement à l'échantillon d'apprentissage :

$$\tilde{\Phi}^{\mathrm{EKLR}}(x, \hat{\alpha}^{\mathrm{EKLR}}) = \sigma\left(\hat{\beta}_0 + \frac{1}{4rd}\sum_{i=1}^{d}\hat{\beta}_i(x_i)\right), \tag{5.26}$$

avec $\hat{\beta}_0$ et $\hat{\beta}_i(x_i)$ défini par les équations (5.24) et (5.25).

Dans l'Algorithme 3, les grandes étapes de l'entraînement de l'EKLR sont détaillées. Tout comme pour la LR PSI LIN et la KLR, nous normalisons les données avant de les transformer par l'application non linéaire $x \mapsto \left(\kappa(x^{(i)}, x)\right)_{i=1}^{N}$ (5.19). Néanmoins, contrairement à la LR PSI LIN et la KLR, le noyau étant indépendant des initialisation $\theta^{(0)}$ du SATURNN que nous souhaitons approximer, il n'est pas nécessaire pour entraîner l'EKLR d'initialiser les paramètres du SATURNN. La Régression Logistique est ensuite entraînée sur les données transformées, à l'aide de la méthode *LogisticRegression* [3] disponible dans la librairie Scikit-Learn. Une fois que les paramètres $\hat{\alpha}^{\mathrm{EKLR}}$ (5.21) sont appris, nous disposons directement de la règle de décision du SATURNN que nous souhaitons approximer. Alors que pour la LR PSI LIN et la KLR il était nécessaire de calculer les paramètres du SATURNN approximé, pour l'EKLR, il suffit de réécrire la règle de décision comme fonction de $\hat{\beta}_0$ (5.24) et des splines univariées estimées $\hat{\beta}_i(x_i)$, pour tout $i \in \{1, \ldots, d\}$ (5.25) pour obtenir le SATURNN entraîné $\tilde{\Phi}^{\mathrm{EKLR}}(x, \hat{\alpha}^{\mathrm{EKLR}})$ défini à l'équation (5.26). Les codes pour pouvoir entraîner l'EKLR sont disponibles sur le Github [4].

Algorithme 3 Apprentissage de $\delta^{\mathrm{EKLR}}(x, \alpha)$

Entrées $X \in \mathbb{R}^{N\times d}$, $Y = \{0,1\}^N$, p

1: Normalisation des données : $X \to \tilde{X} \in \mathcal{B}_2^d(0, r)$, $r > 0$ ▷ Équation 3.19
2: Transformation non linéaire des données : $\tilde{X} \mapsto \mathcal{K}(\tilde{X})$ ▷ Équation 5.19
3: Entraînement de la LR : estimation de $\hat{\alpha}^{\mathrm{EKLR}}$ ▷ Équation 5.21
4: Calcul de $\tilde{\Phi}^{\mathrm{EKLR}}(x, \hat{\alpha}^{\mathrm{EKLR}})$ ▷ Équation 5.26

5.4 Résultats numériques sur données simulées

Dans cette section nous présentons des résultats numériques sur les bases de données simulées Circle et Gaussienne présentées précédemment. Dans un premier temps nous étudions les résultats théoriques d'approximation par les Régressions Logistiques à Noyau $\delta^{\mathrm{KLR}}(x, \alpha)$ (5.6) et $\delta^{\mathrm{EKLR}}(x, \alpha)$ (5.18) en fonction du nombre p de neurones composant le SATURNN que nous souhaitons approximer. Ensuite, nous comparons ces contributions aux Machines à Vecteurs de Support (SVMs), méthodes à noyau traditionnelles, avant de conclure quant à l'unicité de leurs estimations. Puisque les conclusions tirées de ces expériences sur les deux bases de données sont identiques, nous présentons seulement les résultats issus du jeu de données Circle. Les résultats obtenus sur la base de données Gaussienne sont quant à eux disponibles en Annexe E.

3. Méthode *LogisticRegression* dans la librairie Scikit-Learn [Pedregosa *et al.*, 2011]
4. Marie Guyomard - Dépôt SATURNN : https://github.com/GuyomardMarie/SATURNN

5.4.1 Approximation du SATURNN par les Régressions Logistiques à Noyau

Dans ce chapitre, nous avons introduit deux Régression Logistiques à noyau pouvant approximer un SATURNN composé d'un grand nombre p de neurones. Deux noyaux découlant directement de l'architecture du SATURNN ont été proposés. Le premier $\kappa_0 : (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ défini à l'équation (5.4) est dépendant des initialisations $\theta^{(0)}$ du SATURNN dont il découle. En revanche, le second $\kappa : (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ défini à l'équation (5.19) est totalement indépendant du processus d'apprentissage du SATURNN. Nous avons dans un premier temps démontré que le noyau $\kappa_0(x, \tilde{x})$ converge asymptotiquement vers $\kappa(x, \tilde{x})$ à mesure que p augmente. Sur la Figure 5.2, nous affichons les moyennes des normes de Frobenius entre les noyaux $||\kappa_0(x, \tilde{x}) - \kappa(x, \tilde{x})||_F$, obtenues sur 5 différents sous-échantillons pour différentes valeurs de p. Nous pouvons constater d'une part qu'en moyenne la différence de valeurs entre les deux noyaux proposés diminue lorsque p augmente. D'autre part, nous pouvons remarquer que l'écart-type des normes obtenues se réduit à mesure que nous considérons un p grand, ainsi la différence obtenue entre les deux noyaux varie beaucoup moins autour de sa valeur moyenne. Les matrices noyau tendent à devenir égales à mesure que p tend à être grand.

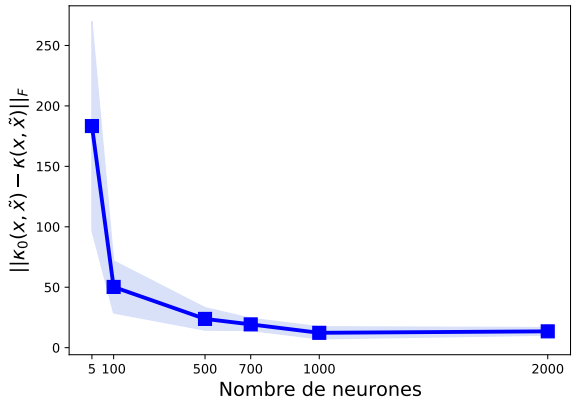

FIGURE 5.2 – Normes de Frobenius moyennes entre les noyaux κ_0 et κ ($||\kappa_0(x, \tilde{x}) - \kappa(x, \tilde{x})||_F$) obtenues sur 5 sous-échantillons de la base de données Circle pour différentes valeurs de p.

Ensuite, nous avons alors établi par la Proposition 2 que considérer la KLR (5.6) transformant les données par l'application du noyau $\kappa_0(x, \tilde{x})$ ou l'EKLR (5.18) appliquant le noyau $\kappa(x, \tilde{x})$ est équivalent pour un p suffisamment grand.

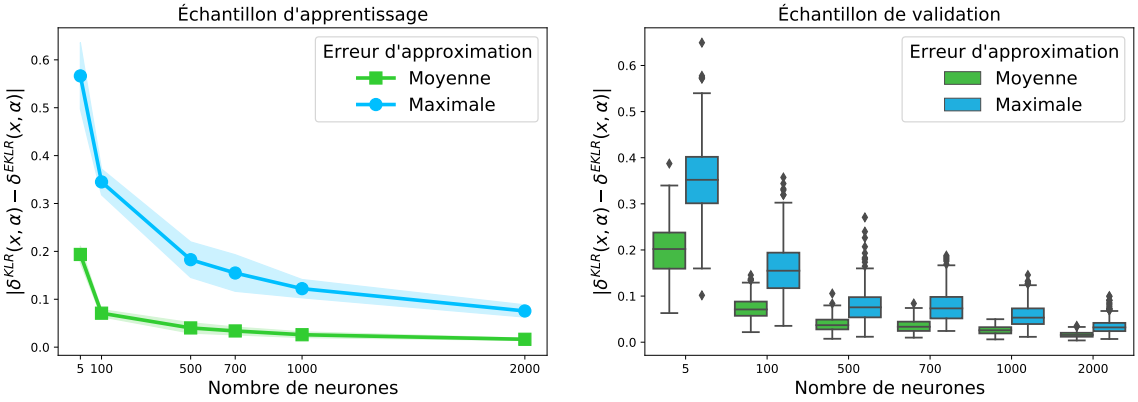

FIGURE 5.3 – Erreurs d'approximation $|\delta^{\text{KLR}}(x, \alpha) - \delta^{\text{EKLR}}(x, \alpha)|$ moyennes (courbes vertes) et maximales (courbes bleues) pour différentes valeurs de p. Ces erreurs ont été calculées par Validation Croisée 5−folds sur les échantillons d'apprentissage (Gauche) et de validation (Droite).

La Figure 5.3 présente les moyennes ainsi que les écart-types des erreurs d'approximation de la KLR par l'EKLR moyennes (courbes vertes) et maximales (courbes bleues) obtenues par Validation Croisée 5−folds pour différentes valeurs de p. Nous pouvons nous rendre compte qu'à la fois les erreurs d'approximation $|\delta^{\text{KLR}}(x, \alpha) - \delta^{\text{EKLR}}(x, \alpha)|$ moyennes et maximales tendent à être nulles à mesure que p augmente sur les échantillons d'apprentissage (Figure 5.3 - Gauche) et de validation (Figure 5.3 - Droite). De plus, sur les échantillons de validation, les boîtes à moustaches moyennes (en vert) et maximales (en bleu) ont tendance à se rétrécir autour de leur médiane. Ainsi, plus p augmente, plus la différence de prédictions obtenue entre la KLR et l'EKLR devient stable, en plus de diminuer. La Proposition 2 est vérifiée, la KLR $\delta^{\text{KLR}}(x, \alpha)$ et l'EKLR $\delta^{\text{EKLR}}(x, \alpha)$ sont équivalentes pour un p suffisamment grand.

Puisque (i) l'EKLR est équivalente à la KLR lorsque p est suffisamment grand (Proposition 2), (ii) la KLR est équivalente à la LR PSI LIN (Proposition 1) et (iii) il est équivalent d'entraîner la LR PSI LIN ou un SATURNN profond (Théorème 3), nous avons proposé d'approximer le SATURNN par l'EKLR. La Figure 5.4-Haute présente les moyennes ainsi que les écart-types des erreurs d'approximation moyennes du SATURNN $\Phi^{\text{SATURNN}}(x, \theta)$ par la KLR $\delta^{\text{KLR}}(x, \eta)$ (traits continus oranges) obtenues par Validation Croisée 5−folds. En pointillé nous retrouvons les moyennes des erreurs $|\Phi^{\text{SATURNN}}(x, \theta) - \delta^{\text{KLR}}(x, \eta)|$ maximales sur les échantillons d'apprentissage (Gauche) et de validation (Droite). Nous pouvons constater qu'à mesure que p augmente, les erreurs d'approximation moyennes et maximales obtenues sur 5−folds à la fois sur les échantillons d'apprentissage (Gauche) et de validation (Droite) diminuent. Ainsi l'équivalence entre entraîner un SATURNN composé d'un grand nombre p de neurones ou la KLR est vérifiée. Lorsque nous nous intéressons à l'approximation du SATURNN par l'EKLR $\delta^{\text{EKLR}}(x, \eta)$, sur la Figure 5.4-Basse, le constat est le même. En moyenne les erreurs d'approximation $|\Phi^{\text{SATURNN}}(x, \theta) - \delta^{\text{EKLR}}(x, \eta)|$ moyennes (traits continus roses) et maximales (pointillés) diminuent à mesure que p augmente.

Néanmoins l'approximation par EKLR est plus rapidement correcte (pour une plus petite

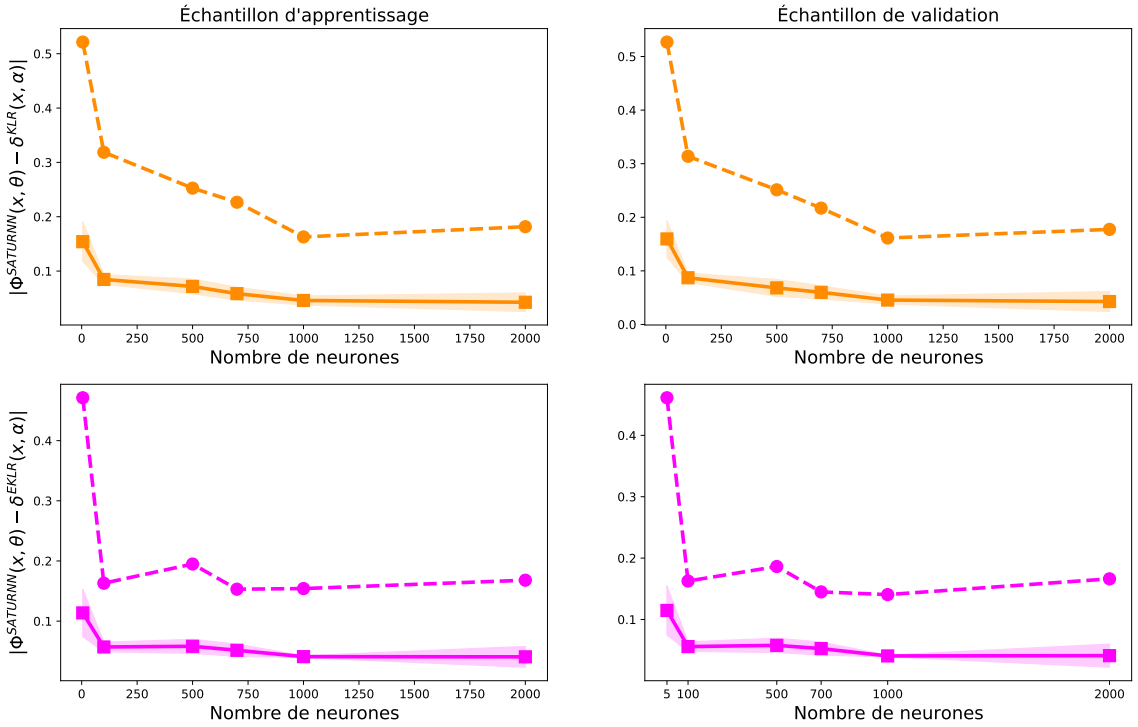

FIGURE 5.4 – Haute - Erreurs d'approximation $|\Phi^{\text{SATURNN}}(x,\theta) - \delta^{\text{KLR}}(x,\eta)|$ moyennes (trait continu) et maximales (en pointillé) pour différentes valeurs de p. Basse - Erreurs d'approximation $|\Phi^{\text{SATURNN}}(x,\theta) - \delta^{\text{EKLR}}(x,\eta)|$ moyennes (trait continu) et maximales (en pointillé) pour différentes valeurs de p. Ces erreurs ont été calculées par Validation Croisée 5−folds sur les échantillons d'apprentissage (Gauche) et de validation (Droite).

valeur de p) que par KLR. En effet, nous pouvons visualiser que pour $p = 100$ l'erreur d'approximation maximale du SATURNN par l'EKLR est de l'ordre de 0.15 en moyenne contre 0.32 en moyenne pour la KLR à la fois sur les échantillons d'apprentissage et de validation. Lorsque nous nous intéressons aux règles de décision obtenues pour différentes valeurs de p par les EKLR (en orange) et les EKLR (en rose) sur la Figure 5.5 ci-dessus, nous pouvons constater que celles issues de l'EKLR sont beaucoup plus performantes dès un petit p contrairement à celles de la KLR qui nécessite $p >> 100$. Le noyau de la KLR étant dépendant des initialisations du SATURNN que nous souhaitons approximer, la règle de décision qui en découle est en partie aléatoire expliquant la nécessité de considérer un très grand nombre de neurones p pour que l'approximation soit correcte. Pour l'EKLR, le noyau est totalement déterministe et est donc moins volatile car soumis à aucune forme d'aléatoire.

5.4.2 Comparaison des KLRs et EKLRs aux méthodes à noyau traditionnelles

Les Machines à Vecteurs de Support

Les méthodes à noyau sont souvent employées pour les problèmes de classification non linéaires [Bermudez *et al.*, 2015]. Les Machines à Vecteur de Support (SVMs) ont été conçues dans ce but. L'idée des SVMs est d'appliquer une transformation noyau non linéaire aux

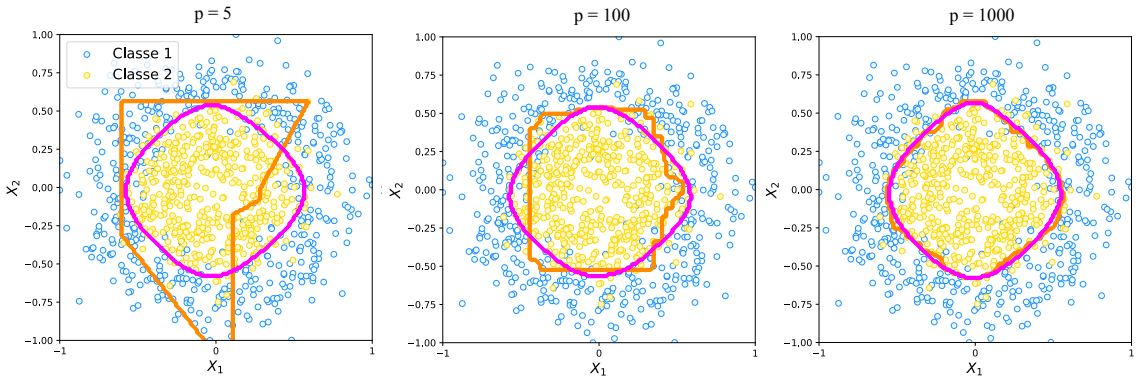

FIGURE 5.5 – Règles de décision des KLRs (orange) et EKLRs (rose) obtenues sur la base de données Circle pour différentes valeurs de p ($p = 5$ - Gauche, $p = 100$ - Milieu et $p = 1000$ - Droite).

données afin de trouver un hyperplan optimal permettant de séparer les différentes classes d'échantillons.

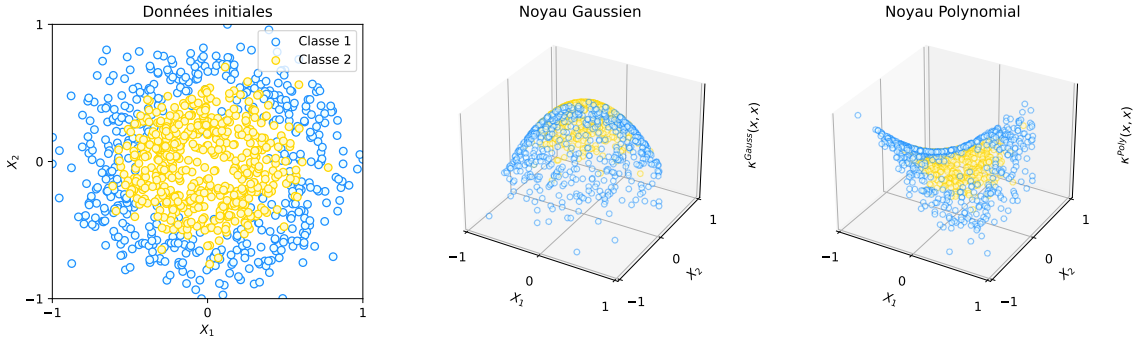

FIGURE 5.6 – Transformation de la base de données Circle (Gauche) par un noyau Gaussien avec $\gamma = 1$ (Milieu) et Polynomial d'ordre $d = 3$ et $\gamma = 1,\ r = 1$ (Droite).

Sur la Figure 5.6, nous pouvons visualiser la transformation de la base de données Circle par des noyaux traditionnels tels que la fonction de base radiale du noyau gaussien (RBF) $\kappa^{\text{Gauss}}(x, \tilde{x}) = \exp(-\gamma||x - \tilde{x}||^2),\ \gamma > 0$ et la transformation Polynomiale d'ordre $d = 3$, tel que $\kappa^{\text{Poly}}(x, \tilde{x}) = \tanh(\gamma x^T \tilde{x} + r)^d,\ \gamma > 0$ et $r \in \mathbb{R}$.

Une fois les données transformées, une règle de décision de classification linéaire est estimée. Pour ce faire, plusieurs noyaux peuvent être considérés et auront un impact directement sur la performance de la règle de décision du SVM.

Comparaison des performances

Nous avons décidé dans cette section de comparer la performance de nos méthodes d'approximation du SATURNN par la KLR et l'EKLR à différents SVMs. Le SATURNN que nous souhaitons approximer a été entraîné avec $p = 50\ 000$ neurones. Les KLR et EKLR ont été entraînées avec différentes valeurs de régularisation ℓ_2 sur les paramètres $(\lambda = [0, 0.01, 10])$.

Le Tableau 5.1 résume les différentes performances prédictives (3.20) et AUC moyennes obtenues par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation pour les différentes méthodes testées. Nous pouvons constater que les performances prédictives des SVMs dépendent grandement du noyau choisi pour opérer la transformation des données. Pour des jeux de données hautement non linéaires telle que la base Circle, les EKLR et EKLR réussissent à modéliser les effets non linéaires tandis que les SVM polynomiaux ont plus de difficultés. En effet, les SVMs appliquant un noyau linéaire ou Polynomial de degrès 3 ne sont pas en mesure de différentier les deux classes (respectivement 50% et 54% de performances prédictives globales moyennes sur les échantillons de validation pour les deux méthodes). En revanche, les SVMs à noyau Gaussien obtiennent en moyenne 95% d'AUC test, soit du même ordre de grandeur que les méthodes à noyau proposées dans ce manuscrit. Les règles de décision issues des KLR et EKLR avec une petite régularisation ($\lambda = 0.01$) ainsi que du SVM avec noyau gaussien sont affichées à la Figure 5.7.

		Échantillon d'apprentissage		**Échantillon de Validation**	
		Perf. Globale	**AUC**	**Perf. Globale**	**AUC**
SATURNN		0.90 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.02)
KLR	$\lambda = 0$	0.89 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.01)
	$\lambda = 0.01$	0.89 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.01)
	$\lambda = 10$	0.88 (0.01)	0.95 (0.01)	0.87 (0.01)	0.94 (0.01)
EKLR	$\lambda = 0$	0.89 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.01)
	$\lambda = 0.01$	0.89 (0.01)	0.96 (0.01)	0.88 (0.01)	0.95 (0.01)
	$\lambda = 10$	0.89 (0.01)	0.95 (0.01)	0.87 (0.02)	0.94 (0.01)
SVM Linéaire		0.54 (0.03)	0.50 (0.01)	0.50 (0.06)	0.50 (0.01)
SVM Gaussien		0.90 (0.01)	0.96 (0.01)	0.88 (0.02)	0.95 (0.01)
SVM Polynomial $d = 3$		0.60 (0.05)	0.48 (0.05)	0.54 (0.06)	0.48 (0.04)

TABLE 5.1 – Performances prédictives et AUC moyennes (écart-types) obtenues par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation. Pour les KLRs et EKLRs, différentes régularisations ont été utilisées : $\lambda = [0, 0.01, 10]$. Nos contributions sont en bleu.

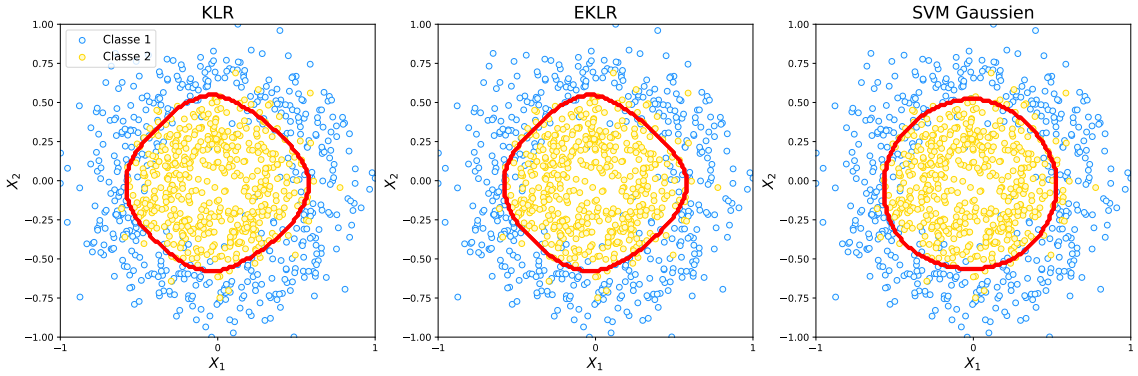

FIGURE 5.7 – Règles de Décision issues de la KLR (Gauche), l'EKLR (Milieu) et du SVM à noyau Gaussien (Droite). Les KLR et EKLR présentés sur cette figure sont ceux entraînés avec une petite régularisation $\lambda = 0.01$.

Interprétabilité

Nous avons constaté que les méthodes à noyau proposées dans ce manuscrit s'adaptent plus facilement à des données non linéaires que des SVMs polynomiaux par exemple. De plus, les règles de décision de nos contributions sont plus facilement interprétables que celles estimées par les SVMs. Contrairement aux SVMs à noyaux traditionnels, il est possible pour la KLR et l'EKLR de réécrire la règle de décision comme une somme additive de splines univariées et ainsi d'afficher les effets non linéaires estimés.

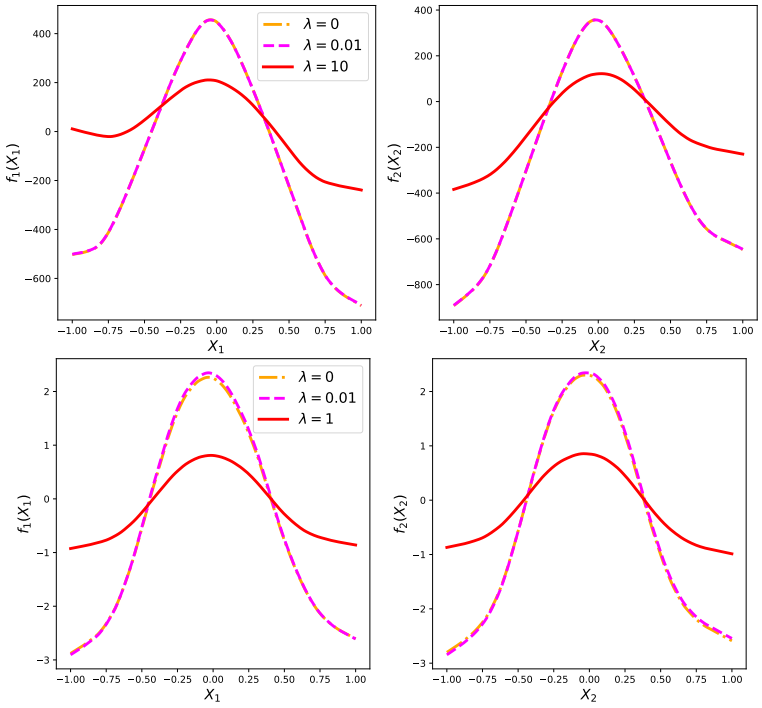

FIGURE 5.8 – Splines estimées pour les variables X_1 (Gauche) et X_2 (Droite) par les EKLR (Haut) et EKLR (Bas) avec différents paramètres de régularisation $\lambda = 0$ en orange, $\lambda = 0.01$ en rose et $\lambda = 10$ en rouge.

La Figure 5.8 présente les splines estimées par les EKLR (Haute) et EKLR (Basse) pour les variables X_1 (Gauche) et X_2 (Droite). Nous les avons tracées pour différentes valeurs de régularisation : $\lambda = 0$ en orange, $\lambda = 0.01$ en rose et $\lambda = 10$ en rouge. Le premier constat que nous pouvons faire est que les splines estimées par les EKLR et EKLR sont de la même forme, seulement leurs ordres de grandeur diffèrent : pour les régularisations $\lambda = [0, 0.01]$ (courbes orange et rose confondues) les EKLR et EKLR modélisent respectivement des splines sur les intervalles $[-900, 400]$ et $[-3, 2.5]$ pour la variable X_2. La seconde conclusion que nous pouvons tirer est que la pénalisation utilisée pour l'apprentissage des méthodes a un rôle important sur les splines estimées. Plus la régularisation ℓ_2 est importante, plus les paramètres sont forcés à être dans une boule centrée en 0 de plus petit rayon, et donc moins les splines sont étendues. Pour la variable X_2, les splines estimées par la KLR et l'EKLR avec une forte régularisation varient respectivement cette fois sur les intervalles $[-400, 100]$ et $[-1, 1]$.

Unicité de la règle de décision

La régularisation ℓ_2 force donc les coefficients estimés à être proches de 0, mais a surtout pour principal avantage de rendre moins volatiles les paramètres estimés quelles que soient leurs valeurs initiales. Ainsi, les règles de décision estimées par les EKLR et EKLR avec une pénalisation suffisamment importante sont uniques. La Figure 5.9 illustre les fréquences des normes des paramètres obtenus par Validation Croisée 5−folds pour les EKLR (Haut) et EKLR (Bas) pour différents paramètres de régularisation $\lambda = [0, 0.01, 10]$. Nous pouvons constater que la distance entre les paramètres estimés $\langle \hat{\alpha}^{\text{EKLR}}_{\text{fold 1}}, \hat{\alpha}^{\text{EKLR}}_{\text{fold 2}}, \hat{\alpha}^{\text{EKLR}}_{\text{fold 3}}, \hat{\alpha}^{\text{EKLR}}_{\text{fold 4}}, \hat{\alpha}^{\text{EKLR}}_{\text{fold 5}} \rangle$ par les deux méthodes à noyau proposées, tend à être nulle lorsque la régularisation est suffisamment grande.

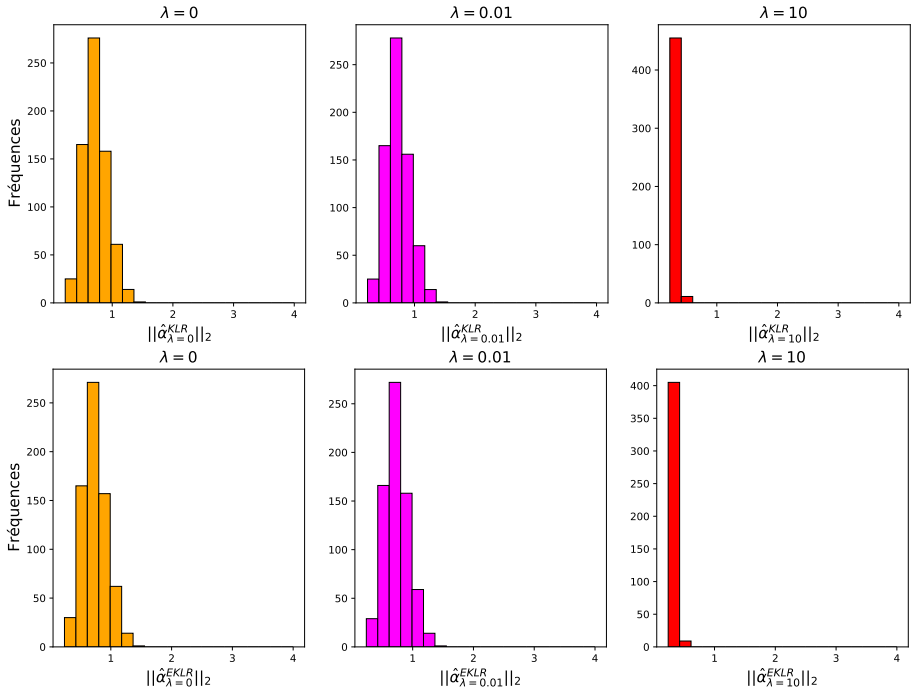

FIGURE 5.9 – Histogramme des normes des paramètres estimées par les KLRs $\hat{\alpha}^{\text{KLR}}$ (Haut) et EKLRs $\hat{\alpha}^{\text{EKLR}}$ (Bas)par Validation Croisée 5−folds pour différentes valeurs de régularisation $\lambda = 0$ (oranges), $\lambda = 0.01$ (roses) et $\lambda = 10$ (rouges).

Rappelons que la KLR estime une règle de décision unique conditionnellement à l'échantillon d'apprentissage mais aussi aux initialisations du SATURNN que nous souhaitons approximer. Tandis que l'EKLR est elle totalement indépendante des $\theta^{(0)}$. De plus, nous avons vu précédemment par la Figure 5.5 que l'EKLR se stabilise plus rapidement que la KLR (pour un plus petit p). Ainsi l'EKLR est une méthode d'approximation du SATURNN fiable puisque d'une part elle obtient de bonnes performances prédictives pour un petit p et elle estime des règles de décision interprétables et uniques conditionnellement seulement à l'échantillon d'apprentissage.

5.5 Synthèse

Dans ce chapitre, nous avons introduit deux méthodes à noyau permettant d'approximer le SATURNN. Les Régressions Logistiques à Noyau proposées (KLR et EKLR) appliquent la Régression Logistique aux données préalablement transformées. Cette transformation non linéaire opérée se fait par l'application d'un noyau. Les deux noyaux proposés découlent directement de l'architecture du SATURNN.

La première méthode KLR permet de retrouver la règle de décision du SATURNN en réinjectant ses paramètres estimés directement dans le modèle que l'on souhaite approximer. Il est alors possible de retrouver les effets non linéaires estimés par variable et de les afficher. Puisque la fonction de coût minimisée pour entraîner la KLR est fortement convexe, nous disposons de garanties quant à l'unicité de ses estimations.

Cependant, le noyau utilisé pour la KLR n'est pas indépendant du processus d'apprentissage du SATURNN que nous cherchons à approximer, puisqu'il dépend de ses paramètres initialisés. Nous avons alors introduit l'EKLR, une Régression Logistique à Noyau Déterministe. Cette méthode d'approximation applique aux données un noyau déterministe qui n'est autre que l'espérance du noyau de la KLR. Ce noyau bien que découlant directement de l'architecture du SATURNN est totalement indépendant de son processus d'entraînement. La règle de décision de l'EKLR peut se réécrire comme une décomposition additive de splines univariées similaire au SATURNN. Ainsi, pour chaque variable d'entrée, nous pouvons afficher les splines estimées. Enfin, nous savons que lorsque nous apprenons l'EKLR avec une régularisation ℓ_2, nous optimisons un problème de minimisation fortement convexe et ainsi nous estimons des paramètres uniques. De ce fait, l'EKLR est un modèle explicable au sens de la Définition 2, puisque d'une part la règle de décision qui en résulte est interprétable et d'autre part nous disposons de garanties de convergence et d'unicité des résultats.

Chapitre 6

Application à des données réelles

Dans ce chapitre nous comparons la fiabilité à la fois prédictive mais aussi d'interprétabilité de nos contributions aux méthodes de l'état de l'art. Pour ce faire, nous avons utilisé trois jeux de données réelles publiques visant à prédire des pathologies. Dans un premier temps nous comparons les performances prédictives des 11 méthodes de classification estimées sur chaque jeu de données. Ensuite, nous nous intéressons à la fiabilité d'interprétation des estimations, à savoir les splines univariées issues des différents modèles. Enfin, nous vérifions les points théoriques importants établis dans ce manuscrit sur les bases de données réelles.

Sommaire

6.1 Présentation des données

Dans ce chapitre, trois jeux de données réelles et publiques ont été utilisées afin de comparer nos contributions aux autres méthodes de l'état de l'art. Les bases de données choisies ont toutes pour but de prédire une pathologie. Néanmoins, elles se démarquent de part la disponibilité des données, à savoir le nombre d'échantillons, mais aussi de variables descriptives à disposition.

Framingham

Le jeu de données Framingham [Mahmood *et al.*, 2014] est très connue dans le domaine de l'application médicale. Elle vise à prédire un évènement cardio-vasculaire à partir de 15 variables descriptives telles que le genre, le taux de cholestérol ou encore l'indice de masse corporelle. La base de données sans valeurs manquantes contient 3658 échantillons dont un peu plus de 15% de patients ayant présenté un évènement cardio-vasculaire (Classe 1). Comme précisé en Introduction de ce manuscrit, le déséquilibre des bases de données, notamment lorsque la population d'intérêt est la moins représentée peut léser la performance des ADS développés. Ce sujet de recherche n'étant pas notre objectif d'étude, nous avons décidé de rééquilibrer le jeu de données. Au final, nous avons à disposition 1114 patients et 15 variables descriptives dont les statistiques sont disponibles dans le Tableau 6.1 ci-dessous :

Variable	Moyenne	Écart-type	Minimum	Médiane	Maximum
Genre	0.48	0.5	0	0	1
Âge	51.5	8.6	34	51.5	69
Éducation	1.9	1.01	1	2	4
Fumeur	0.5	0.5	0	0	1
# Cigarette/jour	9.57	12.4	0	0	60
Traitements hypertension	0.04	0.21	0	0	1
Antécédent AVC	0.01	0.1	0	0	1
Antécédent Hypertension	0.4	0.5	0	0	1
Diabète	0.04	0.2	0	0	1
Taux de Cholestérol	240.9	46.2	124	239	600
Tension Systolique	136.7	24.7	83.5	132	295
Tension Diastolique	84.4	13.3	48	83	140
IMC	26.1	4.3	16	25.7	56.8
Fréquence cardiaque	75.9	12.1	45	75	125
Glucose	84.7	31.69	40	78	394

TABLE 6.1 – Principales statistiques des échantillons par variable descriptive pour la prédiction d'évènements cardio-vasculaires.

Diabete

Le jeu de données Diabete [1] est issue d'une étude de l'Institut National du Diabète et des Maladies Digestives et Rénales des États-Unis. Les patientes présentes dans ce jeu de données sont issues de la Communauté indienne de *Salt River Pima* [Smith *et al.*, 1988]. L'objectif de cette étude est de pouvoir détecter le diabète à partir de 8 mesures clinico-biologiques telles que le nombre de grossesses, le glucose ou encore la pression artérielle. Au

1. Kaggle DataSets - Diabete

départ nous disposions de 768 patientes dont seulement 268 atteintes de diabète soit moins de 35%. Nous avons rééquilibré la base de données et réalisé nos études sur 536 patientes pour lesquelles le Tableau 6.2 détaille les principales statistiques de leurs caractéristiques clinico-biologiques.

Variable	Moyenne	Écart-type	Minimum	Médiane	Maximum
# Grossesses	4.1	3.5	0	3	17
Glucose	125.6	33.3	0	122	199
Pression Artérielle	69.5	19.7	0	72	122
Épaisseur de la peau	20.9	16.4	0	24	99
Insuline	80.1	114.8	0	20.5	846
IMC	32.6	8.1	0	32.6	67.1
Diabete Pedigree	0.5	0.3	0.1	0.4	2.4
Âge	34.1	11.8	21	31	81

TABLE 6.2 – Principales statistiques des patientes composant la base de données Diabete par caractéristique clinico-biologique.

Parkinson

Le jeu de données Parkinson [2] [Little *et al.*, 2008] est composée d'une série de mesures effectuées sur 195 enregistrements vocaux de personnes. Ce jeu de données visent à prédire la maladie de Parkinson à partir des variations des fréquences de voix des personnes. La base de données étant une fois encore très déséquilibrée (74.5% - 24.6%) nous avons décidé de la rééquilibrer. Finalement nous disposons de 96 échantillons et nous avons gardé 16 mesures bio-médicales de voix telles que la fréquence maximale, minimale mais aussi des mesures de variation de la voix en fréquence fondamentale et en amplitude. Le Tableau 6.3 ci-dessous décrit les statistiques de la base de données à disposition pour apprendre nos règles de décision.

Variable	Moyenne	Écart-type	Minimum	Médiane	Maximum
Fréquence vocale moyenne	159.2	48.5	88.3	148.5	260.1
Fréquence vocale maximale	191.9	80.2	102.3	178.2	592
Fréquence vocale minimale	127.4	50.1	68.4	108.8	239.2
Variation de la fréquence (%)	0.006	0.005	0.002	0.004	0.033
Variation de la fréquence (Abs)	0	0	0	0	0.0003
Variation de la fréquence (RAP)	0.003	0.003	0.0009	0.002	0.0214
Variation de la fréquence (PPQ)	0.003	0.003	0.001	0.002	0.02
Variation de la fréquence (DDP)	0.009	0.01	0.0028	0.006	0.064
Variation en amplitude	0.027	0.02	0.01	0.02	0.12
Variation en amplitude (dB)	0.252	0.2	0.085	0.185	1.3
Variation en amplitude (APQ3)	0.014	0.01	0.005	0.012	0.056
Variation en amplitude (APQ5)	0.016	0.012	0.006	0.012	0.079
Variation en amplitude (APQ)	0.022	0.018	0.007	0.016	0.138
Variation en amplitude (DDA)	0.042	0.03	0.014	0.032	0.169
NHR	0.023	0.045	0.0006	0.008	0.315
HNR	22.62	4.68	8.44	23.45	33.05

TABLE 6.3 – Principales Statistiques des échantillons disponibles par variable descriptive pour différentier les patients sains et ceux atteints de la maladie de Parkinson.

2. UCI Machine Learning - Parkinson

6.2 Méthodes testées

Nous comparons la performance du SATURNN (Section 3.3) et de ses méthodes d'approximation LR PSI LIN (Section 4.3), KLR (Section 5.2) et EKLR (Section 5.3) à diverses méthodes de l'état de l'art. Pour les méthodes gloutonnes nous avons utilisé un algorithme de *gridsearch* pour sélectionner les hyperparamètres optimaux. Le nombre d'arbres ainsi que leur profondeur pour les RF et EBM (Section 2.3.3) ont été sélectionnés par Validation Croisée $5-$folds de sorte à maximiser l'AUC moyenne obtenue sur les échantillons de validation afin de limiter le sur-apprentissage des méthodes. Pour les MARS (2.3.1) et les GAM (Section 2.3.2), les algorithmes implémentés permettent d'optimiser le nombre d'effets non linéaires ajoutés sans avoir à procéder au préalable à une recherche de paramètres optimaux. Ainsi, toutes les méthodes gloutonnes ont été estimées dans le but de limiter le sur-apprentissage. Afin de pouvoir comparer équitablement les méthodes, nous n'avons pas intégré d'effets d'intération entre les variables pour les GAM et les EBM.

Pour les Réseaux de Neurones ReLU (Section 2.4) nous avons choisi le nombre de neurones de manière à obtenir de bonnes performances de généralisation. Trop peu de neurones aurait pour conséquence de ne pas modéliser suffisamment d'effets non linéaires et donc de fournir une règle de décision dont les performances prédictives seraient inférieures à celles que nous pourrions prétendre. En revanche, un trop grand nombre de neurones entraînerait un risque de sur-apprentissage. En ce qui concerne les NAMs (Section 2.4.4), l'algorithme de *gridsearch* proposé par les auteurs est trop long à entraîner. Ainsi nous avons tout comme pour les RN ReLU et le SATURNN fixé le nombre de neurones (c-à-d pour les NAMs le nombre de splines par variable) de sorte à trouver le juste équilibre entre sous-apprentissage et sur-apprentissage. Enfin, tout comme pour les GAM et les EBM, les NAMs réalisent un partitionnement univarié de l'espace d'entrée et n'intègrent donc aucun effets d'intération entre les variables afin de pouvoir comparer équitablement les performances de toutes les méthodes testées. Finalement, les paramètres utilisés pour l'entraînement des RN sur chaque base de données sont résumés dans le tableau ci-dessous :

	RN ReLU	**NAM**	**SATURNN**	**SATURNN$_\infty$**
FRAMINGHAM				
Nombre de Neurones	8	20	40	50 0000
Pas de Gradient	1e-1	1e-4	1e-1	2e-1
Nombre d'itérations	20 000	20 000	30 000	30 000
DIABETE				
Nombre de Neurones	12	20	40	50 000
Pas de Gradient	1e-1	1e-4	1e-1	2e-1
Nombre d'itérations	20 000	20 000	30 000	30 000
PARKINSON				
Nombre de Neurones	8	30	60	50 000
Pas de Gradient	1e-1	1e-4	1e-1	2e-1
Nombre d'itérations	20 000	20 000	30 000	30 000

TABLE 6.4 – Paramètres d'entraînement des Réseaux de Neurones : nombre de neurones, pas de gradient et nombres d'itérations pour la descente de gradient.

Les méthodes d'approximation LR PSI LIN, KLR et EKLR découlent du SATURNN$_\infty$ composé d'un grand nombre de neurones ($p = 50\ 000$). Ainsi, lorsque nous étudions leur pouvoir d'approximation, nous comparons leurs performances respectives au SATURNN$_\infty$.

Les données ont été normalisées (équation (3.19)) afin que les valeurs prises par les variables soit comprises dans la boule ouverte de rayon 1. Toutes les méthodes testées ont été entraînées par Validation Croisée 5−folds. Les résultats présentés dans les Tableaux qui vont suivre sont moyennisés. Nous proposons aussi leurs écart-types respectifs afin de donner une idée de la variabilité des performances prédictives obtenues.

6.3 Comparaison des méthodes

6.3.1 Performances prédictives

Le Tableau 6.5 résume les performances globales (3.20) et AUC moyennes obtenues lors de la CV 5−folds sur les bases de données Framingham et Diabete. Toutes les méthodes testées obtiennent relativement les mêmes performances globales sur l'échantillon de validation. Pour le jeu de données Framingham, le SATURNN, le NAM et la LR PSI LIN obtiennent 66% de performances prédictives en moyenne sur les échantillons de validation, les méthodes à noyau KLR et EKLR atteignent 67% et les MARS et EBM 68%. En ce qui concerne la base de données Diabete, les performances globales moyennes obtenues sur les échantillons de validation sont de 73% pour les MARS, GAM, EBM, RF et les méthodes à noyau KLR et EKLR, de 74% pour le NAM et de 75% pour le SATURNN. Néanmoins, les méthodes gloutonnes ont tendance à sur-apprendre la base de données d'apprentissage. Nous constatons que les différences de performances globales moyennes obtenues sur les bases d'entraînement et de test pour Framingham sont de 8% pour les EBM et GAM, et de 11% pour le RF. Nos contributions se généralisent mieux : le SATURNN obtient respectivement 69% en moyenne de performances prédictives sur les bases d'apprentissage et 66% sur celles de validation soit un différentiel de 3%. Le même constat est fait pour les méthodes à noyau KLR et EKLR qui obtiennent en moyenne seulement 2% de différence. Sur la base de données Diabete les mêmes conclusions sont faites puisque les performances varient sur les échantillons de 6% pour les EBM et RF, 8% pour le GAM et seulement 2% pour les méthodes à noyau KLR et EKLR, là où le SATURNN obtient en moyenne 75% de bonnes classifications à la fois sur les échantillons d'apprentissage et de validation.

Lorsque nous nous intéressons plus particulièrement aux performances obtenues par nos contributions, nous nous rendons compte que le SATURNN obtient des performances similaires au NAM, algorithme dont l'architecture se rapproche le plus de notre modèle, bien qu'étant plus simple à optimiser. Le SATURNN et le NAM obtiennent tous les deux en moyenne 66% de performances prédictives et 71% d'AUC sur les échantillons de validation issus du jeu de données Framingham. Sur la base de données Diabete, le SATURNN obtient en moyenne respectivement 75% et 84% de performances prédictives et d'AUC test contre 74% et 82% pour le NAM. Finalement, les méthodes à noyau KLR et EKLR approximent correctement le SATURNN_∞. En effet, pour Framingham le SATURNN_∞ obtient 67% de bonnes classifications sur les bases de validation et la KLR et l'EKLR réalisent de bonnes prédictions sur respectivement 65% et 66% d'échantillons de test. En ce qui concerne la prédiction du diabète le SATURNN_∞, la KLR et l'EKLR obtiennent respectivement 81%, 82% et 80% d'AUC test. De plus, ces méthodes à noyau sont très rapides à entraîner (1 à 2 millisecondes par fold) et obtiennent des performances similaires aux NAMs qui nécessitent un temps de calcul de 152 secondes sur Framingham et 88 sur le jeu de données du Diabete pour obtenir les mêmes performances prédictives.

Méthodes	Apprentissage				Validation				Temps de Calcul
	Perf. Globales	Perf. Classe 1	Perf. Classe 2	AUC	Perf. Globales	Perf. Classe 1	Perf. Classe 2	AUC	
FRAMINGHAM									
MARS	0.69 (0.01)	0.69 (0.02)	0.69 (0.03)	0.74 (0.01)	0.68 (0.02)	0.66 (0.02)	**0.70 (0.02)**	**0.73 (0.02)**	0.2
GAM	0.73 (0.01)	0.73 (0.02)	0.72 (0.02)	0.80 (0.01)	0.65 (0.01)	0.64 (0.01)	0.65 (0.03)	0.69 (0.02)	0.9
EBM	0.76(0.01)	0.78 (0.02)	0.74 (0.02)	0.83 (0.01)	**0.68 (0.01)**	**0.68 (0.03)**	0.60 (0.03)	0.74 (0.01)	0.1
RF	0.74 (0.01)	0.73 (0.02)	0.75 (0.03)	0.82 (0.01)	0.63 (0.02)	0.62 (0.05)	0.66 (0.02)	0.70 (0.02)	0.1
RN RELU	0.73 (0.01)	0.76 (0.07)	0.71 (0.09)	0.81 (0.01)	0.62 (0.03)	0.61 (0.08)	0.64 (0.08)	0.68 (0.03)	472
NAM	0.72 (0.01)	0.73 (0.01)	0.71 (0.02)	0.78 (0.01)	0.66 (0.01)	0.66 (0.02)	0.66 (0.03)	0.71 (0.02)	152
SATURNN	0.69 (0.01)	0.69 (0.02)	0.70 (0.02)	0.75 (0.01)	0.66 (0.01)	0.64 (0.02)	0.68 (0.02)	0.71 (0.02)	716
SATURNN$_\infty$	0.71 (0.01)	0.71 (0.01)	0.71 (0.01)	0.77 (0.01)	0.67 (0.01)	0.67 (0.05)	0.68 (0.04)	0.72 (0.01)	762.5
LR PSI LIN	0.74 (0.01)	0.75 (0.02)	0.73 (0.02)	0.81 (0.01)	0.66 (0.01)	0.64 (0.03)	0.69 (0.02)	0.71 (0.01)	50.4
KLR	0.69 (0.01)	0.67 (0.2)	0.70 (0.02)	0.74 (0.01)	0.67 (0.02)	0.65 (0.02)	0.69 (0.03)	**0.73 (0.02)**	0.2
EKLR	0.69 (0.01)	0.67 (0.02)	0.70 (0.02)	0.74 (0.01)	0.67 (0.02)	0.66 (0.02)	0.68 (0.03)	**0.73 (0.02)**	0.2
DIABETE									
MARS	0.76 (0.01)	0.76 (0.02)	0.75 (0.02)	0.85 (0.01)	0.73 (0.02)	0.73 (0.05)	0.75 (0.05)	0.80 (0.02)	0.4
GAM	0.81 (0.01)	0.80 (0.01)	0.81 (0.01)	0.90 (0.01)	0.73 (0.03)	0.72 (0.05)	0.75 (0.07)	0.81 (0.02)	0.1
EBM	0.79 (0.01)	0.75 (0.02)	0.83 (0.04)	0.88 (0.01)	0.73 (0.02)	0.72 (0.08)	0.74 (0.05)	**0.84 (0.02)**	6.1
RF	0.79 (0.01)	0.74 (0.02)	0.84 (0.02)	0.88 (0.01)	0.73 (0.02)	0.67 (0.05)	**0.79 (0.03)**	0.81 (0.02)	0.3
NN RELU	0.83 (0.01)	0.84 (0.05)	0.82 (0.04)	0.92 (0.01)	0.69 (0.01)	0.64 (0.06)	0.74 (0.09)	0.77 (0.02)	252
NAM	0.80 (0.01)	0.79 (0.01)	0.81 (0.01)	0.89 (0.01)	0.74 (0.03)	0.72 (0.04)	0.77 (0.03)	0.82 (0.02)	88
SATURNN	0.75 (0.01)	0.75 (0.02)	0.75 (0.01)	0.85 (0.01)	**0.75 (0.01)**	**0.76 (0.05)**	0.75 (0.07)	**0.84 (0.01)**	393
SATURNN$_\infty$	0.80 (0.02)	0.80 (0.02)	0.80 (0.03)	0.88 (0.01)	0.73 (0.01)	0.73 (0.01)	0.74 (0.06)	0.81 (0.03)	403
LR PSI LIN	0.82 (0.01)	0.82 (0.01)	0.83 (0.01)	0.92 (0.01)	0.72 (0.01)	0.70 (0.02)	0.75 (0.03)	0.80 (0.02)	12.5
KLR	0.75 (0.02)	0.73 (0.02)	0.77 (0.02)	0.85 (0.01)	0.73 (0.03)	0.71 (0.02)	0.77 (0.06)	0.83 (0.02)	0.1
EKLR	0.75 (0.01)	0.73 (0.02)	0.77 (0.02)	0.85 (0.01)	0.73 (0.02)	0.69 (0.02)	0.77 (0.05)	0.80 (0.02)	0.1

TABLE 6.5 – Résultats des expériences obtenus par Validation Croisée 5−folds sur les bases de données Framingham et Diabete pour les méthodes testées : performances globales, par classe et AUC moyennes (écart-types) sur les échantillons d'apprentissage et de validation, ainsi que les temps de calcul (en secondes). Nos contributions sont en bleu.

Concernant les résultats obtenus sur la base de données Parkinson (Tableau 6.6), les méthodes gloutonnes ont encore une fois tendance à sur-apprendre. Les variations de performances prédictives entre les bases d'apprentissage et de validation sont sur cette base encore plus importantes : en moyenne le différentiel de bonnes classifications entre les échantillons est de 12% pour les MARS, 18% pour les EBM et même 20% pour les RF, tandis que le SATURNN ne perd que 4% de pouvoir de généralisation. Du fait de la petite base de données d'entraînement à disposition pour estimer la règle de décision (67 patients par *fold*), les ADS dont le processus d'estimation est glouton ont tendance à sur-apprendre. Le SATURNN est quant à lui, ici en mesure d'estimer une règle de décision généralisable malgré le faible nombre de données. Puisque les KLR et EKLR appliquent une transformation non linéaire aux données sous forme de noyau, le nombre d'échantillons a évidemment un impact : moins nous disposons de données, moins d'effets non linéaires seront modélisés. Ainsi, comme nous pouvions l'anticiper, la base de données Parkinson est trop petite pour espérer approximer le SATURNN_∞ par les méthodes à noyaux. Le KLR et l'EKLR obtiennent en moyenne 67% de bonnes classifications et 83% d'AUC sur les échantillons de validation, tandis que le SATURNN∞ que l'on cherche à approximer, atteint respectivement 78% et 85%.

	Apprentissage		**Validation**		**Temps de**
Méthodes	**Perf.**	**AUC**	**Perf.**	**AUC**	**Calcul**
MARS	0.94 (0.04)	0.98 (0.02)	**0.82 (0.07)**	0.86 (0.08)	0.2
GAM	0.76 (0.02)	0.84 (0.02)	0.74 (0.02)	0.77 (0.04)	0.75
EBM	0.97 (0.01)	0.99 (0.01)	0.79 (0.1)	**0.90 (0.05)**	18
RF	0.99 (0.01)	0.99 (0.01)	0.79 (0.05)	0.89 (0.07)	0.01
NN RELU	0.95 (0.06)	0.99 (0.02)	0.79 (0.06)	0.85 (0.07)	75
NAM	0.98 (0.03)	0.99 (0.01)	0.81 (0.04)	0.88 (0.04)	87
SATURNN	0.76 (0.05)	0.86 (0.03)	0.72 (0.1)	0.87 (0.08)	128
SATURNN_∞	0.92 (0.01)	0.97 (0.01)	0.78 (0.05)	0.85 (0.03)	136
LR PSI LIN	0.81 (0.03)	0.92 (0.01)	0.74 (0.11)	0.87 (0.07)	0.9
KLR	0.66 (0.09)	0.82 (0.05)	0.67 (0.07)	0.83 (0.1)	0.007
EKLR	0.66 (0.09)	0.81 (0.05)	0.67 (0.07)	0.83 (0.1)	0.009

TABLE 6.6 – Résultats des expériences obtenus sur la base de données Parkinson par Validation Croisée 5−folds : performances globales et AUC moyennes (écart-types) sur les échantillons d'apprentissage et de validation et temps de calcul (en seconde). Nos contributions sont en bleu.

6.3.2 Explicabilité des résultats

Nos contributions sont alors tout aussi performantes que les méthodes gloutonnes ou les NAMs lorsque les bases de données sont suffisamment grandes, mais ont pour principal atout d'être explicables. Les splines estimées par les méthodes d'approximation LR PSI LIN, KLR et EKLR sont uniques. Contrairement aux méthodes itératives et aux RN qui minimisent des problèmes d'optimisation non convexes, nous pouvons alors avoir confiance en l'interprétation que nous faisons des splines obtenues par nos méthodes d'approximation du SATURNN_∞. Sur les Figures 6.1 et 6.2 nous retrouvons les splines obtenues par les différentes méthodes réalisant un partitionnement univarié de l'espace d'entrée. Il est à noter que les échelles de valeurs prises par les splines sont très différentes, elles ont alors été normalisées afin de pouvoir les comparer et les visualiser sur une même figure de la

manière suivante :

$$\tilde{f}_i(x_i) = \frac{f_i(x_i)}{\max(|f_i(x_i)|)}, \tag{6.1}$$

pour tout $i \in \{1, \ldots, d\}$. Cette normalisation a pour avantage de conserver les signes et donc de ne pas biaiser l'interprétation des effets non linéaires estimés pour chaque variable.

Nous avons mentionné précédemment que les méthodes gloutonnes ont tendance à sur-apprendre la base de données d'apprentissage, notamment sur la base de données Parkinson. Ce constat se vérifie lorsque nous nous intéressons aux splines estimées par ces algorithmes. Sur la Figure 6.1 nous retrouvons les effets non linéaires estimés pour la prédiction de la maladie de Parkinson pour les variables fréquence vocale moyenne (haute) et maximale (basse). Lorsque nous nous intéressons à celles estimées par l'EBM (courbes bleues) et plus particulièrement le GAM (courbes vertes), nous pouvons constater que les effets non linéaires sont très complexes du fait du sur-apprentissage, contrairement aux splines issues du NAM (courbes roses) ou du SATURNN (courbes rouges) qui sont moins détaillées.

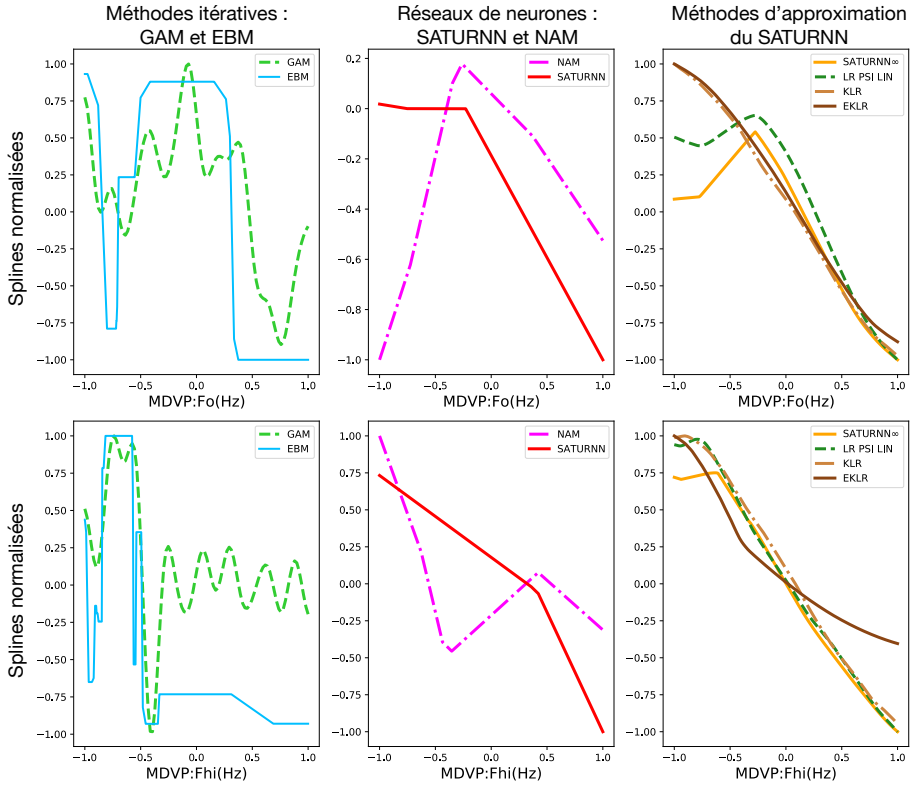

FIGURE 6.1 – Splines normalisées pour la Fréquence Vocale Moyenne (Haute) et Maximale (Basse) estimées pour la prédiction de la maladie de Parkinson. À gauche nous retrouvons celles obtenues par les méthodes itératives GAM (vert) et EBM (bleu). Au milieu, celles estimées par le NAM (rose) et le SATURNN composé de 40 neurones (rouge) sont affichées. Finalement à droite nous comparons celles du SATURNN$_\infty$ composé de 50 000 neurones (orange) et de ses méthodes d'approximation LR PSI LIN (vert), KLR (marron clair) et EKLR (marron foncé).

Sur les bases de données Framingham et Diabete, le GAM a plus de difficultés à identifier des effets non linéaires (Figure 6.2). De plus, l'impact estimé pour cette variable sur la prédiction des pathologies respectives n'est pas pertinent biologiquement. Sur la Figure 6.2 nous pouvons visualiser les splines estimées pour la variable Glucose sur les bases de données Framingham (Haute) et Diabete (Basse) pour les différentes méthodes testées. Pour la prédiction d'évènements cardio-vasculaires, nous pouvons constater que l'impact du glucose estimé par le GAM (Figure 6.2 - Haute, courbe verte), bien que croissant, est néanmoins toujours négatif. Les autres méthodes telles que l'EBM (courbe bleue), le SATURNN$_\infty$ composé de 50 000 neurones (courbe orange), ainsi que ses méthodes d'approximation estiment qu'il existe un seuil à partir du quel le taux de glucose augmente les risques d'AVC (les splines deviennent positives). Sur la base de données Diabete, l'incohérence de la spline estimée par le GAM pour le Glucose est encore plus importante : selon le GAM un petit taux de Glucose augmente le risque d'être atteint de diabète alors que pour un seuil très élevé, les patients en seraient protégés. Ainsi, bien que performantes, les méthodes gloutonnes ne sont pas toujours en mesure de fournir une interprétation des effets non linéaires pertinente ; elles ont tendance à sur-apprendre la base de données d'apprentissage et à estimer des effets non linéaires parfois peu cohérents biologiquement.

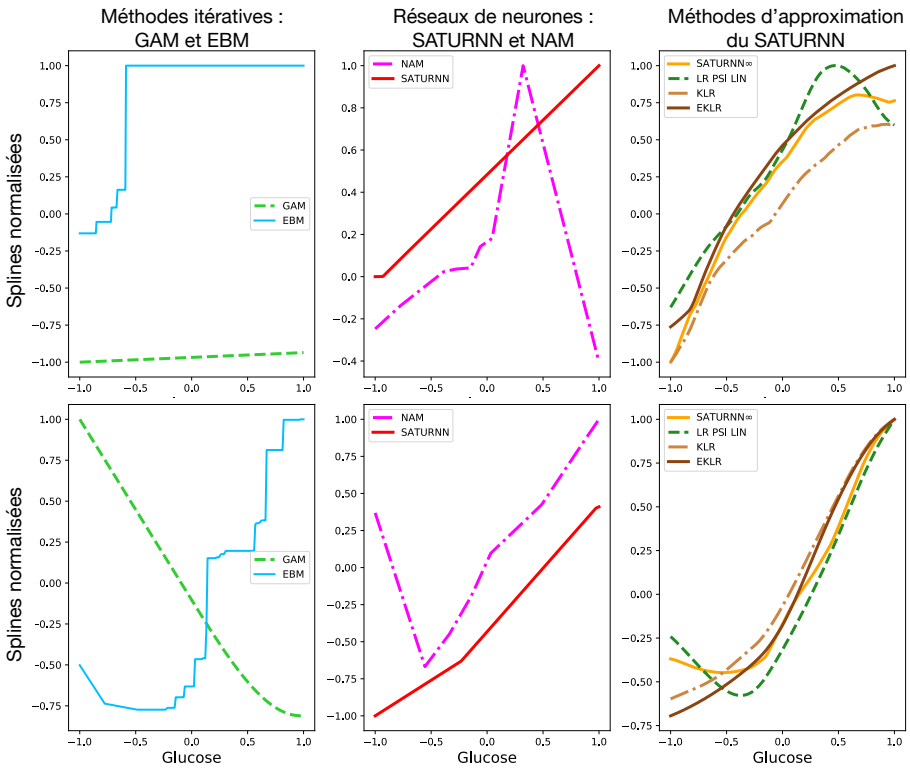

FIGURE 6.2 – Splines normalisées pour le Glucose sur la base de données Framingham (Haute) et Diabete (Basse). À gauche, nous retrouvons celles obtenues par les méthodes itératives GAM (vert) et EBM (bleu). Au milieu, celles estimées par le NAM (rose) et le SATURNN composé de 40 neurones (rouge) sont affichées. Finalement, à droite, nous comparons celles du SATURNN$_\infty$ composé de 50 000 neurones (orange) et de ses méthodes d'approximation LR PSI LIN (vert), KLR (marron clair) et EKLR (marron foncé).

Les Réseaux de Neurones bien qu'utilisant un critère d'optimisation global ne sont

pas en mesure de fournir des résultats explicables. Sur la Figure 6.3 nous retrouvons les splines estimées pour les variables Pression Artérielle (Haute) et Insuline (Basse) dans le but de prédire le diabète. Une pression artérielle trop faible ou trop élevée est propice à entraîner des problèmes de santé, comme la LR PSI LIN (courbe verte), la KLR (courbe marron clair) et l'EKLR (courbe marron foncé) le modélisent. En revanche le GAM (courbe bleue) et le SATURNN composé de 40 neurones (courbe rouge) estiment qu'à mesure que la pression artérielle augmente, le risque de développer du diabète diminue. De plus, ce SATURNN estime une spline strictement positive ; la pression artérielle quelque soit sa valeur est un facteur de risque selon ce réseau de neurones. Ainsi, l'interprétation que l'on fait des paramètres estimés par les méthodes pour lesquelles nous ne disposons d'aucune garantie de convergence peut s'avérer peu pertinente biologiquement. Ce constat est aussi vérifié lorsque nous nous intéressons à l'impact de l'insuline pour la prédiction de diabète (Figure 6.3 - Basse). Les méthodes convergentes (LR PSI LIN, KLR et EKLR) modélisent correctement qu'en dessous d'une certaine valeur de taux d'insuline, le patient est protégé et qu'il existe un seuil à partir duquel la valeur insulinaire est un facteur à risque. Tandis que le GAM (courbe bleue) estime que plus l'insuline augmente, moins le patient a de chance d'être diabétique. Le NAM (courbe rose) quant à lui estime seulement qu'à partir d'un certain seuil les risques de présenter un diabète augmente, mais ne modélise en aucun cas le caractère protecteur d'une faible dose (spline toujours positive).

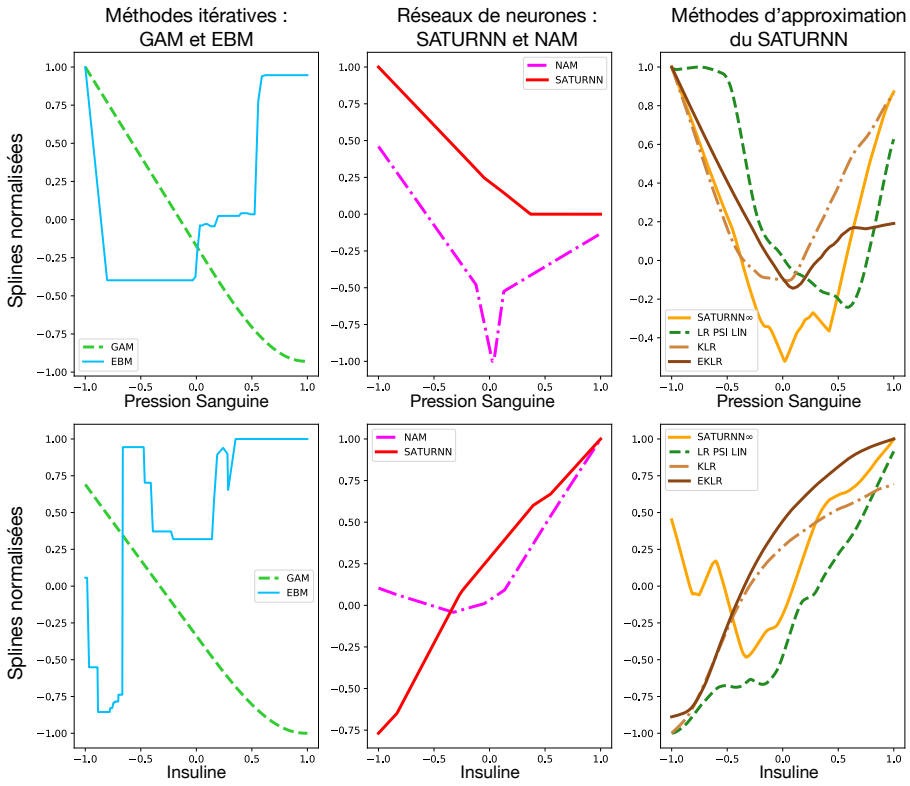

FIGURE 6.3 – Splines normalisées pour les variables Pression Artérielle (Haute) et Insuline (Basse) estimées pour la prédiction de diabète. À gauche nous retrouvons celles obtenues par les méthodes itératives GAM (vert) et EBM (bleu). Au milieu, celles estimées par le NAM (rose) et le SATURNN composé de 40 neurones (rouge) sont affichées. Finalement à droite nous comparons celles du SATURNN∞ composé de 50 000 neurones (orange) et de ses méthodes d'approximation LR PSI LIN (vert), KLR (marron clair) et EKLR (marron foncé).

6.4 Résultats théoriques de nos contributions

La LR PSI LIN, la KLR et l'EKLR ont démontré une grande qualité d'approximation du SATURNN sur les bases de données Framingham et Diabete (Tableau 6.5). En revanche, sur la petite base de données Parkinson, d'une part les performances obtenues par ces méthodes d'approximation (Tableau 6.6) sont légèrement inférieures à celles escomptées et d'autre part les splines qui en résultent ont des comportements linéaires moins prononcés. Nous remarquons que sur les trois bases de données, les KLR et EKLR obtiennent des performances similaires. Ces résultats confirment la théorie développée dans le Chapitre 5. Nous avions démontré à l'équation (5.16) que le noyau appliqué par la KLR κ_0 (5.4) converge asymptotiquement vers celui appliqué par l'EKLR κ (5.19) à mesure que le nombre p de neurones composant le SATURNN que l'on souhaite approximer augmente. Ces résultats théoriques sont vérifiés sur les trois bases de données. Sur la Figure 6.4 - Gauche, nous pouvons observer que la norme de Frobenius entre les deux noyaux calculés sur les échantillons de validation au cours d'une Validation Croisée 5−folds tend à diminuer lorsque p augmente, pour les bases de données Framingham (en vert), Diabete (en bleu) et Parkinson (en orange). De plus, il est établi dans la Proposition 2 qu'il est équivalent de considérer la KLR ou l'EKLR lorsque le nombre p de neurones composant le SATURNN que l'on souhaite approximer. Sur la Figure 6.4 - Droite, les moyennes d'erreurs obtenues par Validation Croisée 5−folds entre $\delta^{\text{KLR}}(x, \alpha)$ et $\delta^{\text{EKLR}}(x, \alpha)$ sont affichées. Nous pouvons constater que ces erreurs décroissent à mesure que p augmente, comme théoriquement établi à la fois sur les bases de données Framingham (courbe verte), Diabete (courbe bleue) et Parkinson (courbe orange). Ainsi, les résultats théoriques établis sont vérifiés justifiant les similarités des performances prédictives obtenues entre les KLR et EKLR sur les trois bases de données réelles.

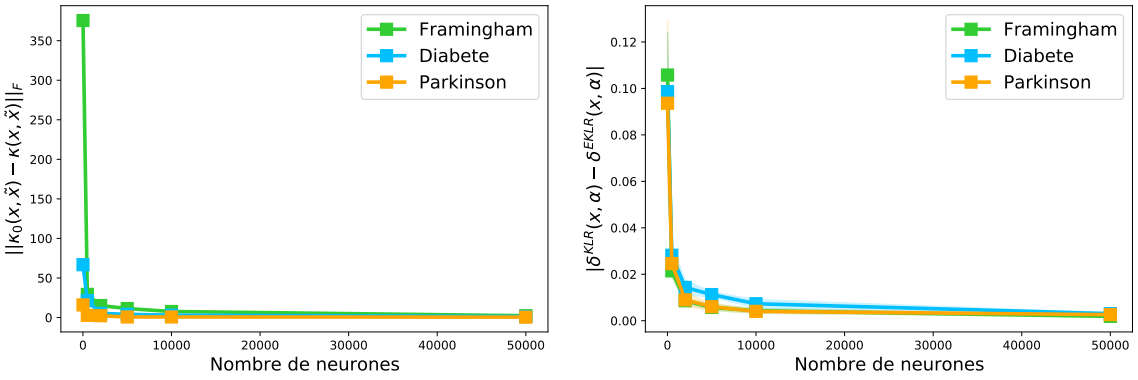

FIGURE 6.4 – Gauche : Normes de Frobenius moyennes entre les Noyaux κ_0 et κ obtenues par Validation Croisée 5−folds sur les bases de données Framingham (courbe verte), Diabete (courbe bleue) et Parkinson (courbe orange). Droite : Erreurs d'approximation moyennes de la KLR $\delta^{\text{KLR}}(x, \alpha)$ (5.6) par l'EKLR $\delta^{\text{EKLR}}(x, \alpha)$ (5.18) obtenues par Validation Croisée 5−folds sur les échantillons de validation pour différentes valeurs de p pour les bases de données Framingham (courbe verte), Diabete (courbe bleue) et Parkinson (courbe orange).

C'est lorsque nous nous intéressons aux erreurs d'approximation entre les méthodes à noyaux et le SATURNN que des différences sur les trois bases de données s'observent. Sur la Figure 6.5 les erreurs moyennes entre les méthodes à noyaux KLR (Gauche) et EKLR (Droite) obtenues par Validation Croisée 5−folds sont représentées sous forme de

boîtes à moustache. Nous pouvons constater qu'à mesure que p augmente, la KLR et l'EKLR approximent davantage correctement le SATURNN$_\infty$ sur les les bases de données Framingham (boîtes vertes) et Diabete (boîtes bleues). En revanche, les erreurs obtenues sur la base de données Parkinson (boîtes oranges) sont plus élevées que pour les deux autres jeux de données comme vérifié précédemment par les performances obtenues dans le Tableau 6.6. De plus, les décroissances attendues des erreurs d'approximation entre les méthodes à noyau et le SATURNN$_\infty$ à mesure que p augmente ne sont pas vérifiées pour la base de données Parkinson. Enfin, nous pouvons remarquer que les erreurs d'approximation du SATURNN$_\infty$ par les méthodes à noyau sont inférieures sur Framingham, qui est la base de données composées du plus grand nombre d'échantillons. Ainsi le nombre p de neurones, mais aussi la taille de l'échantillon d'apprentissage semblent avoir un impact sur le pouvoir d'approximation du SATURNN$_\infty$ par les méthodes à noyau KLR et EKLR.

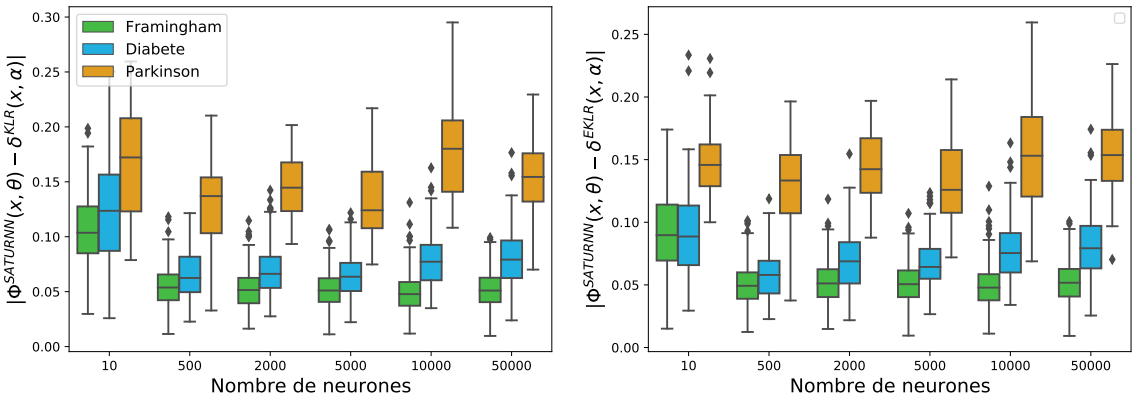

FIGURE 6.5 – Erreurs d'approximation moyennes du SATURNN par KLR (Gauche) et par EKLR (Droite) obtenues sur les échantillons de validation par Validation Croisée 5−folds pour les bases de données Framingham (vert), Diabete (bleu) et Parkinson (orange).

6.5 Synthèse

Les méthodes d'approximation du SATURNN proposées dans ce manuscrit, ont démontré à la fois leur performance sur des bases de données réelles mais aussi leur fiabilité en termes d'interprétation des résultats. Sur les trois bases de données testées, le SATURNN ainsi que ses méthodes d'approximation LR PSI LIN, la KLR et l'EKLR obtiennent des performances prédictives similaires aux autres méthodes de l'état de l'art sur les échantillons de validation. Néanmoins, nos contributions semblent davantage stables puisqu'elles ne présentent aucun sur-apprentissage contrairement aux méthodes itératives. De plus, à travers les trois bases de données testées nous avons mis en évidence que les méthodes d'approximation du SATURNN proposées dans ce manuscrit font preuve d'une plus grande fiabilité dans l'interprétation faite des non-linéarités estimées. En effet, les méthodes non convergentes telles que les GAM, les NAMs ou même les SATURNN ont montré certaines incohérences biologiques au niveau de l'explicabilité des résultats obtenus. La LR PSI LIN, la KLR et l'EKLR ont quant à elles fourni des estimations interprétables, uniques et biologiquement fiables.

Sur la plus petite base de données Parkinson, le pouvoir d'approximation des méthodes à noyau s'est révélé être inférieur à celui que nous pouvions escompter. Les performances

prédictives restent néanmoins du même ordre de grandeur que celles obtenues par les méthodes de l'état de l'art. En théorie, les résultats de convergence des noyaux respectifs et des estimations obtenues par la Régression Logistique appliquée à ces noyaux sont vérifiés. Les transformations non linéaires des données opérées par la KLR et l'EKLR résultent directement des noyaux considérés et donc du nombre d'échantillons composant les bases d'apprentissage. Ainsi, la faible disponibilité d'échantillons dans la base de données Parkinson est une possible explication aux résultats d'approximation légèrement inférieurs à ceux que nous pouvions espérer.

Chapitre 7

Résultats des collaborations

Dans ce chapitre, nous présentons les résultats issus des collaborations avec l'IPMC sur la prédiction de la bipolarité. La Section 7.1 présente le travail réalisé sur les bases de données à disposition. Ensuite, nous détaillons en Section 7.2 les résultats issus du modèle complet à savoir construit avec toutes les variables à disposition. Dans les Sections 7.3 et 7.4 nous présentons le processus de sélection de variables ainsi que les performances obtenues pour des modèles constitués des variables les plus discriminantes. Avec le même procédé, nous détaillons en Section 7.5 les résultats issus de modèles sexes-spécifiques. Une synthèse et une discussion concernant les résultats obtenus sont proposées en Section 7.6.

Sommaire

7.1 Préparation des données

Dans cette section nous détaillons les bases de données dont nous disposons ainsi que le traitement qu'elles ont nécessité avant de pouvoir entraîner les ADS. Finalement sur la base de données finale, nous réalisons plusieurs tests statistiques de corrélations et de répartition des variables par rapport à l'étiquette à prédire.

7.1.1 Jointure des bases de données

Nous disposons de deux bases de données provenant de deux centres médicaux distincts pour développer un ADS permettant de distinguer les patients dépressifs et bipolaires à partir de variables clinico-biologiques. La première base de données constituée par le CHU de Marseille contient 114 patients dont 29% de bipolaires. Le second jeu de données provenant d'un centre hospitalier de Montpellier contient quant à lui plus de données, mieux réparties entre les classes mais qui présente un grand nombre de valeurs manquantes relativement à la taille de la base de données : nous disposons de 462 patients dont 43% de bipolaires et 184 échantillons avec des valeurs manquantes.

Centre de collecte	**Marseille**	**Montpellier**
Nombre de patients	114	462
Valeurs Manquantes	0	184
Répartition en %	[71, 29]	[60, 40]

TABLE 7.1 – Résumé des bases de données de Marseille et de Montpellier : nombre total de patients, nombre de valeurs manquantes et répartition des classes (1-Dépressifs, 2-Bipolaires).

Ces deux jeux de données sont de parfaits exemples aux différents challenges rencontrés pour l'application médicale. Lorsque nous ne retenons que les patients ne présentant aucune valeur manquante, nous disposons de seulement 278 patients pour la base de données de Montpellier. Les bases de données à disposition sont alors de petites tailles. De plus, la base de Marseille est déséquilibrée ; beaucoup plus de patients dépressifs sont présents dans la base de données. Afin de palier à ces deux grandes limites, nous avons dans un premier temps joint les deux bases à disposition. Ainsi, nous disposons de 576 échantillons, dont 392 sans valeurs manquantes contenant un peu plus de 40% de bipolaires. Afin de différentier les patients dépressifs des bipolaires, nous disposons de 32 variables clinico-biologiques : 13 variables biologiques telles que le genre, l'âge, l'IMC ou encore la consommation de drogues douces et 19 cytokines (protéines).

7.1.2 Nettoyage de la base de données

Dans cette section, nous présentons les différentes étapes de préparation de la base de données jointe nécessaires afin de disposer d'un jeu de données sur lequel entraîner les ADS.

Suppression des outliers

La première étape de notre analyse consiste à analyser la présence de valeurs extrêmes pouvant fausser l'analyse. Il arrive comme précisé en Introduction de ce manuscrit que

certaines valeurs peuvent être faussées au moment de la mesure de la caractéristique ou bien de l'ajout dans le fichier de données. En affichant différentes statistiques, telles que la moyenne des variables ainsi que les valeurs minimales et maximales nous pouvons identifier plusieurs "*outliers*" c'est à dire des patients présentant des valeurs aberrantes. Par exemple, pour la cytokine IL6, nous avons une valeur moyenne de 0.91, un quantile 75% de 0.81 et une valeur maximale de 55, ce qui laisse penser qu'au moins un échantillon présente une valeur extrême, qui plus est biologiquement peu cohérente. Sur la Figure 7.1-Gauche, nous pouvons effectivement constater que deux échantillons (en rouge) ont des valeurs bien plus élevées que le reste de l'échantillon. Sur cette même figure nous pouvons retrouver les valeurs extrêmes pour les cytokines CCL22 (Milieu) et TNFα (Droite). De plus, en Annexe F.1, le Tableau F.1 répertorie les statistiques qui ont permis de repérer les valeurs aberrantes présentes pour 13 cytokines. Afin de ne pas fausser les ADS développés, nous avons décidé de supprimer les 12 patients présentant des valeurs extrêmes. Ainsi, après ce nettoyage de la base de données jointe, nous disposons de 380 patients dont un peu plus de 43% de bipolaires.

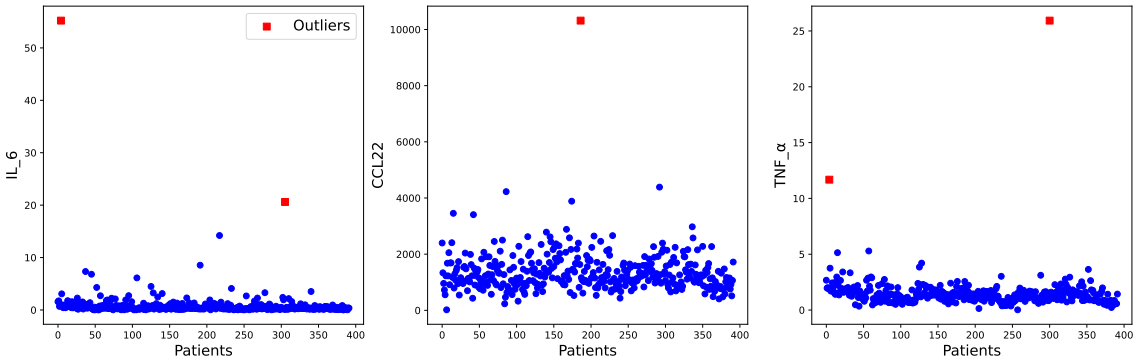

FIGURE 7.1 – Valeurs extrêmes repérées (carrés rouges) pour les variables IL6, CCL22 et TNFα.

Analyse Statistique de la base de données

La seconde étape de notre analyse consiste à étudier les répartitions de patients dans la base de données. Pour ce faire, nous calculons à nouveau diverses métriques, telles que par exemple la moyenne, la médiane, les valeurs minimales et maximales par variable sur les jeux bases de données. En comparant les différentes statistiques, nous nous apercevons d'une part que certaines caractéristiques des patients inclus dans les études diffèrent selon le centre de collecte des données. Tout d'abord, les échantillons marseillais ne comprennent aucune consommation d'alcool ou de drogues douces. Il en découle que seulement 16% des patients du jeu de données joint consomment alors des drogues. Ce déséquilibre de représentation dans la base de données peut entraîner des discriminations lors de l'apprentissage des ADS comme expliqué en Section 1.2. D'autre part, un problème bien plus contraignant peut être mis en évidence. Les valeurs moyennes et médianes des cytokines diffèrent grandement entre les échantillons montpelliérains et marseillais. Dans le Tableau 7.2 ci-dessous, les 10 plus grandes différences de répartition sont présentées.

	Montpellier			Marseille		
	Moyenne	**Écart-type**	**Médiane**	**Moyenne**	**Écart-type**	**Médiane**
IDSC30	**30.46**	**15.28**	**32.0**	**43.08**	**6.13**	**42.0**
CCL3	**42.61**	**66.68**	**36.72**	**19.4**	**29.67**	**14.2**
CCL4	84.26	47.01	76.28	103.35	68.27	87.57
CCL11	219.2	114.21	197.32	191.77	167.35	140.09
CCL17	566.99	460.12	465.98	318.29	241.81	242.03
CXCL10	180.64	164.11	146.56	337.07	221.11	271.0
IL7	16.06	7.63	14.88	7.28	9.3	5.67
IL8	**10.13**	**12.84**	**8.25**	**160.96**	**568.41**	**13.64**
IL16	218.05	182.11	171.8	156.63	80.44	146.41
IFNγ	3.91	14.83	1.75	4.93	6.46	3.05

TABLE 7.2 – Statistiques (moyennes, écart-types, médianes) des 10 cytokines présentant de grandes différences de répartition entre les bases de données de Montpellier et de Marseille. La répartition des trois variables en gras est affichée à la Figure 7.2.

Lorsque nous visualisons ces données sur la Figure 7.2 en fonction du centre de collecte des données (rouge pour Marseille et bleu pour Montpellier), la différence de répartition est évidente. La différence de répartitions des cytokines s'explique par de nombreuses phases successives de congélation et décongélation des cytokines qui ont pu altérer leurs mesures.

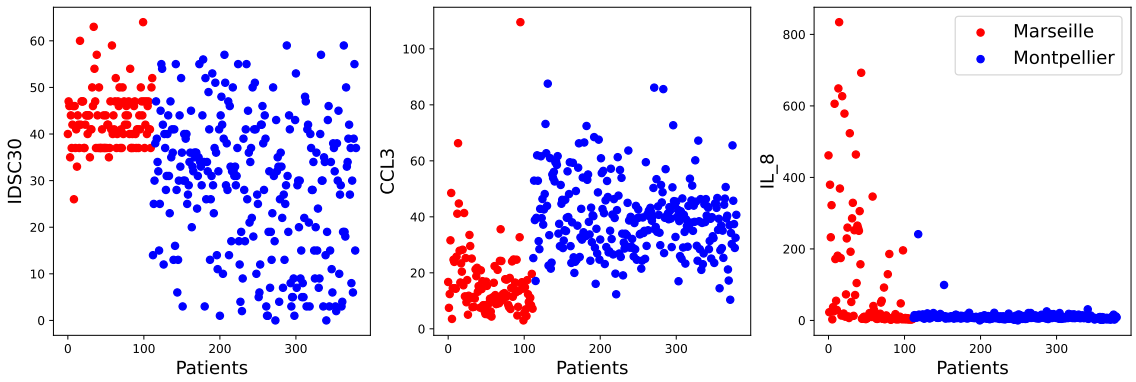

FIGURE 7.2 – Répartitions des variables IDSC30, CCL3 et IL8 en fonction des centres de collecte de données : en rouge nous retrouvons les échantillons marseillais et en bleu ceux issus du CHU de Montpellier.

Base de données de Montpellier

Afin de ne pas introduire de biais dans l'apprentissage d'un ADS, nous avons décidé de ne se concentrer que sur les échantillons collectés par le centre hospitalier de Montpellier. Une fois le nettoyage de la base de données opéré, c'est à dire les échantillons ayant des valeurs manquantes et aberrantes supprimés, la base de données résultante reflète les différentes problématiques auxquelles nous sommes confrontés pour l'application à la médecine de précision. La base de données est de petite taille, seulement 268 patients sont présents. Néanmoins, pour ce cas d'étude la base de données est équilibrée puisque 50% des patients sont dépressifs et 50% sont bipolaires. Le Tableau 7.3 détaille les principales statistiques des 13 variables biologiques et 19 cytokines à disposition pour différencier les

patients dépressifs de ceux atteints de bipolarité.

Variable	Moyenne	Écart-type	Minimum	Médiane	Maximum
Genre	1.65	0.48	1	2	2
Age	41.96	13.51	18	43	78
BMI	24.21	5.26	12.89	23.77	51.47
Tabac	0.82	0.7	0	1	2
Alcool	0.31	0.46	0	0	1
Drogues récréatives	0.2	0.4	0	0	1
Anti-dépresseurs	0.57	0.5	0	1	1
Benzodiazepines	0.5	0.5	0	0	1
Anti-psychotiques	0.37	0.48	0	0	1
Lithium	0.13	0.33	0	0	1
Anti-convulsants	0.26	0.44	0	0	1
Autres médicaments psychotiques	0.15	0.36	0	0	1
Anti-inflammatoires	0.02	0.15	0	0	1
IDSC30	28.73	14.98	0	31	59
CCL2	270.85	112.87	89.13	254.68	741.52
CCL3	39.94	13.11	10.38	39.34	87.56
CCL4	84.21	35.26	26.01	79.61	255.82
CCL11	222.78	109.84	40.27	204.26	879.06
CCL13	143.12	72.93	16.99	130.07	423.43
CCL17	568.20	394.57	73.80	470.27	2710.97
CCL22	1373.93	575.76	407.32	1240.61	4386.89
CXCL10	187.76	133.50	30.56	159.64	1252.51
IL6	0.60	1.05	0.03	0.35	14.21
IL7	16.31	7.44	0.06	15.58	52.37
IL8	10.37	15.82	0.04	8.29	241.24
IL10	0.23	0.47	0.02	0.15	6.97
IL12p40	89.42	52.35	0.16	77.24	333.88
IL15	1.76	0.55	0.08	1.77	3.45
IL16	216.43	159.71	3.67	174.18	1197.49
IL27	1003.21	531.88	135.66	933.03	6119.83
IFNγ	3.35	8.74	0.18	1.59	95.38
TNFα	1.29	0.57	0.02	1.25	3.85

TABLE 7.3 – Principales Statistiques des échantillons montpelliérains par variable descriptive pour différencier les patients dépressifs des patients bipolaires.

7.1.3 Tests statistiques

Le test statistique du chi-2 [Rakotomalala, 2013] permet de savoir si deux variables discrètes ou discrétisées sont dépendantes l'une de l'autre. Il s'agit d'un test d'hypothèse, telle que si l'hypothèse H1 est retenue alors l'hypothèse nulle H0 d'indépendance des variables est rejetée. Lorsque nous avons appliqué ce test à nos données, nous nous sommes aperçus que seulement 10 des 32 variables sont corrélées au label à savoir : l'âge, la consommation de tabac, la prise d'antidépresseurs, de benzodiazépines, d'antipsychotiques, de lithium et d'anticonvulsants mais aussi certaines cytokines telles que l'IDSC30, le CCL3 et l'IL10. Toutes les dépendances identifiées entre les variables sont visibles en Annexe F.2.1 sur la Figure F.1.

De plus lorsque nous comparons les moyennes (Test de Student) et les variances (Test de Levene) entre les deux populations (dépressifs et bipolaires), nous nous rendons compte que peu de variables présentent des répartitions différentes. Seuls l'âge et la cytokine

IDSC30 ont des répartitions différentes entre les pathologies à la fois en moyenne et variance. Nous pouvons nous rendre compte sur la Figure 7.3-Gauche ci-dessous que la médiane de l'âge des échantillons bipolaires est plus élevée que pour les dépressifs et que leur répartition est plus centrée autour de cette valeur. Pour ce qui est de la cytokine IDSC30 (Figure 7.3-Milieu), le constat inverse peut être fait : la médiane est plus élevée pour les dépressifs et la répartition plus étirée pour les bipolaires. Enfin, le test de Levene rejette l'hypothèse nulle de variances similaires entre les deux populations d'intérêt pour les cytokines CCL3 et IL15 bien que le test de Student accepte l'hypothèse nulle de moyenne similaire.

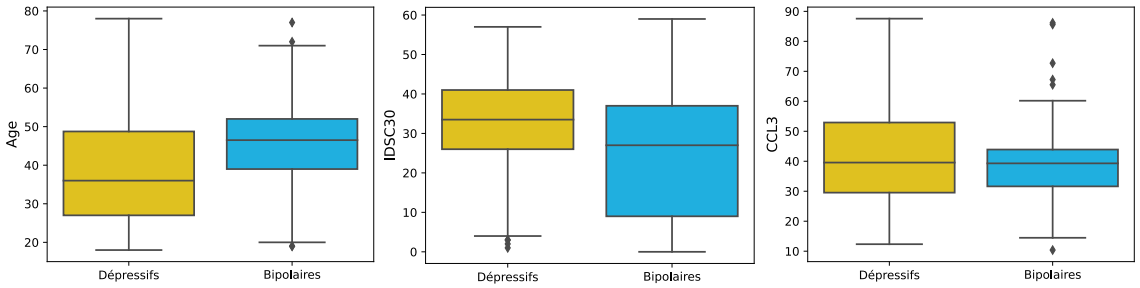

FIGURE 7.3 – Boxplots de répartition des variables Âge (Gauche), IDSC30 (Milieu) et CCL3 (Droite) en fonction du label à prédire : les dépressifs en jaune et bipolaires en bleu.

Il y a alors peu de différence de répartition entre les deux populations, combiné au fait que peu de variables sont corrélées au label, le développement d'un ADS performant pour différentier les échantillons bipolaires des dépressifs s'annonce difficile.

7.2 Modèle Complet

Dans cette section nous entraînons des modèles sur l'ensemble des variables dont nous disposons, soit pour rappel 32 au total (13 variables biologiques et 19 cytokines). Dans un premier temps nous présentons les différentes méthodes testées sur la base de données de Montpellier. Dans un second temps, nous comparons les différentes performances obtenues.

7.2.1 Méthodes testées

Nous comparons nos contributions à deux méthodes de l'état de l'art : une linéaire et une autre modélisant des effets de seuil. Nous avons dans un premier temps estimé une Régression Logistique (LR) définie en Section 2.2.1 avec une pénalisation de Ridge sur les paramètres. Cette régularisation nous permet d'obtenir des garanties de convergence et d'unicité des paramètres estimés conditionnellement à la base d'apprentissage. La LR de Ridge est entraînée avec le paramètre de régularisation maximisant l'AUC moyenne obtenue sur l'échantillon. Afin de comparer nos contributions à un modèle de classification non linéaire, nous avons entraîné un EBM (Section 2.3.3). Le nombre d'arbres ainsi que leur profondeur composant l'EBM ont été sélectionnés de sorte à limiter le sur-apprentissage. Les paramètres retenus sont ceux maximisant l'AUC moyenne obtenue sur les échantillons de validation. Finalement nous avons entraîné deux SATURNN, un composé de 40 neurones et un autre noté SATURNN_∞ composé de 50 000 neurones, une LR PSI LIN, une KLR et

un EKLR. Les méthodes, ainsi que leur algorithme de *gridsearch* sont entraînés et évalués par Validation Croisée 5−folds.

7.2.2 Résultats Globaux

Le Tableau 7.4 résume les performances globales (3.20) et AUC moyennes obtenues par Validation Croisée 5−folds. Le premier constat important est que la LR, l'EBM, le SATURNN$_\infty$ et la LR PSI LIN sur-apprennent. Plus précisément l'EBM réalise 94% de bonnes classifications sur la base d'apprentissage et seulement 67% sur celle de validation. La LR PSI LIN présente une différence moyenne d'AUC de plus de 17% entre les deux bases de données. De plus, toutes les méthodes entraînées sont instables notamment pour la prédiction de patients bipolaires (Classe 2), leurs écart-types étant très élevés.

	Apprentissage		**Validation**			
	Perf.	**AUC**	**Globale**	**Classe 1**	**Classe 2**	**AUC**
LR	83.7 (3.9)	91.5 (2.3)	67.0 (2.7)	67.6 (5.3)	66.9 (7.9)	74.1 (3.2)
EBM	94.1 (1.6)	98.5 (0.9)	67.2 (4.8)	71.8 (3.4)	63.1 (10.0)	77.5 (5.5)
SATURNN	72.3 (3.4)	80.5 (3.4)	69.4 (4.7)	69.5 (12.6)	69.8 (11.7)	74.5 (5.6)
SATURNN$_\infty$	85.2 (2)	92.4 (1.5)	66.7 (4.7)	71.1 (5.6)	62.9 (6.0)	76.4 (5.0)
LR PSI LIN	89.5 (1.5)	96.2 (0.6)	70.9 (3.3)	70.0 (7.2)	72.0 (3.4)	78.8 (2.2)
KLR	75.2 (1.4)	83.7 (1.3)	73.3 (5.1)	73.4 (8.1)	73.7 (11.9)	82.2 (3.4)
EKLR	75.2 (1.5)	83.7 (1.3)	72.4 (5.2)	72.4 (8.1)	74.2 (12.1)	82.1 (3.5)

TABLE 7.4 – Performances prédictives et AUC moyennes (écart-types) obtenues par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation pour le modèle complet composé de 32 variables descriptives. Pour les échantillons de validation, nous présentons aussi les performances obtenues par classe (1-Dépressifs et 2-Bipolaires).

L'instabilité des méthodes ne se retrouve pas seulement dans les performances, mais aussi dans les coefficients estimés. Rappelons qu'une régularisation ℓ_2 a été ajoutée pour que la LR fournisse des garanties d'unicité. La Figure 7.4 illustre les boîtes à moustache des coefficients estimés par la LR pour les 32 variables inclues dans le modèle. Nous pouvons constater d'une part que les boîtes obtenues sont larges et donc que les coefficients estimés varient grandement en fonction de la base d'apprentissage utilisée. D'autre part, pour la majorité des variables explicatives, les coefficients associés obtiennent des signes contraires en fonction des échantillons d'apprentissage. Cette volatilité rend le modèle trop instable, peu fiable mais aussi son interprétabilité difficile.

S'intéresser aux coefficients estimés par nos contributions revient à se concentrer sur les splines qui en résultent. Sur la Figure 7.5, nous retrouvons les splines estimées par les méthodes proposées dans le manuscrit pour l'IDSC30 (Gauche), le BMI (Centre) et la cytokine CCL22 (Droite). Ces splines ont été normalisées (6.1) afin de pouvoir les comparer et les visualiser sur une même figure. Tout d'abord, nous remarquons que le SATURNN estime des splines nulles, s'apparentant à un processus de sélection des variables importantes, pour 16 variables dont 11 cytokines telles que l'IDSC30 (Figure 7.5-Gauche) et la CCL22 (Figure 7.5-Droite). Ensuite, nous constatons que beaucoup de variables ont un impact plus ou moins linéaire sur le risque d'être bipolaire, comme l'IDSC30 (Figure 7.5-Gauche), pouvant expliquer le peu de différences de performances prédictives entre la LR linéaire et les méthodes non linéaires (EBM, SATURNN$_\infty$). En revanche, l'impact de certaines caractéristiques biologiques est fortement non linéaire. Par exemple, pour la CCL22

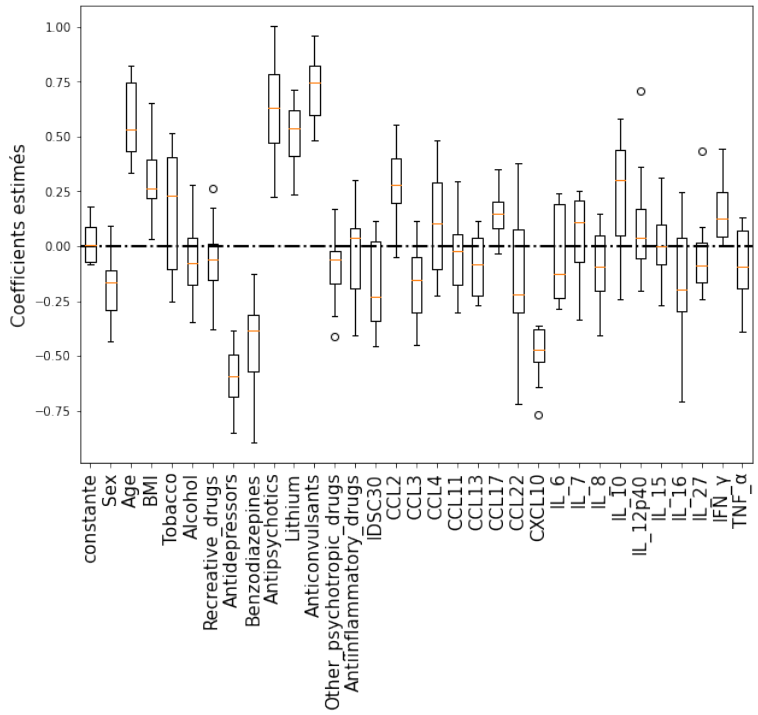

FIGURE 7.4 – Boîtes à moustache des coefficients estimés par Validation Croisée 5−folds pour les 32 variables composant le modèle complet.

(Figure 7.5-Droite), la LR PSI LIN (vert) et la KLR (marron clair) estiment que pour une valeur normalisée de la cytokine entre environ $[-0.5, 0.25]$, les patients ont tendance à être dépressifs. En effet, sur cet intervalle, l'impact de la cytokine sur la probabilité estimée d'être bipolaire est négatif. De plus, pour une valeur de la CCL22 inférieure à peu près à -0.5 et supérieure à 0, la LR PSI LIN et la KLR estiment que le risque d'être bipolaire augmente. En revanche, le SATURNN$_\infty$ (orange) et l'EKLR (marron foncé) estiment qu'au delà d'une valeur normalisée de 0, le risque diminue constamment. Ainsi, les splines estimées par les méthodes explicables sont relativement différentes, reflétant la volatilité des performances prédictives obtenues (Tableau 7.4) et le manque de stabilité des modèles.

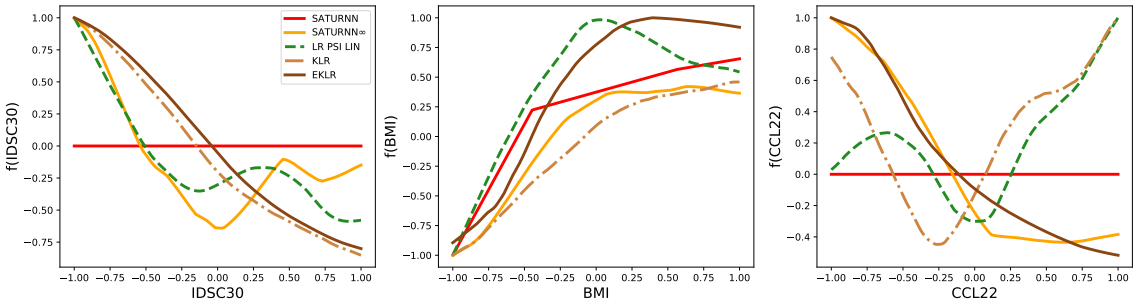

FIGURE 7.5 – Splines normalisées estimées pour le Modèle Complet par le SATURNN (rouge), le SATURNN$_\infty$ (orange), la LR PSI LIN (vert), la KLR (marron clair) et l'EKLR (marron foncé). A gauche nous retrouvons celles estimées pour l'IDSC30, au milieu le BMI et à droite la cytokine CCL2.

7.3 Sélection de variables

Afin d'éviter le sur-apprentissage et rendre les modèles plus stables nous allons au préalable, dans cette section sélectionner les variables les plus discriminantes permettant de différencier les patients bipolaires des dépressifs. Limiter le nombre de variables explicatives dans le modèle a aussi pour conséquence de rendre les modèles plus simples à interpréter et à appliquer. Dans cette section nous détaillons dans un premier temps la méthodologie employée pour sélectionner les variables les plus discriminantes, tout en prenant en compte les corrélations existantes entre elles. Dans un second temps, nous présentons les variables retenues.

7.3.1 Méthodologie employée

Variable Inclusion Probability

Afin de pouvoir sélectionner les variables les plus discriminantes, nous avons entraîné une Régression Logistique (LR). La LR, détaillée dans l'état de l'art $\delta(x, \beta)$ (Section 2.2.1) a été entraînée avec une pénalisation Elastic-Net sur les paramètres :

$$\begin{aligned} L(\delta(\beta, x), y) = &-y \log\left(\delta(x, \beta)\right) + (1 - y) \log\left(1 - \delta(x, \beta)\right) \\ &-\lambda_1 \sum_{j=1}^{d} |\beta_j| - \lambda_2 \sum_{j=1}^{d} \beta_j^2. \end{aligned} \tag{7.1}$$

La régularisation Elastic-Net se compose de deux pénalisations et donc de deux hyper-paramètres : λ_1 pour la pénalisation de Lasso et λ_2 pour celle de Ridge. Les avantages quant à utiliser cette régularisation sont multiples. Tout d'abord la pénalisation de Ridge permet de limiter les variances des prédictions (rendre les estimations uniques) tandis que celle de Lasso permet de fournir un modèle parcimonieux. Plus le paramètre λ_1 sera élevé, plus l'apprentissage sera contraint d'estimer des paramètres β nuls. Ainsi, lorsque les paramètres sont nuls, les variables associées sont écartées de la règle de décision. Néanmoins, de trop fortes régularisations peuvent nuire à la performance prédictive du modèle estimé. Ainsi, nous avons dans un premier temps optimisé les hyperparamètres sur une Validation Croisée (CV) à $25-$folds. Les valeurs optimales λ_1^* et λ_2^* ont été définies par maximisation de l'AUC sur les échantillons de validation. Nous avons ensuite entraîné la LR Elastic-Net avec ces paramètres optimaux et calculé la probabilité qu'une variable soit gardée par le modèle appelée VIP (*Variable Inclusion Probability*). Cela revient à comptabiliser le nombre de fois où une variable se voit affecter un coefficient β non nul (à un ϵ près) sur une CV $50-$folds. Les résultats des VIPs obtenus sont disponibles en Annexe F.2.2 dans le Tableau F.2. Cinq variables ont obtenu un VIP $> 90\%$ à savoir l'âge, la prise d'antidépresseurs, d'antipsychotiques, de lithium et d'anticonvulsants. Néanmoins, la pénalisation de Lasso, bien que permettant de rendre les modèles parcimonieux en écartant des variables du modèle de prédiction ne prend pas en compte les corrélations entre ces dernières. Lorsque des variables sont corrélées, le Lasso sélectionne aléatoirement quelle variable garder. De ce fait, la pénalisation Elastic-Net n'est pas mesure de limiter le problème de *Confounding*.

Limiter le Confounding

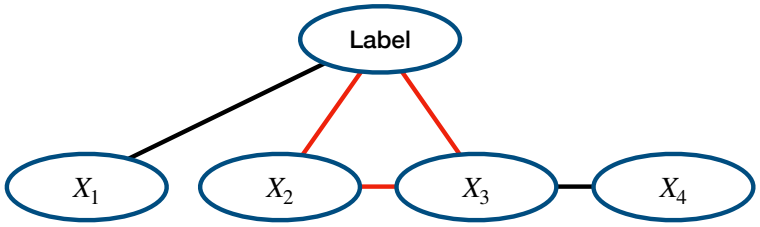

FIGURE 7.6 – Illustration du *Confounding* : les traits noirs représentent des corrélations non dangereuses tandis que les traits rouges illustrent le problème triangulaire de *Confounding* entre deux variables et le label.

Le *Confounding* se caractérise comme un problème dangereux de corrélation entre deux variables et le label [Greenland et Morgenstern, 2001, McNamee, 2005]. Sur la Figure 7.6 ci-dessus nous illustrons le phénomène. Lorsqu'une variable est corrélée seulement au label (X_1) ou à une autre variable sans être dépendante du label (X_4) alors ces relations ne constituent aucun risque. En revanche, le problème de *Confounding* apparaît lorsque le triangle de corrélation entre deux variables et le label simultanément (X_2, X_3 et le label) intervient. En cas de suppression d'une de ces variables du modèle, alors la règle de décision perd son interprétabilité. Le coefficient estimé pour la variable restante intégrera l'impact de cette caractéristique mais aussi de celles corrélées à la fois avec elle et au label. Ainsi, l'interprétation du coefficient, à savoir de son signe et de sa valeur ne sera pas révélatrice de l'impact de cette variable en question. Or dans notre cas d'étude, le test du Chi-2 (Annexe F.2.1) a révélé de nombreuses corrélations entre des variables mais aussi avec le label. Puisque la pénalisation de Lasso ne tient pas compte du *Confounding*, à partir des VIPs nous avons ajouté des variables dans le modèle pour en limiter les effets et obtenir une règle de décision interprétable et fiable. Pour ce faire, nous avons calculé les corrélations linéaires pour les variables catégorielles et appliqué le test du Chi-2 pour les variables continues sur la base de données de Montpellier. Plus précisément, nous avons ajouté les variables pour lesquelles le test du Chi-2 a démontré une forte corrélation avec une caractéristique composant le modèle, car retenue par le VIP.

7.3.2 Variables sélectionnées

Nous partons des cinq variables retenues par le VIP et pour chacune d'entre elles nous vérifions les corrélations linéaires et non linéaires existantes avec d'autres variables et le label simultanément.

— Âge : la variable est corrélée au label et à l'IMC, la consommation de tabac, la prise de lithium, d'anticonvulsants ainsi que les cytokines CCL2, CCL11, CXCL10, IL10 et TFNα. Néanmoins, seules les variables tabac, lithium, anticonvulsants et IL10 sont aussi corrélées au label. Puisque le lithium et les anticonvulsants sont déjà sélectionnés par le VIP, nous ajoutons seulement la consommation de tabac et la cytokine IL10 dans le modèle.

— Antidépresseurs : la variable est corrélée au label et à la prise de benzodiazépines, de lithium et d'autres médicaments psychotiques. Le lithium et les benzodiazépines sont corrélées au label. Le lithium est déjà contenu dans le modèle, nous ajoutons alors seulement les benzodiazépines.

— Antipsychotiques : la variable est corrélée au label, au genre, à la consommation d'alcool et de drogues douces, mais aussi à la prise de benzodiazépine et d'autres médicaments psychotiques, ainsi qu'aux cytokines CCL11, IL27. Seule la variable benzodiazépine est aussi corrélée au label. Puisqu'elle est déjà contenue dans le modèle, nous n'ajoutons aucune nouvelle variable.

— Lithium : la variable est corrélée au label ainsi qu'à l'âge, l'IMC, le tabac, la prise d'antidépresseurs, d'anticonvulsants et les cytokines CCL2, CCL3, CXCL10. Parmi ces variables, seuls l'âge, le tabac, les antidépresseurs, les anticonvulsants et la cytokine CCL3 sont aussi corrélés au label. Nous ajoutons la cytokine CCL3 jusque là non incluse dans le modèle.

— Anticonvulsants : la variable est corrélée au label ainsi qu'à l'âge, le tabac, le lithium, les cytokines CCL2 et IL27. Les variables aussi corrélées au label (Âge, Tabac, Lithium) sont déjà incluses au modèle.

Finalement, nous retenons 8 variables dans le modèle : l'âge, la prise d'antidépresseurs, de lithium, d'anticonvulsants, de benzodiazépines, la consommation de tabac et les cytokines IL10 et CCL3.

7.4 Modèle avec sélection de variables

Dans cette section, nous comparons les performances prédictives obtenues par les modèles composés seulement des 8 variables retenues. Les méthodes testées ainsi que la méthodologie employée sont les mêmes que pour le modèle Complet présenté en Section 7.2. Dans un premier temps nous avons retravaillé la base de données initiales afin de maximiser le nombre d'échantillons à disposition. Ensuite, nous présentons les résultats issus des méthodes testées. Dans un dernier temps, nous introduisons des zones grises pour les différentes LRs estimées.

7.4.1 Préparation de la base de données

Avant d'optimiser le modèle final, nous retravaillons la base de données initiale. Les échantillons d'apprentissage sont restreints puisqu'en se concentrant seulement sur le jeu issus du CHU de Montpellier, sans valeurs manquantes ni valeurs extrêmes nous disposions de seulement 268. Lorsque nous avons supprimé les valeurs manquantes la première fois, nous avons retiré de l'étude des patients qui disposaient des informations nécessaires pour les 8 variables retenues, mais manquantes pour d'autres. Ainsi, nous sommes repartis de la base de données initiale et nous avons gardé seulement les 462 patients montpelliérains et les 8 variables d'intérêt. Nous avons supprimé 78 patients ayant des valeurs manquantes et les 2 *outliers* ayant des valeurs extrêmes pour les cytokines CCL3 et IL10. Finalement nous disposons alors d'une base de données composée de 382 patients pour apprendre le modèle final, avec un peu plus de 48% de bipolaires.

7.4.2 Résultats globaux

Le Tableau 7.5 résume les performances et AUC moyennes obtenues par Validation Croisée 5−folds avec seulement les 8 variables retenues lors de la phase de sélection.

	Apprentissage		Validation			
	Perf.	**AUC**	**Perf.**	**Classe 1**	**Classe 2**	**AUC**
LR	76.0 (2.7)	82.3 (2.5)	70.9 (2.5)	71.6 (5.3)	70.3 (5.6)	76.4 (3.0)
EBM	73.0 (2.7)	83.1 (0.8)	67.5 (4.2)	85.1 (10)	50.6 (15)	78.2 (2.3)
SATURNN	75.1 (0.9)	79.6 (0.9)	75.6 (2.4)	80.5 (6.7)	71.4 (5.9)	82.9 (1.6)
SATURNN$_\infty$	75.5 (1.6)	81.4 (1.3)	76.7 (3.3)	78.1 (5.9)	75.9 (5.6)	81.8 (3.2)
LR PSI LIN	78.7 (2.2)	85.5 (1.7)	74.9 (4.8)	81.2 (6.5)	68.9 (10.6)	80.8 (3.9)
KLR	73.9 (2.6)	78.8 (2.7)	75.3 (5.1)	80.1 (8.5)	70.9 (8.6)	82.0 (5.2)
EKLR	73.9 (2.5)	78.8 (2.6)	75.3 (5.1)	79.7 (8.4)	71.2 (8.1)	82.1 (5.0)

TABLE 7.5 – Performances prédictives et AUC moyennes (écart-types) obtenues par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation pour le modèle composé des 8 variables pré-sélectionnées. Pour les échantillons de validation, nous présentons aussi les performances obtenues par classe (1-Dépressifs et 2-Bipolaires).

Les méthodes estimées se généralisent beaucoup mieux que précédemment. En effet, les EBM obtiennent une différence de performances globales entre les échantillons d'apprentissage et de validation de seulement 5.5% contre 27% pour le modèle complet. Le même constat est fait pour les LR PSI LINs qui atteignent 85% d'AUC moyene en apprentissage et 81% en test quand le différentiel était de plus de 17% précédemment. La sélection de variables a donc été bénéfique pour limiter le sur-apprentissage des méthodes, mais aussi pour obtenir de meilleures performances globales. Toutes les méthodes ont gagné en performance prédictive : le SATURNN classifie correctement 75.6% des échantillons de validation en moyenne (contre moins de 70% précédemment) et les KLR et EKLR 75% contre respectivement 73% et 72% avec le modèle complet. La méthode la plus performante est le SATURNN$_\infty$, atteignant 76.7% de performance globale. Ensuite, nous constatons que toutes les méthodes obtiennent de meilleures performances pour les dépressifs que précédemment. Par exemple, le SATURNN et la LR PSI LIN classifient correctement plus de 80% de ces patients contre 70% avec le modèle complet. Pour les dépressifs, l'EBM est la méthode la plus performante atteignant plus de 85% de performance. En revanche, ce taux de bonnes prédictions est très élevé au détriment des bipolaires qui sont lésés par cet algorithme avec seulement 50%($\pm$15%) de bonnes classifications. La méthode obtenant le meilleur équilibre de classification, soit l'AUC la plus élevée est le SATURNN (presque 83% d'AUC test), suivi de près par les méthodes à noyau KLR et EKLR (82%). Enfin, nous pouvons aussi remarquer que les performances conditionnelles à la classe des bipolaires sont davantage stables que précédemment, sauf pour la LR PSI LIN et l'EBM.

Pour faire la comparaison avec les résultats obtenus précédemment, nous pouvons nous intéresser dans un dernier temps à la stabilité des coefficients estimés par la LR. Sur la Figure 7.7 les boîtes à moustache des valeurs des coefficients sont affichées. Ces boîtes sont d'une part beaucoup moins élargies que précédemment, les coefficients estimés sont donc plus stables. D'autre part, sauf pour quelques valeurs maximales, nous n'observons pas le phénomène de changement de signe.

Les splines normalisées (6.1) estimées par les méthodes proposées dans ce manuscrit sont elles aussi plus stables. Sur la Figure 7.8, nous retrouvons les splines estimées par le SATURNN (rouge), le SATURNN$_\infty$ (orange), la LR PSI LIN (vert), la KLR (marron clair) et l'EKLR (marron foncé). Tout d'abord, la phase de sélection de variables semble pertinente puisque le SATURNN n'estime plus aucune spline nulle. Ainsi, chaque caractéristique

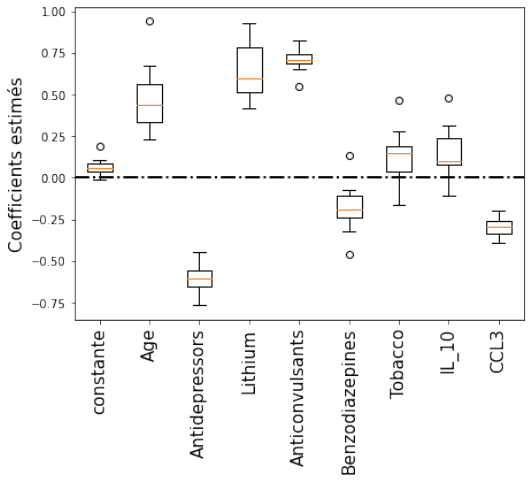

FIGURE 7.7 – Boîtes à moustache des coefficients estimés par Validation Croisée 5−folds pour les 8 variables composant le modèle.

clinico-biologique composant le modèle a un impact sur la prédiction de la bipolarité. Ensuite, nous constatons que les variables catégorielles binaires (Oui / Non) ne sont pas segmentées inutilement par les modèles. Sur la Figure 7.8-Gauche nous pouvons constater que l'impact estimé de la prise d'anticonvulsants est linéaire. En effet, pour l'absence d'anticonvulsants les coefficients estimés sont négatifs, tandis que la prise d'anticonvulsants augmente la probabilité d'être bipolaire. De plus, tout comme le modèle complet, l'impact de certaines variables continues ne nécessite que peu de modélisation non linéaire. Par exemple, sur la Figure 7.8-Centre, nous pouvons constater que les splines estimées par les différentes méthodes pour la variable âge est relativement linéaire. L'impact de l'âge estimé par les méthodes explicables (LR PSI LIN, KLR et EKLR) est continuellement croissant. Ainsi plus un patient vieillit, plus il développe des risques d'être atteint de troubles bipolaires. Plus précisément, en dessous d'un certain âge (0 en valeur normalisée), les splines sont négatives caractérisant les patients de dépressifs, tandis qu'au dessus de ce seuil, le risque d'être bipolaire est présent. Ces splines reflètent les répartitions présentes dans la base de données : les patients bipolaires composant le jeu de données sont plus âgés que les patients dépressifs (Figure 7.3-Gauche). D'autres variables, notamment les cytokines IL10 et CCL3 ont des impacts non linéaires sur la probabilité estimée de développer des troubles bipolaires. Sur la Figure 7.8-Droite, nous retrouvons les splines estimées pour la protéine CCL3. Le premier constat que nous pouvons faire est que le SATURNN estime une spline négative, signifiant que quelque soit la valeur de cette caractéristique biologique le patient est sujet à être dépressif. Cette incohérence biologique, confirmée par les splines des méthodes explicables provient de la non convergence du processus d'apprentissage des réseaux de neurones, pouvant remettre en question la fiabilité de l'interprétation de leurs estimations. En effet, la LR PSI LIN, la KLR et l'EKLR s'accordent sur le fait qu'une petite valeur de CCL3 (jusqu'à une valeur normalisée de 0.5) augmente les risques d'un patient à être sujet à des troubles bipolaires. Ensuite, la LR PSI LIN et l'EKLR modélisent un risque décroissant dépassé ce seuil et même un effet protecteur pour une grande valeur de cette cytokine : la spline devient négative pour une valeur normalisée supérieure à 0.75.

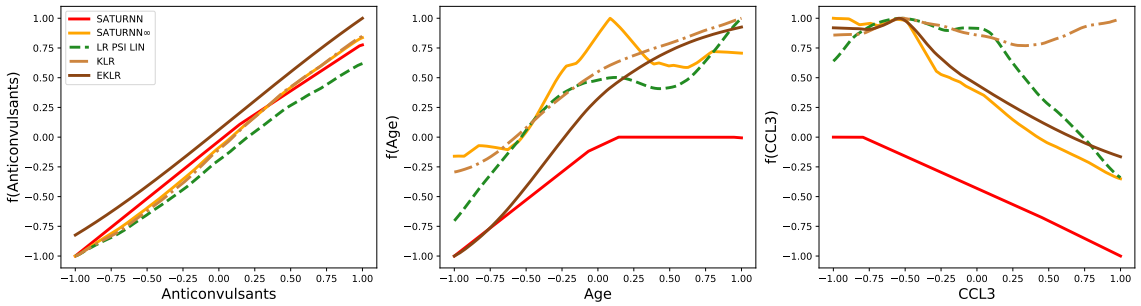

FIGURE 7.8 – Splines normalisées estimées pour le Modèle avec sélection de variables par le SATURNN (rouge), le SATURNN$_\infty$ (orange), la LR PSI LIN (vert), la KLR (marron clair) et l'EKLR (marron foncé). A gauche nous retrouvons celles estimées pour la prise d'anticonvulsants, au milieu l'âge et enfin la cytokine CCL3 à droite.

7.4.3 Zone Grise

Ainsi, les modèles sont plus stables bien que leurs écart-types restent néanmoins relativement élevés lorsque nous nous concentrons sur les performances obtenues par classe. Dans [Alshamaa *et al.*, 2017, Alshamaa *et al.*, 2018], les auteurs proposent un algorithme évaluant des fonctions de croyance, de sorte à fournir des intervalles de confiance dans la classification d'un nouvel échantillon. Une autre solution couramment employée en médecine de précision afin de rendre les modèles davantage performants et stables est de considérer une troisième classe appelée "zone grise" (ZG) pour laquelle l'ADS ne prend aucune décision [Grandvalet *et al.*, 2008, Hanczar et Dougherty, 2008, Clertant *et al.*, 2019, Hanczar, 2019]. Pour rappel, la LR estime la probabilité dans notre cas d'appartenir à la classe des bipolaires. Si cette probabilité est supérieure ou égale à 0.5 alors le patient sera classifié comme bipolaire par l'algorithme (équation 2.3). Néanmoins il est possible de faire varier ce seuil mais aussi d'en considérer deux au moment de la classification : les seuils de *rule in* et *rule out*.

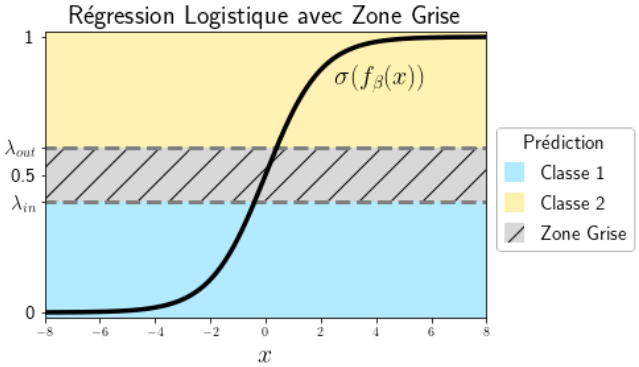

FIGURE 7.9 – Illustration de la règle de décision issue d'une Régression Logistique binaire. La sigmoïde est tracée en noire. En bleu nous avons $\hat{\mathbb{P}}(Y = 1) < \lambda_{\text{in}}$ donc $\hat{y} = 1$ et en jaune $\hat{\mathbb{P}}(Y = 1) > \lambda_{\text{out}}$ ainsi $\hat{y} = 2$. Finalement la zone hachurée grise représente la "zone grise", c'est à dire l'intervalle pour lequel aucune décision n'est préconisée par l'ADS $\lambda_{\text{in}} < \hat{\mathbb{P}}(Y = 1) < \lambda_{\text{out}}$ et pour lequel des études supplémentaires doivent être réalisées.

Le seuil de *rule out* noté λ_{out} définira le seuil à partir duquel nous classifierons les pa-

tients comme étant bipolaires (Figure 7.9, zone jaune) tandis que celui de *rule in* noté λ_{in} permettra quant à lui d'écarter l'hypothèse de bipolarité (Figure 7.9, zone bleue). Entre ces deux seuils, aucune décision n'est prise par l'algorithme ; les échantillons sont classifiés comme appartenant à la zone grise (Figure 7.9, zone hachée grise). Finalement la nouvelle règle de décision estimée par l'ADS se réécrit :

— Dépressif : $\hat{Y} = 0$ si $\hat{\mathbb{P}}(Y = 1) < \lambda_{\text{in}}$,

— Bipolaire : $\hat{Y} = 1$ si $\hat{\mathbb{P}}(Y = 1) > \lambda_{\text{out}}$,

— Zone grise : $\hat{Y} = \text{ZG}$ si $\lambda_{\text{in}} < \hat{\mathbb{P}}(Y = 1) < \lambda_{\text{out}}$.

Nous pouvons par exemple considérer que les patients en ZG pourront subir davantage d'analyses complémentaires par les médecins avant de ne pouvoir poser le diagnostic. Ainsi, en jouant sur les seuils de *rule in* et *rule out*, nous pouvons améliorer la spécificité du modèle, c'est à dire le taux de dépressifs bien classifiés, mais aussi sa sensibilité afin d'éviter de mal classifier les patients bipolaires. Néanmoins, puisque l'objectif de l'utilisation de l'IA est d'améliorer les parcours de soin sans avoir recours à des méthodes invasives ou coûteuses, un modèle comprenant une ZG trop importante est difficilement utilisable. En pratique, les seuils de *rule in* et *rule out* sont définis à partir des taux de spécificité et de sensibilité. Plus précisément le λ_{in} retenu est le seuil maximal en dessous duquel nous obtenons moins de 90% de Sensibilité tandis que le λ_{out} optimal est le seuil minimal nécessaire pour obtenir 90% de Spécificité.

			Vrais Labels	
			Dépressifs	**Bipolaires**
Prédictions	**Dépressifs**	**LR**	34.5	9.15
		LR PSI LIN	41.4	8.14
		KLR	45.7	8.12
		EKLR	45.3	7.79
	Zone Grise	**LR**	56.2	57.8
		LR PSI LIN	49.8	45.3
		KLR	46.4	38.7
		EKLR	46.4	38.6
	Bipolaires	**LR**	9.36	39.1
		LR PSI LIN	8.87	46.6
		KLR	7.9	53.2
		EKLR	8.17	53.5

TABLE 7.6 – Matrice de confusion des performances prédictives moyennes en présence d'une zone grise obtenues par Validation Croisée 5−folds. En bleu nous retrouvons les pourcentages de patients classifiés comme Dépressifs, en jaune ceux classifiés comme Bipolaires et en gris ceux classés en Zone Grise en fonction de leur vrai label. Aide à la lecture : pour la LR linéaire, 39% des patients bipolaires sont bien classifiés et 56% des dépressifs se trouvent en Zone Grise.

Les résultats issus de la classification avec ZG se présentent sous la forme d'une matrice de confusion telle que nous retrouvons les vraies étiquettes ainsi que celles prédites. Nous disposons de 3 classes de prédiction à savoir les dépressifs, les bipolaires et la Zone Grise pour laquelle des analyses supplémentaires sont nécessaires pour émettre un diagnostique. Dans le Tableau 7.6, nous retrouvons les pourcentages moyens d'attribution de classe obtenus pour les différentes LRs par Validation Croisée 5−folds lors de l'instauration de la Zone Grise sur les échantillons de validation à partir des modèles appris sur les bases d'apprentissage. L'instauration de cette zone d'indécision entraîne pour tous les modèles

moins d'erreurs de classification. Pour les deux classes, les erreurs de la LR avec zone grise s'élèvent à un peu plus de 9% contre près de 30% pour le modèle précédent. Nous pouvons faire le même constat pour nos contributions. Pour l'EKLR par exemple, moins de 8% des bipolaires sont classifiés comme dépressifs contre près de 29% sans zone grise. Bien que nos contributions présentent des zones grises moins grandes que la LR linéaire, aucune décision n'est prise pour plus 38% des bipolaires et 46% des dépressifs pour les KLR et EKLR.

7.5 Modèles Sexe-Spécifiques

Afin de limiter les incertitudes des ADS développés pour la médecine de précision, il est courant de développer des modèles sexe-spécifiques. Le genre des patients ayant un impact sur leur parcours de soin, il est souvent préférable d'établir des règles de décision différentes en fonction de ce paramètre lorsque cela est possible. Dans la suite de cette section nous présenterons les résultats obtenus pour l'élaboration d'ADS sexe-spécifiques.

7.5.1 Méthodologie

Tout comme pour le modèle sexes-confondus présenté précédemment, nous avons d'abord sélectionné les variables les plus discriminantes pour les deux bases de données hommes et femmes à l'aide de l'une LR régularisée avec pénalisation Elastic-Net par CV 10−folds. Nous avons dans un second temps ajouté les variables pour lesquelles le risque de *Confounding* était présent. Une fois les variables sélectionnées, nous avons pour les deux bases de données conservé les échantillons ayant des valeurs manquantes seulement pour les caractéristiques clinico-biologiques retenues afin d'optimiser le nombre d'échantillons à disposition pour l'entraînement du modèle final. Enfin, nous avons appris les modèles et sélectionné les seuils permettant de définir les ZGs comme précédemment.

Naturellement en divisant la base de données initiale en fonction du genre des patients nous disposons de beaucoup moins d'échantillons pour apprendre les modèles. Le Tableau 7.7 répertorie le nombre d'échantillons ainsi que les proportions de représentation des classes pour les différentes étapes de l'élaboration des ADS. Pour le modèle sexes confondus nous disposions de 268 patients sans valeurs manquantes dont 175 femmes et seulement 93 hommes. Une fois les patients ayant des valeurs manquantes pour les variables sélectionnées, nous disposons de seulement 181 échantillons pour l'apprentissage du modèle Femme et 97 pour celui des hommes. Il y a alors très peu d'échantillons masculins dans la base de données d'apprentissage, ce qui rend très compliqué l'élaboration d'un modèle stable.

Hommes		**Femmes**	
# Échantillons	**% Dép / Bip**	**# Échantillons**	**% Dép / Bip**
Sélection des variables			
93	58.1 / 41.9	175	54.3 / 45.7
Apprentissage du modèle final			
97	57.7 / 42.3	181	54.1 / 45.9

TABLE 7.7 – Nombre d'échantillons et proportions des classes (dépressifs et bipolaires) chez les hommes et les femmes pour les étapes de sélection de variables et d'apprentissage du modèle final.

De plus, les échantillons à disposition sont légèrement déséquilibrés, par exemple un peu

moins de 42% des hommes sont bipolaires dans la base de données utilisée pour réaliser la sélection des variables. Afin de limiter l'impact de cette sous-représentation, nous avons décidé de repondérer la fonction de coût que nous minimisons [1]. Pour les étapes de sélection de variables et de construction du modèle final, nous utilisons des LRs respectivement pénalisées par l'Elastic-Net et Ridge. La repondération appliquée à la fonction de coût d'une LR sans pénalisation pour les tâches de classification binaire (2.11) se définit comme :

$$L(\delta(\beta, x), y) = -w_1 y \log\left(\delta(x, \beta)\right) + w_2 y\left(1 - y\right) \log\left(1 - \delta(x, \beta)\right), \tag{7.2}$$

avec w_1 et w_2 les poids inversement proportionnels à la fréquence d'apparition de chaque classe ($y = 1$ et $y = 2$) dans l'échantillon :

$$w_k = \frac{N}{\sum_{i=1}^{N} \mathbb{1}_{\{Y_i = k\}}}. \tag{7.3}$$

Si $w_1 = w_2 = 1$, alors nous retrouvons la BCE classique (2.9). En revanche, si une classe est moins représentée qu'une autre, le poids qui lui sera attribuée dans la fonction de coût sera plus important. Dans notre cas, le fait de mal classifier les patients bipolaires moins présents que les dépressifs aura un impact plus important sur la valeur de la fonction de coût que nous cherchons à minimiser.

7.5.2 Modèles Finaux

Lors de la phase de sélection de variables, 8 ont été retenues chez les hommes et 7 chez les femmes. La sélection des caractéristiques clinico-biologiques permettant de différentier les dépressifs des bipolaires confirme la nécessité de développer des ADS sexes-spécifiques. Certaines variables sont à la fois retenues dans le modèle sexes-confondus mais aussi dans les deux modèles sexes-spécifiques telles que l'âge, la consommation de tabac et la prise d'antidépresseurs. En revanche, d'autres diffèrent, mettant en avant la nécessité de prendre en compte le genre des patients lorsque nous cherchons à personnaliser les parcours de soin. Aux trois variables citées précédemment s'ajoutent chez les hommes la prise de benzodiazépines ainsi que les cytokines CCL4, IL6, IL15 et IDSC30. Tandis que chez les femmes, nous avons retenu en plus la prise d'antipsychotiques, de lithium, d'anticonvulsants et la cytokine IL15.

	Hommes				**Femmes**			
	Apprentissage		**Validation**		**Apprentissage**		**Validation**	
Méthodes	**Perf.**	**AUC**	**Perf.**	**AUC**	**Perf.**	**AUC**	**Perf.**	**AUC**
LR	73.1 (4.5)	83.3 (4.0)	62.2 (3.4)	69.2 (4.5)	77.9 (3.5)	86.9 (2.7)	71.2 (3.7)	79.3 (3.6)
EBM	85.4 (4.0)	95.2 (2.0)	66.7 (2.4)	70.2 (3.4)	90.8 (5.8)	97.5 (2.3)	68.0 (4.6)	76.5 (4.2)
SATURNN	75.2 (2.5)	84.9 (2.0)	67.3 (4.9)	76.3 (4.7)	75.2 (2.3)	84.4 (2.0)	71.0 (4.3)	82.1 (4.0)
SATURNN$_\infty$	81.8 (4.3)	91.0 (2.1)	64.7 (8.7)	73.4 (6.7)	78.6 (2.5)	88.0 (0.9)	74.2 (5.2)	82.4 (4.0)
LR PSI LIN	88.4 (1.6)	95.1 (1.3)	65.3 (8.0)	77.2 (4.0)	82.2 (1.3)	91.1 (5.7)	68.4 (4.7)	80.4 (2.6)
KLR	71.9 (2.2)	80.6 (1.2)	64.7 (3.0)	77.6 (5.2)	77.9 (1.7)	84.6 (1.3)	71.6 (3.3)	80.7 (3.4)
EKLR	72.2 (2.3)	80.4 (1.1)	64.7 (3.0)	77.7 (5.5)	78.1 (1.4)	84.6 (1.2)	71.6 (3.3)	80.7 (3.4)

TABLE 7.8 – Performances prédictives et AUC moyennes (écart-types) obtenues par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation pour les modèles Hommes et Femmes.

1. Scikit-Learn propose une option *classweight='balanced'*.

Le Tableau 7.8 résume les performances globales (3.20) et AUC moyennes obtenues par Validation Croisée 5−folds pour les modèles sexes-spécifiques sur les échantillons d'apprentissage et de validation. Les modèles sexes-spécifiques obtiennent tous de meilleures AUC d'apprentissage par rapport aux modèles sexes-confondus. En effet, l'EBM atteint respectivement 95.2% et 97.5% d'AUC sur les échantillons d'apprentissage pour les modèles masculins et féminins contre 83% précédemment avec le modèle sexes-confondus. Le SATURNN quant à lui passe d'un peu moins de 80% d'AUC à plus de 84% pour les deux sexes. Ainsi, l'apprentissage engendre moins de différences de performances entre les classes. Néanmoins, ces performances sont moins élevées sur les échantillons de validation. Les modèles ont tendance à sur-apprendre du fait de la faible taille des échantillons (97 hommes et 181 femmes). L'EBM qui présentait moins de 6% de différence entre les performances d'apprentissage et de validation pour le modèle sexes-confondus, obtient un différentiel de 29% chez les hommes et 23% chez les femmes. Le même constat est fait pour la LR PSI LIN dont les différentiels sont de 14% pour le modèle féminin et 23% pour celui estimé sur la population masculine (contre moins de 4% lorsque les deux genres étaient confondus).

De plus, nous pouvons constater que les performances de généralisation sont davantage élevées chez les femmes que les hommes, du fait du plus grand nombre d'échantillons féminins disponibles pour apprendre les modèles. Pour les hommes, les meilleures performances de validation sont obtenues par l'EBM (près de 67% de bonnes classifications), bien que ce modèle soit celui présentant le plus de sur-apprentissage, après la LR PSI LIN. Néanmoins, ce modèle lèse grandement les dépressifs (seulement 50% en moyenne bien prédits contre 80% pour le bipolaires). Ainsi l'AUC de validation obtenue est avec celle de la LR linéaire la plus petite (70%). Les meilleures AUC de validation pour les hommes est obtenue par l'EKLR (77.7%), la KLR (77.6%) et la LR PSI LIN (77.2%). Pour ce qui est des modèles féminins, le SATURNN_∞ obtient les meilleures performances de généralisation (74.2%) et AUC (82.4%), équilibrant le plus les performances par classes.

Enfin, nous pouvons remarquer que les méthodes à noyau proposées dans ce manuscrit ont perdu en performances prédictives lors de l'estimation de modèles sexes-spécifiques par rapport au modèle sexes-confondus. La KLR qui obtenait plus de 75% de bonnes classifications, atteint désormais relativement moins de 72% et 65% de performances de généralisation chez les femmes et les hommes. De plus, l'EKLR obtient moins de 81% d'AUC test chez les femmes et 78% chez les hommes, tandis que l'EKLR appris sur la base de données sexes-confondus s'élevait à plus de 82%. Le nombre de transformations non linéaires opérées par ces méthodes dépend du nombre d'échantillons composant la base d'apprentissage. Ainsi, moins de comportements non linéaires ont été estimés, les bases de données sexes-spécifiques étant plus petite, expliquant cette perte de performance. Néanmoins, ce sont aussi les seules méthodes ayant gagné en stabilité. Quand tous les autres modèles obtiennent des écart-types plus élevées pour les performances de généralisation, les méthodes à noyau proposée estiment des modèles sexes-spécifiques davantage fiables. Les écart-types obtenus pour l'AUC de validation sont en revanche plus élevées, la volatilité des performances par classe est alors plus importante que pour les autres méthodes. L'instauration d'une zone grise prend alors tout son sens ici du fait de la stabilité du modèle en terme de performances globales.

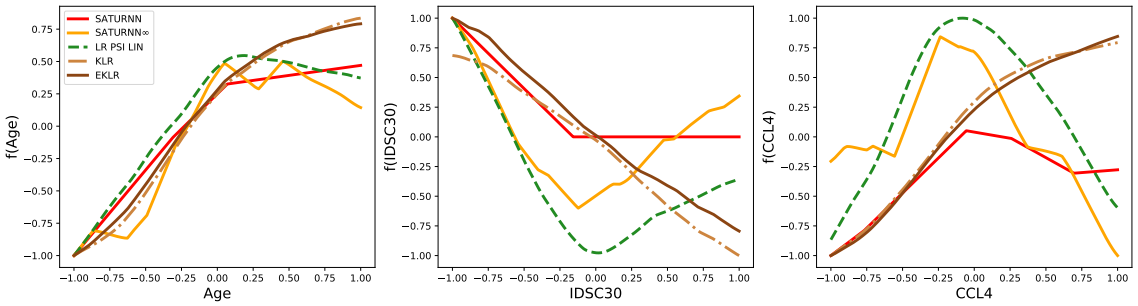

FIGURE 7.10 – Splines normalisées estimées pour le Modèle Hommes par le SATURNN (rouge), le SATURNN$_\infty$ (orange), la LR PSI LIN (vert), la KLR (marron clair) et l'EKLR (marron foncé). A gauche nous retrouvons celles estimées pour l'âge, au milieu la cytokine IDSC30 et enfin la protéine CCL4 à droite.

Étudier la stabilité et la fiabilité des modèles estimés nécessite de s'intéresser aux splines qui en résultent. Nous avons constaté précédemment que les comportements non linéaires sont estimés le plus souvent pour les variables biologiques, telles que les cytokines. Sur la Figure 7.10 nous nous intéressons alors aux splines normalisées (6.1) estimées pour les modèles masculins, comprenant plus de protéines (CCL4, IL6, IL15 et IDSC30) que les femmes (seulement IL15 et CCL11). Les splines estimées par le SATURNN sont en rouge, celles du SATURNN$_\infty$ en orange, la LR PSI LIN en vert et respectivement la KLR et l'EKLR en marron clair et foncé. Tout d'abord l'âge (Figure 7.10-Gauche) a toujours un impact croissant, les patients bipolaires composant la base de données étant plus âgés que ceux dépressifs. De plus, nous remarquons que l'impact des cytokines sur la prédiction de troubles bipolaires est non linéaire, justifiant que la LR linéaire obtienne des performances prédictives relativement plus faibles que les méthodes non linéaires. Le SATURNN$_\infty$ et la LR PSI LIN estiment qu'au plus la valeur de la protéine IDSC30 est petite, au plus les patients sont sujets à être bipolaires (Figure 7.10-Centre). Plus précisément, en dessous d'une valeur normalisée de -0.6 le patient est à risque car la spline est positive. Au dessus de 0 à peu près le risque augmente à nouveau : les courbes redeviennent croissantes sans pour autant changer de signe. La KLR et l'EKLR quant à elles estiment constamment des splines décroissantes, tel qu'au delà d'une valeur normalisée de 0, les patients sont moins sujets à développer des troubles bipolaires. Pour la protéine CCL4 (Figure 7.10-Droite), l'effet inverse est constaté pour le SATURNN$_\infty$ et la LR PSI LIN : les risques d'être bipolaire augmentent pour une petite valeur de CCL4 (jusqu'à -0.25) avant de décroître à nouveau. De plus, pour des valeurs inférieures à -0.7 à peu près et supérieures à 0.3 pour le SATURNN$_\infty$ et 0.75 pour la LR PSI LIN les splines sont négatives et donc les patients sont sujets à être dépressifs plus que bipolaires. Les splines des KLR et EKLR sont quant à elles constamment croissantes telles qu'en dessous d'une valeur normalisée de -0.2 l'impact sur la probabilité d'être bipolaire est négatif. Ainsi, pour les deux protéines, les splines des méthodes à noyaux sont linéaires. Puisque les transformations non linéaires opérées sur les variables clinico-biologiques dépendent seulement de l'échantillon d'apprentissage, la petite taille de la base de données ne permet pas aux noyaux de modéliser suffisamment d'effets de seuil.

7.5.3 Zones Grises

Les modèles sexes-spécifiques produisent des AUC de validation relativement variables (grands écart-types), les performances par classe sont instables. Ainsi, nous avons réalisé des modèles sexes-spécifiques avec zones grises (ZGs) pour rendre les modèles davantage stables. Les résultats issus de la classification avec ZG sont présentés dans le Tableau 7.9 pour les hommes et les femmes.

			Hommes Vrais Labels		**Femmes Vrais Labels**	
			Dépressifs	**Bipolaires**	**Dépressifs**	**Bipolaires**
Prédictions	**Dépressifs**	**LR**	26.3	6.7	44.2	8.26
		LR PSI LIN	31.3	5.93	50.6	8.2
		KLR	38.08	5.93	46.2	8.2
		EKLR	38.08	5.93	46.2	8.2
	Zone Grise	**LR**	66	61.8	46.9	48.16
		LR PSI LIN	62.22	47.2	41.5	49.6
		KLR	53.9	42.1	46.7	52.8
		EKLR	53.9	42.1	45.3	51.4
	Bipolaires	**LR**	7.71	31.5	8.8	43.6
		LR PSI LIN	6.48	46.9	7.9	42.15
		KLR	8.02	50.85	7.1	39.02
		EKLR	8.02	52.0	8.5	40.35

TABLE 7.9 – Matrice de confusion des performances prédictives en présence d'une zone grise obtenues par Validation Croisée 5−folds pour les modèles sexes-spécifiques (Hommes puis Femmes). En bleu nous retrouvons les pourcentages de patients classifiés comme Dépressifs, en jaune ceux classifiés comme Bipolaires et en gris ceux classés en Zone Grise en fonction de leur vrai label. Aide à la lecture : pour les hommes, les klr et eklr estiment 53.9% des dépressifs en zone grise et se trompent pour 8.2% des bipolaires.

Nous constatons dans un premier temps que pour les femmes dépressives les ZGs sont comparables à celles des modèles sexes-confondus (KLR et EKLR), voire se sont réduites pour certaines méthodes. Par exemple, la LR PSI LIN classifie 41.5% des femmes souffrant de dépressions en ZG contre plus de 49% avec le modèle sexes-confondus. En revanche, chez les femmes atteintes de troubles bipolaires et les hommes (dépressifs et bipolaires), les ZGs de nos contributions se sont agrandies. Les modèles sexes-confondus estimés par les KLR et EKLR prédisent 38.7% des patients en ZGs, quand les modèles hommes proposent une ZG supérieure à 50%.
Puisque davantage d'hommes se trouvent en zone grise, le pourcentage de bipolaires mal classifiés a diminué. Les KLR et EKLR estimées avec une zone d'indécision sur les hommes entraînent moins de 6% de mauvaises classifications des patients d'intérêt, contre respectivement plus de 7% et 8% pour les modèles sexes-confondus par les deux méthodes à noyaux. Chez les femmes, les taux de mauvaise classification des patientes atteintes de troubles bipolaires n'ont pas changé par rapport aux modèles sexes-confondus (8%). En revanche, puisque les ZGs des femmes bipolaires ont augmenté et le taux de mauvaise classification n'a pas changé, le taux de bonne classification a nécessairement baissé. Les KLR et EKLR estimées pour le modèle femme classifient correctement environ 40% des femmes bipolaires alors que plus de 53% des patients bipolaires sont bien classifiés par ces méthodes estimés sur l'ensemble des patients.
Pour les patients dépressifs, nous constatons que le taux de mal classifiées a baissé chez les hommes et les femmes pour toutes les méthodes. Par exemple, 6% des hommes et moins de 8% des femmes souffrant de dépressions sont mal classifiés par les LR PSI LIN

sexes-spécifiques contre un peu moins de 9% pour le modèle sexes-confondus. Le taux de dépressives bien classifiées par les méthodes à noyaux a augmenté et dépassent les 46%. En revanche, pour les hommes, la zone grise étant très large (près de 54% des dépressifs pour les méthodes à noyau), le taux de dépressifs bien classifiés a grandement diminué : les KLR et EKLR hommes prédisent correctement 38% des hommes dépressifs contre plus de 45% pour les modèles sexes-confondus.

Ainsi, les modèles avec ZG sexes-spécifiques ne sont pas convaincants. Par rapport aux modèles sexes-confondus, moins de patients sont directement classifiés correctement et se retrouvent en ZG. Puisque notre objectif principal est de bien prédire les bipolaires, nous proposons alors de changer les seuils de *rule in* et *rule out*. Pour les hommes, les prédictions n'ont pas pu être améliorées, les méthodes développées étant trop instables du fait de la petite taille de la base de données. En revanche, pour les femmes, des seuils fixés tels que $\lambda_{\text{in}} = 0.2$ et $\lambda_{\text{out}} = 0.4$ permettent de réduire considérablement les ZGs, sans détériorer les taux de mauvaise classification des femmes atteintes de troubles bipolaires. Favoriser la bonne prédiction des patientes bipolaires entraîne nécessairement une dégradation des performances prédictives des dépressives. Il est acceptable de dégrader légèrement la prédiction des dépressives pour réduire considérablement les ZGs. En effet, un modèle comprenant une trop large ZG ne peut être utilisée sur le terrain.

Dans le Tableau 7.10, nous retrouvons les performances obtenues par les différentes méthodes. Tout d'abord nous constatons une baisse considérable des ZGs. En effet, la LR PSI LIN classifie 14% des femmes dépressives et 13.75% des femmes bipolaires en ZG avec des seuils $\lambda_{\text{in}} = 0.2$ et $\lambda_{\text{out}} = 0.4$ contre respectivement 41.5% et 49.6% avec des seuils définis par Spécificité et Sensibilité. Les méthodes à noyau KLR et EKLR estiment des ZGs relativement plus élevées pour les femmes dépressives (un peu moins de 26%) mais plus petites pour les bipolaires (moins de 13%). Le taux de femmes présentant des troubles bipolaires bien classifiées a aussi considérablement augmenté : un peu moins de 79% pour

			Vrais Labels	
			Dépressifs	**Bipolaires**
Prédictions	**Dépressifs**	**LR**	47.2	13.92
		LR PSI LIN	51.68	8.64
		KLR	44.54	8.34
		EKLR	44.54	8.34
	Zone Grise	**LR**	18.2	5.7
		LR PSI LIN	14.04	13.75
		KLR	25.9	12.8
		EKLR	25.9	12.8
	Bipolaires	**LR**	34.64	80.4
		LR PSI LIN	34.27	77.61
		KLR	29.55	78.9
		EKLR	29.55	78.9

TABLE 7.10 – Matrice de confusion des performances prédictives moyennes en présence d'une zone grise obtenues par Validation Croisée 5−folds avec $\lambda_{\text{in}} = 0.2$ et $\lambda_{\text{in}} = 0.4$ pour les modèles Femmes. En bleu nous retrouvons les pourcentages de patientes classifiées comme Dépressives, en jaune celles classifiées comme Bipolaires et en gris celles classées en Zone Grise en fonction de leur vrai label. Aide à la lecture : les KLR et EKLR classifient 25.9% des femmes dépressives en zone grise et prédisent correctement 78.9% des femmes bipolaires.

les KLR et EKLR et même 80% pour la LR linéaire. En revanche, le taux de femmes bipolaires mal classifiées a augmenté pour la LR (près de 14%), tandis qu'il est resté aux alentours de 8% pour les méthodes proposées dans ce manuscrit. De plus, favoriser la bonne classification de bipolaires et la minimisation des ZGs entraîne nécessairement une baisse de performance pour les femmes dépressives. En revanche les erreurs de classification pour cette catégorie de femmes est plus basse pour les méthodes à noyau (moins de 30%) que pour la LR linéaire (près de 35%). Ainsi, les résultats des méthodes à noyau estimées avec une ZG dont les seuils sont égaux à $\lambda_{\text{in}} = 0.2$ et $\lambda_{\text{out}} = 0.4$ sont prometteurs.

7.6 Synthèse et discussion

Synthèse

Dans ce chapitre nous avons présenté les résultats obtenus pour la prédiction de la bipolarité, en collaboration avec l'IPMC. Les données que nous avons à disposition ont nécessité un travail de pré-traitement important (données manquantes et *outliers*). Concernant l'apprentissage d'ADS, plusieurs constats ont pu être établis. Tout d'abord, les modèles contenant beaucoup de variables prédictives ont tendance à sur-apprendre et être peu stables. Ensuite, une fois les variables les plus discriminantes sélectionnées, nous avons remarqué que les méthodes non linéaires obtiennent des performances prédictives plus élevées que la Régression Logistique linéaire. Il est alors nécessaire de développer des ADS modélisant des effets non linéaires pour l'application à la médecine de précision. Parmi les méthodes non linéaires estimées, les KLR et EKLR sont davantage généralisables que les EBM, méthodes à apprentissage itératif. De plus, les méthodes à noyau proposées dans ce manuscrit obtiennent des performances prédictives similaires au SATURNN, bien qu'étant beaucoup plus rapides à estimer. Enfin, en terme de fiabilité d'interprétation des règles de décision, les KLR et EKLR ont estimé des effets non linéaires plus cohérents biologiquement que les EBM et SATURNN. Ainsi, les méthodes à noyau proposées dans ce manuscrit semblent davantage adaptées aux problèmes de classification appliqués à la médecine.

Il est courant d'estimer des modèles sexes-spécifiques pour l'application médicale. En effet, ces modèles peuvent à l'image des résultats obtenus sur les femmes être très performants. En revanche, comme discuté lors de l'introduction du manuscrit, il est souvent compliqué d'obtenir un modèle stable puisque la sous-division des bases de données déjà petites pour développer ces modèles selon le genre rend les échantillons d'apprentissage trop restreints.
Un deuxième procédé couramment employé pour l'application au domaine médical, est l'ajout d'une zone d'indétermination appelée Zone Grise. Nous avons remarqué que le choix des seuils utilisés pour déterminer cette zone est crucial ; si la Zone Grise est trop grande (contient beaucoup de patients), alors l'ADS ne pourra que difficilement être utilisable sur le terrain. Afin de lever l'incertitude sur le diagnostic de patients, les seuils de Zone Grise peuvent être ajustés afin de limiter la proportions d'échantillon pour lesquels aucune décision n'est prise, mais aussi ne pas manquer la classification des patients d'intérêt (dans notre cas bipolaire). Pour le modèle féminin avec Zone Grise, les KLR et EKLR sont les modèles réalisant le meilleur compromis pour la prédiction de la bipolarité.

Discussion

Lors des phases de sélection de variables, peu de cytokines sont conservées alors que l'idée était de développer des ADS à partir de la combinaison de variables cliniques et biologiques. Lorsque nous entraînons un modèle seulement avec ces caractéristiques biologiques, les performances obtenues ne sont pas satisfaisantes. Il s'avère que dans la base de données que nous avons à disposition, les patients sont d'une part identifiés comme dépressifs ou bipolaires, mais sont d'autre part suivis et reçoivent un traitement pour leur pathologie. C'est pourquoi les traitements par antidépresseurs, lithium, benzodiazépines ou encore anticonvulsants sont si importants dans les modèles. Puisque ces médicaments impactent leur caractéristiques biologiques en stabilisant leur humeur et donc en altérant la valeur de leurs cytokines, il n'est pas possible de différentier ces deux catégories à l'aide de ces protéines. Pour qu'un ADS puisse différentier les patients à partir de ces cytokines il conviendrait de les mesurer avant la prise d'un traitement adéquat. Or cela nécessiterait de devoir suivre un grand nombre de patients sur une longue durée, en attendant que certains se déclarent, soit dépressifs, soit bipolaires. Néanmoins, constituer ces bases de données demande beaucoup de temps mais aussi des moyens financiers importants.

Chapitre 8

Conclusion et Travaux Futurs

Ce chapitre synthétise dans un premier temps les différents challenges et objectifs (Section 8.1) de la thèse. Dans un second temps, la Section 8.2 détaille les différentes contributions théoriques présentées dans ce manuscrit. Nous proposons en Section 8.3 différentes pistes de recherche afin de renforcer, améliorer et compléter les différentes méthodes développées pendant la thèse. Pour finir, la Section 8.4 liste les différents articles de recherche liés aux travaux de thèse mais aussi aux collaborations.

Sommaire

8.1 Synthèse du contexte et des objectifs

L'utilisation de l'intelligence artificielle pour la médecine de précision est de plus en plus répandue. La médecine de précision vise à personnaliser les parcours de soin des patients en fonction des caractéristiques clinico-biologiques qui leur sont propres. L'apprentissage automatique est de ce fait un bon outil d'aide à la décision pour un médecin. L'idée n'est pas de remplacer les experts du domaine mais de leur fournir des outils leur permettant de simplifier leur diagnostic. Le développement d'Algorithmes de Support de Décision (ADS) par les méthodes de *Machine Learning* (ML) est répandu notamment pour répondre à deux principaux objectifs. Le premier consiste naturellement à prédire des pathologies. Nous nous sommes concentrés dans ce manuscrit sur l'exploitation de données clinico-biologiques. De part nos échanges avec des Professeurs d'Immunologie et d'Hépatologie, nous nous sommes rendus compte qu'il est nécessaire pour prédire certaines pathologies, d'acquérir des données par des méthodes qui sont à la fois coûteuses pour les systèmes hospitaliers, mais aussi invasives et douloureuses pour les patients (biopsies par exemple). Les ADS pourraient alors être utiles pour écarter toute indétermination sur un grand nombre de patients, pour lesquels des caractéristiques clinico-biologiques facilement calculables suffiraient. Ainsi, il ne resterait qu'une partie des patients pour lesquels des études complémentaires, par méthodes invasives notamment, seraient nécessaires. Le second objectif pour lequel le ML est de plus en plus utilisé en médecine de précision consiste à personnaliser des traitements. Comme détaillé en Introduction, il y a des pathologies qu'il est difficile de diagnostiquer, mais aussi pour lesquelles l'ajustement d'un traitement adéquat prend du temps. Ainsi les ADS développés dans ce but visent à fournir des recommandations de traitements en fonction des caractéristiques clinico-biologiques de chaque patient.

Afin de répondre aux attentes des praticiens, il est nécessaire d'adapter les méthodes de ML aux problématiques bien spécifiques rencontrées lorsque nous travaillons avec des données bio-médicales. Dans un premier temps, les bases de données à disposition sont d'une part relativement petites (peu d'échantillons) et déséquilibrées. Cette sous-représentation de certaines catégories de patients dans les jeux d'apprentissage peut entraîner indirectement des discriminations. Dans ce manuscrit, lorsque nous avons présenté les résultats issus de nos collaborations nous avons par exemple pris le parti de créer des modèles sexe-spécifiques afin de ne pas léser une certaine population sous-représentée. La deuxième grande difficulté pour l'application du ML à la médecine de précision réside dans la nécessité de développer des modèles éthiques et fiables. Puisqu'il s'agit de personnaliser des parcours de soin de patients, il est alors primordial de développer des algorithmes performants mais aussi interprétables. Il n'est pas concevable pour un médecin de prendre appui sur une prédiction issue d'un algorithme, dont on ne peut expliquer le cheminement qui a mené à une décision précise, plutôt qu'à une autre. Or dans le ML il y a souvent un compromis à faire entre performances prédictives et interprétabilité des modèles. Les méthodes souvent les plus performantes, telles que les réseaux de neurones sont souvent qualifiées de "boîtes noires" tant il est difficile de comprendre la règle de décision estimée.

Dans ce manuscrit nous nous sommes intéressés au développement d'une méthode de classification adaptée aux problématiques médicales. Dans un premier temps la méthode développée doit modéliser des effets non linéaires. En effet, les experts des domaines médicaux sont convaincus, qu'intégrer des effets de seuils par exemple dans les règles de décision

pourrait permettre de gagner en performances prédictives. L'impact de certaines variables clinico-biologiques pour la prédiction d'une pathologie ou d'un traitement adéquat est non linéaires. Si nous prenons l'Indice de Masse Corporelle (IMC) par exemple, il existe un seuil au-delà duquel le patient sera atteint de co-morbidité, mais aussi un seuil en dessous duquel l'individu est considéré en sous-poids. Il existe alors deux intervalles pour lesquels la variable IMC pourra être un facteur à risque, d'où la nécessité de segmenter les variables et intégrer des effets de seuil dans la modélisation. La seconde nécessité de la méthode développée consiste à pouvoir interpréter le cheminement menant à la décision. Il est nécessaire de pouvoir visualiser les différentes transformations non linéaires opérées par la méthode sur les données afin que les médecins puissent contrôler la pertinence biologique de la règle de décision. Beaucoup de méthodes de l'état de l'art modélisent des effets non linéaires univariés tels que pour chaque variable nous puissions observer les transformations, et ainsi comprendre leur impact respectif sur la règle de décision. Néanmoins, ces méthodes utilisent des algorithmes d'apprentissage itératifs, dont l'optimalité globale ne peut être établie et dont la fiabilité peut alors être discutée. La méthode développée doit être optimisée par un algorithme offrant des garanties d'unicité des résultats afin d'écarter tout effet aléatoire lié par exemple à l'initialisation des paramètres estimés. Finalement, l'ADS développé nécessite la modélisation d'une règle de décision non linéaire interprétable et unique conditionnellement à l'échantillon d'apprentissage.

8.2 Synthèse de nos contributions

Dans le Chapitre 3, nous avons tout d'abord introduit le modèle RN-MARS. Ce Réseau de Neurones (RN) s'inspire des avantages des méthodes de classification non linéaires itératives de l'état de l'art présentées dans le Chapitre 2, tout en s'affranchissant de leur limite d'optimisation. La règle de décision de ce RN se modélise comme celle issue d'une Régression Logistique (LR) non linéaire et est estimée dans sa globalité. Le nombre de neurones composant ce RN est fixé afin de contrôler le partitionnement de l'espace d'entrée. En effet, le nombre de transformations non linéaires appliquées aux données est limitée, tel qu'au maximum deux intervalles soient créés par caractéristique clinico-biologique. De plus l'architecture du RN-MARS est contrainte. Contrairement aux RN ReLU traditionnels qui mélangent toutes les variables entre elles, le RN-MARS traite chaque variable indépendamment les unes des autres. La segmentation de l'espace d'entrée se fait ainsi par des orthotopes facilement interprétables. Nous avons également introduit dans ce chapitre le SATURNN pour *Splines Approximation Throught Univariate ReLU Neural Network*, une généralisation du RN-MARS. Il arrive que segmenter davantage chaque variable descriptive et ne pas se limiter à seulement deux intervalles permette de gagner en performance prédictive. Ainsi le SATURNN se compose d'une couche cachée de p neurones auxquels est appliquée l'activation ReLU et d'une couche de sortie Sigmoïde afin de répondre à un problème de classification binaire. Comme pour le RN-MARS les variables explicatives sont segmentées individuellement. Pour ce faire, chaque neurone composant la couche cachée du SATURNN se voit attribuer aléatoirement une variable, à laquelle il applique une transformation ReLU. Toutes les variables ont la même probabilité d'être tirées au sort afin que le partitionnement induit par le SATURNN soit contrôlé.

Comme pour tous les RN ReLU, estimer le RN-MARS et le SATURNN revient à minimiser un problème d'optimisation non convexe. Ainsi, bien qu'interprétables, ces réseaux n'offrent aucune garantie de convergence vers une solution unique. Nous avons

alors proposé dans le Chapitre 4 de linéariser partiellement le SATURNN par rapport à ses paramètres. Nous savions de part la littérature qu'il n'était pas possible de le linéariser dans sa globalité. Nous nous sommes alors concentrés sur la linéarisation de sa couche cachée aussi appelée fonction de score. Nous avons par la suite démontré que la composition par la Sigmoïde de la fonction de score linéarisée ne change en rien la qualité de l'approximation. Plus précisément il est établi que l'erreur d'approximation du SATURNN par la sigmoïde appliquée à la fonction de score linéarisée tend à devenir nulle à une vitesse $O(\frac{1}{\sqrt{p}})$. Ensuite, nous avons démontré qu'à mesure que p augmente il devient équivalent d'entraîner le SATURNN ou une LR appliquée à la fonction de score linéarisée (LR PSI LIN). La LR est entraînée sur des données préalablement transformées par l'application non linéaire découlant directement de l'architecture du SATURNN, et plus précisément de sa fonction de score linéarisée. Après entraînement, il est possible de réinjecter les paramètres estimés par la LR PSI LIN pour retrouver la règle de décision du SATURNN associé et visualiser les splines univariées. L'avantage d'entraîner la LR PSI LIN plutôt que le SATURNN ne se résume pas seulement à sa rapidité d'apprentissage. Lorsque nous entraînons la LR PSI LIN en ajoutant une contrainte sur ses paramètres (régularisation ℓ_2), nous minimisons un problème d'optimisation fortement convexe. Ainsi, les estimations résultant de la LR PSI LIN et de ce fait du SATURNN associé sont uniques. La LR PSI LIN répond ainsi aux grands objectifs énumérés précédemment. En approximant le SATURNN par la LR PSI LIN nous disposons d'une règle de décision non linéaire, interprétable et unique. Néanmoins, la LR PSI LIN reste dépendante du processus d'entraînement du SATURNN que l'on souhaite approximer. La fonction de score linéarisée appliquée aux données avant l'apprentissage de la LR est dépendante des initialisations du SATURNN. La règle de décision estimée par la LR PSI LIN est alors unique conditionnellement aux paramètres initialisés du SATURNN que nous souhaitons approximer.

Dans le Chapitre 5, nous démontrons qu'il est possible de faire apparaître un noyau dans la LR PSI LIN. Il devient alors équivalent d'entraîner la LR PSI LIN ou une Régression Logistique à Noyau (KLR). Tout comme pour la LR PSI LIN, la KLR applique une LR aux données préalablement transformées à travers l'application d'un noyau κ_0. Cette transformation non linéaire des données dépend du nombre p de neurones composant le SATURNN que l'on souhaite approximer mais aussi du nombre d'échantillons composant la base d'entraînement. Une fois la KLR entraînée, il est aussi possible de retrouver la règle de décision du SATURNN associé. Néanmoins le noyau κ_0 proposé reste dépendant des initialisations du SATURNN. Ainsi, la règle de décision issue de la KLR bien qu'interprétable est une fois encore unique conditionnellement à l'initialisation du SATURNN associé. Afin de s'affranchir de cette limite nous avons démontré par la Loi Forte des Grands Nombres que le noyau κ_0 converge asymptotiquement pour un p suffisamment grand vers son espérance κ. Ce nouveau noyau κ est quant à lui totalement indépendant du processus d'entraînement du SATURNN que l'on souhaite approximer. Ainsi, notre principale contribution du manuscrit est l'introduction d'une Régression Logistique à Noyau Déterministe (EKLR) pouvant approximer un SATURNN composé d'un grand nombre p de neurones. L'EKLR ne dépendant plus des paramètres initiaux du SATURNN, il n'est pas possible de réinjecter ses paramètres estimés pour retrouver le SATURNN approximé. Néanmoins la transformation non linéaire des données κ découle directement de l'architecture du SATURNN. Il en résulte que la règle de décision estimée par l'EKLR peut se réécrire comme une somme additive de fonctions splines univariées. Enfin, l'optimisation de l'EKLR avec une régularisation ℓ_2 offre des

garanties de convergence et donc d'unicité des résultats. Ainsi lorsque nous considérons un SATURNN composé d'un suffisamment grand nombre de neurones p, il est possible de l'approximer correctement pas l'EKLR dont (i) le noyau découle directement du SATURNN bien qu'étant totalement indépendant de son processus d'entraînement, (ii) la règle de décision issue est interprétable car se modélisant comme une somme des splines univariées, (iii) nous disposons de garanties d'unicité des estimations conditionnellement cette fois seulement à l'échantillon d'apprentissage.

Dans chaque chapitre détaillant les contributions de la thèse nous avons réalisé des expériences sur données simulées afin de valider les résultats théoriques établis. Les Chapitres 6 et 7 sont consacrés à la présentation de résultats expérimentaux sur données réelles.
Plus précisément, dans le Chapitre 6, nous avons comparé les performances prédictives et la fiabilité des algorithmes proposés aux méthodes de l'état de l'art sur trois jeux de données publics. Les bases de données choisies ont toutes pour objectif de prédire des pathologies (Accident Vasculaire Cérébral, Diabète et Parkinson), mais se démarquent de part la disponibilité des données (nombres d'échantillons et de variables). Lorsque les échantillons d'apprentissage sont suffisamment nombreux, les méthodes proposées dans le manuscrit obtiennent des performances prédictives similaires à celles des méthodes non linéaires de l'état de l'art. Néanmoins, nos contributions se distinguent sur deux points. D'une part, les méthodes proposées dans ce manuscrit semblent davantage stables, car elles ne présentent pas de sur-apprentissage contrairement aux méthodes à apprentissage glouton de l'état de l'art. D'autre part, l'interprétation des règles de décision des méthodes à noyau proposées (KLR et EKLR) est davantage fiable. En effet, les méthodes ne disposant pas de garantie de convergence (MARS, GAM, RF, EBM ou RN) estiment des effets non linéaires parfois peu cohérents biologiquement. Les méthodes à noyau proposées KLR et EKLR confirment la nécessité de développer des méthodes explicables pour l'application médicale. En revanche, lorsque peu d'échantillons sont disponibles dans les bases de données nous avons constaté que les méthodes à noyau atteignent des performances prédictives globales plus faibles que celles des méthodes de l'état de l'art. Les transformations non linéaires opérées par les noyaux dépendent directement du nombre d'échantillons d'apprentissage disponibles. Ainsi, lorsque nous disposons de trop peu de données, les méthodes proposées ont davantage de difficultés à modéliser des effets non linéaires.
Finalement, dans le Chapitre 7, nous avons introduit les résultats obtenus pour la prédiction de la bipolarité, en collaboration avec l'IPMC. Nous avons comparé les performances prédictives et l'interprétabilité de nos contributions à une méthode linéaire (Régression Logistique) et une autre modélisant des effets non linéaires (EBM). Puisque tous les modèles composés d'un grand nombre de variables présentent du sur-apprentissage et des résultats peu stables, nous avons réalisé une sélection des variables les plus discriminantes. Pour ce faire, nous avons utilisé la pénalisation Elastic-Net tout en faisant attention que le problème de *Confounding* souvent rencontré dans l'application médicale n'intervienne pas. La première conclusion que nous avons pu faire est que les méthodes proposées dans ce manuscrit atteignent des performances prédictives similaires aux méthodes non linéaires boostées (EBM) ou aux RN (SATURNN), bien qu'étant plus stables. De plus, les effets non linéaires estimés par l'EBM et le SATURNN révèlent des incohérences biologiques. L'interprétation des règles de décision des KLR et EKLR est là encore davantage fiable. Nous avons intégré aux méthodes explicables proposées dans ce manuscrit une Zone Grise, c'est à dire une zone d'indécision pour laquelle l'algorithme n'émet pas de diagnostic. Les patients se retrouvant dans cette zone nécessitent des études complémentaires (par méthodes invasives par exemple) pour pouvoir être correctement classifiés. Les méthodes à noyau proposées

dans ce manuscrit, KLR et EKLR offrent le meilleur compromis entre minimisation de la zone grise et maximisation des bonnes classifications des patients bipolaires. Enfin, nous avons entraîné des modèles sexes-spécifiques. Les hommes étant trop peu nombreux, les méthodes testées n'ont pas été en mesure d'améliorer leurs résultats. En revanche, pour les femmes, les résultats issus des méthodes proposées KLR et EKLR avec Zone Grise sont très prometteurs.

8.3 Perspectives futures

Le SATURNN et son approximation par l'EKLR offrent de nombreuses pistes de recherche tant sur le plan théorique que sur le plan applicatif que nous détaillons dans cette section.

8.3.1 Travaux théoriques futurs

Nous aimerions par la suite étudier davantage certaines propriétés de l'EKLR. Tout d'abord, étudier le noyau proposé pourrait être intéressant. Cette application non linéaire pourrait avoir des connections avec des noyaux traditionnellement utilisés. De plus, nous avons établi que la règle de décision issue de l'EKLR se réécrit comme une somme additive de splines univariées. Il pourrait être intéressant d'étudier davantage la forme de ces splines afin de faire le lien avec celles présentées dans l'état de l'art de ce manuscrit.
Ensuite, le noyau κ introduit dans le Chapitre 5 dépend du nombre p de neurones composant le SATURNN que nous souhaitons approximer mais surtout des échantillons d'apprentissage. Nous avons démontré dans le Chapitre 6 que cette méthode d'approximation obtient des performances prédictives similaires aux méthodes de l'état de l'art tout en s'affranchissant de leurs contraintes sur deux bases de données relativement grandes. En revanche, sur une plus petite base de données nous avons pu constater que les performances obtenues sont inférieures à celles espérées mais aussi que les splines résultantes ont des comportements linéaires moins prononcés. Puisque ces transformations non linéaires découlent directement du noyau κ et donc du nombre d'échantillons composant la base d'entraînement, nous supposons que la petite taille de la base de données en est à l'origine. Il pourrait alors être intéressant d'étudier théoriquement l'impact de la taille de la base de données sur le pouvoir d'approximation du SATURNN par l'EKLR. Nous pourrions comparer la fonction de coût du SATURNN à celle de l'EKLR, comme réalisé pour le Théorème 2. Cette fois nous étudierons néanmoins une borne en fonction du nombre d'échantillons et non pas seulement en fonction de p.

Nous souhaiterions par la suite ajouter à l'EKLR la possibilité de sélectionner les variables les plus pertinentes. Néanmoins, l'application directe d'une pénalisation Lasso n'est pas envisageable dans notre cas pour deux raisons : (i) les paramètres appris par l'EKLR sont associés à un échantillon composant le jeu d'entraînement et non pas une variable, et (ii) les échantillons étant transformés par l'application noyau, nous ne cherchons pas simplement à sélectionner une variable, mais une fonction de variable (spline). Dans les résultats présentés en collaboration avec l'IPMC nous avons sélectionné les variables d'intérêt avec un modèle linéaire (RL) et ensuite appliquer notre méthode. Nous pensons que sélectionner directement les variables transformées non linéairement permettrait d'améliorer les résultats prédictifs ; avec le modèle linéaire nous avons

certainement sélectionné des variables qui n'auraient pas été retenues par une méthode de sélection non linéaire et à l'inverse écarté des caractéristiques non linéaires qui auraient permis d'améliorer les performances prédictives. Ainsi, nous aimerions nous intéresser aux travaux élaborés dans la littérature quant à la sélection de variables fonctionnelles (*functional features*) afin d'apprendre en une seule étape une sélection de variables pertinente mais aussi la règle de décision qui en découle, pour gagner en performance prédictive.

Enfin, tout ce travail a été développé dans le cadre de la classification binaire. La prédiction de deux classes peut s'avérer limité notamment lorsque nous travaillons sur la personnalisation d'un traitement médical. Ainsi, il serait intéressant par la suite de pouvoir reprendre le cheminement théorique à partir de la linéarisation partielle du SATURNN mais cette fois pour l'application multi-classes. Nous avons l'intuition que remplacer la couche de sortie Sigmoïde par une couche *Softmax* dans le SATURNN devrait légèrement compliquer l'étude. Néanmoins approximer un SATURNN pour la classification multi-classes par une méthode à noyau déterministe semble réalisable.

8.3.2 Travaux futurs pour l'application médicale

En médecine de précision, modéliser des effets non linéaires est crucial pour prédire des pathologies ou des traitements adéquats. Créer des effets de seuils est intéressant, mais créer des effets de ratio pourrait aussi permettre d'augmenter les performances prédictives des algorithmes. Par exemple, le poids et la taille sont certes des caractéristiques clinico-biologiques pertinentes pour la prédiction de pathologie, mais c'est surtout l'Indice de Masse Corporelle (poids (kg) / taille^2 (m)) qui est le plus utilisé, car le plus informateur. Il pourrait alors être intéressant de pouvoir modéliser des effets d'intéraction entre les variables. Ajouter une deuxième couche au SATURNN pourrait être une piste intéressante, mais seulement si elle ne remet pas en question l'interprétabilité de la règle de décision qui en découle.

Toujours dans l'optique de ne pas se restreindre à la prédiction de deux classes, il pourrait être intéressant d'améliorer la classification avec Zone Grise. En effet, pour l'application médicale, il est courant d'ajouter une zone d'incertitude dans les problèmes de classification binaire. Lors de la présentation des résultats issus des collaborations, nous avons mis en évidence que définir des seuils de *rule in* et *rule out* à partir de la Sensibilité et la Spécificité du classifieur n'était pas toujours optimal. En effet, il serait avantageux de pouvoir optimiser les seuils définissant la Zone Grise en fonction de plusieurs paramètres. Dans [Hanczar et Dougherty, 2008, Hanczar, 2019], les auteurs proposent un problème d'optimisation sous contrainte telles que les performances par classe soient maximisées et la Zone Grise minimisée. Nous pourrions sinon imaginer une fonction de coût visant à minimiser les taux de mauvaises classification par classe mais aussi les proportions de patients se trouvant en Zone Grise. Cette nouvelle règle d'optimisation pourrait aussi favoriser la classe d'intérêt au détriment d'une perte de performance globale. En effet, si nous prenons l'exemple de la prédiction de la pathologie, il est préférable pour les experts du domaine de bien classifier les patients malades, même si cela engendre davantage de fausses alertes, à savoir prédire un patient sain comme étant malade.

8.4 Liste des publications

Cette dernière Section répertorie les différents articles de recherche découlant des travaux liés à la thèse (puces ■) mais aussi aux collaborations (puces □).

8.4.1 Revue Scientifique de Machine Learning

□ Cyprien Gilet, Marie Guyomard, Sébastien Destercke, Lionel Fillatre, Softmin discrete minimax classifier for imbalanced classes and prior probability shifts, *Machine Learning, Springer*, 2023.

8.4.2 Conférences de Machine Learning Internationales

■ Marie Guyomard, Susana Barbosa, Lionel Fillatre, Kernel Logistic Regression Approximation of an Understandable ReLU Neural Network, *International Conference on Machine Learning (ICML)*, 2023.

■ Marie Guyomard, Susana Barbosa, Lionel Fillatre, Understandable ReLU Neural Network for signal classification, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.

□ Cyprien Gilet, Marie Guyomard, Susana Barbosa, Lionel Fillatre, Multiclass Minimax Learning for Deep Neural Networks, *European Signal Processing Conference (EUSIPCO)*, 2023.

□ Cyprien Gilet, Marie Guyomard, Susana Barbosa, Lionel Fillatre, Adjusting Decision Trees for Uncertain Class Proportions, *Workshop WUML at ECML*, 2020.

8.4.3 Conférences de Machine Learning Françaises

■ Marie Guyomard, Susana Barbosa, Lionel Fillatre, Régression Logistique à Noyau Équivalente à un Réseau de Neurones Interprétable, *Groupe de Recherche et d'Etudes de Traitement du Signal et des Images (Gretsi)*, 2023.

□ Cyprien Gilet, Marie Guyomard, Sébastien Destercke, Lionel Fillatre, Classifieur minimax discret randomisé pour la détection de classes rares et la présence de probabilités a priori imprécises, *Groupe de Recherche et d'Etudes de Traitement du Signal et des Images (Gretsi)*, 2023.

■ Marie Guyomard, Susana Barbosa, Lionel Fillatre, Approximation d'un Réseau de Neurones ReLU interprétable par une Régression Logistique à Noyau, *Journées de la Statistique (JDS)*, 2023.

□ Cyprien Gilet, Marie Guyomard, Sébastien Destercke, Lionel Fillatre, Apprentissage d'un classifieur Minimax Randomisé pour risques d'erreur par classe déséquilibrés et probabilités a-priori incertaines, *Journées de la Statistique (JDS)*, 2023.

■ Marie Guyomard, Susana Barbosa, Lionel Fillatre, Adaptive splines-based logistic regression with a ReLU neural network, *Journées ouvertes en biologie, informatique et mathématiques (JOBIM)*, 2022.

- ■ Marie Guyomard, Susana Barbosa, Lionel Fillatre, Régression logistique à base de splines adaptatives avec un réseau de neurones ReLU, *Groupe de Recherche et d'Etudes de Traitement du Signal et des Images (Gretsi)*, 2022.
- □ Cyprien Gilet, Marie Guyomard, Susana Barbosa, Lionel Fillatre, Apprentissage minimax pour les réseaux de neurones, *Groupe de Recherche et d'Etudes de Traitement du Signal et des Images (Gretsi)*, 2022.
- ■ Marie Guyomard, Cyprien Gilet, Susana Barbosa, Lionel Fillatre, Sur l'équivalence entre la régression logistique à base de splines et l'apprentissage profond, *Congrès des Jeunes Chercheuses et Chercheurs en Mathématiques Appliquées (CJC-MA)*, 2021.
- □ Marie Guyomard, Cyprien Gilet, Susana Barbosa, Lionel Fillatre, Réseaux de Neurones Convolutifs avec Apprentissage Minimax pour des Proportions par classe incertaines et déséquilibrées, *Journées francophones des jeunes chercheurs en vision par ordinateur (ORASIS)*, 2021.

8.4.4 Conférence Médicale Française

- □ Marie Guyomard, Dann Ouizeman, Renaud Schiappa, Cyprien Gilet, Jocelyn Gal, Emmanuel Chamorey, Stéphanie Patouraux, Thierry Piche, Albert Tran, Philippe Gual, Antonio Iannelli, Lionel Fillatre, Rodolphe Anty, Diagnostic non invasif de la NASH fibrosante à l'aide de l'intelligence artificielle, *AFEF (Société Française d'Hépatologie)*, 2020.

Bibliographie

[Adadi et Berrada, 2018] Adadi, A. et Berrada, M. (2018). Peeking inside the black-box : a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.

[Adam *et al.*, 2020] Adam, G., Rampášek, L., Safikhani, Z., Smirnov, P., Haibe-Kains, B. et Goldenberg, A. (2020). Machine learning approaches to drug response prediction : challenges and recent progress. *NPJ precision oncology*, 4(1):19.

[Agarwal *et al.*, 2021] Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R. et Hinton, G. E. (2021). Neural additive models : Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34:4699–4711.

[Ahmad *et al.*, 2018] Ahmad, M. A., Eckert, C. et Teredesai, A. (2018). Interpretable machine learning in healthcare. *In Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 559–560.

[Albahri *et al.*, 2023] Albahri, A., Duhaim, A. M., Fadhel, M. A., Alnoor, A., Baqer, N. S., Alzubaidi, L., Albahri, O., Alamoodi, A., Bai, J., Salhi, A. *et al.* (2023). A systematic review of trustworthy and explainable artificial intelligence in healthcare : Assessment of quality, bias risk, and data fusion. *Information Fusion*.

[Ali *et al.*, 2023] Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., Guidotti, R., Del Ser, J., Díaz-Rodríguez, N. et Herrera, F. (2023). Explainable artificial intelligence (xai) : What we know and what is left to attain trustworthy artificial intelligence. *Information fusion*, 99:101805.

[Alshamaa *et al.*, 2018] Alshamaa, D., Chehade, F. M. et Honeine, P. (2018). A hierarchical classification method using belief functions. *Signal Processing*, 148:68–77.

[Alshamaa *et al.*, 2017] Alshamaa, D., Mourad-Chehade, F. et Honeine, P. (2017). Classification paramétrique multi-classes à croyance. *In Actes du 26-ème Colloque GRETSI sur le Traitement du Signal et des Images*.

[Amos *et al.*, 2017] Amos, B., Xu, L. et Kolter, J. Z. (2017). Input convex neural networks. *In International Conference on Machine Learning*, pages 146–155. PMLR.

[Apley et Zhu, 2020] Apley, D. W. et Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B : Statistical Methodology*, 82(4):1059–1086.

[Aronszajn, 1950] Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404.

[Aung *et al.*, 2021] Aung, Y. Y., Wong, D. C. et Ting, D. S. (2021). The promise of artificial intelligence : a review of the opportunities and challenges of artificial intelligence in healthcare. *British medical bulletin*, 139(1):4–15.

[Balestriero *et al.*, 2018] Balestriero, R. *et al.* (2018). A spline theory of deep learning. *In International Conference on Machine Learning*, pages 374–383. PMLR.

[Balestriero *et al.*, 2019] Balestriero, R., Cosentino, R., Aazhang, B. et Baraniuk, R. (2019). The geometry of deep networks : Power diagram subdivision. *Advances in Neural Information Processing Systems*, 32.

[Beaudouin *et al.*, 2020] Beaudouin, V., Bloch, I., Bounie, D., Clémençon, S., d'Alché Buc, F., Eagan, J., Maxwell, W., Mozharovskyi, P. et Parekh, J. (2020). Flexible and context-specific ai explainability : a multidisciplinary approach. *arXiv preprint arXiv :2003.07703*.

[Bengio *et al.*, 2005] Bengio, Y., Roux, N., Vincent, P., Delalleau, O. et Marcotte, P. (2005). Convex neural networks. *Advances in neural information processing systems*, 18.

[Bermudez *et al.*, 2015] Bermudez, J. C., Honeine, P., Tourneret, J.-Y. et Richard, C. (2015). Kernel-based nonlinear signal processing.

[Bharati *et al.*, 2023] Bharati, S., Mondal, M. R. H. et Podder, P. (2023). A review on explainable artificial intelligence for healthcare : Why, how, and when ? *IEEE Transactions on Artificial Intelligence*.

[Boyd *et al.*, 2004] Boyd, S., Boyd, S. P. et Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

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

[Breiman, 2017] Breiman, L. (2017). *Classification and regression trees*. Routledge.

[Brutzkus *et al.*, 2017] Brutzkus, A., Globerson, A., Malach, E. et Shalev-Shwartz, S. (2017). Sgd learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv :1710.10174*.

[Cabitza *et al.*, 2019] Cabitza, F., Campagner, A. et Ciucci, D. (2019). New frontiers in explainable ai : understanding the gi to interpret the go. *In Machine Learning and Knowledge Extraction : Third IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2019, Canterbury, UK, August 26–29, 2019, Proceedings 3*, pages 27–47. Springer.

[Chaddad *et al.*, 2023] Chaddad, A., Peng, J., Xu, J. et Bouridane, A. (2023). Survey of explainable ai techniques in healthcare. *Sensors*, 23(2):634.

[Cho et Saul, 2009] Cho, Y. et Saul, L. (2009). Kernel methods for deep learning. *Advances in neural information processing systems*, 22.

[Clertant *et al.*, 2019] Clertant, M., Sokolovska, N., Chevaleyre, Y. et Hanczar, B. (2019). Interpretable cascade classifiers with abstention. *In The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2312–2320. PMLR.

[Cortes et Vapnik, 1995] Cortes, C. et Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.

[Costello *et al.*, 2014] Costello, J. C., Heiser, L. M., Georgii, E., Gönen, M., Menden, M. P., Wang, N. J., Bansal, M., Ammad-Ud-Din, M., Hintsanen, P., Khan, S. A. *et al.* (2014). A community effort to assess and improve drug sensitivity prediction algorithms. *Nature biotechnology*, 32(12):1202–1212.

[Daubechies *et al.*, 2022] Daubechies, I., DeVore, R., Foucart, S., Hanin, B. et Petrova, G. (2022). Nonlinear approximation and (deep) relu networks. *Constructive Approximation*, 55(1):127–172.

[De Boor, 1978] De Boor, C. (1978). *A practical guide to splines*, volume 27. springer-verlag New York.

[Delacour *et al.*, 2005] Delacour, H., Servonnet, A., Perrot, A., Vigezzi, J. et Ramirez, J. (2005). La courbe roc (receiver operating characteristic) : principes et principales applications en biologie clinique. *In Annales de biologie clinique*, volume 63, pages 145–154.

[Doshi-Velez et Kim, 2017] Doshi-Velez, F. et Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv :1702.08608*.

[Dwivedi *et al.*, 2023] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G. *et al.* (2023). Explainable ai (xai) : Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9):1–33.

[Eckle et Schmidt-Hieber, 2019] Eckle, K. et Schmidt-Hieber, J. (2019). A comparison of deep networks with relu activation function and linear spline-type methods. *Neural Networks*, 110:232–242.

[Esteva *et al.*, 2017] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. et Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118.

[Eyal *et al.*, 2019] Eyal, G., Sabatello, M., Tabb, K., Adams, R., Jones, M., Lichtenberg, F. R., Nelson, A., Ochsner, K., Rowe, J., Stiles, D. *et al.* (2019). The physician–patient relationship in the age of precision medicine. *Genetics in Medicine*, 21(4):813–815.

[Fel et Vigouroux, 2020] Fel, T. et Vigouroux, D. (2020). Representativity and consistency measures for deep neural network explanations.

[Fisher *et al.*, 2018] Fisher, A., Rudin, C. et Dominici, F. (2018). Model class reliance : Variable importance measures for any machine learning model class, from the "rashomon" perspective. *arXiv preprint arXiv :1801.01489*, 68.

[Friedman, 1991] Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67.

[Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation : a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

[Gilpin *et al.*, 2018] Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M. et 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.

[Goldstein *et al.*, 2015] Goldstein, A., Kapelner, A., Bleich, J. et Pitkin, E. (2015). Peeking inside the black box : Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, 24(1):44–65.

[Goodfellow *et al.*, 2020] Goodfellow, I., Bengio, Y. et Courville, A. (2020). Deep learning book, 2018.

[Grandvalet *et al.*, 2008] Grandvalet, Y., Rakotomamonjy, A., Keshet, J. et Canu, S. (2008). Support vector machines with a reject option. *Advances in neural information processing systems*, 21.

[Greenland et Morgenstern, 2001] Greenland, S. et Morgenstern, H. (2001). Confounding in health research. *Annual review of public health*, 22(1):189–212.

[Guidotti *et al.*, 2018] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F. et Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42.

[Gunning et Aha, 2019] Gunning, D. et Aha, D. (2019). Darpa's explainable artificial intelligence (xai) program. *AI magazine*, 40(2):44–58.

[Habehh et Gohel, 2021] Habehh, H. et Gohel, S. (2021). Machine learning in healthcare. *Current Genomics*, 22(4):291.

[Hanczar, 2019] Hanczar, B. (2019). Performance visualization spaces for classification with rejection option. *Pattern Recognition*, 96:106984.

[Hanczar et Dougherty, 2008] Hanczar, B. et Dougherty, E. R. (2008). Classification with reject option in gene expression data. *Bioinformatics*, 24(17):1889–1895.

[Hanczar *et al.*, 2020] Hanczar, B., Zehraoui, F., Issa, T. et Arles, M. (2020). Biological interpretation of deep neural network for phenotype prediction based on gene expression. *BMC bioinformatics*, 21:1–18.

[Hannah et Dunson, 2013] Hannah, L. A. et Dunson, D. B. (2013). Multivariate convex regression with adaptive partitioning. *The Journal of Machine Learning Research*, 14(1): 3261–3294.

[Hastie et Tibshirani, 1987a] Hastie, T. et Tibshirani, R. (1987a). Generalized additive models : some applications. *Journal of the American Statistical Association*, 82(398):371–386.

[Hastie et Tibshirani, 1987b] Hastie, T. et Tibshirani, R. (1987b). Non-parametric logistic and proportional odds regression. *Journal of the Royal Statistical Society : Series C (Applied Statistics)*, 36(3):260–276.

[Hastie *et al.*, 2009] Hastie, T., Tibshirani, R., Friedman, J. H. et Friedman, J. H. (2009). *The elements of statistical learning : data mining, inference, and prediction*, volume 2. Springer.

[Hastie, 2017] Hastie, T. J. (2017). Generalized additive models. *In Statistical models in S*, pages 249–307. Routledge.

[Hawkins, 1972] Hawkins, D. M. (1972). On the choice of segments in piecewise approximation. *IMA Journal of Applied Mathematics*, 9(2):250–256.

[Hoerl et Kennard, 1970a] Hoerl, A. E. et Kennard, R. W. (1970a). Ridge regression : applications to nonorthogonal problems. *Technometrics*, 12(1):69–82.

[Hoerl et Kennard, 1970b] Hoerl, A. E. et Kennard, R. W. (1970b). Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

[Hood et Flores, 2012] Hood, L. et Flores, M. (2012). A personal view on systems medicine and the emergence of proactive p4 medicine : predictive, preventive, personalized and participatory. *New biotechnology*, 29(6):613–624.

[Hornik *et al.*, 1989] Hornik, K., Stinchcombe, M. et White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

[J. Boursier et all, 2018] J. Boursier, R. Anty, L. V. et all (2018). Screening for therapeutic trials and treatment indication in clinical practice : Mack-3, a new blood test for the diagnosis of fibrotic nash. *Alimentary Pharmacology and Therapeutics, n°48*, pages 1387–1396.

[Jacot *et al.*, 2018] Jacot, A., Gabriel, F. et Hongler, C. (2018). Neural tangent kernel : Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

[Katzman *et al.*, 2018] Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T. et Kluger, Y. (2018). Deepsurv : personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):1–12.

[Kingma et Ba, 2017] Kingma, D. P. et Ba, J. (2017). Adam : A method for stochastic optimization.

[Kolmogorov et Bharucha-Reid, 2018] Kolmogorov, A. N. et Bharucha-Reid, A. T. (2018). *Foundations of the theory of probability : Second English Edition*. Courier Dover Publications.

[Küchenhoff, 1996] Küchenhoff, H. (1996). An exact algorithm for estimating breakpoints in segmented generalized linear models.

[Kulesza *et al.*, 2013] Kulesza, T., Stumpf, S., Burnett, M., Yang, S., Kwan, I. et Wong, W.-K. (2013). Too much, too little, or just right ? ways explanations impact end users' mental models. *In 2013 IEEE Symposium on visual languages and human centric computing*, pages 3–10. IEEE.

[LeCun *et al.*, 2015] LeCun, Y., Bengio, Y. et Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[Lee *et al.*, 2018] Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J. et Sohl-Dickstein, J. (2018). Deep neural networks as gaussian processes. *In International Conference on Learning Representations*.

[Lee *et al.*, 2019] Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J. et Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32.

[Leshno *et al.*, 1993] Leshno, M., Lin, V. Y., Pinkus, A. et Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867.

[Li et Liang, 2018] Li, Y. et Liang, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in neural information processing systems*, 31.

[Linardatos *et al.*, 2020] Linardatos, P., Papastefanopoulos, V. et Kotsiantis, S. (2020). Explainable ai : A review of machine learning interpretability methods. *Entropy*, 23(1):18.

[Little *et al.*, 2008] Little, M., McSharry, P., Hunter, E., Spielman, J. et Ramig, L. (2008). Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *Nature Precedings*, pages 1–1.

[Liu *et al.*, 2020a] Liu, C., Zhu, L. et Belkin, M. (2020a). On the linearity of large non-linear models : when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33:15954–15964.

[Liu *et al.*, 2020b] Liu, C., Zhu, L. et Belkin, M. (2020b). Toward a theory of optimization for over-parameterized systems of non-linear equations : the lessons of deep learning. *arXiv preprint arXiv :2003.00307*.

[Lou *et al.*, 2012] Lou, Y., Caruana, R. et Gehrke, J. (2012). Intelligible models for classification and regression. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158.

[Lou *et al.*, 2013] Lou, Y., Caruana, R., Gehrke, J. et Hooker, G. (2013). Accurate intelligible models with pairwise interactions. *In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631.

[Lundberg et Lee, 2017] Lundberg, S. M. et Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

[MacEachern et Forkert, 2021] MacEachern, S. J. et Forkert, N. D. (2021). Machine learning for precision medicine. *Genome*, 64(4):416–425.

[Magnani et Boyd, 2009] Magnani, A. et Boyd, S. P. (2009). Convex piecewise-linear fitting. *Optimization and Engineering*, 10:1–17.

[Mahmood *et al.*, 2014] Mahmood, S. S., Levy, D., Vasan, R. S. et Wang, T. J. (2014). The framingham heart study and the epidemiology of cardiovascular disease : a historical perspective. *The lancet*, 383(9921):999–1008.

[Matthews *et al.*, 2018] Matthews, A. G. d. G., Hron, J., Rowland, M., Turner, R. E. et Ghahramani, Z. (2018). Gaussian process behaviour in wide deep neural networks. *In International Conference on Learning Representations*.

[McNamee, 2005] McNamee, R. (2005). Regression modelling and other methods to control confounding. *Occupational and environmental medicine*, 62(7):500–506.

[Meijering *et al.*, 2022] Meijering, E., Calhoun, V. D., Menegaz, G., Miller, D. J. et Ye, J. C. (2022). Deep learning in biological image and signal processing [from the guest editors]. *IEEE Signal Processing Magazine*, 39(2):24–26.

[Miller, 2019] Miller, T. (2019). Explanation in artificial intelligence : Insights from the social sciences. *Artificial intelligence*, 267:1–38.

[Minka, 2003] Minka, T. P. (2003). A comparison of numerical optimizers for logistic regression. *Unpublished draft*, pages 1–18.

[Molnar, 2020] Molnar, C. (2020). *Interpretable machine learning*.

[Montavon *et al.*, 2017] Montavon, G., Lapuschkin, S., Binder, A., Samek, W. et Müller, K.-R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222.

[Montavon *et al.*, 2018] Montavon, G., Samek, W. et Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15.

[Montufar *et al.*, 2014] Montufar, G. F., Pascanu, R., Cho, K. et Bengio, Y. (2014). On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27.

[Muggeo, 2003] Muggeo, V. M. (2003). Estimating regression models with unknown break-points. *Statistics in medicine*, 22(19):3055–3071.

[Murphy, 2012] Murphy, K. P. (2012). *Machine learning : a probabilistic perspective*. MIT press.

[Nayyar *et al.*, 2021] Nayyar, A., Gadhavi, L. et Zaman, N. (2021). Chapter 2 - machine learning in healthcare : review, opportunities and challenges. *In* Singh, K. K., Elhoseny, M., Singh, A. et Elngar, A. A., éditeurs : *Machine Learning and the Internet of Medical Things in Healthcare*, pages 23–45. Academic Press.

[Neal, 2012] Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

[Neyshabur *et al.*, 2018] Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y. et Srebro, N. (2018). Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv :1805.12076*.

[Neyshabur *et al.*, 2014] Neyshabur, B., Tomioka, R. et Srebro, N. (2014). In search of the real inductive bias : On the role of implicit regularization in deep learning. *arXiv preprint arXiv :1412.6614*.

[Novak *et al.*, 2018a] Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J. et Sohl-Dickstein, J. (2018a). Sensitivity and generalization in neural networks : an empirical study. *arXiv preprint arXiv :1802.08760*.

[Novak *et al.*, 2018b] Novak, R., Xiao, L., Lee, J., Bahri, Y., Yang, G., Hron, J., Abolafia, D. A., Pennington, J. et Sohl-Dickstein, J. (2018b). Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv :1810.05148*.

[Ortiz-Jiménez *et al.*, 2021] Ortiz-Jiménez, G., Moosavi-Dezfooli, S.-M. et Frossard, P. (2021). What can linearized neural networks actually say about generalization ? *Advances in Neural Information Processing Systems*, 34:8998–9010.

[Pedregosa *et al.*, 2011] 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. et Duchesnay, E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Philip N Newsome et all., 2020] Philip N Newsome, Magali Sasso, J. J. D. et all. (2020). Fibroscan-ast (fast) score for the non-invasive identification of patients with non-alcoholis steatohepatitis with significant activity and fibrosis : a prospective derivation and global validation study. *Lancet Gastroenterol Hepatol, n°48*, pages 362–373.

[Pineau *et al.*, 2021] Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché Buc, F., Fox, E. et Larochelle, H. (2021). Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *The Journal of Machine Learning Research*, 22(1):7459–7478.

[R. Anty et all., 2010] R. Anty, A. lannelli, S. et all. (2010). A new composite model including metabolic syndrome, alanine aminotransferase and cytokeratin-18 for the diagnosis of non)alcoholic steatohepatitis in morbidly obese patients. *Alimentary Pharmacology and Therapeutics, n°32*, pages 1315–1322.

[Rajpurkar *et al.*, 2022] Rajpurkar, P., Chen, E., Banerjee, O. et Topol, E. J. (2022). Ai in health and medicine. *Nature medicine*, 28(1):31–38.

[Rajpurkar *et al.*, 2017] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K. *et al.* (2017). Chexnet : Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv :1711.05225*.

[Rakotomalala, 2013] Rakotomalala, R. (2013). Comparaison de populations : Tests paramétriques. *Bartlett test*, 7:27–29.

[Raschka et Mirjalili, 2019] Raschka, S. et Mirjalili, V. (2019). *Python machine learning : Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd.

[Riachi *et al.*, 2021] Riachi, E., Mamdani, M., Fralick, M. et Rudzicz, F. (2021). Challenges for reinforcement learning in healthcare. *arXiv preprint arXiv :2103.05612*.

[Ribeiro *et al.*, 2016] Ribeiro, M. T., Singh, S. et 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.

[Ribeiro *et al.*, 2018] Ribeiro, M. T., Singh, S. et Guestrin, C. (2018). Anchors : High-precision model-agnostic explanations. *In Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Samek *et al.*, 2021] Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J. et Müller, K.-R. (2021). Explaining deep neural networks and beyond : A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278.

[Saporta, 2006] Saporta, G. (2006). *Probabilités, analyse des données et statistique*. Editions technip.

[Schapire et Freund, 2012] Schapire, R. E. et Freund, Y. (2012). *Boosting : Foundations and Algorithms*. MIT Press.

[Schmidt *et al.*, 2017] Schmidt, M., Le Roux, N. et Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162:83–112.

[Schölkopf *et al.*, 2001] Schölkopf, B., Herbrich, R. et Smola, A. J. (2001). A generalized representer theorem. *In International conference on computational learning theory*, pages 416–426. Springer.

[Serra *et al.*, 2018] Serra, T., Tjandraatmadja, C. et Ramalingam, S. (2018). Bounding and counting linear regions of deep neural networks. *In International Conference on Machine Learning*, pages 4558–4566. PMLR.

[Shailaja *et al.*, 2018] Shailaja, K., Seetharamulu, B. et Jabbar, M. (2018). Machine learning in healthcare : A review. *In 2018 Second international conference on electronics, communication and aerospace technology (ICECA)*, pages 910–914. IEEE.

[Siddiqui *et al.*, 2020] Siddiqui, M. K., Morales-Menendez, R., Huang, X. et Hussain, N. (2020). A review of epileptic seizure detection using machine learning classifiers. *Brain informatics*, 7(1):1–18.

[Smith *et al.*, 1988] Smith, J. W., Everhart, J. E., Dickson, W., Knowler, W. C. et Johannes, R. S. (1988). Using the adap learning algorithm to forecast the onset of diabetes mellitus. *In Proceedings of the annual symposium on computer application in medical care*, page 261. American Medical Informatics Association.

[Stasinopoulos et Rigby, 1992] Stasinopoulos, D. et Rigby, R. (1992). Detecting break points in generalised linear models. *Computational Statistics & Data Analysis*, 13(4):461–471.

[Stone, 1985] Stone, C. J. (1985). Additive regression and other nonparametric models. *The annals of Statistics*, 13(2):689–705.

[Sun et Medaglia, 2019] Sun, T. Q. et Medaglia, R. (2019). Mapping the challenges of artificial intelligence in the public sector : Evidence from public healthcare. *Government Information Quarterly*, 36(2):368–383.

[Tishler et Zang, 1981a] Tishler, A. et Zang, I. (1981a). A maximum likelihood method for piecewise regression models with a continuous dependent variable. *Journal of the Royal Statistical Society : Series C (Applied Statistics)*, 30(2):116–124.

[Tishler et Zang, 1981b] Tishler, A. et Zang, I. (1981b). A new maximum likelihood algorithm for piecewise regression. *Journal of the American Statistical Association*, 76(376):980–987.

[Ulm, 1991] Ulm, K. (1991). A statistical method for assessing a threshold in epidemiological studies. *Statistics in medicine*, 10(3):341–349.

[Wanner *et al.*, 2021] Wanner, J., Herm, L.-V., Heinrich, K. et Janiesch, C. (2021). Stop ordering machine learning algorithms by their explainability ! an empirical investigation of the tradeoff between performance and explainability. *In Conference on e-Business, e-Services and e-Society*, pages 245–258. Springer.

[Wiens et Shenoy, 2018] Wiens, J. et Shenoy, E. S. (2018). Machine learning for healthcare : on the verge of a major shift in healthcare epidemiology. *Clinical Infectious Diseases*, 66(1):149–153.

[Williams, 1996] Williams, C. (1996). Computing with infinite networks. *Advances in neural information processing systems*, 9.

[Woldaregay *et al.*, 2019] Woldaregay, A. Z., Årsand, E., Botsis, T., Albers, D., Mamykina, L. et Hartvigsen, G. (2019). Data-driven blood glucose pattern classification and anomalies detection : machine-learning applications in type 1 diabetes. *Journal of medical Internet research*, 21(5):e11030.

[Xing *et al.*, 2020] Xing, L., Giger, M. L. et Min, J. K. (2020). *Artificial intelligence in medicine : technical basis and clinical applications*. Academic Press.

[Yarotsky, 2017] Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114.

[Zintgraf *et al.*, 2017] Zintgraf, L. M., Cohen, T. S., Adel, T. et Welling, M. (2017). Visualizing deep neural network decisions : Prediction difference analysis. *arXiv preprint arXiv :1702.04595*.

Annexes

Annexe A

Compléments de résultats numériques pour le Chapitre 3

Dans cette Annexe, des résultats expérimentaux supplémentaires pour le Chapitre 3 sont présentés. La Figure A.1 illustre les splines estimées par les différentes méthodes estimées dans la Section 3.4 sur la base de données Circle. Les Figures A.2 et A.3 présentent les différentes règles de décision estimées respectivement sur les bases de données Gaussienne et Circle.

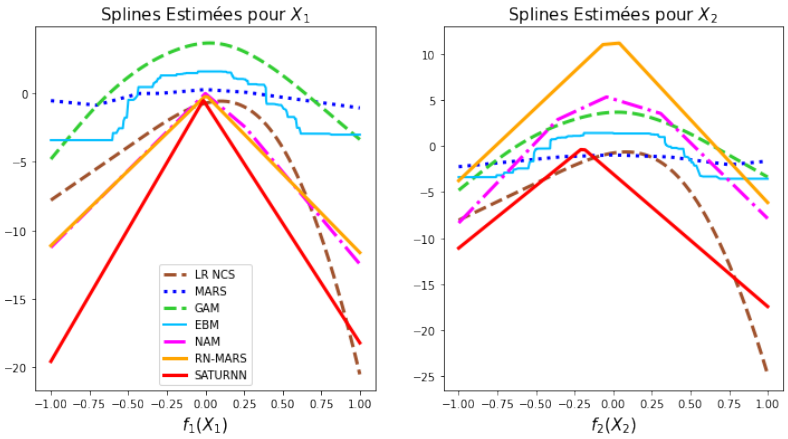

FIGURE A.1 – Splines estimées sur le jeu de données simulé Circle par les différentes méthodes testées estimant des splines univariées : LR NCS (pointillé marron), MARS (pointillé bleu foncé), GAM (pointillé vert), EBM (trait continu bleu clair), NAM (pointillé rose), RN-MARS (trait orange) et SATURNN (trait rouge). À gauche nous retrouvons la spline estimée pour X_1 et à droite celle pour X_2. Les splines univariées affichées sont issues des modèles ayant obtenu la meilleure AUC sur l'échantillon de validation lors de la Validation Croisée 5−folds.

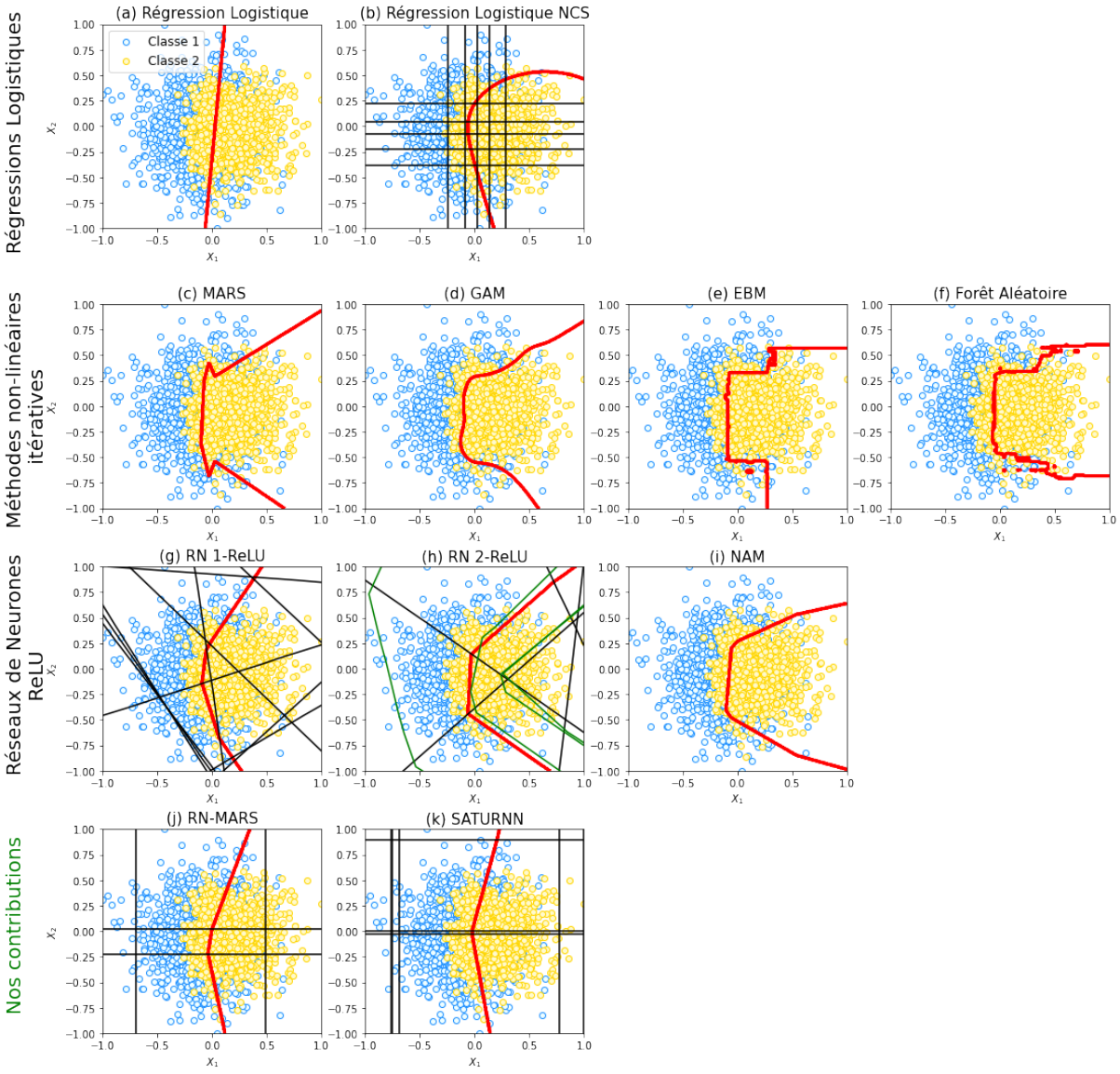

FIGURE A.2 – Règles de décisions estimées (en rouge) sur le jeu de données simulé gaussien pour les différentes méthodes testées : LR (a), LR NCS (b), MARS (c), GAM (d), EBM (e), RF (f), RN ReLU 1 et 2 couches (resp. g et h), NAM (i), le RN-MARS (j) et le SATURNN (k). Pour la LR NCS, le RN ReLU à 1 couche, le RN-MARS ainsi que le SATURNN le partitionnement est affiché en trait noir. Pour le RN ReLU à 2 couches il est affiché la segmentation opérée par la première couche en trait noir et la seconde en courbe verte. Les règles de décision affichées sont issues des modèles ayant obtenu la meilleure AUC sur l'échantillon de validation lors de la Validation Croisée 5−folds.

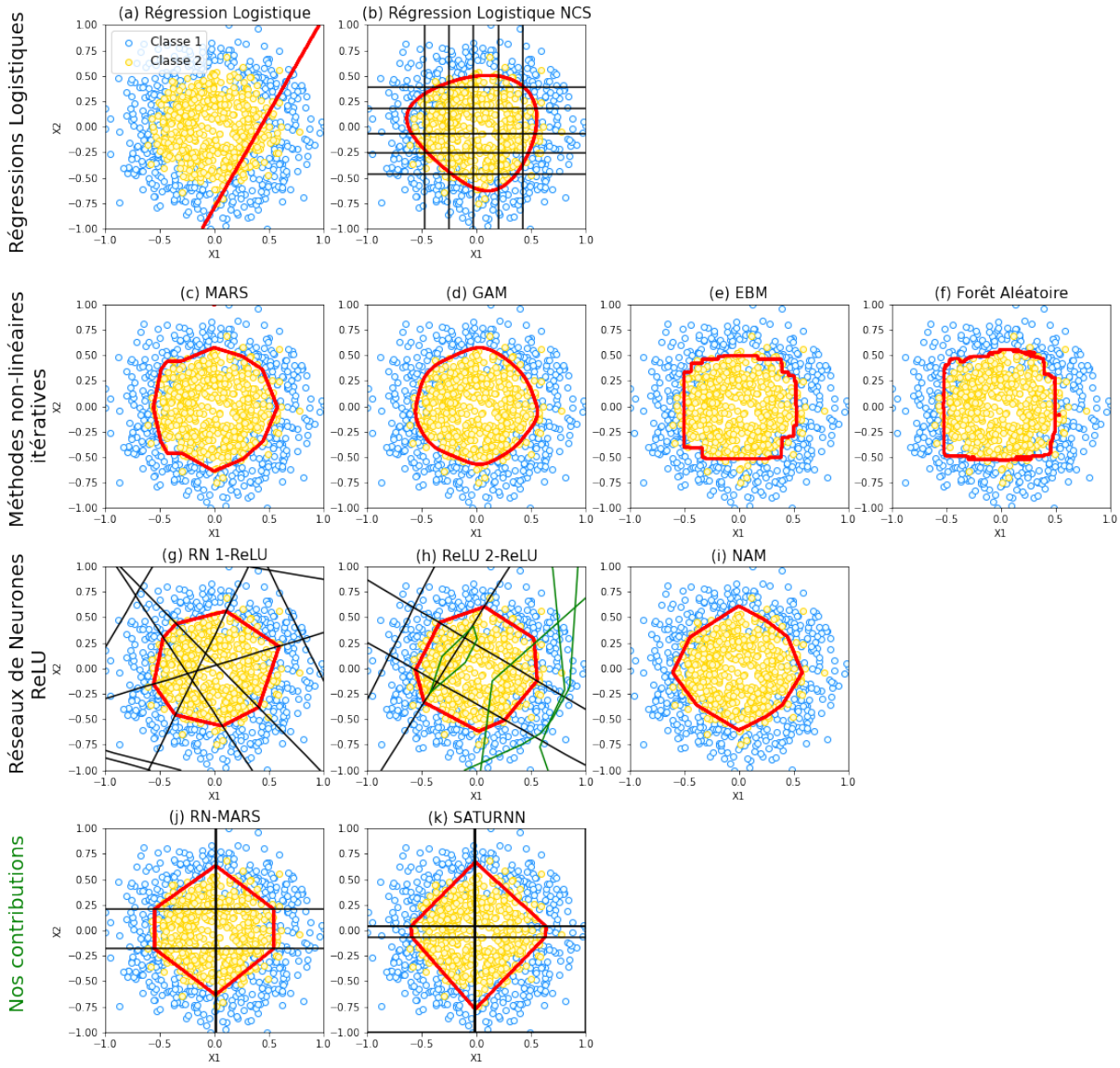

FIGURE A.3 – Règles de décisions estimées (en rouge) sur le jeu de données simulé Circle pour les différentes méthodes testées : LR (a), LR NCS (b), MARS (c), GAM (d), EBM (e), RF (f), RN ReLU 1 et 2 couches (resp. g et h), NAM (i), le RN-MARS (j) et le SATURNN (k). Pour la LR NCS, le RN ReLU à 1 couche, le RN-MARS ainsi que le SATURNN le partitionnement est affiché en trait noir. Pour le RN ReLU à 2 couches il est affiché la segmentation opérée par la première couche en trait noir et la seconde en courbe verte. Les règles de décision affichées sont issues des modèles ayant obtenu la meilleure AUC sur l'échantillon de validation lors de la Validation Croisée 5−folds.

Annexe B

Compléments de preuves pour le Chapitre 4

B.1 Approximation de Taylor

B.1.1 Théorème de Taylor d'ordre 2

Théorème 4 (Théorème de Taylor d'ordre 2)**.** *Soient $f : \mathbb{R}^d \to \mathbb{R}$ une fonction $\mathcal{C}^2$ définie sur une boule ouverte $\mathcal{B}_2^d(c, r)$ de centre $c \in \mathbb{R}^d$ et de rayon $r > 0$ et un point $h < r$. Il existe un réel $s \in \{0, 1\}$ tel que :*

$$f(c+h) = f(c) + \nabla f(c)h + \frac{1}{2}h^T H_f(c+sh)h. \tag{B.1}$$

B.1.2 Différentiabilité de la fonction de score

Supposons un échantillon x fixé. Lorsque nous souhaitons approximer la fonction de score du SATURNN définie à l'équation 3.7, nous supposons que $\psi(x, \theta)$ est différentiable presque partout. Les dérivées partielles de l'activation ReLU (2.17) ne sont pas continues au voisinage des points $s_k x_{v(k)} + b_k^{(0)} = 0$, pour $k \in \{1, \dots, p\}$. Ainsi en ces points, la fonction de score n'est pas différentiable et l'approximation de Taylor pourrait ne pas être correcte. Néanmoins, nous pouvons démontrer qu'il est possible avec une probabilité certaine de choisir des initialisations des paramètres permettant de considérer une fonction de score différentiable.

Les coefficients $\beta_0, \beta_1, \dots, \beta_p$ n'influencent pas la différentiabilité de la fonction considérée. Nous allons alors seulement nous intéresser à l'impact de l'initialisation des biais $[b_1^{(0)}, \dots b_p^{(0)}]$ sur la non-différentiabilité de la fonction de score. Nous savons que pour tout $k \in \{1, \dots, p\}$, les biais sont initialisés selon une loi Uniforme (Hypothèse 1) tel que $b_k \sim \mathcal{U}[-r, r]$. Par définition, calculer la probabilité que la variable aléatoire b_k appartienne à un booléen A revient à considérer la mesure de Lesbegue suivante :

$$\mathbb{P}(b_k \in A) := \int_{-r}^{r} \mathbb{1}_A(x)dx. \tag{B.2}$$

De plus, les points pour lesquels la fonction de score n'est pas différentiable sont dénombrables. En effet pour tout $k \in \{1, \dots, p\}$, nous avons $s_k x_{\upsilon(k)} + b_k^{(0)} = 0$ pour $b_k = -x_{\upsilon(k)}$ (si $s_k = 1$) et $b_k = x_{\upsilon(k)}$ (si $s_k = -1$). Nous nous intéressons à la probabilité de générer un b_k égal à une de ces valeurs :

$$\mathbb{P}(b_k^{(0)} \in \{-x_{\upsilon(k)}, x_{\upsilon(k)}\}) = \int_{\{-x_{\upsilon(k)}, x_{\upsilon(k)}\} \cap [-r,r]} dx \tag{B.3}$$

$$\leq \int_{\{-x_{\upsilon(k)}, x_{\upsilon(k)}\}} dx = 0. \tag{B.4}$$

Puisque $\{-x_{\upsilon(k)}, x_{\upsilon(k)}\} \cap [-r, r] \subset \{-x_{\upsilon(k)}, x_{\upsilon(k)}\}$ et que l'intégrale d'un point est nulle, nous avons :

$$0 \leq \mathbb{P}(b_k^{(0)} \in \{-x_{\upsilon(k)}, x_{\upsilon(k)}\}) \leq 0 \Rightarrow \mathbb{P}(b_k^{(0)} \in \{-x_{\upsilon(k)}, x_{\upsilon(k)}\}) = 0. \tag{B.5}$$

Ainsi, nous pouvons en déduire que l'on peut choisir une initialisation des biais $b_k^{(0)}$ pour $k \in \{1, \dots, p\}$ tels qu'ils soient tous différents de $\{-x_{\upsilon(k)}, x_{\upsilon(k)}\}$ avec une probabilité certaine :

$$\mathbb{P}(b_k^{(0)} \notin \{-x_{\upsilon(k)}, x_{\upsilon(k)}\}) = 1. \tag{B.6}$$

En conclusion, il est possible de trouver des initialisations telle que la fonction de score soit différentiable en tout point avec une probabilité certaine.

B.2 Étude du Gradient de la fonction de score

B.2.1 Définition du Gradient

Le gradient de la fonction de score $\psi(x, \theta)$ (3.7) par rapport aux paramètres θ au point $\theta^{(0)}$ est défini par :

$$\nabla_\theta \psi(x, \theta^{(0)}) = \left[\frac{\partial \psi(x, \theta^{(0)})}{\partial \beta_0}, \frac{\partial \psi(x, \theta^{(0)})}{\partial \beta_1}, \dots, \frac{\partial \psi(x, \theta^{(0)})}{\partial \beta_p}, \frac{\partial \psi(x, \theta^{(0)})}{\partial b_1}, \dots, \frac{\partial \psi(x, \theta^{(0)})}{\partial b_p} \right]^T. \tag{B.7}$$

De plus, nous avons pour tout $k \in \{1, \dots, p\}$:

$$\begin{aligned}
\frac{\partial \psi(x, \theta^{(0)})}{\partial \beta_0} &= \frac{1}{\sqrt{p}} \\
\frac{\partial \psi(x, \theta^{(0)})}{\partial \beta_k} &= \frac{1}{\sqrt{p}} \phi(s_k x_{\upsilon(k)} + b_k^{(0)}) \\
\frac{\partial \psi(x, \theta^{(0)})}{\partial b_k} &= \frac{1}{\sqrt{p}} \beta_k^{(0)} \frac{\partial \phi(s_k x_{\upsilon(k)} + b_k^{(0)})}{\partial b_k}.
\end{aligned}$$

Puisque $\phi(\cdot) = ReLU(\cdot)$ défini à l'équation (2.17), nous avons :

$$\frac{\partial \psi(x, \theta^{(0)})}{\partial b_k} = \frac{1}{\sqrt{p}} \beta_k^{(0)} \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}}.$$

Nous pouvons ainsi réécrire le gradient défini précédemment comme étant :

$$\begin{aligned}
\nabla_\theta \psi(x, \theta^{(0)}) = \frac{1}{\sqrt{p}} \Big[& 1, \phi(s_1 x_{\upsilon(1)} + b_1^{(0)}), \dots, \phi(s_p x_{\upsilon(p)} + b_p^{(0)}), \\
& \beta_1^{(0)} \mathbb{1}_{\{s_1 x_{\upsilon(1)} + b_1^{(0)} > 0\}}, \dots, \beta_p^{(0)} \mathbb{1}_{\{s_p x_{\upsilon(p)} + b_p^{(0)} > 0\}} \Big]^T.
\end{aligned} \tag{B.8}$$

B.2.2 Constance du Gradient

Dans cette sous-section, nous proposons une preuve pour le Lemme 1. Nous démontrons que le Gradient de la fonction de score du SATURNN par rapport à ses paramètres θ pris au point des initialisations $\theta^{(0)}$ noté $\nabla_\theta \psi(x, \theta^{(0)})$ est constant.

Démonstration.
Pour rappel $\nabla_\theta \psi \left(x, \theta^{(0)}\right)$ est défini par :

$$\begin{aligned}
\nabla_\theta \psi(x, \theta^{(0)}) &= \left[\frac{\partial \psi(x, \theta^{(0)})}{\partial \beta_0}, \frac{\partial \psi(x, \theta^{(0)})}{\partial \beta_1}, \dots, \frac{\partial \psi(x, \theta^{(0)})}{\partial \beta_p}, \frac{\partial \psi(x, \theta^{(0)})}{\partial b_1}, \dots, \frac{\partial \psi(x, \theta^{(0)})}{\partial b_p} \right]^T \\
&= \frac{1}{\sqrt{p}} \Big[1, \ \phi(s_1 x_{\upsilon(1)} + b_1^{(0)}), \phi(s_2 x_{\upsilon(2)} + b_2^{(0)}) \ , \dots, \ \phi(s_p x_{\upsilon(p)} + b_p^{(0)}), \\
&\qquad \beta_1^{(0)} \mathbb{1}_{\{s_1 x_{\upsilon(1)} + b_1^{(0)} > 0\}}, \ \dots, \ \beta_p^{(0)} \mathbb{1}_{\{s_p x_{\upsilon(p)} + b_p^{(0)} > 0\}} \Big]^T.
\end{aligned} \tag{B.9}$$

Afin d'étudier la constance du gradient $\nabla_\theta \psi\left(x, \theta^{(0)}\right)$ nous calculons sa norme :

$$\begin{aligned}\langle \nabla_\theta \psi(x, \theta^{(0)}), \nabla_\theta \psi(x, \theta^{(0)}) \rangle &= \nabla_\theta \psi(x, \theta^{(0)})^T \nabla_\theta \psi(x, \theta^{(0)}) \\ &= \frac{1}{p}\left[1 + \sum_{i=1}^{p} \phi(s_i x_{\upsilon(i)} + b_i^{(0)})^T \phi(s_i x_{\upsilon(i)} + b_i^{(0)}) \right. \\ &\left. + \sum_{i=1}^{p} {\beta_i^{(0)}}^2 \mathbb{1}_{\{s_i x_{\upsilon(i)} + b_i^{(0)} > 0\}} \mathbb{1}_{\{s_i x_{\upsilon(i)} + b_i^{(0)} > 0\}} \right]. \end{aligned} \tag{B.10}$$

D'après l'Hypothèse 1, nous avons pour tout $k \in \{1, \ldots, p\}$, $b_k^{(0)} \in [-r, +r]$ et donc $s_k x_{\upsilon(k)} \in [-r, +r]$. Ainsi, $s_k x_{\upsilon(k)} + b_k^{(0)} \in [-2r, 2r]$ et $\phi(s_k x_{\upsilon(k)} + b_k^{(0)}) \in [0, 2r]$. Ainsi nous avons :

$$\begin{aligned}\langle \nabla_\theta \psi(x, \theta^{(0)}), \nabla_\theta \psi(x, \theta^{(0)}) \rangle &= \frac{1}{p}\left[1 + \underbrace{\sum_{i=1}^{p} \phi(s_i x_{\upsilon(i)} + b_i^{(0)})^T \phi(s_i x_{\upsilon(i)} + b_i^{(0)})}_{\in [0, 4pr^2]} \right. \\ &\left. + \underbrace{\sum_{i=1}^{p} {\beta_i^{(0)}}^2 \mathbb{1}_{\{s_i x_{\upsilon(i)} + b_i^{(0)} > 0\}} \mathbb{1}_{\{s_i x_{\upsilon(i)} + b_i^{(0)} > 0\}}}_{\in [0, p \max(\beta)^2]} \right] \\ &\leq \frac{1}{p}[1 + 4pr^2 + p \max(\beta)^2] \\ &= O\left(\frac{1}{p}[O(1 + 4pr^2 + p \max(\beta)^2)]\right) \\ &= O\left(\frac{1}{p} O(p)\right) \\ &= O(1). \end{aligned} \tag{B.11}$$

□

B.3 Hessien de la fonction de score

B.3.1 Définition du Hessien

Soit $H_\theta\psi((x,(1-\tau)\theta^{(0)}+\tau\theta))$ la matrice Hessienne de la fonction de score (3.7) au point $\tilde{\theta} := (1-\tau)\theta^{(0)}+\tau\theta$:

$$H_\theta(\psi(x,\tilde{\theta})) = \left(\begin{array}{c|ccc|ccc} \frac{\partial^2\psi(x,\tilde{\theta})}{\partial^2\beta_0} & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_0\partial\beta_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_0\partial\beta_p} & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_0\partial b_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_0\partial b_p} \\ \hline \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_1\partial\beta_0} & \frac{\partial^2\psi(x,(\tilde{\theta})}{\partial^2\beta_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_1\partial\beta_p} & \frac{\partial^2\psi(x,(\tilde{\theta})}{\partial\beta_1\partial b_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_1\partial b_p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_p\partial\beta_0} & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_p\partial\beta_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial^2\beta_p} & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_p\partial b_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_p\partial b_p} \\ \hline \frac{\partial^2\psi(x,\theta)}{\partial b_1\partial\beta_0} & \frac{\partial^2\psi(x,\theta)}{\partial b_1\partial\beta_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial b_1\partial\beta_p} & \frac{\partial^2\psi(x,\theta)}{\partial^2 b_1} & \cdots & \frac{\partial^2\psi(x,\theta a)}{\partial b_1\partial b_p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2\psi(x,\tilde{\theta})}{\partial b_p\partial\beta_0} & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial b_p\partial\beta_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial b_p\partial\beta_p} & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial b_p\partial b_1} & \cdots & \frac{\partial^2\psi(x,\tilde{\theta})}{\partial^2 b_p} \end{array}\right). \quad \text{(B.12)}$$

Nous pouvons calculer les différentes dérivées secondes composant la matrice hessienne. Nous avons pour $i,j\in\{1,\dots,p\}^2$, $i\neq j$ et $\phi(\cdot)=\mathrm{ReLU}(\cdot)$ (2.17) :

$$\begin{aligned}
&\frac{\partial^2\psi(x,\tilde{\theta})}{\partial^2\beta_0} = \frac{\partial^2\psi(x,\tilde{\theta})}{\partial^2\beta_j} = \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_i\partial\beta_j} = 0,\\
&\frac{\partial^2\psi(x,\tilde{\theta})}{\partial b_i\partial b_j} = \frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_i\partial b_j} = 0,\\
&\frac{\partial^2\psi(x,\tilde{\theta})}{\partial^2 b_j} = \frac{1}{\sqrt{p}}\beta_j\frac{\partial^2\phi\left(s_j x_{v(j)}+(1-\tau)b_j^{(0)}+\tau b_j\right)}{\partial^2 b_j} = 0,\\
&\frac{\partial^2\psi(x,\tilde{\theta})}{\partial\beta_j\partial b_j} = \frac{\tau}{\sqrt{p}}\frac{\partial\phi\left(s_j x_{v(j)}+(1-\tau)b_j^{(0)}+\tau b_j\right)}{\partial b_j} = \frac{\tau^2}{\sqrt{p}}\mathbb{1}_{\left\{s_j x_{v(j)}+(1-\tau)b_j^{(0)}+\tau b_j>0\right\}}.
\end{aligned}$$

Finalement, la matrice hessienne de la fonction de score au point $\tilde{\theta} := (1-\tau)\theta^{(0)}+\tau\theta)$ se réécrit :

$$\left(\begin{array}{c|ccc|ccc} 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \hline \vdots & & & & \ddots & & 0 \\ \vdots & & 0 & & & \frac{\tau^2}{\sqrt{p}}\mathbb{1}_{\left\{s_k x_{v(k)}+(1-\tau)b_k^{(0)}+\tau b_k>0\right\}} & \\ \vdots & & & & 0 & & \ddots \\ \hline \vdots & \ddots & & 0 & & & \\ \vdots & & \frac{\tau^2}{\sqrt{p}}\mathbb{1}_{\left\{s_k x_{v(k)}+(1-\tau)b_k^{(0)}+\tau b_k>0\right\}} & & & 0 & \\ 0 & 0 & & \ddots & & & \end{array}\right). \quad \text{(B.13)}$$

B.3.2 Comportement asymptotique du Hessien

Dans cette sous-section, nous proposons une preuve pour le Lemme 2. Nous démontrons que le Hessien de la fonction de score du SATURNN par rapport à ses paramètres θ pris au point $(1-\tau)\theta^{(0)}+\tau\theta)$ avec $\tau \in \{0,1\}$ s'approche de 0 au fur et à mesure que la profondeur du SATURNN augmente ($p \to \infty$).

Démonstration.
Pour rappel, $H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))$ est défini par :

$$\left(\begin{array}{c|ccc|ccc} 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \hline \vdots & & & & \ddots & & 0 \\ \vdots & & 0 & & & \frac{\tau^2}{\sqrt{p}}\mathbb{1}_{\left\{s_k x_{\upsilon(k)}+(1-\tau)b_k^{(0)}+\tau b_k>0\right\}} & \\ \vdots & & & & 0 & & \ddots \\ \hline \vdots & \ddots & & 0 & & 0 & \\ \vdots & & \frac{\tau^2}{\sqrt{p}}\mathbb{1}_{\left\{s_k x_{\upsilon(k)}+(1-\tau)b_k^{(0)}+\tau b_k>0\right\}} & & & 0 & \\ 0 & 0 & & \ddots & & & \end{array}\right). \tag{B.14}$$

Afin d'étudier le comportement asymptotique de $H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))$, nous calculons sa norme spectrale. La norme spectrale d'une matrice A se définit comme étant sa plus grande valeur singulière, soit la racine carrée de la valeur propre maximale $\lambda_{\max}(A^TA)$:

$$\|H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))\|_2 = \sqrt{\lambda_{\max}\left[H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))^T H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))\right]}.$$

Puisque $H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))$ est symétrique, nous avons :

$$\begin{aligned}\|H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))\|_2 &= \sqrt{\lambda_{\max}\left[H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))\right]^2} \\ &= \lambda_{\max}\left[H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))\right]. \end{aligned} \tag{B.15}$$

Pour déduire les valeurs propres de la matrice hessienne, nous remarquons dans un premier temps que sa structure est parcimonieuse puisque chaque ligne ne possède que deux entrées non-nulles. Soit $u = [0,\dots,0,1,0,\dots,0,\alpha,0,\dots,0]^T \in \mathbb{R}^{2p+1}$ un vecteur à 2 entrées non-nulles $(1,\alpha)$ aux coordonnées k et $k+p$. Si nous supposons que u est le vecteur propre associé à la valeur propre λ, nous devons alors avoir :

$$H_\theta(\psi(x,(1-\tau)\theta^{(0)}+\tau\theta))u = \lambda u.$$

Nous obtenons le système d'équations à deux paramètres inconnus (λ,α) suivant :

$$\begin{cases} \lambda = 0, \\ \lambda = \alpha\frac{\tau^2}{\sqrt{p}}\mathbb{1}_{\left\{s_k x_{\upsilon(k)}+(1-\tau)b_k^{(0)}+\tau b_k>0\right\}}, \\ \alpha\lambda = \frac{\tau^2}{\sqrt{p}}\mathbb{1}_{\left\{s_k x_{\upsilon(k)}+(1-\tau)b_k^{(0)}+\tau b_k>0\right\}}. \end{cases}$$

En négligeant les $2p+1$ solutions nulles ($\lambda = 0$), nous résolvons le système suivant :

$$\begin{aligned}
&\lambda = \left(\frac{\tau^2}{\sqrt{p}}\right)^2 \frac{1}{\lambda} \mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}} \\
&\Leftrightarrow \lambda^2 = \frac{1}{p} \tau^4 \mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}} \\
&\Leftrightarrow \lambda = \pm \frac{\tau^2}{\sqrt{p}} \mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}} \\
&\Leftrightarrow \lambda = \frac{\tau^2}{\sqrt{p}} \mathbb{1}_{\{s_k x_{\upsilon(k)} + (1-\tau) b^{(0)} + \tau b_k > 0\}}, \text{ car } p \in \mathbb{R}^+.
\end{aligned} \tag{B.16}$$

En glissant les entrées non-nulles de u aux positions k et $k+p$, pour $k \in \{1, \ldots, p\}$, nous retrouvons $2p$ valeurs propres. Puisque la hessienne (4.6) est une matrice avec $2p$ entrées non-nulles, nous retrouvons de ce fait toutes les valeurs propres. Les paires de valeurs propres possibles sont pour $k \in \{1, \ldots, p\}$:

$$\lambda_k, \lambda_{k+p} = \frac{\tau^2}{\sqrt{p}} \mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}}. \tag{B.17}$$

Puisque la norme spectrale de la matrice hessienne est sa plus grande valeur propre (B.15), nous pouvons alors la déduire :

$$\begin{aligned}
\|H_\theta(\psi(x, (1-\tau)\theta^{(0)} + \tau\theta))\|_2 &= \max_{k=\{1,\ldots,p\}} \frac{\tau^2}{\sqrt{p}} \mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}} \\
&= \frac{1}{\sqrt{p}} \max_{k=\{1,\ldots,p\}} \tau^2 \mathbb{1}_{\left\{s_k x_{\upsilon(k)} + (1-\tau) b_k^{(0)} + \tau b_k > 0\right\}} \\
&= \frac{1}{\sqrt{p}} \\
&= O\left(\frac{1}{\sqrt{p}}\right).
\end{aligned} \tag{B.18}$$

□

B.4 Étude de $\psi(x, \theta^{(0)})$

Dans cette partie, nous nous intéressons au calcul de l'espérance et de la variance de $c_0(x) = \psi(x, \theta^{(0)})$ soit la fonction de score du SATURNN en son initialisation $\theta^{(0)} \in \mathbb{R}^{2p+1}$. Nous supposons que les variables d'entrée prennent des valeurs dans une boule ouverte de rayon $r > 0$, $x \in \mathcal{B}_2^p(0, r)$. De plus, d'après l'Hypothèse 1, les paramètres initialisés sont indépendants et suivent les distributions suivantes : $\beta = [\beta_0, \dots, \beta_p] \sim \mathcal{N}(0, 1)$ et $b = [b_1, \dots, b_p] \sim \mathcal{U}[-r, r]$.

B.4.1 Espérance

Dans un premier temps, nous pouvons montrer que l'espérance de $c_0(x)$ est nulle :

$$\begin{aligned}
\mathbb{E}(c_0(x)) &= \mathbb{E}\left(\frac{1}{\sqrt{p}}\left(\beta_0^{(0)} + \sum_{k=1}^{p} \beta_k^{(0)} \phi(s_k x_{\upsilon(k)} + b_k^{(0)})\right)\right) \\
&= \frac{1}{\sqrt{p}}\left(\underbrace{\mathbb{E}\left(\beta^{(0)}\right)}_{=0} + \sum_{k=1}^{p} \underbrace{\mathbb{E}\left(\beta_k^{(0)}\right)}_{=0} \mathbb{E}\left(\phi(s_k x_{\upsilon(k)} + b_k^{(0)})\right)\right) \\
&= 0. \qquad \text{(B.19)}
\end{aligned}$$

B.4.2 Variance

Dans un second temps, nous démontrons que la variance de $c_0(x)$ est bornée par $\frac{1}{p} + 4r^2$. Pour rappel, puisque $\beta \sim \mathcal{N}(0, 1)$, nous avons $\mathbb{E}(\beta^0) = 0$ et $\mathbb{E}(\beta^{(0)^2}) = 1$. Ainsi, nous obtenons :

$$\begin{aligned}
\mathbb{V}(c_0(x)) &= \mathbb{V}\left(\frac{1}{\sqrt{p}}\left(\beta_0^{(0)} + \sum_{k=1}^{p} \beta_k^{(0)} \phi(s_k x_{\upsilon(k)} + b_k^{(0)})\right)\right) \\
&= \frac{1}{p}\mathbb{E}\left(\left(\beta_0^{(0)} + \sum_{k=1}^{p} \beta_k^{(0)} \phi(s_k x_{\upsilon(k)} + b_k^{(0)})\right)^2\right) - \frac{1}{p}\underbrace{\mathbb{E}\left(\beta_0^{(0)} + \sum_{k=1}^{p} \beta_k^{(0)} \phi(s_k x_{\upsilon(k)} + b_k^{(0)})\right)^2}_{=0 \text{ (Annexe B.4.1)}} \\
&= \frac{1}{p}\mathbb{E}\left(\beta_0^{(0)^2} + 2\beta_0^{(0)} \sum_{k=1}^{p} \beta_k^{(0)} \phi(s_k x_{\upsilon(k)} + b_k^{(0)}) + \sum_{k=1}^{p} \beta_k^{(0)^2} \phi(s_k x_{\upsilon(k)} + b_k^{(0)})^2\right) \\
&= \frac{1}{p}\underbrace{\mathbb{E}\left(\beta_0^{(0)^2}\right)}_{=1} + \frac{2}{p}\underbrace{\mathbb{E}\left(\beta_0^{(0)}\right)}_{=0} \sum_{k=1}^{p} \mathbb{E}\left(\beta_k^{(0)} \phi(s_k x_{\upsilon(k)} + b_k^{(0)})\right) + \sum_{k=1}^{p} \underbrace{\mathbb{E}\left(\beta_k^{(0)^2}\right)}_{=1} \mathbb{E}\left(\psi(s_k x_{\upsilon(k)} + b_k^{(0)})^2\right) \\
&= \frac{1}{p} + \frac{1}{p}\sum_{k=1}^{p} \mathbb{E}\left(\psi(s_k x_{\upsilon(k)} + b_k^{(0)})^2\right) \\
&\leq \frac{1}{p} + \frac{1}{p}\sum_{k=1}^{p} \mathbb{E}\left((s_k x_{\upsilon(k)} + b_k^{(0)})^2\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{p} + \frac{1}{p}\sum_{k=1}^{p} \mathbb{E}\left(\underbrace{(s_k x_{\upsilon(k)})}_{\leq r}^2 + 2\, \underbrace{b_k^{(0)}}_{\leq r^2} \underbrace{s_k x_{\upsilon(k)}}_{\leq r} + \underbrace{b_k^{(0)^2}}_{\leq r^2} \right) \\
&\leq \frac{1}{p} + \frac{1}{p}\sum_{k=1}^{p} \mathbb{E}(r^2 + 2r^2 + r^2) \\
&= \frac{1}{p} + \frac{1}{p}\sum_{k=1}^{p} 4r^2 \\
&= \frac{1}{p} + 4r^2. \qquad \text{(B.20)}
\end{aligned}
$$

B.5 Équivalence entre le SATURNN et la LR PSI LIN

Le Théorème 3 établit qu'il devient équivalent de minimiser la fonction de coût du SATURNN notée $\mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D})$ et définie par (3.12) ou celle de la LR PSI LIN $\mathcal{L}^{\text{LR PSI LIN}}(\eta)$ définie par (4.26) lorsque le nombre p de neurones composant le SATURNN est suffisamment grand. Plus précisément, nous pouvons démontrer que :

$$\sup_{\substack{\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R) \\ \eta \in \mathcal{B}_2^{2p+1}(0, R)}} \left| \mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D}) - \mathcal{L}^{\text{LR}}(\eta, \mathcal{D}) \right| \leq \frac{R^2}{2\sqrt{p}}. \tag{B.21}$$

Nous détaillons dans cette section la preuve complète aboutissant à cette borne supérieure d'erreur.

Démonstration. Supposons que nous disposons de N échantillons tels que pour chaque échantillon, les variables descriptives $x^{(i)}$ prennent des valeurs dans une boule ouverte de rayon $r > 0 : x \in \mathcal{B}_2^d(0, r)$. Soit $\Phi^{\text{SATURNN}}(x, \theta) = \sigma(\psi(x, \theta))$ le SATURNN, tel que σ réfère à la sigmoïde (2.4). Nous supposons que les initialisations du SATURNN pour son processus d'optimisation respectent l'Hypothèse 1, à savoir $\theta^{(0)} = [\beta_0^{(0)}, \beta_1^{(0)}, \ldots, \beta_p^{(0)}, b_1^{(0)}, \ldots, b_p^{(0)}]$ avec $\beta_k^{(0)} \sim \mathcal{N}(0, 1)$ et $b_k^{(0)} \sim \mathcal{U}[-r, +r]$. De plus, nous supposons que l'apprentissage du SATURNN est contraint de sorte à ce que les paramètres estimés $\hat{\theta}$ ne s'éloignent pas trop de ceux initialisés $\theta^{(0)}$, à une distance $R > 0$ maximale (Hypothèse 2). Ainsi, entraîner le SATURNN revient à minimiser la fonction de coût de $\mathcal{L}^{\text{SATURNN}}(x, \theta)$ sachant que $\hat{\theta} \in \mathcal{B}_2^{2p+1} := \{\|\theta - \theta^{(0)}\|_2 \leq R\}$:

$$\mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} L(\sigma(\psi(x^{(i)}, \theta), y^{(i)})),$$

avec L la Cross-Entropie binaire (2.9). Enfin, nous considérons la Régression Logistique appliquée à la fonction de score linéarisée du SATURNN $\delta^{\text{LR PSI LIN}}(x, \eta) = \sigma(\psi^{\text{lin}}(x, \theta, \theta^{(0)})\eta)$, tel que $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ est le modèle linéaire approximant la fonction de score du SATURNN. Afin d'apprendre les estimateurs $\hat{\eta}$, il convient de minimiser la fonction de coût suivante :

$$\mathcal{L}^{\text{LR}}(\eta, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} L(\sigma(\psi^{\text{lin}}(x^{(i)}, \theta, \theta^{(0)}), y^{(i)})),$$

Puisque nous nous intéressons à l'équivalence entre le SATURNN et la LR PSI LIN, nous cherchons donc à étudier :

$$\left| \mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D}) - \mathcal{L}^{\text{LR}}(\eta, \mathcal{D}) \right| = \left| \frac{1}{N} \sum_{i=1}^{N} L\left(\sigma\left(\psi(x^{(i)}, \theta)\right), y^{(i)}\right) - L\left(\sigma\left(\psi^{\text{lin}}(x^{(i)}, \theta, \theta^{(0)})\right), y^{(i)}\right) \right|. \tag{B.22}$$

Nous supposons d'autre part que la fonction de score linéarisée $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ est $\mathcal{C}^2$ presque partout et se définie par :

$$\psi^{\text{lin}}(x, \theta, \theta^{(0)}) = \psi(x, \theta^{(0)}) + g_0(x)^T(\theta - \theta^{(0)}),$$

avec $g_0(x) = \nabla_\theta \psi(x, \theta^{(0)})$ le gradient de $\psi(x, \theta)$ par rapport à ses paramètres θ calculé au point $\theta^{(0)}$ définit par l'équation 4.3. Nous avons :

$$\psi(x, \theta) = \psi^{\text{lin}}(x, \theta, \theta^{(0)}) + \epsilon(x, \theta, \theta^{(0)}), \tag{B.23}$$

avec $\epsilon(x, \theta, \theta^{(0)})$ l'erreur d'approximation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$. Nous savons d'après le Théorème 1 que l'erreur d'approximation est bornée et tend à être nulle lorsque le nombre p de neurones est suffisamment grand : $|\epsilon(x, \theta, \theta^{(0)})| \leq \frac{R^2}{2\sqrt{p}}$. Ainsi le problème que nous considérons peut se réécrire :

$$\begin{aligned}\left|\mathcal{L}^{\text{SATURNN}}(\theta, \mathcal{D}) - \mathcal{L}^{\text{LR}}(\eta, \mathcal{D})\right| = \Bigg| \frac{1}{N} \sum_{i=1}^{N} L\left(\sigma\left(\psi^{\text{lin}}(x^{(i)}, \theta, \theta^{(0)}) + \epsilon(x^{(i)}, \theta, \theta^{(0)})\right), y^{(i)}\right) \\ - L\left(\sigma\left(\psi^{\text{lin}}(x^{(i)}, \theta, \theta^{(0)})\right), y^{(i)}\right)\Bigg|. \end{aligned} \tag{B.24}$$

Pour simplifier les notations, nous considérerons $z^{(i)} = \psi^{\text{lin}}(x^{(i)}, \theta, \theta^{(0)})$ et $\epsilon^{(i)} = \epsilon(x^{(i)}, \theta, \theta^{(0)})$. Dans un premier temps, nous considérons le problème avec un échantillon i fixé.

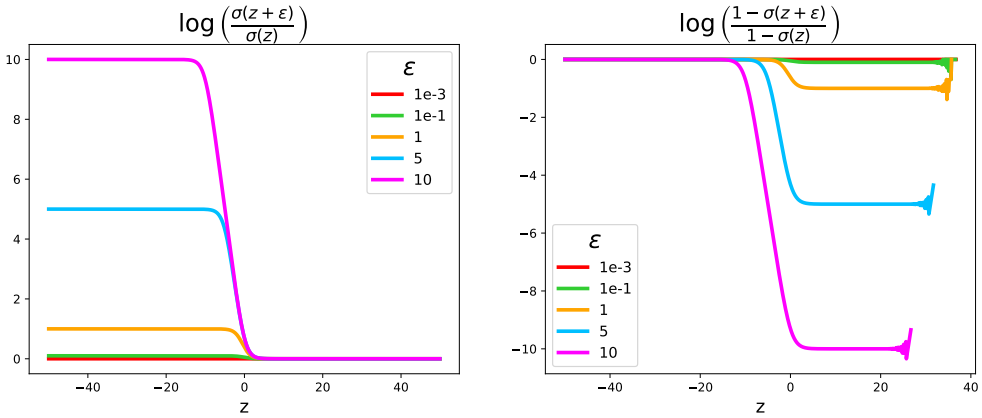

FIGURE B.1 – Valeurs de $|L(\sigma(z+\epsilon), y) - L(\sigma(z), y)|$ (B.25) pour différentes valeurs de ϵ selon l'étiquette : $y = 0$ (Gauche) and $y = 1$ (Droite).

Deux cas sont à considérer selon la valeur du label :

$$|L(\sigma(z+\epsilon), y) - L(\sigma(z), y)| = \begin{cases} \left|\log\left(\frac{1-\sigma(z+\epsilon)}{1-\sigma(z)}\right)\right| & \text{si } y = 0, \\ \left|\log\left(\frac{\sigma(z+\epsilon)}{\sigma(z)}\right)\right| & \text{si } y = 1. \end{cases} \tag{B.25}$$

Considérons dans un premier temps le cas $y = 0$ et $f_\epsilon(z) = \log\left(\frac{1-\sigma(z+\epsilon)}{1-\sigma(z)}\right)$ (Figure B.1-gauche). La dérivée de $f_\epsilon(z)$ est égale à :

$$f_\epsilon'(z) = \frac{1}{1+\exp(-z)} - \frac{1}{1+\exp(-z-\epsilon)}. \tag{B.26}$$

Nous supposons $\epsilon > 0$ (le cas $\epsilon < 0$ sera considéré plus tard). Nous avons alors $f_\epsilon'(z) < 0$. Maintenant que nous avons établi que $f_\epsilon(z)$ est strictement décroissante (comme confirmé

par la Figure B.1-gauche), nous étudions ses bornes.
Quand z tend vers $-\infty$, nous obtenons :

$$\begin{aligned}\lim_{z\to-\infty}\left|\log\left(\frac{1-\sigma\left(z+\epsilon\right)}{1-\sigma\left(z\right)}\right)\right| &= \lim_{z\to-\infty}\left|\log\left(\frac{1-\frac{1}{1+\exp(-z-\epsilon)}}{1-\frac{1}{1+\exp(-z)}}\right)\right| \\ &= \lim_{z\to-\infty}\left|\log\left(1-\frac{1}{1+\exp(-z-\epsilon)}\right)-\log\left(1-\frac{1}{1+\exp(-z)}\right)\right| \\ &= 0. \end{aligned} \tag{B.27}$$

Quand z tend vers $+\infty$, nous avons :

$$\begin{aligned}\lim_{z\to+\infty}\left|\log\left(\frac{1-\sigma\left(z+\epsilon\right)}{1-\sigma\left(z\right)}\right)\right| &= \lim_{z\to+\infty}\left|\log\left(\frac{1-\frac{1}{1+\exp(-z-\epsilon)}}{1-\frac{1}{1+\exp(-z)}}\right)\right| \\ &= \lim_{z\to+\infty}\left|\log\left(\frac{\frac{\exp(-z-\epsilon)}{1+\exp(-z-\epsilon)}}{\frac{\exp(-z)}{1+\exp(-z)}}\right)\right| \\ &= \lim_{z\to+\infty}\left|\log\left(\frac{\exp(-z-\epsilon)}{1+\exp(-z-\epsilon)}\times\frac{1+\exp(-z)}{\exp(-z)}\right)\right| \\ &= \lim_{z\to+\infty}\left|\log\left(\exp(-\epsilon)\times\frac{1+\exp(-z)}{1+\exp(-z-\epsilon)}\right)\right| \\ &= \lim_{z\to+\infty}\left|-\epsilon+\log\left(1+\exp(-z)\right)-\log\left(1+\exp(-z-\epsilon)\right)\right| \\ &= \epsilon. \end{aligned} \tag{B.28}$$

Puisque $f_\epsilon(z)$ est strictement décroissante et admet des bornes inférieure (B.27) et supérieure (B.28), nous pouvons conclure que :

$$0 < \left|L(\sigma(z+\epsilon),y)-L(\sigma(z),y)\right| < \epsilon \leq \frac{R^2}{2\sqrt{p}}\ \forall z\in\mathbb{R},\ \text{quand } y=0. \tag{B.29}$$

Quand $\epsilon < 0$, nous obtenons les mêmes bornes. La preuve est simple bien qu'il faille modifier légèrement les calculs.

De plus, avec exactement le même raisonnement, nous obtenons pour le cas $y = 1$:

$$0 < \left|L(\sigma(z+\epsilon),y)-L(\sigma(z),y)\right| < \epsilon \leq \frac{R^2}{2\sqrt{p}}\ \forall z\in\mathbb{R},\ \text{quand } y=1. \tag{B.30}$$

Maintenant que nous avons établi des bornes pour $|L(\sigma(z+\epsilon),y)-L(\sigma(z),y)|$ pour un échantillon i fixé, tel que $i \in \{1,\dots,N\}$ dans les deux cas $y = 0$ and $y = 1$, nous allons étudier la borne du problème global défini à l'équation (B.24) :

$$\sup_{\substack{\theta\in\mathcal{B}_2^{2p+1}(\theta^{(0)},R)\\ \eta\in\mathcal{B}_2^{2p+1}(0,R)}}\left|\mathcal{L}^{\text{SATURNN}}(\theta,\mathcal{D})-\mathcal{L}^{\text{LR}}(\eta,\mathcal{D})\right| = \sup_{\theta\in\mathcal{B}_2^{2p+1}(\theta^{(0)},R)}\left|\frac{1}{N}\sum_{i=1}^{N}L\left(\sigma\left(z^{(i)}+\epsilon^{(i)}\right),y^{(i)}\right)\right.$$

$$
\begin{aligned}
&-L\left(\sigma\left(z^{(i)}\right), y^{(i)}\right)\Big| \\
&= \frac{1}{N}\sum_{i=1}^{N} \sup_{\theta \in \mathcal{B}_2^{2p+1}(\theta^{(0)}, R)} \Big| L\left(\sigma\left(z^{(i)} + \epsilon^{(i)}\right), y^{(i)}\right) \\
&\qquad\qquad -L\left(\sigma\left(z^{(i)}\right), y^{(i)}\right)\Big| \\
&\leq \frac{1}{N}\sum_{i=1}^{N} \epsilon \\
&\leq \epsilon \\
&\leq \frac{R^2}{2\sqrt{p}} \qquad \text{(B.31)} \\
&= O\left(\frac{1}{\sqrt{p}}\right). \qquad \text{(B.32)}
\end{aligned}
$$

□

Annexe C

Compléments de résultats numériques pour le Chapitre 4

Dans cette annexe, nous présentons des résultats expérimentaux complémentaires pour le Chapitre 4 du manuscrit.

C.1 Linéarisation de la fonction de score du SATURNN

Dans un premier temps en Section C.1 nous vérifions les résultats théoriques établis concernant la linéarisation de la fonction de score, en vérifiant tout d'abord la constance du gradient (sous-section C.1.1) et ensuite le comportement asymptotique du Hessien (sous-section C.1.2).

C.1.1 Constance du Gradient

Afin que l'approximation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ soit correcte, nous avons établi que le gradient de la fonction de score par rapport à ses paramètres pris au point $\theta^{(0)}$ que l'on note $\nabla_\theta \psi(x, \theta^{(0)})$ et défini à l'équation (4.3) devient constant à mesure que le nombre p de neurones composant le SATURNN augmente (Lemme 1).

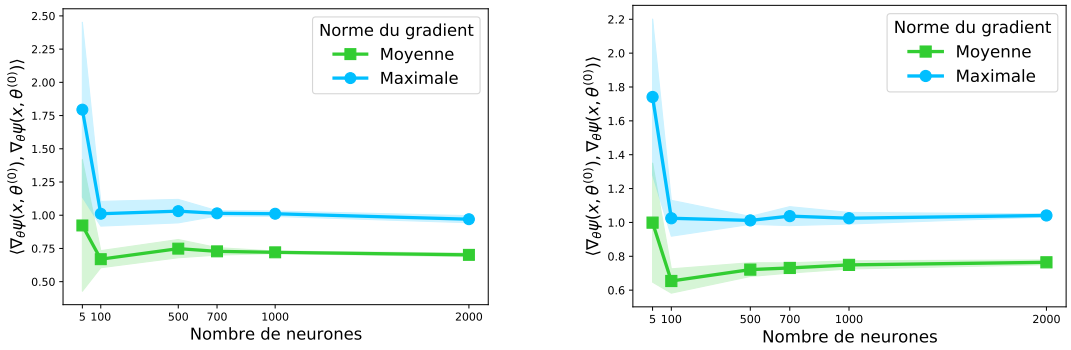

FIGURE C.1 – Moyennes et écart-types des normes $|\psi(x, \theta) - \psi^{\text{lin}}(x, \theta, \theta^{(0)})|$ moyennes (courbe verte) et maximales (courbe bleue) obtenues sur une Validation Croisée 5-folds sur la base de données Gaussienne (gauche) et Circle (droite) pour différentes valeurs de p.

Sur la Figure C.1, nous pouvons constater qu'à mesure que la profondeur du SATURNN augmente ($p \to \infty$), la courbe moyenne des normes moyennes (courbe verte) devient constante. Les normes maximales obtenues (courbe bleue) ne dépassent pas la valeur de 1 dès une profondeur du SATURNN de $p = 100$. De plus, nous pouvons constater sur cette même figure que les écart-types des normes moyennes et maximales obtenues sur 5-folds se resserrent à mesure que p augmente. Enfin, sur les Histogrammes C.2 et C.3, ces résultats sont vérifiés puisqu'à la fois les normes moyennes et maximales tendent à être bornées par 1 sur les deux jeux de données à mesure que p augmente.

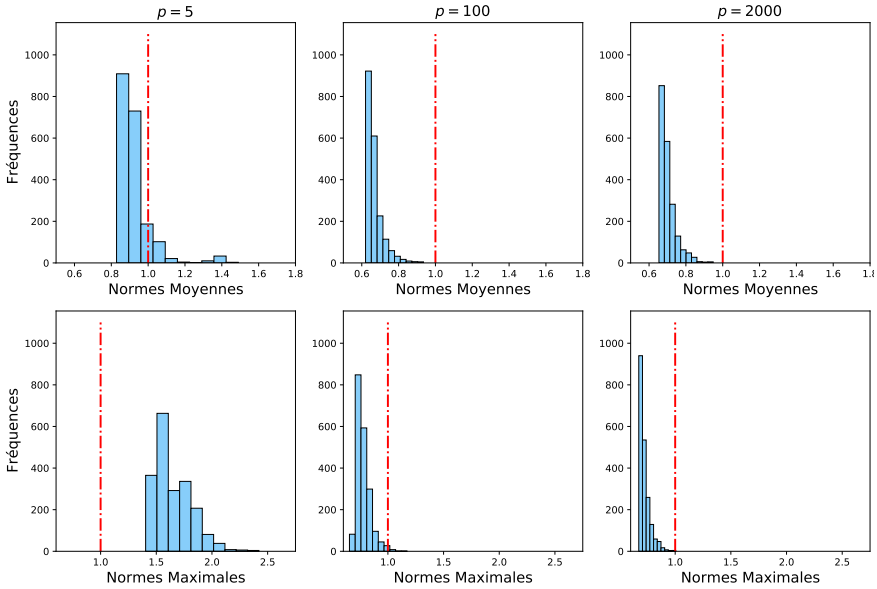

FIGURE C.2 – Histogrammes des normes du gradient $\nabla_\theta \psi(x, \theta^{(0)})$ obtenues pour différents échantillons x sur une Validation Croisée 5-folds sur la base de données Gaussienne. Sur la partie haute de la Figure, nous retrouvons la fréquence des normes moyennes obtenues sur les 5-folds et sur la partie basse les normes maximales. Les deux normes sont calculées pour différentes valeurs de $p = [5, 100, 2000]$.

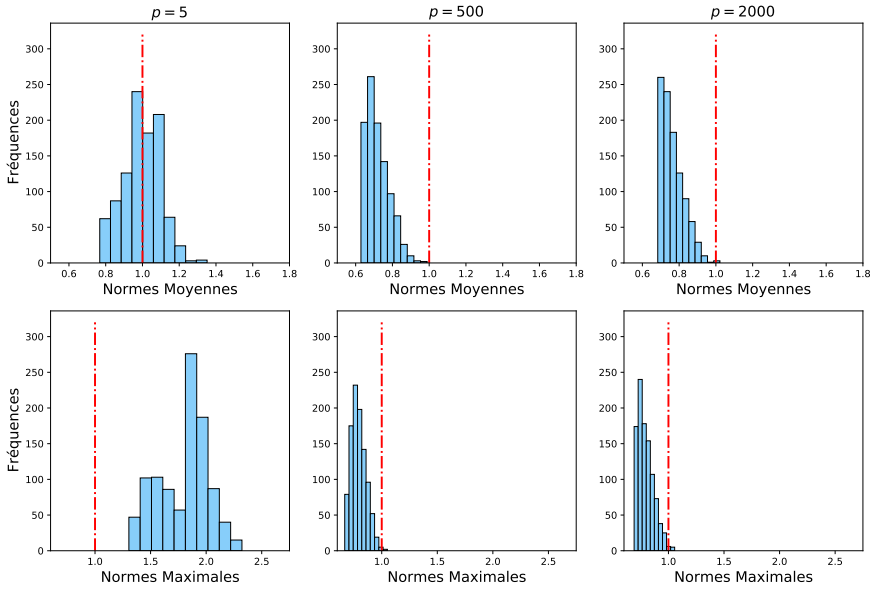

FIGURE C.3 – Histogrammes des normes du gradient $\nabla_\theta \psi(x, \theta^{(0)})$ obtenues pour différents échantillons x sur une Validation Croisée 5-folds sur la base de données Circle. Sur la partie haute de la Figure, nous retrouvons la fréquence des normes moyennes obtenues sur les 5-folds et sur la partie basse les normes maximales. Les deux normes sont calculées pour différentes valeurs de $p = [5, 100, 2000]$.

C.1.2 Comportement asymptotique de la Hessienne

Pour que l'approximation de $\psi(x, \theta)$ par $\psi^{\text{lin}}(x, \theta, \theta^{(0)})$ soit correcte, nous avons établi que la norme spectrale de la hessienne de la fonction de score par rapport à ses paramètres pris au point $(1-\tau)\theta^{(0)} + \tau\theta$ avec $\tau \in \{0, 1\}$ que l'on note $H_\theta(\psi(x, (1-\tau)\theta^{(0)} + \tau\theta))$ et définie à l'équation 4.6 tend à être nulle à mesure que le nombre p de neurones composant le SATURNN augmente.

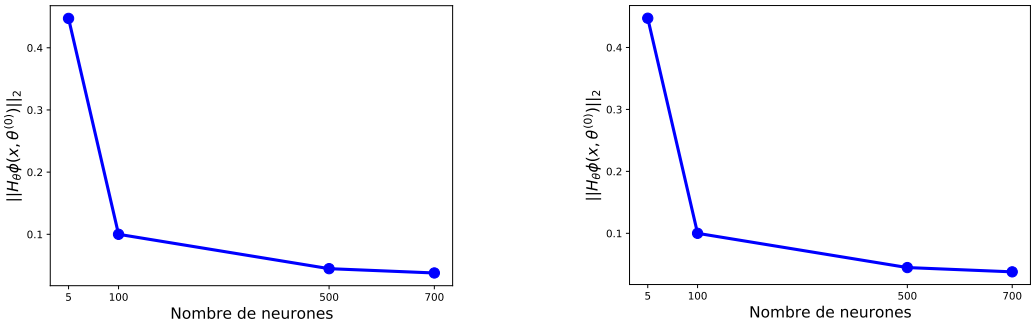

FIGURE C.4 – Normes spectrales maximales de la Hessienne $H_\theta(\psi(x, \theta^{(0)}))$ obtenues pour différentes valeurs de p sur 5-folds sur les base de données Gaussienne (gauche) et Circle (droite).

Afin de vérifier ce point, nous avons calculé la norme spectrale de la hessienne au point $\theta^{(0)}$ pour différentes valeurs de $p = [5, 100, 500, 700]$. Sur la Figure C.4 - Gauche (resp. Droite) nous pouvons constater que la norme spectrale de la hessienne $H_\theta(\psi(x, \theta^{(0)}))$ décroît à une vitesse $\frac{1}{\sqrt{p}}$ sur la base de données Gaussienne (resp. Circle).

C.2 Équivalence entre le SATURNN et la Régression Logistique

Les résultats expérimentaux présentés dans cette annexe sont complémentaires à ceux introduits en Section 4.1, ils proviennent de la base de données Circle.

Étude de l'impact de p

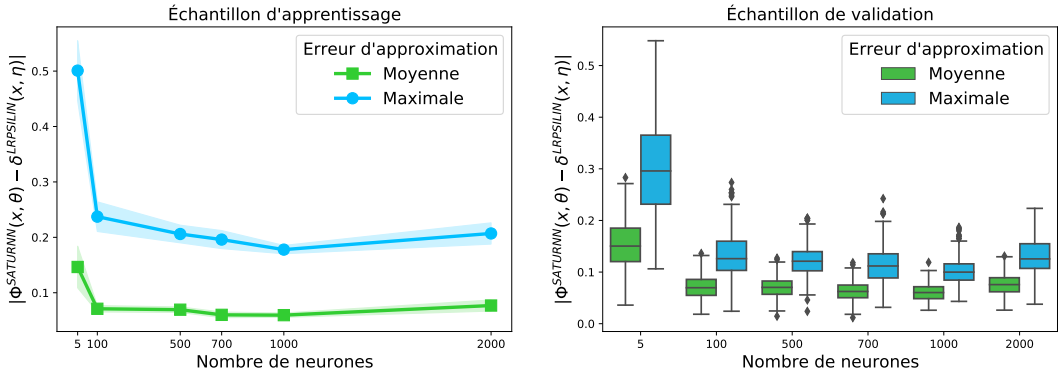

FIGURE C.5 – Erreurs d'approximation $|\Phi^{\text{SATURNN}}(x,\eta) - \delta^{\text{LR PSI LIN}}(x,\eta)|$ pour différentes valeurs de p. Ces erreurs ont été calculées par Validation Croisée 5-folds sur les échantillons d'apprentissage (Gauche) et de validation (Droite).

	Apprentissage		**Validation**	
p	**Accuracy**	**AUC**	**Accurracy**	**AUC**
5	0.72 (0.06)	0.77 (0.06)	0.70 (0.08)	0.76 (0.07)
100	0.87 (0.01)	0.94 (0.01)	0.86 (0.02)	0.94 (0.01)
500	0.88 (0.01)	0.95 (0.01)	0.89 (0.03)	0.95 (0.01)
700	0.89 (0.01)	0.95 (0.01)	0.87 (0.03)	0.95 (0.01)
1000	0.89 (0.01)	0.95 (0.01)	0.89 (0.03)	0.95 (0.01)
2000	0.88 (0.01)	0.95 (0.01)	0.89 (0.03)	0.96 (0.01)

TABLE C.1 – Résultats des LRs PSI LIN sur le jeu de données : performances globales et AUC moyennes (écart-types) obtenues par Validation Croisée 5-folds sur les échantillons d'apprentissage et de validation pour différentes valeurs de p.

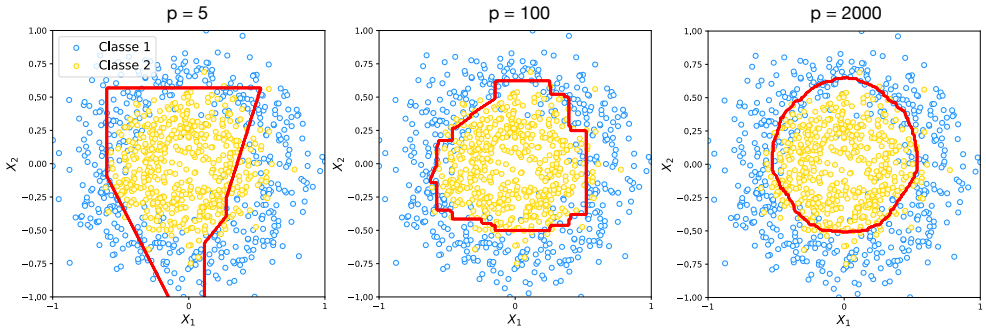

FIGURE C.6 – Règles de décision (courbe rouge) obtenues par différentes LRs PSI LIN $\delta^{\text{LR PSI LIN}}(x,\eta)$ pour différentes valeurs de p.

Étude de l'impact de la régularisation ℓ_2 sur les paramètres $\hat{\eta}$

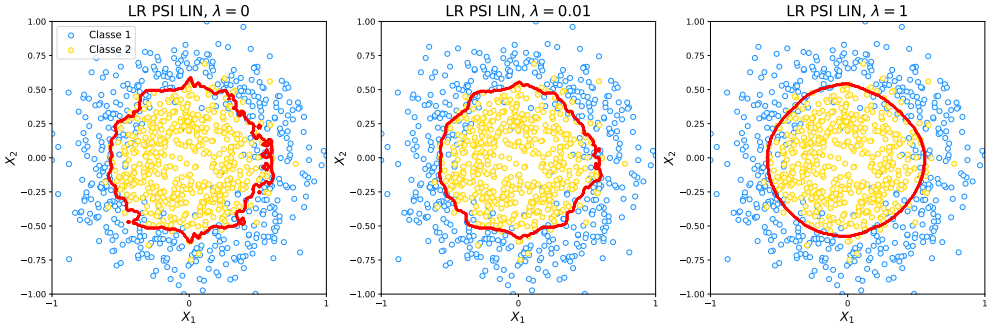

FIGURE C.7 – Règles de décision (courbe rouge) issues des différentes LRs PSI LIN $\delta^{\text{LR PSI LIN}}(x, \eta)$ pour différentes valeurs de paramètres de régularisation $\lambda = [0, 0.01, 1]$.

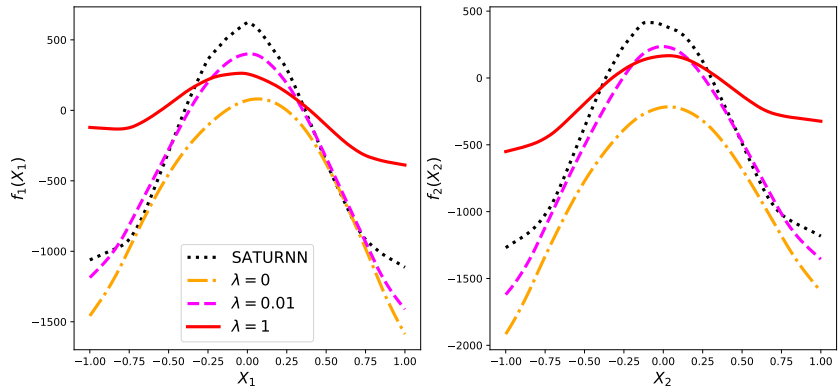

FIGURE C.8 – Splines estimées pour X_1 (Gauche) et X_2 (Droite) par le SATURNN composé de $p = 50\,000$ neurones en noire, la LR PSI LIN avec $\lambda = 0$ en orange, celle entraînée avec $\lambda = 0.01$ en rose et finalement pour $\lambda = 1$ en rouge.

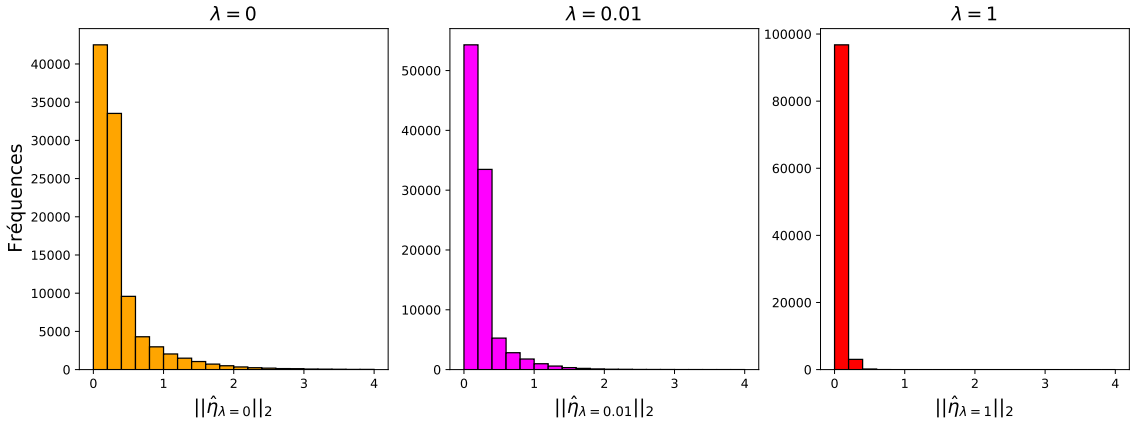

FIGURE C.9 – Histogramme des normes des $\hat{\eta}$ obtenues par Validation Croisée 5−folds $(\langle \hat{\eta}_{\text{ fold 1}}, \hat{\eta}_{\text{ fold 2}}, \hat{\eta}_{\text{ fold 3}}, \hat{\eta}_{\text{ fold 4}}, \hat{\eta}_{\text{ fold 5}} \rangle)$.

Annexe D

Compléments théoriques pour le Chapitre 5

D.1 Espérance du noyau κ_0

Le Lemme 3 établit que l'espérance de l'application noyau $\kappa_0(x, \tilde{x}) : (x, \tilde{x}) \in (\mathbb{R}^d, \mathbb{R}^d) \to \mathbb{R}$ est égale à :

$$\mathbb{E}(\kappa_0(x, \tilde{x})) = \frac{1}{p} + \frac{r^2}{6} + \frac{1}{4rd} \sum_{i=1}^{d} 2r(x_i \tilde{x}_i + 1) - |x_i - \tilde{x}_i| + \frac{1}{6}|x_i - \tilde{x}_i|^3. \tag{D.1}$$

Démonstration.
Nous supposons que le vecteur de paramètres initialisés $\theta^{(0)}$ respecte l'Hypothèse 1, à savoir pour tout $k \in \{1, \dots, p\}$, $\beta_k^{(0)} \sim \mathcal{N}(0, 1)$ et $b_k^{(0)} \sim \mathcal{U}[-r, r]$, avec $r > 0$ le rayon de la boule ouverte sur laquelle sont définies les variables descriptives $x \in \mathcal{B}_2^d(0, r)$. De plus, rappelons que les paramètres fixés du SATURNN sont initialisés selon certaines distributions : $s_k \in \{-1, 1\} \sim \mathcal{B}(1/2)$ et $\upsilon(k) \sim \mathcal{U}[\![1, d]\!]$, pour tout $k \in \{1, \dots, p\}$. Soit $\kappa_0 : (x, \tilde{x}) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ l'application suivante :

$$\kappa_0(x, \tilde{x}) = \frac{1}{p}\left[1 + \sum_{k=1}^{p} \phi\left(s_k x_{\upsilon(k)} + b_k^{(0)}\right) \phi\left(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}\right) + {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}}\right],$$

avec $\phi(\cdot) = \max\{0, \cdot\}$ l'activation ReLU.

Nous calculons l'espérance de l'activation $\kappa_0(x, \tilde{x})$:

$$\begin{aligned}
\mathbb{E}(\kappa_0(x, \tilde{x})) &= \mathbb{E}\left(\frac{1}{p}\left[1 + \sum_{k=1}^{p} \phi\left(s_k x_{\upsilon(k)} + b_k^{(0)}\right) \phi\left(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}\right)\right.\right. \\
&\qquad \left.\left. + {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}}\right]\right) \\
&= \frac{1}{p} + \frac{1}{p} \sum_{k=1}^{p} \mathbb{E}\left(\phi\left(s_k x_{\upsilon(k)} + b_k^{(0)}\right) \phi\left(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}\right)\right.
\end{aligned}$$

$$
\begin{aligned}
&\qquad + {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}} \Big) \\
&= \frac{1}{p} + \frac{1}{p} \sum_{k=1}^{p} \mathbb{P}(s_k = -1) \mathbb{E}\left(\phi\left(-x_{\upsilon(k)} + b_k^{(0)} \right) \phi\left(-\tilde{x}_{\upsilon(k)} + b_k^{(0)} \right) \right) \\
&\qquad + \mathbb{P}(s_k = 1) \mathbb{E}\left(\phi\left(x_{\upsilon(k)} + b_k^{(0)} \right) \phi\left(\tilde{x}_{\upsilon(k)} + b_k^{(0)} \right) \right) \\
&\qquad + \mathbb{P}(s_k = -1) \mathbb{E}\left({\beta_k^{(0)}}^2 \right) \mathbb{E}\left(\mathbb{1}_{\{-x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{-\tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}} \right) \\
&\qquad + \mathbb{P}(s_k = 1) \mathbb{E}\left({\beta_k^{(0)}}^2 \right) \mathbb{E}\left(\mathbb{1}_{\{x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{\tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}} \right),
\end{aligned}
$$

avec $\mathbb{P}(s_k = -1) = \mathbb{P}(s_k = 1) = \frac{1}{2}$ car $s_k \in \{-1, 1\} \sim \mathcal{B}(1/2)$, $\upsilon(k) \sim \mathcal{U}[\![1, d]\!]$, pour tout $k \in \{1, \ldots, p\}$ et d'après l'Hypothèse 1, nous avons $\beta_k^{(0)} \sim \mathcal{N}(0, 1)$ et donc $\mathbb{E}\left({\beta_k^{(0)}}^2 \right) = 1$. Ainsi nous obtenons :

$$
\begin{aligned}
\mathbb{E}(\kappa_0(x, \tilde{x})) &= \frac{1}{p} + \frac{1}{2p} \sum_{k=1}^{p} \sum_{i=1}^{d} \mathbb{P}(\upsilon(k) = i) \mathbb{E}\left(\phi\left(-x_i + b_k^{(0)} \right) \phi\left(-\tilde{x}_i + b_k^{(0)} \right) \right) \\
&\qquad + \mathbb{P}(\upsilon(k) = i) \mathbb{E}\left(\phi\left(x_i + b_k^{(0)} \right) \phi\left(\tilde{x}_i + b_k^{(0)} \right) \right) \\
&\qquad + \mathbb{P}(\upsilon(k) = i) \mathbb{E}\left(\mathbb{1}_{\{-x_i + b_k^{(0)} > 0\}} \mathbb{1}_{\{-\tilde{x}_i + b_k^{(0)} > 0\}} \right) \\
&\qquad + \mathbb{P}(\upsilon(k) = i) \mathbb{E}\left(\mathbb{1}_{\{x_i + b_k^{(0)} > 0\}} \mathbb{1}_{\{\tilde{x}_i + b_k^{(0)} > 0\}} \right) \\
&= \frac{1}{p} + \frac{1}{2pd} \sum_{k=1}^{p} \sum_{i=1}^{d} \mathbb{E}\left(\left[-x_i + b_k^{(0)} \right] \left[-\tilde{x}_i + b_k^{(0)} \right] \mathbb{1}_{\{b_k^{(0)} > x_i, b_k^{(0)} > \tilde{x}_i\}} \right) \\
&\qquad + \mathbb{E}\left(\left[x_i + b_k^{(0)} \right] \left[\tilde{x}_i + b_k^{(0)} \right] \mathbb{1}_{\{b_k^{(0)} > -x_i, b_k^{(0)} > -\tilde{x}_i\}} \right) \\
&\qquad + \mathbb{E}\left(\mathbb{1}_{\{b_k^{(0)} > x_i\}} \mathbb{1}_{\{b_k^{(0)} > \tilde{x}_i\}} \right) + \mathbb{E}\left(\mathbb{1}_{\{b_k^{(0)} > -x_i\}} \mathbb{1}_{\{b_k^{(0)} > -\tilde{x}_i\}} \right).
\end{aligned}
$$

Soit $\mathrm{f}_b(t)$ la fonction de densité de probabilité de b. Puisque nous avons supposé pour tout $k \in \{1, \ldots, p\}$, $b_k^{(0)} \sim \mathcal{U}[-r, +r]$ (Hypothèse 1), nous avons de fait $\mathrm{f}_b(t) = \frac{1}{2r}$.

$$
\begin{aligned}
\mathbb{E}(\kappa_0(x, \tilde{x})) &= \frac{1}{p} + \frac{1}{2pd} \sum_{k=1}^{p} \sum_{i=1}^{d} \int_{-r}^{r} [-x_i + t][-\tilde{x}_i + t] \mathrm{f}_b(t) \mathbb{1}_{\{t > x_i, t > \tilde{x}_i\}} \, \mathrm{d}t \\
&\qquad + \int_{-r}^{r} [x_i + t][\tilde{x}_i + t] \mathrm{f}_b(t) \mathbb{1}_{\{t > -x_i, t > -\tilde{x}_i\}} \, \mathrm{d}t \\
&\qquad + \int_{-r}^{r} \mathrm{f}_b(t) \mathbb{1}_{\{t > x_i, t > \tilde{x}_i\}} \, \mathrm{d}t + \int_{-r}^{r} \mathrm{f}_b(t) \mathbb{1}_{\{t > -x_i, t > -\tilde{x}_i\}} \, \mathrm{d}t \\
&= \frac{1}{p} + \frac{1}{2pd} \sum_{k=1}^{p} \sum_{i=1}^{d} \int_{\max(x_i, \tilde{x}_i)}^{r} \frac{1}{2r} [-x_i + t][-\tilde{x}_i + t] \, \mathrm{d}t \\
&\qquad + \int_{\max(-x_i, -\tilde{x}_i)}^{r} \frac{1}{2r} [x_i + t][\tilde{x}_i + t] \, \mathrm{d}t \\
&\qquad + \int_{\max(x_i, \tilde{x}_i)}^{r} \frac{1}{2r} \, \mathrm{d}t + \int_{\max(-x_i, -\tilde{x}_i)}^{r} \frac{1}{2r} \, \mathrm{d}t
\end{aligned}
$$

$$= \frac{1}{p} + \frac{1}{2pd}\sum_{k=1}^{p}\sum_{i=1}^{d}\frac{1}{2r}\left[\left[x_i\tilde{x}_i t - \frac{t^2}{2}(x_i+\tilde{x}_i) + \frac{t^3}{3}\right]_{\max(x_i,\tilde{x}_i)}^{r} + \left[t\right]_{\max(x_i,\tilde{x}_i)}^{r}\right.$$
$$\left. + \left[x_i\tilde{x}_i t + \frac{t^2}{2}(x_i+\tilde{x}_i) + \frac{t^3}{3}\right]_{\max(-x_i,-\tilde{x}_i)}^{r} + \left[t\right]_{\max(-x_i,-\tilde{x}_i)}^{r}\right].$$

Avec les notations, $\underline{m}_i = \min(x_i, \tilde{x}_i)$ et $\overline{m}_i = \max(x_i, \tilde{x}_i)$, l'espérance de $\kappa_0(x, \tilde{x})$ se réécrit :

$$\begin{aligned}\mathbb{E}(\kappa_0(x,\tilde{x})) &= \frac{1}{p} + \frac{1}{4rd}\sum_{i=1}^{d}\left[2r(x_i\tilde{x}_i+1) + \frac{2r^3}{3} + \underline{m}_i - \overline{m}_i - x_i\tilde{x}_i\left(\overline{m}_i - \underline{m}_i\right)\right.\\ &\qquad \left. + \frac{1}{2}\left(x_i+\tilde{x}_i\right)\left(\overline{m}_i^2 - \underline{m}_i^2\right) - \frac{1}{3}\left(\overline{m}_i^3 - \underline{m}_i^3\right)\right]\\ &= \frac{1}{p} + \frac{r^2}{6} + \frac{1}{4rd}\sum_{i=1}^{d}\left[2r(x_i\tilde{x}_i+1) + \underline{m}_i - \overline{m}_i - x_i\tilde{x}_i(\overline{m}_i - \underline{m}_i)\right.\\ &\qquad \left. + \frac{1}{2}(x_i+\tilde{x}_i)(\overline{m}_i^2 - \underline{m}_i^2) - \frac{1}{3}(\overline{m}_i^3 - \underline{m}_i^3)\right].\end{aligned}$$

Puisque $\max(a,b) = \frac{a+b+|a-b|}{2}$ et $\min(a,b) = \frac{a+b-|a-b|}{2}$ pour $a, b \in \mathbb{R}^2$, nous obtenons finalement :

$$\mathbb{E}(\kappa_0(x,\tilde{x})) = \frac{1}{p} + \frac{r^2}{6} + \frac{1}{4rd}\sum_{i=1}^{d} 2r(x_i\tilde{x}_i+1) - |x_i - \tilde{x}_i| + \frac{1}{6}|x_i - \tilde{x}_i|^3. \tag{D.2}$$

□

D.2 Variance du noyau κ_0

Le Lemme 4 propose une borne supérieure de la variance de l'application noyau $\kappa_0(x, \tilde{x})$: $(x, \tilde{x}) \in (\mathbb{R}^d, \mathbb{R}^d) \to \mathbb{R}$ défini par :

$$\kappa_0(x, \tilde{x}) = \frac{1}{p} + \frac{1}{p} \sum_{k=1}^{p} \phi(s_k x_{\upsilon(k)} + b_k^{(0)}) \phi(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}) + {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)}\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}\}}. \tag{D.3}$$

Nous pouvons démontrer que d'une part $\mathbb{V}(\kappa_0(x, \tilde{x}))$ est bornée et tend à être nulle à mesure que p augmente :

$$\mathbb{V}(\kappa_0(x, \tilde{x})) \leq \frac{1}{p^2} + \frac{1}{p} \left[8r^4 + \frac{r^3}{3} + \frac{7r^2}{3} + 13 \right] \tag{D.4}$$

$$= O\left(\frac{1}{p}\right). \tag{D.5}$$

Démonstration.
Nous supposons que le vecteur de paramètres initialisés $\theta^{(0)}$ respecte l'Hypothèse 1, à savoir pour tout $k \in \{1, \ldots, p\}$, $\beta_k^{(0)} \sim \mathcal{N}(0, 1)$ et $b_k^{(0)} \sim \mathcal{U}[-r, r]$, avec $r > 0$ le rayon de la boule ouverte sur laquelle sont définies les variables descriptives $x \in \mathcal{B}_2^d(0, r)$. De plus, rappelons que les paramètres fixés du SATURNN sont initialisés selon certaines distributions : $s_k \in \{-1, 1\} \sim \mathcal{B}(1/2)$ et $\upsilon(k) \sim \mathcal{U}[\![1, d]\!]$, pour tout $k \in \{1, \ldots, p\}$.

La variance de $\kappa_0(x, \tilde{x})$ s'écrit :

$$\begin{aligned}
\mathbb{V}(\kappa_0(x, \tilde{x})) &= \mathbb{E}(\kappa_0(x, \tilde{x})^2) - \mathbb{E}(\kappa_0(x, \tilde{x}))^2 \\
&= \mathbb{E}\left(\left(\frac{1}{p} + \frac{1}{p} \sum_{k=1}^{p} \phi(s_k x_{\upsilon(k)} + b_k^{(0)}) \phi(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}) + {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)}\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}\}} \right)^2 \right) \\
&= \mathbb{E}\left(\frac{1}{p^2} + \frac{1}{p^2} \sum_{k=1}^{p} \Big(\phi(s_k x_{\upsilon(k)} + b_k^{(0)})^2 \phi(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)})^2 + {\beta_k^{(0)}}^4 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}} \right. \right. \\
&\qquad + 2\phi(s_k x_{\upsilon(k)} + b_k^{(0)}) \phi(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}) + 2{\beta_k^{(0)}}^2 \phi(s_k x_{\upsilon(k)} + b_k^{(0)}) \phi(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}) \\
&\qquad \left. \left. + 2{\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}} \Big) \right) \right) \\
&= \frac{1}{p^2} + \frac{1}{p^2} \sum_{k=1}^{p} \mathbb{E}\left(\phi(s_k x_{\upsilon(k)} + b_k^{(0)})^2 \phi(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)})^2 + {\beta_k^{(0)}}^4 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}} \right) \\
&\qquad + 2\mathbb{E}\left(\phi(s_k x_{\upsilon(k)} + b_k^{(0)}) \phi(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}) + {\beta_k^{(0)}}^2 \phi(s_k x_{\upsilon(k)} + b_k^{(0)}) \phi(s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)}) \right. \\
&\qquad\qquad \left. {\beta_k^{(0)}}^2 \mathbb{1}_{\{s_k x_{\upsilon(k)} + b_k^{(0)} > 0\}} \mathbb{1}_{\{s_k \tilde{x}_{\upsilon(k)} + b_k^{(0)} > 0\}} \right)
\end{aligned}$$

D'après l'Hypothèse 1, nous avons $s_k \in \{-1, 1\} \sim \mathcal{B}(1/2)$ pour tout $k \in \{1, \ldots, p\}$. Nous pouvons alors réécrire la variance comme suivant :

$$\begin{aligned}
\mathbb{V}(\kappa_0(x,\tilde{x})) = \frac{1}{p^2} + \frac{1}{p^2}\sum_{k=1}^{p}\mathbb{P}(s_k=1)\Big[\mathbb{E}\Big(&\phi(x_{\upsilon(k)}+b_k^{(0)})^2\phi(\tilde{x}_{\upsilon(k)}+b_k^{(0)})^2 \\
&+\beta_k^{(0)^4}\mathbb{1}_{\{x_{\upsilon(k)}+b_k^{(0)}>0\}}\mathbb{1}_{\{\tilde{x}_{\upsilon(k)}+b_k^{(0)}>0\}}\Big) + 2\mathbb{E}\Big(\phi(x_{\upsilon(k)}+b_k^{(0)})\phi(\tilde{x}_{\upsilon(k)}+b_k^{(0)}) \\
&+\beta_k^{(0)^2}\phi(x_{\upsilon(k)}+b_k^{(0)})\phi(\tilde{x}_{\upsilon(k)}+b_k^{(0)}) + \beta_k^{(0)^2}\mathbb{1}_{\{x_{\upsilon(k)}+b_k^{(0)}>0\}}\mathbb{1}_{\{\tilde{x}_{\upsilon(k)}+b_k^{(0)}>0\}}\Big)\Big] \\
+\mathbb{P}(s_k=-1)\Big[\mathbb{E}\Big(&\phi(-x_{\upsilon(k)}+b_k^{(0)})^2\phi(-\tilde{x}_{\upsilon(k)}+b_k^{(0)})^2 + \beta_k^{(0)^4}\mathbb{1}_{\{-x_{\upsilon(k)}+b_k^{(0)}>0\}}\mathbb{1}_{\{-\tilde{x}_{\upsilon(k)}+b_k^{(0)}>0\}}\Big) \\
&+2\mathbb{E}\Big(\phi(-x_{\upsilon(k)}+b_k^{(0)})\phi(-\tilde{x}_{\upsilon(k)}+b_k^{(0)}) + \beta_k^{(0)^2}\phi(-x_{\upsilon(k)}+b_k^{(0)})\phi(-\tilde{x}_{\upsilon(k)}+b_k^{(0)}) \\
&+\beta_k^{(0)^2}\mathbb{1}_{\{-x_{\upsilon(k)}+b_k^{(0)}>0\}}\mathbb{1}_{\{-\tilde{x}_{\upsilon(k)}+b_k^{(0)}>0\}}\Big)\Big] \\
= \frac{1}{p^2} + \frac{1}{2p^2}\sum_{k=1}^{p}\mathbb{E}\Big(&\phi(x_{\upsilon(k)}+b_k^{(0)})^2\phi(\tilde{x}_{\upsilon(k)}+b_k^{(0)})^2 + \phi(-x_{\upsilon(k)}+b_k^{(0)})^2\phi(-\tilde{x}_{\upsilon(k)}+b_k^{(0)})^2\Big) \\
&+\beta_k^{(0)^4}\mathbb{1}_{\{x_{\upsilon(k)}+b_k^{(0)}>0\}}\mathbb{1}_{\{\tilde{x}_{\upsilon(k)}+b_k^{(0)}>0\}} + \beta_k^{(0)^4}\mathbb{1}_{\{-x_{\upsilon(k)}+b_k^{(0)}>0\}}\mathbb{1}_{\{-\tilde{x}_{\upsilon(k)}+b_k^{(0)}>0\}}\Big) \\
+\frac{1}{p^2}\sum_{k=1}^{p}\mathbb{E}\Big(&\phi(x_{\upsilon(k)}+b_k^{(0)})\phi(\tilde{x}_{\upsilon(k)}+b_k^{(0)}) + \phi(-x_{\upsilon(k)}+b_k^{(0)})\phi(-\tilde{x}_{\upsilon(k)}+b_k^{(0)}) \\
&+\beta_k^{(0)^2}\phi(x_{\upsilon(k)}+b_k^{(0)})\phi(\tilde{x}_{\upsilon(k)}+b_k^{(0)}) + \beta_k^{(0)^2}\phi(-x_{\upsilon(k)}+b_k^{(0)})\phi(-\tilde{x}_{\upsilon(k)}+b_k^{(0)}) \\
&+\beta_k^{(0)^2}\mathbb{1}_{\{x_{\upsilon(k)}+b_k^{(0)}>0\}}\mathbb{1}_{\{\tilde{x}_{\upsilon(k)}+b_k^{(0)}>0\}} + \beta_k^{(0)^2}\mathbb{1}_{\{-x_{\upsilon(k)}+b_k^{(0)}>0\}}\mathbb{1}_{\{-\tilde{x}_{\upsilon(k)}+b_k^{(0)}>0\}}\Big)
\end{aligned}$$

De plus, nous savons que chaque variable x_i, $i \in \{1,\dots,d\}$ a la même probabilité d'être tirée par les sélecteurs de variables $\upsilon(k)$ pour tout $k \in \{1,\dots,p\}$.

$$\begin{aligned}
\mathbb{V}(\kappa_0(x,\tilde{x})) = \frac{1}{p^2} + \sum_{k=1}^{p}\sum_{i=1}^{d}\frac{1}{2p^2d}\mathbb{E}\Big(&\phi(x_i+b_k^{(0)})^2\phi(\tilde{x}_i+b_k^{(0)})^2 + \phi(-x_i+b_k^{(0)})^2\phi(-\tilde{x}_i+b_k^{(0)})^2 \\
&+\beta_k^{(0)^4}\mathbb{1}_{\{x_i+b_k^{(0)}>0\}}\mathbb{1}_{\{\tilde{x}_i+b_k^{(0)}>0\}} + \beta_k^{(0)^4}\mathbb{1}_{\{-x_i+b_k^{(0)}>0\}}\mathbb{1}_{\{-\tilde{x}_i+b_k^{(0)}>0\}}\Big) \\
&+\frac{1}{p^2d}\mathbb{E}\Big(\phi(x_i+b_k^{(0)})\phi(\tilde{x}_i+b_k^{(0)}) + \phi(-x_i+b_k^{(0)})\phi(-\tilde{x}_i+b_k^{(0)}) \\
&+\beta_k^{(0)^2}\phi(x_i+b_k^{(0)})\phi(\tilde{x}_i+b_k^{(0)}) + \beta_k^{(0)^2}\phi(-x_i+b_k^{(0)})\phi(-\tilde{x}_i+b_k^{(0)}) \\
&+\beta_k^{(0)^2}\mathbb{1}_{\{x_i+b_k^{(0)}>0\}}\mathbb{1}_{\{\tilde{x}_i+b_k^{(0)}>0\}} + \beta_k^{(0)^2}\mathbb{1}_{\{-x_i+b_k^{(0)}>0\}}\mathbb{1}_{\{-\tilde{x}_i+b_k^{(0)}>0\}}\Big)
\end{aligned}$$

Nous supposons conformément à l'Hypothèse 1 que les biais b_k pour tout $k \in \{1,\dots,p\}$ sont indépendamment et identiquement distribués : pour tout $i,j \in \{1,\dots,p\}^2$ nous avons $\mathbb{E}(b_i) = \mathbb{E}(b_j)$, nous posons alors un b quelconque tiré selon une loi uniforme sur $[-r,r]$. De plus, nous avons supposé les coefficients $\beta_k^{(0)} \sim \mathcal{N}(0,1)$, pour tout $k \in \{1,\dots,p\}$. Ainsi, nous savons que $\mathbb{E}(\beta_k^{(0)^2}) = 1$ et $\mathbb{E}(\beta_k^{(0)^4}) = 3$. Nous obtenons alors :

$$\begin{aligned}\mathbb{V}(\kappa_0(x,\tilde{x})) = \frac{1}{p^2}\sum_{i=1}^{d}\frac{1}{2pd}\Big[&\mathbb{E}\left(\phi(x_i+b)^2\phi(\tilde{x}_i+b)^2+\phi(-x_i+b)^2\phi(-\tilde{x}_i+b)^2\right)\\ &+3\mathbb{E}\left(\mathbb{1}_{\{x_i+b>0\}}\mathbb{1}_{\{\tilde{x}_i+b>0\}}+\beta_k^{(0)^4}\mathbb{1}_{\{-x_i+b>0\}}\mathbb{1}_{\{-\tilde{x}_i+b>0\}}\right)\Big]\\ +\frac{1}{pd}\Big[&2\mathbb{E}\left(\phi(x_i+b)\phi(\tilde{x}_i+b)+\phi(-x_i+b)\phi(-\tilde{x}_i+b)\right)\\ &+\mathbb{E}\left(\mathbb{1}_{\{x_i+b>0\}}\mathbb{1}_{\{\tilde{x}_i+b>0\}}+\beta_k^{(0)^4}\mathbb{1}_{\{-x_i+b>0\}}\mathbb{1}_{\{-\tilde{x}_i+b>0\}}\right)\Big]\end{aligned}$$

Nous pouvons majorer cette équation puisque nous savons que le maximum atteint pour l'application ReLU est :

$$\max(\phi(x_i+b)) = \max(x_i+b).$$

Nous obtenons alors :

$$\begin{aligned}\mathbb{V}(\kappa_0(x,\tilde{x})) &\leq \frac{1}{p^2}+\sum_{i=1}^{d}\frac{1}{2pd}\left[\mathbb{E}\left((x_i+b)^2(\tilde{x}_i+b)^2+(-x_i+b)^2(-\tilde{x}_i+b)^2\right)+6\right]\\ &\quad+\frac{1}{pd}\left[2\mathbb{E}\left((x_i+b)(\tilde{x}_i+b)+(-x_i+b)(-\tilde{x}_i+b)\right)+2\right]\\ &= \frac{1}{p^2}+\sum_{i=1}^{d}\frac{1}{2pd}\left[\mathbb{E}\left(2x_i^2\tilde{x}_i^2+2x_i^2b+8x_i\tilde{x}_ib^2+2\tilde{x}_i^2b+2b^2\right)+6\right]\\ &\quad+\frac{2}{pd}\left[\mathbb{E}(2x_i\tilde{x}_i+2b^2)+2\right]\end{aligned}$$

De plus, les variables prennent des valeurs dans une boule ouverte de rayon $r>0$, ainsi nous savons que pour tout $i\in\{1,\dots,d\}$, $x_i<r$.

$$\begin{aligned}\mathbb{V}(\kappa_0(x,\tilde{x})) &\leq \frac{1}{p^2}+\sum_{i=1}^{d}\frac{1}{2pd}\left[\mathbb{E}(2r^4+2r^2b+8r^2b^2+2r^2b+2b^2)+6\right]+\frac{2}{pd}\left[\mathbb{E}(2r^2+2b^2)+2\right]\\ &= \frac{1}{p^2}+\frac{1}{2p}\left[2r^4+6+4r^2\mathbb{E}(b)+(8r^2+2)\mathbb{E}(b^2)\right]+\frac{1}{p}\left[r^2+1+\mathbb{E}(b^2)\right]\end{aligned}$$

Finalement, puisque pour tout $k\in\{1,\dots,p\}$, $b_k^{(0)}\sim\mathcal{U}[-r,r]$, nous avons $\mathbb{E}(b)=0$ et $\mathbb{E}(b^2)=\mathbb{V}(b)=\frac{r^2}{3}$. Nous pouvons alors majorer la variance par :

$$\begin{aligned}\mathbb{V}(\kappa_0(x,\tilde{x})) &\leq \frac{1}{p^2}+\frac{1}{2p}\left[2r^4+6+(8r^3+2)\frac{r^2}{3}\right]+\frac{1}{p}\left[r^2+2+\frac{2r^2}{3}\right]\\ &= \frac{1}{p^2}+\frac{1}{p}\left[8r^4+\frac{r^3}{3}+\frac{7r^2}{3}+13\right] \qquad \text{(D.6)}\\ &= O\left(\frac{1}{p}\right). \qquad \text{(D.7)}\end{aligned}$$

□

Annexe E

Compléments de résultats numériques pour le Chapitre 5

Dans cette annexe, des résultats expérimentaux supplémentaires pour le Chapitre 5 sont présentés. Tous ces résultats sont issus d'expériences menées sur la base de données Gaussienne.

E.1 Approximation du SATURNN par les Régressions Logistiques à Noyau

Équivalence entre la KLR et l'EKLR

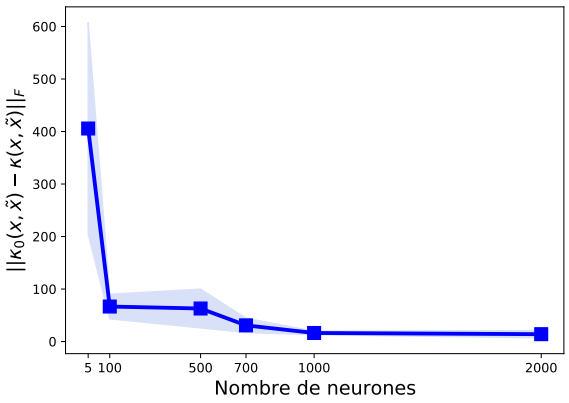

FIGURE E.1 – Normes de Frobenius moyennes entre les noyaux κ_0 et κ ($||\kappa_0(x,\tilde{x}) - \kappa(x,\tilde{x})||_F$) obtenues sur 5 sous-échantillons de la base de données Circle pour différentes valeurs de p.

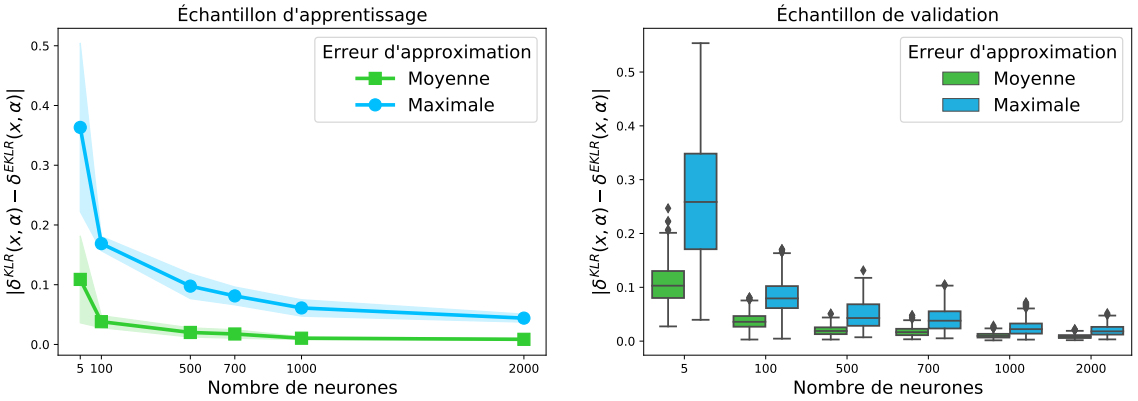

FIGURE E.2 – Erreurs d'approximation $|\delta^{\mathrm{KLR}}(x,\alpha) - \delta^{\mathrm{EKLR}}(x,\alpha)|$ moyennes (courbe verte) et maximales (courbe bleue) pour différentes valeurs de p. Ces erreurs ont été calculées par Validation Croisée 5−folds sur les échantillons d'apprentissage (Gauche) et de validation (Droite).

Équivalence entre le SATURNN et les méthodes à noyau KLR et EKLR

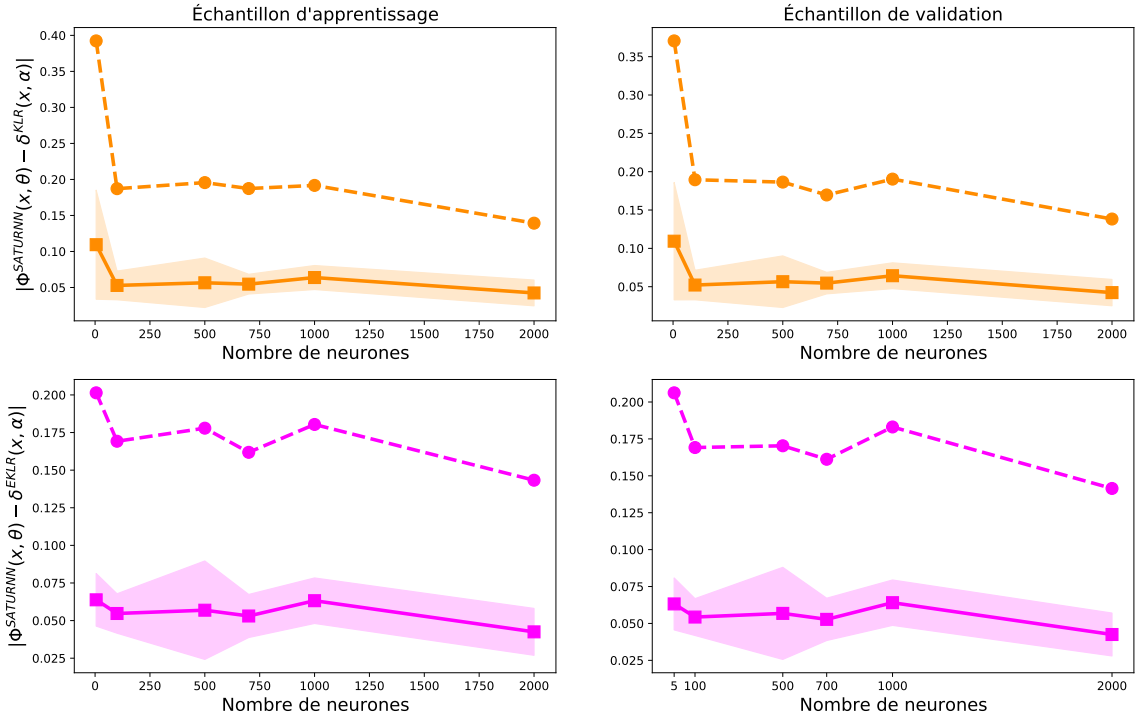

FIGURE E.3 – Haute - Erreurs d'approximation $|\phi^{\mathrm{SATURNN}}(x,\theta) - \delta^{\mathrm{KLR}}(x,\eta)|$ moyenne (trait continu) et maximales (en pointillé) pour différentes valeurs de p. Basse - Erreurs d'approximation $|\phi^{\mathrm{SATURNN}}(x,\theta) - \delta^{\mathrm{EKLR}}(x,\eta)|$ moyenne (trait continu) et maximales (en pointillé) pour différentes valeurs de p. Ces erreurs ont été calculées par Validation Croisée 5−folds sur les échantillons d'apprentissage (Gauche) et de validation (Droite).

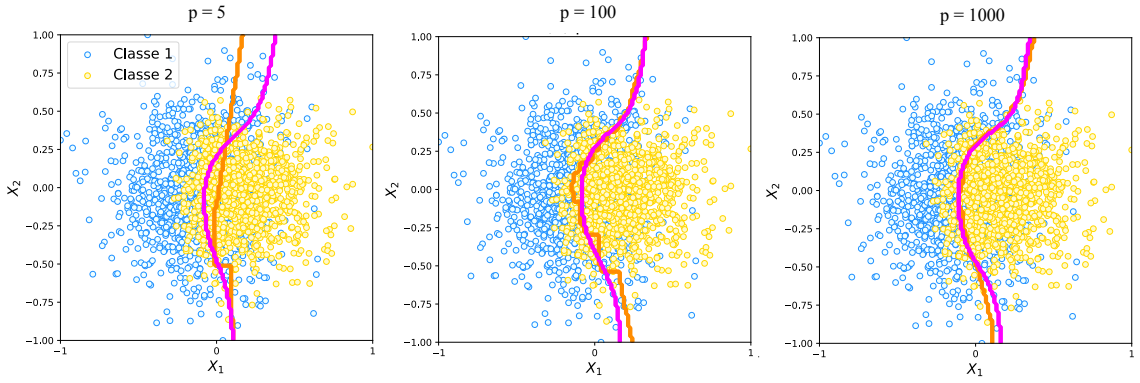

FIGURE E.4 – Règles de décision des KLRs (orange) et EKLRs (rose) obtenues sur la base de données Circle pour différentes valeurs de p ($p = 5$ - Gauche, $p = 100$ - Milieu et $p = 1000$ - Droite).

E.2 Comparaison des KLRs et EKLRs aux méthodes à noyau traditionnelles

Comparaison des performances

		Échantillons d'apprentissage		Échantillon de Validation	
		Perf. Globale	**AUC**	**Perf. Globale**	**AUC**
SATURNN		0.77 (0.01)	0.85 (0.01)	0.78 (0.01)	0.86 (0.02)
KLR	$\lambda = 0$	0.76 (0.01)	0.85 (0.01)	0.76 (0.01)	0.84 (0.01)
	$\lambda = 0.01$	0.76 (0.01)	0.85 (0.01)	0.76 (0.01)	0.83 (0.01)
	$\lambda = 10$	0.75 (0.01)	0.84 (0.01)	0.74 (0.02)	0.83 (0.01)
EKLR	$\lambda = 0$	0.76 (0.01)	0.85 (0.01)	0.76 (0.01)	0.83 (0.01)
	$\lambda = 0.01$	0.76 (0.01)	0.85 (0.01)	0.75 (0.01)	0.83 (0.01)
	$\lambda = 10$	0.75 (0.01)	0.84 (0.01)	0.74 (0.02)	0.82 (0.01)
SVM Linéaire		0.73 (0.01)	0.82 (0.01)	0.71 (0.06)	0.81 (0.01)
SVM Gaussien		0.78 (0.01)	0.85 (0.01)	0.76 (0.01)	0.83 (0.01)
SVM Polynomial $d = 3$		0.70 (0.03)	0.81 (0.01)	0.68 (0.07)	0.80 (0.04)

TABLE E.1 – Performances prédictives et AUC moyennes (écart-types) obtenues par Validation Croisée 5−folds sur les échantillons d'apprentissage et de validation. Pour les KLRs et EKLRs, différentes régularisations ont été utilisées : $\lambda = [0, 0.01, 10]$. Nos contributions sont en bleu.

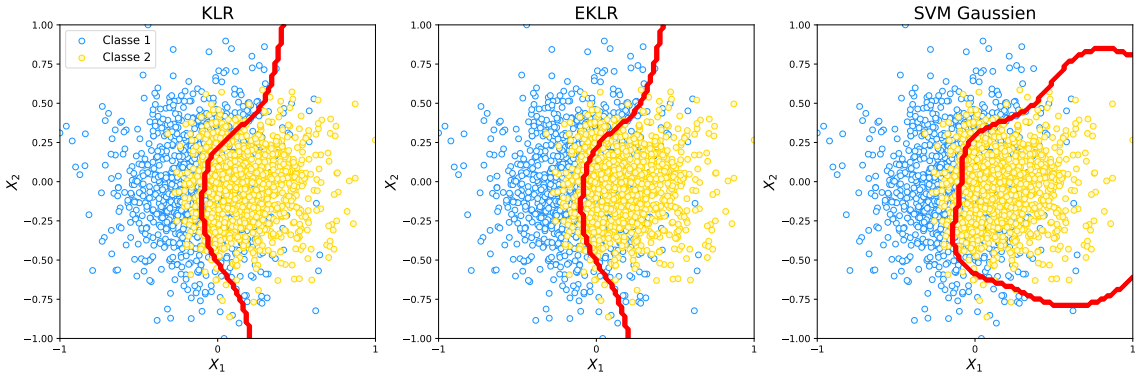

FIGURE E.5 – Règles de Décision issues de la KLR (Gauche), l'EKLR (Milieu) et du SVM à noyau Gaussien (Droite). Les KLR et EKLR présentés sur cette figure sont ceux entraînés avec une petite régularisation $\lambda = 0.01$.

Explicabilité de la règle de décision

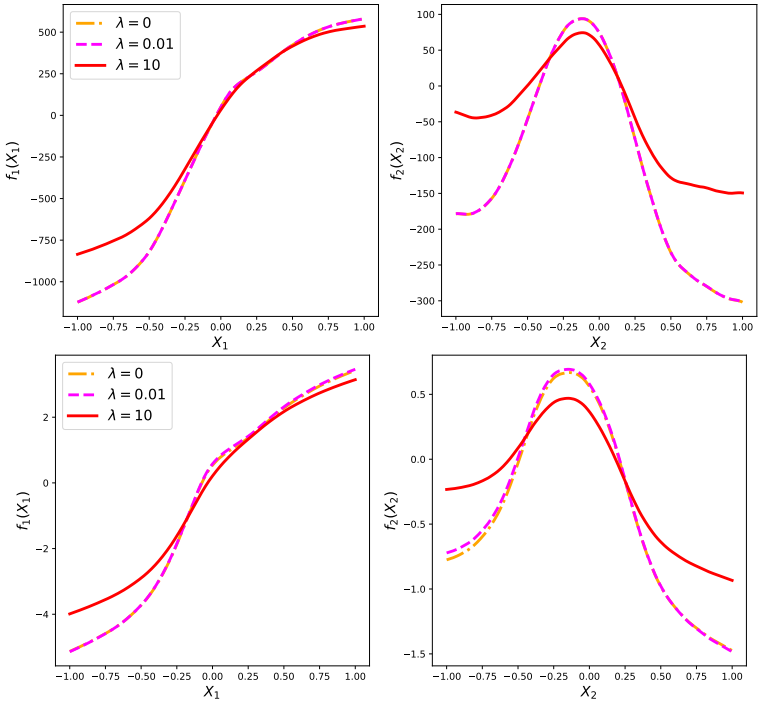

FIGURE E.6 – Splines estimées pour les variables X_1 (Gauche) et X_2 (Droite) par les KLR (Haut) et EKLR (Bas) avec différents paramètres de régularisation $\lambda = 0$ en orange, $\lambda = 0.01$ en rose et $\lambda = 10$ en rouge.

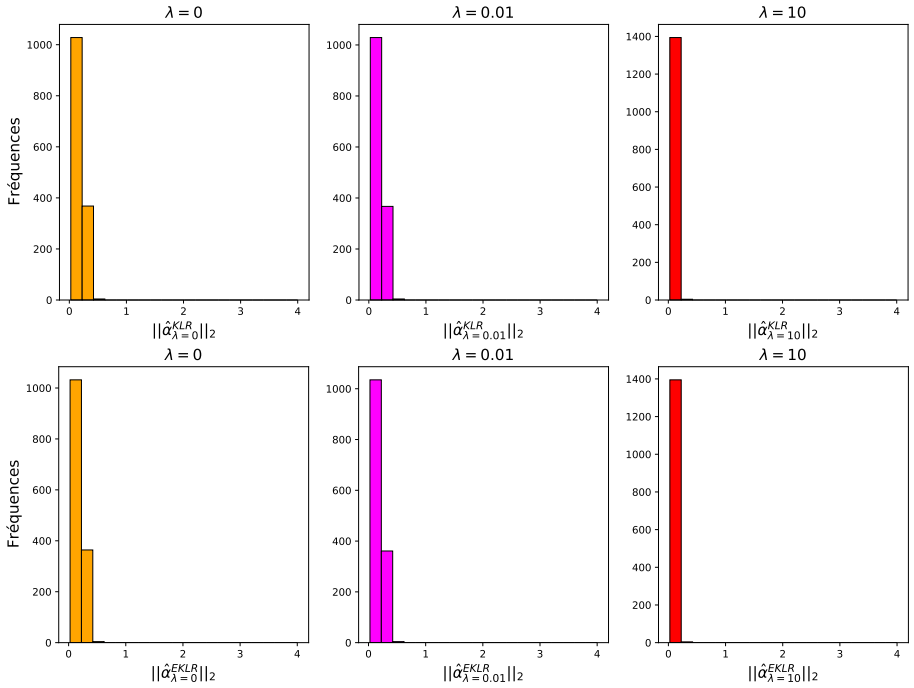

FIGURE E.7 – Histogramme des normes des paramètres estimées par les KLRs $\hat{\alpha}^{\text{KLR}}$ (Haut) et EKLRs $\hat{\alpha}^{\text{EKLR}}$ (Bas)par Validation Croisée 5−folds pour différentes valeurs de régularisation $\lambda = 0$ (orange), $\lambda = 0.01$ (rose) et $\lambda = 10$ (rouge).

Annexe F

Compléments de résultats pour le diagnostique de la bipolarité

Dans cette annexe, des résultats expérimentaux complémentaires pour le Chapitre 7, issus de la collaboration avec l'IPMC sur la prédiction de la bipolarité sont présentés.

F.1 Outliers

	Moyenne	**Min**	**Max**	**75%**
CCL3	34.33	3.0	299.79	43.71
CCL17	511.64	10.83	5405.51	624.62
CCL22	**1363.4**	**18.48**	**10316.15**	**1633.15**
IL6	**0.91**	**0.03**	**55.21**	**0.81**
IL7	13.78	0.06	70.59	19.19
IL8	54.36	0.04	5871.19	13.89
IL10	0.32	0.02	19.8	0.27
IL12p40	88.56	0.16	510.53	111.6
IL15	1.65	0.08	5.15	1.97
IL16	205.35	3.67	2120.94	224.69
IL27	1186.46	135.66	69446.68	1242.63
IFNγ	4.47	0.18	263.11	3.48
TNFα	**1.48**	**0.02**	**25.93**	**1.68**

TABLE F.1 – Statistiques (Moyenne, Valeur Minimale, Valeur Maximale et 3e quantile) permettant de mettre en avant la présence de valeurs extrêmes pour 13 cytokines du jeu de données. Pour les trois variables en gras, leur répartition ainsi que les *outliers* sont mis en évidence sur la Figure 7.1.

F.2 Sélection de variables

F.2.1 Test du chi-2

Le test du chi-2 teste les hypothèses :

— L'hypothèse nulle H0 : Les deux variables sont indépendantes

— L'hypothèse H1 : Non H0

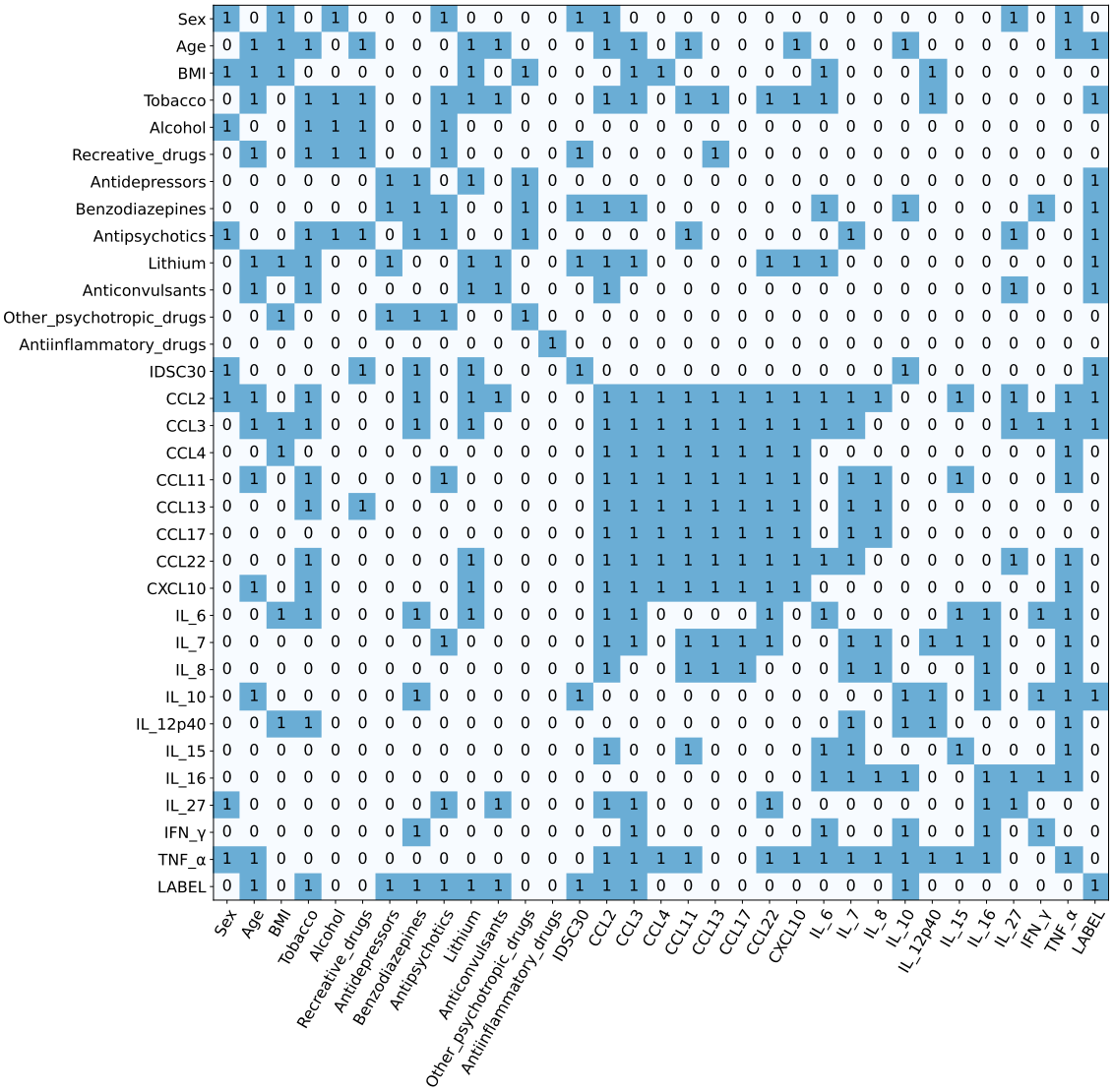

FIGURE F.1 – Résultats du chi-2 sur la base de données de Montpellier. Les tests rejetant l'hypothèse nulle d'indépendance des variables sont en bleu.

F.2.2 VIP

	Occurence	Pourcentage
Sex	5.0	50.0
Age	10.0	**100.0**
BMI	6.0	60.0
Tobacco	6.0	60.0
Alcohol	1.0	10.0
Recreative drugs	2.0	20.0
Antidepressors	10.0	**100.0**
Benzodiazepines	8.0	80.0
Antipsychotics	10.0	**100.0**
Lithium	10.0	**100.0**
Anticonvulsants	10.0	**100.0**
Other psychotropic drugs	2.0	20.0
Antiinflammatory drugs	3.0	30.0
IDSC30	7.0	70.0
CCL2	6.0	60.0
CCL3	3.0	30.0
CCL4	2.0	20.0
CCL11	0.0	0.0
CCL13	0.0	0.0
CCL17	1.0	10.0
CCL22	6.0	60.0
CXCL10	8.0	80.0
IL6	3.0	30.0
IL7	1.0	10.0
IL8	3.0	30.0
IL10	7.0	70.0
IL12p40	2.0	20.0
IL15	1.0	10.0
IL16	5.0	50.0
IL27	1.0	10.0
IFNγ	2.0	20.0
TNFα	0.0	0.0

TABLE F.2 – VIPs obtenus par Régression Logistique Elastic-Net avec les hyperparamètres optimaux sur Validation Croisée 10−folds. Les variables retenues sont en bleu.

Annexe G

Collaboration sur le classifieur Minimax

Pendant ces trois années de thèse, nous avons étroitement collaboré avec Cyprien Gilet, Maître de Conférence à l'Université Technologique de Compiègne. Ses travaux de thèse et de post-doctorat ont porté sur l'élaboration d'un classifieur Minimax adapté à des enjeux rencontrés notamment dans le domaine bio-médical.

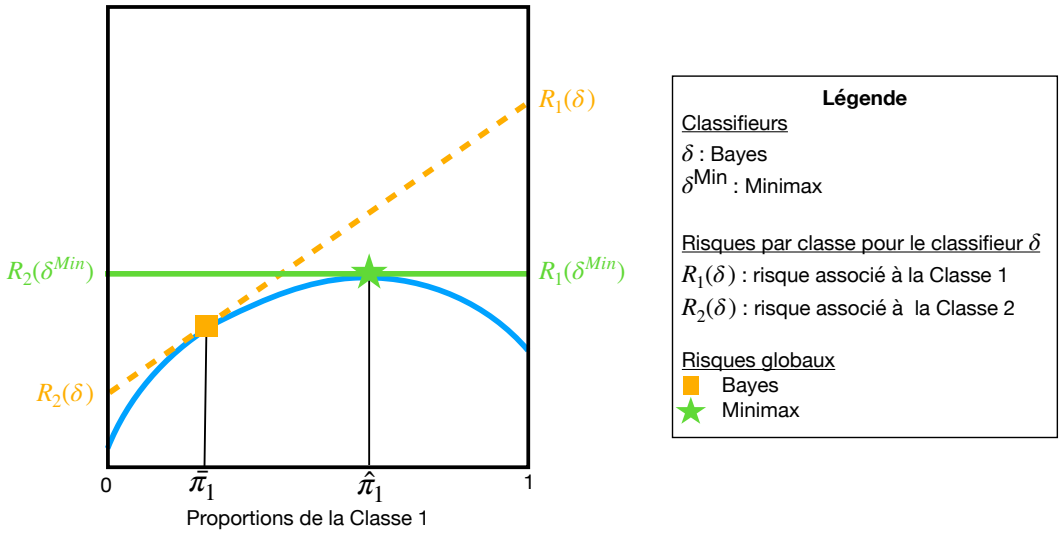

FIGURE G.1 – Illustration du Minimax dans le cas d'un problème de classification binaire. En abscisse nous avons les proportions de la Classe 1, et en ordonnées les risques associés à la Classe 2 (Gauche) et la Classe 1 (Droite). En bleu nous avons le risque de Bayes calculé pour différentes proportions de la Classe 1. Le rectangle orange représente le risque global issu du classifieur de Bayes δ avec des proportions $\bar{\pi}_1$ et l'étoile verte celui du classifieur Minimax δ^{Min}.

Les bases de données médicales ont tendance à être fortement déséquilibrées, les différentes classes de patients sont souvent inégalement réparties. Les algorithmes ont alors tendance à léser les populations sous-représentées dans leur règle de décision. Bien que toute une catégorie de patients soit mal classifiée, la performance globale des algorithmes peut s'avérer très élevée si cette classe minoritaire est présente en très petite proportion. Or pour l'application médicale et notamment pour la prédiction de

pathologies, il s'avère que dans la majorité des cas, les patients minoritaires sont ceux d'intérêt qu'il convient de détecter. Sur la Figure G.1, nous considérons 2 classes telle que la Classe 1 est moins représentée, soit en proportion $\bar{\pi}_1$ (ainsi $\bar{\pi}_2 = 1 - \bar{\pi}_1 > \bar{\pi}_1$). Le risque de Bayes est le risque minimum d'erreur que nous pouvons espérer. Pour ces proportions données, le classifieur de Bayes δ obtient un risque d'erreur global faible représenté par le carré orange. En revanche ses risques par classe que l'on retrouve aux extrémités de la courbe orange, la tangente au risque de Bayes au point $\bar{\pi}_1$ sont quant à eux très déséquilibrés. Le risque d'erreur pour la Classe 2 $R_2(\delta)$ majoritairement représentée dans la base d'apprentissage est très faible, tandis que celui associé la Classe 1 $R_1(\delta)$ est très élevé. Le classifieur Minimax a été développé afin d'égaliser les risques d'erreur entre les classes. Pour ce faire, le Minimax va chercher les proportions $\hat{\pi}_1$ maximisant le risque de Bayes (courbe bleue). En ce point, le risque global obtenu (étoile verte) est supérieur à celui résultant du classifieur de Bayes (carré orange). En revanche, les risques par classe sont quant à eux égalisés : $R_1(\delta^{\text{Min}}) \simeq R_2(\delta^{\text{Min}})$. Ainsi, il existe un compromis entre minimiser le risque d'erreur global et égaliser les risques conditionnels. Comme illustré sur la Figure G.1, obtenir des performances prédictives par classe du même ordre de grandeur entraîne une performance globale moins élevée : la prédiction des patients les plus représentés se détériore au profit de celle des patients sous-représentés.

Une autre problématique à laquelle le Minimax répond concerne le *probability shift*, à savoir le changement des proportions par classe au cours du temps. Pour l'application médicale, le problème de *probability shift* est souvent rencontré. Par exemple, la proportion de personnes présentant des symptômes grippaux est plus élevée en hiver qu'au printemps. Il est tout aussi possible que les proportions par classe varient d'un centre hospitalier à un autre, et ainsi que l'entraînement d'un modèle de classification sur une base de données issue d'un centre de collecte ne soit pas généralisable aux autres. Lorsque les proportions par classe changent au cours du temps, les risques d'erreur globaux des classifieurs évoluent linéairement. Sur la Figure G.1, si les proportions $\bar{\pi}_1$ évoluent au cours du temps, le risque global issu du classifieur de Bayes va évoluer linéairement sur la courbe orange. Si la proportion de patients appartenant à la Classe 1 augmente, le classifieur de Bayes engendrera un risque d'erreur supérieur à celui du Minimax. Le Minimax quant à lui est construit de sorte à ce que même si les proportions évoluent dans les bases données, le risque d'erreur de classification restera lui constant (évoluera sur la courbe verte).

Ainsi, le Minimax est robuste à la fois aux problématiques liées au déséquilibre des bases de données et au *probability shift* souvent rencontrés dans le domaine bio-médical. Ma principale implication dans cette collaboration a été de réaliser l'implémentation de cet algorithme en Python et de pouvoir tester cet algorithme à la fois sur des données clinico-biologiques mais aussi sur des images médicales.